

SIMULATION-BASED PROCESS ANALYSIS AND OPTIMIZATION  
FOR CONTINUOUS PHARMACEUTICAL MANUFACTURING

By

ZILONG WANG

A dissertation submitted to the

School of Graduate Studies

Rutgers, The State University of New Jersey

In partial fulfillment of the requirements

For the degree of

Doctor of Philosophy

Graduate Program in Chemical and Biochemical Engineering

Written under the direction of

Marianthi G. Ierapetritou, PhD

And approved by

---

---

---

---

New Brunswick, New Jersey

May, 2018

## ABSTRACT OF THE DISSERTATION

### Simulation-based Process Analysis and Optimization for Continuous Pharmaceutical Manufacturing

by ZILONG WANG

Dissertation Director:

Marianthi G. Ierapetritou, PhD

This thesis explores to apply simulation-based mathematical tools to understand and improve continuous pharmaceutical manufacturing processes. On the basis of a continuous direct compaction process, we propose a framework consisting of global sensitivity analysis, feasibility analysis, and optimization to systematically extract process knowledge from simulation models. Under this framework, efficient analysis and optimization methods have been developed for both deterministic systems and stochastic systems. Specifically, mathematical techniques including surrogate modeling and adaptive sampling are adopted to address difficult problems involving computationally expensive black-box systems. The developed algorithms in this thesis can be used to obtain a comprehensive knowledge of solids-based pharmaceutical processes.

## **Acknowledgment**

I have been encouraged by many people along the journey of my Ph.D. studies over the past four years.

I would like to first express my sincerest gratitude and appreciation to my advisor, Dr. Marianthi Ierapetritou. You have always been there for me whenever I need your support. Thank you for putting your trust in me and allowing me to explore a variety of interesting projects. Thank you for pushing me to always think as a Ph.D. does, to do better when I am not good enough, and to aim higher when I have reached a milestone. Your wisdom has inspired me in many ways as I searched for a better idea in my research. Your positive energies have given me strength whenever I feel frustrated. Your advising and mentoring will always guide me in my future career.

I would also like to thank my committee members, Dr. Rohit Ramachandran, Dr. Ioannis P. Androulakis, and Dr. Amanda Rogers. I have had many valuable discussions with you in my research. Through these discussions, I was able to understand better the meaning of my projects, and explore alternative methods that are better suitable for the problem I was trying to solve. My research was greatly enlightened by the work of Dr. Amanda Rogers, who has been greatly supportive since I first started my Ph. D. journey.

My projects were funded by FDA (DHHS - FDA - 1 U01 FD005295-01) as well as National Science Foundation Engineering Research Center on Structured Organic Particulate Systems (NSF-ECC 0540855). In my research, I have had many WebEx meetings with Michael and Thomas from FDA. I am greatly thankful to their valuable suggestions on my work.

I have been fortunate to get the chance to work with a group of wonderful people at Rutgers: Nikisha, Jinjun, Zhaojia, Nihar, Ravendra, Parham, Sebastian, Nirupa, Lisia, Abhay, Atharv, Ou, Pooja, Jierui, Xian, Charles, Shishir, Shu, Praneeth, Siddharth, Yue, Hao, Tianyi, Huiyi, Yexuan. It has been a great pleasure to know you in my life, and I know we will always keep in touch.

My interests in process systems engineering started from my M.S. studies at Carnegie Mellon University. I am greatly thankful to Dr. Nick Sahinidis, who was my advisor and taught me a great deal in doing computational studies as well as doing literature reviews. Those basic research skills have been extremely beneficial to my later Ph.D. studies. While I was in Carnegie Mellon, I was fortunate to make friends with many wonderful people: Qi, Xue, Jun, Yajun, Zixi, Deyang. They have also helped me a lot later in my Ph.D. studies. Also, I want to express special thanks to my lifetime friends, Yifan, Wenjia, and Xinyi.

Finally, I need to thank my parents. Throughout my life, they have supported me with their unconditional love and caring. They always have faith in me, and always encourage me to be a better person. I feel proud to have such parents, and I hope I have made them feel proud, too. To my wife, thank you for putting your trust in me and for encouraging me to take challenges in life. It is my greatest luck to have you in my life.

## Table of Contents

<b>ABSTRACT OF THE DISSERTATION .....</b>	<b>ii</b>
<b>Acknowledgment.....</b>	<b>iii</b>
<b>List of figures.....</b>	<b>x</b>
<b>List of tables.....</b>	<b>xvii</b>
<b>I. Introduction .....</b>	<b>1</b>
<b>1 Introduction to Continuous Pharmaceutical Manufacturing .....</b>	<b>2</b>
<b>1.1 An Overview of models for the Continuous Direct Compaction Process.....</b>	<b>3</b>
<b>1.2 Process analysis and optimization for deterministic systems.....</b>	<b>12</b>
<b>1.3 Process analysis and optimization for stochastic systems.....</b>	<b>13</b>
<b>II. Literature Review .....</b>	<b>15</b>
<b>2 Global Sensitivity and Feasibility Analysis of Continuous Pharmaceutical Manufacturing Processes .....</b>	<b>16</b>
<b>2.1 Introduction .....</b>	<b>16</b>
<b>2.2 Global Sensitivity Analysis .....</b>	<b>18</b>
2.2.1 Methods.....	19
<b>2.3 Feasibility Analysis.....</b>	<b>28</b>
2.3.1 Methods.....	29
<b>2.4 Software.....</b>	<b>35</b>
<b>2.5 Summary .....</b>	<b>35</b>
<b>3 Applications of Optimization in Continuous Pharmaceutical Manufacturing .....</b>	<b>36</b>
<b>3.1 Introduction .....</b>	<b>36</b>

<b>3.2</b>	<b>Different optimization objectives in the pharmaceutical process development ...</b>	<b>38</b>
3.2.1	Single-objective optimization .....	39
3.2.2	Multi-objective optimization.....	41
<b>3.3</b>	<b>Applications of data-driven models in optimization .....</b>	<b>43</b>
3.3.1	Sampling plan .....	44
3.3.2	Building a data-driven model.....	45
3.3.3	Model validation .....	50
3.3.4	Data-driven models in support of optimization needs .....	52
<b>3.4</b>	<b>Optimization methods in pharmaceutical processes .....</b>	<b>53</b>
3.4.1	Derivative-based methods.....	54
3.4.2	Derivative-free methods.....	56
<b>3.5</b>	<b>Summary .....</b>	<b>60</b>
<b>III.</b>	<b>Process analysis and optimization for deterministic systems .....</b>	<b>62</b>
<b>4</b>	<b>A Novel Feasibility Analysis Method for Black-Box Processes using a Radial Basis Function Adaptive Sampling Approach .....</b>	<b>63</b>
4.1	Introduction .....	63
4.2	Surrogate-based feasibility analysis.....	67
4.3	Surrogate Model Accuracy .....	75
4.4	Computational studies .....	77
4.5	Case study of a roller compaction process .....	102
4.6	Summary and future work .....	106
<b>5</b>	<b>Process Analysis and Optimization of Continuous Pharmaceutical Manufacturing using Flowsheet Models .....</b>	<b>107</b>
5.1	Introduction .....	107
5.2	Sensitivity Analysis.....	111

5.2.1	Morris Method .....	112
5.2.2	PRCC Method.....	113
5.2.3	Sobol' Method.....	115
<b>5.3</b>	<b>Feasibility Analysis.....</b>	<b>116</b>
5.3.1	Surrogate-based method.....	117
<b>5.4</b>	<b>Analysis of the Direct Compaction line for production of pharmaceutical tablets</b>	<b>120</b>
5.4.1	Sensitivity Analysis Results.....	120
5.4.2	Surrogate-based Feasibility Analysis Results .....	127
<b>5.5</b>	<b>Optimization of the Direct Compaction line using flowsheet models.....</b>	<b>132</b>
<b>5.6</b>	<b>Conclusion .....</b>	<b>136</b>
<b>IV.</b>	<b>Process analysis and optimization for stochastic systems .....</b>	<b>138</b>
<b>6</b>	<b>Surrogate-based feasibility analysis for black-box stochastic simulations with</b>	
	<b>heteroscedastic noise variances.....</b>	<b>139</b>
<b>6.1</b>	<b>Introduction .....</b>	<b>139</b>
<b>6.2</b>	<b>Surrogate-based approaches for feasibility analysis.....</b>	<b>146</b>
6.2.1	Stochastic Kriging.....	146
6.2.2	Adaptive sampling methods.....	149
<b>6.3</b>	<b>Convergence.....</b>	<b>160</b>
<b>6.4</b>	<b>Computational experiments .....</b>	<b>162</b>
6.4.1	Test problems and noise scenarios.....	162
6.4.2	Performance measures .....	164
6.4.3	Implementation details.....	165
<b>6.5</b>	<b>Results.....</b>	<b>166</b>
<b>6.6</b>	<b>Application to a pharmaceutical manufacturing process model .....</b>	<b>175</b>

6.7	Summary and future work .....	178
<b>7</b>	<b>A novel surrogate-based optimization method for black-box stochastic simulations with heteroscedastic noises.....</b>	<b>180</b>
7.1	Introduction .....	181
7.2	Stochastic Kriging Model .....	187
7.3	Infill criteria .....	189
7.3.1	Expected Improvement approach.....	190
7.3.2	One stage approach .....	193
7.4	Computational studies .....	196
7.4.1	Noise scenarios .....	196
7.4.2	Performance measures and data visualization .....	197
7.4.3	Test problems and implementation details.....	200
7.5	Results.....	201
7.5.1	Interpretation on the data figures .....	201
7.5.2	Influence of noise structure on algorithms’ performance .....	205
7.6	Case study of a pharmaceutical manufacturing process .....	208
7.7	Summary and future work .....	212
<b>8</b>	<b>A Kriging-based method with feasibility enhancements for constrained optimization of black-box stochastic systems.....</b>	<b>213</b>
8.1	Introduction .....	213
8.2	Kriging-based optimization approaches .....	219
8.2.1	Stochastic Kriging Model .....	219
8.2.2	Infill criteria .....	222
8.2.3	Framework of optimization.....	234
8.3	Computational studies .....	238

8.3.1	Test problems and additional competitive solvers .....	238
8.3.2	Noise scenarios .....	239
8.3.3	Performance measures .....	240
8.3.4	Implementation details .....	241
<b>8.4</b>	<b>Results.....</b>	<b>241</b>
8.4.1	Capabilities of returning a truly feasible solution .....	241
8.4.2	Accuracies of “locating” the feasible global optimum .....	245
8.4.3	Accuracies of “predicting” the objective value at the near-optimal solution ....	251
<b>8.5</b>	<b>Optimization of a continuous pharmaceutical manufacturing process .....</b>	<b>253</b>
<b>8.6</b>	<b>Summary and future work .....</b>	<b>255</b>
<b>V.</b>	<b>Conclusions .....</b>	<b>258</b>
<b>9</b>	<b>Conclusions and Future Work.....</b>	<b>259</b>
9.1	Major contributions .....	259
9.2	Future work .....	260
	<b>Acknowledgment of previous publications.....</b>	<b>263</b>
	<b>Appendix.....</b>	<b>265</b>
	<b>Bibliography .....</b>	<b>273</b>

## List of figures

Figure 1. Flowsheet of Direct Compaction.....	4
Figure 2. Example of the trajectory in a 3D input space. ....	20
Figure 3. Example of the radial design in a 3D input space .....	20
Figure 4. Algorithm of the Kriging-based adaptive sampling approach .....	34
Figure 5. Two approaches that use a data-driven model to support optimization. (a) sequential approach; (b) adaptive approach.....	52
Figure 6. Framework for Kriging-based and RBF-based feasibility analysis algorithm.....	75
Figure 7. Illustrating figure for the model accuracy metrics .....	76
Figure 8. Contour of Feasibility function for branincon. (Feasible regions are shaded with dots.) .....	78
Figure 9. Contour of Feasibility function for ex3. (Feasible region is shaded with dots.) .....	79
Figure 10. Contour of feasibility function for sasenacon. (Feasible regions are shaded with dots.) .....	80
Figure 11. Contour of feasibility function for camelback. (Feasible regions are shaded with dots.) .....	81
Figure 12. 3D plot of feasible region boundary for qcp4con. (Feasible regions are shaded with dots.).....	82
Figure 13. Contour of surrogate models for branincon. (a) RBF model for branincon; (b) Kriging model for branincon. (Initial sample points are noted with dots. Adaptive sampling points are noted with circle points.).....	84
Figure 14. Contour of surrogate models for ex3. (a) RBF model for ex3; (b) Kriging model for ex3. (Initial sample points are noted with dots. Adaptive sampling points are noted with circle points.) .....	84

Figure 15. Contour of surrogate models for sasenacon. (a) RBF model for sasenacon; (b) Kriging model for sasenacon. (Initial sample points are noted with dots. Adaptive sampling points are noted with circle points.)..... 85

Figure 16. Contour of Surrogate models for camelback. (a) RBF model for camelback; (b) Kriging model for camelback. (Initial sample points are noted with dots. Adaptive sampling points are noted with circle points.)..... 86

Figure 17. 3D plot of feasible region boundaries from surrogate models for qcp4con. (a) RBF model for qcp4con; (b) Kriging model for qcp4con. (Initial sample points are noted with circle points. Adaptive sampling points are noted with asterisk points.) ..... 87

Figure 18. Contour of surrogate models for sasenacon using different scale factors. (a) RBF model using a larger scale factor; (b) RBF model using a smaller scale factor (Initial sample points are noted with dots. Adaptive sampling points are noted with circle points.) ..... 88

Figure 19. Comparison of surrogate accuracy by iteration of adaptive sampling for branincon. (a) CF% (b)CIF (c) NC% ..... 89

Figure 20. Comparison of surrogate accuracy by iteration of adaptive sampling for ex3. (a) CF% (b)CIF (c) NC% ..... 90

Figure 21. Comparison of surrogate accuracy by iteration of adaptive sampling for sasenacon. (a) CF% (b)CIF (c) NC% ..... 90

Figure 22. Comparison of surrogate accuracy by iteration of adaptive sampling for camelback. (a) CF% (b)CIF (c) NC% ..... 91

Figure 23. Comparison of surrogate accuracy by iteration of adaptive sampling for qcp4con. (a) CF% (b)CIF (c) NC% ..... 91

Figure 24. Number of adaptive sampling points v.s. number of initial sample points using the RBF-based and the Kriging-based method. (a) branincon; (b) ex3; (c) sasenacon; (d) camelback; (e) qcp4con..... 94

Figure 25. Initial surrogate models for “branincon” with 4 ( $=2^2$ ) initial sample points. (a) Initial RBF model. (b) Initial Kriging model ..... 96

Figure 26. Final surrogate models for “branincon” after adaptive sampling. (a) Final RBF model after 90 iterations. (b) Final Kriging model after 300 iterations. Circle points represent adaptive sampling points..... 96

Figure 27. Comparison of surrogate accuracy by iteration of adaptive sampling for g4con. (a) CF% (b)CIF (c) NC% ..... 102

Figure 28. Comparison of surrogate accuracy by iteration of adaptive sampling for t3con. (a) CF% (b)CIF (c) NC% ..... 102

Figure 29. Contour of surrogate models for feasibility function values of roller compaction model. (a) RBF model for roller compaction; (b) Kriging model for roller compaction. (Black dash-dot line represents the feasible region boundary of the original model; blue dashed line represents the feasible region boundary predicted by the surrogate model.)..... 105

Figure 30. Algorithm of the RBF-based adaptive sampling approach for feasibility analysis.... 120

Figure 31. PRCC results on the DC line. .... 124

Figure 32. Sobol’ s total effect sensitivity index ( $ST_i$ ) on the DC line..... 127

Figure 33. Contour plots of the feasibility of the DC line flowsheet model. (The red dotted line is the feasible region boundary. The area within the red dotted is the feasible region with feasibility function values less than or equal to zero.)..... 131

Figure 34. Dynamic simulation results of the concentration at inlet and outlet of the blender. (a) API concentration; (b) Excipient concentration; (c) Lubricant concentration..... 135

Figure 35. Dynamic simulation results of tablet properties at the optimal operation conditions. (a) API concentration; (b) tablet weight; (c) tablet hardness; (d) waste index. (Solid line: simulation results. Dashed lines: upper and lower bounds of the quality constraints)..... 136

Figure 36. Demonstration of a one-dimensional Kriging model. .... 151

Figure 37. Illustration of the geometric meaning of <i>EIfeas</i> .....	153
Figure 38. Algorithm of the <i>EIfeas</i> method.....	154
Figure 39. A case study of "branin" using the <i>EIfeas</i> approach for feasibility analysis. (a) feasible region boundary (thick-dashed line) of the true function, and heteroscedastic noise (filled contour). (b) predicted feasible region boundary (thick-solid line) with the initial SK model, and <i>EIfeas</i> (filled contour). (c) predicted feasible region boundary (thick-solid) line with the updated SK model, and <i>EIfeas</i> (filled contour).....	155
Figure 40. A case study of "branin" using <i>EIfeas</i> and <i>AEIfeas</i> . (a) Results of <i>EIfeas</i> ; (b) Results of <i>AEIfeas</i> .....	157
Figure 41. The difference between <i>EQIfeas</i> , 0.1 and <i>EQIfeas</i> , 0.9. (1) Results of <i>EQIfeas</i> , 0.1; (2) Results of <i>EQIfeas</i> , 0.9.....	160
Figure 42. Results of "example3" (plan A for sampling allocation). (a) CF%; (b) CIF%; (c) NC%.....	167
Figure 43. Results of "g4con" (plan A for sampling allocation). (a) CF%; (b) CIF%; (c) NC%.....	167
Figure 44. Results of "branin" (plan A for sampling allocation). (a) CF%; (b) CIF%; (c) NC%.....	169
Figure 45. Results of "sasena" (plan A for sampling allocation). (a) CF%; (b) CIF%; (c) NC%.....	170
Figure 46. Results of "camelback" (plan A for sampling allocation). (a) CF%; (b) CIF%; (c) NC%.....	171
Figure 47. Results of "example3" . (a) CF%; (b) CIF%; (c) NC%.....	172
Figure 48. Results of "g4con" . (a) CF%; (b) CIF%; (c) NC%.....	173
Figure 49. Results of "branin" . (a) CF%; (b) CIF%; (c) NC%.....	173

Figure 50. Results of “sasena” . (a) CF%; (b) CIF%; (c) NC% .....	174
Figure 51. Results of “camelback” . (a) CF%; (b) CIF%; (c) NC% .....	174
Figure 52. Feasible region for the RC process.....	178
Figure 53. SK-EI algorithm .....	192
Figure 54. Illustration of 4 noise scenarios with D1 test problem. Solid line: $fx$ ; Dashed line: $fx$ +/- $2\xi(x)$ .....	197
Figure 55. Illustration of two performance measures with the D1 test problem. Black dot-dashed line: objective function $fx$ ; red solid line: SK prediction $yx$ .....	199
Figure 56. Boxplot and data profiles of “D1” test problem. (a) boxplot of $yorigx$ after 100 iterations of adaptive sampling; (b) boxplot of $gap(x)$ after 100 iterations of adaptive sampling; (c) data profiles of $\eta y95$ for easy noise scenarios; (d) data profiles of $\eta gap0.2$ for easy noise scenarios; (e) data profiles of $\eta y95$ for hard noise scenarios; (f) data profiles of $\eta gap0.2$ for hard noise scenarios. ....	203
Figure 57. Boxplot and data profiles of “Rosenbrock” test problem. (a) boxplot of $yorigx$ after 100 iterations of adaptive sampling; (b) boxplot of $gap(x)$ after 100 iterations of adaptive sampling; (c) data profiles of $\eta y95$ for easy noise scenarios; (d) data profiles of $\eta gap0.2$ for easy noise scenarios; (e) data profiles of $\eta y95$ for hard noise scenarios; (f) data profiles of $\eta gap0.2$ for hard noise scenarios. ....	204
Figure 58. Flowsheet of DC line.....	209
Figure 59. Noise-free landscape of objective function. Left: 3D surface. Right: contour plot ...	210
Figure 60. Performance measures and estimated variance at returned optimal solution by iteration. .....	212
Figure 61. 2D Test problem “constrained Branin” . (a) Objective and constraint. Contours represent the objective function; diamond points are three global minima of the objective function; white-shaded areas are infeasible regions; thick solid lines are feasibility boundaries. (b)	

Noises in the objective function. Contours represent the standard deviation of the noise in the objective; diamond points are three global minima of the objective function. (c) Noises in the constraint function. Contours represent the standard deviation of the noise in the constraint; thick solid lines are feasibility boundaries..... 226

Figure 62. Infill criteria. (a)  $EI$ . (b)  $AEI$ . (c)  $cAEIf$ . Contour represent the infill criteria; dot points are initial points; thick dashed lines are predicted feasibility boundaries; white-shaded areas are predicted infeasible regions. .... 227

Figure 63. 2D Test problem “G24” . (a) Objective and constraint. Contours represent the objective function; diamond point is the global minimum of the objective function; white-shaded areas are infeasible regions; thick solid lines are feasibility boundaries. (b) Noises in the objective function. Contours represent the standard deviation of the noise in the objective; the diamond point is the global minimum of the objective function. (c) Noises in the constraint function. Contours represent the standard deviation of the noise in the constraint; thick solid lines are feasibility boundaries. .... 232

Figure 64. Infill criteria. (a)  $cAEIf$ ; (b)  $EQIFg$ ; (c)  $FEI$ . Contours represent the surface of the infill criteria. Dot points are initial point. Thick dashed lines are predicted feasibility boundaries. White-shaded areas are predicted infeasible regions. The thin dotted line in (b) is the contour of the predicted objective equal to  $f^{***}$ . The thin solid line in (b) is the contour of the predicted constraint equal to  $g^{**}$ ..... 233

Figure 65. Optimization framework of the Kriging-based approach..... 237

Figure 66. Contours of constraint functions. (a) new Branin; (b) Sasena. Thick solid lines are feasible region boundaries ..... 245

Figure 67. Boxplots of  $\log_{10}gap1$ ..... 249

Figure 68. Profiles of  $\log_{10}gap1$ . Thick solid lines represent median profiles for “constrained EI” ; thick dashed line represent median profiles for “Feasibility-enhanced EI” ; lower and

upper thin solid lines represent the 1<sup>st</sup> quartile and 3<sup>rd</sup> quartile profiles for “constrained EI” ;  
lower and upper thin dashed lines represent the 1<sup>st</sup> quartile and 3<sup>rd</sup> quartile profiles for  
“feasibility-enhanced EI” ..... 250

Figure 69. Boxplots of  $\log_{10}gap2$ ..... 252

Figure 70. Flowsheet of a Continuous Direct Compaction process..... 253

Figure 71. “Branin” test function. (a) “easy” noise; (b) “hard” noise. Thick-dashed line:  
feasible region boundaries; filled contour: standard deviation of the noise term ..... 266

Figure 72. “Camelback” test function. (a) “easy” noise; (b) “hard” noise. Thick-dashed  
line: feasible region boundaries; filled contour: standard deviation of the noise term ..... 266

Figure 73. “Example3” test function. (a) “easy” noise; (b) “hard” noise. Thick-dashed  
line: feasible region boundaries; filled contour: standard deviation of the noise term ..... 267

Figure 74. “Sasena” test function. (a) “easy” noise; (b) “hard” noise. Thick-dashed line:  
feasible region boundaries; filled contour: standard deviation of the noise term ..... 267

## List of tables

Table 1. Correlation kernel functions .....	31
Table 2. Regression Models.....	69
Table 3. Correlation Models .....	69
Table 4. Results on 2D and 3D test problems.....	82
Table 5. Minimum number of total samples.....	97
Table 6. Results on 5D and 6D test problems with grid sampling plan for initial points.....	99
Table 7. Results on 5D and 6D test problems with LHS plan for initial points .....	100
Table 8. Parameter values and boundaries in the roller compactor case study.....	102
Table 9. Comparison of accuracy between RBF-based method and Kriging-based method .....	105
Table 10. 22 input factors for sensitivity analysis .....	121
Table 11. 20 output variables for sensitivity analysis.....	122
Table 12. Ranges for 6 uncertain variables and nominal values for the remaining 4 material properties.....	128
Table 13. Constraints for 20 output variables.....	128
Table 14. Bounds of the decision variables .....	133
Table 15. Optimal operation conditions.....	134
Table 16. Test problems.....	162
Table 17. Noise functions .....	164
Table 18. Summary of allocating the sampling budget .....	165
Table 19. Number fraction of 30 runs identifying all feasible regions for “camelback” .....	171
Table 20. Parameters and bounds for the roller compaction model .....	176
Table 21. Performance measures for the RC test problem .....	177
Table 22. Summary of 7 test problems .....	200
Table 23. Summary of comparison between SK-EI and SK-OS with <i>yorigx</i> and $\eta y95$ .....	205

Table 24. Summary of comparison between SK-EI and SK-OS with $gap(x)$ and $\eta gap0.2$ .....	207
Table 25. Summary of optimization results .....	211
Table 26. Differences between the “constrained EI” method and the “feasibility-enhanced EI” method.....	237
Table 27. Summary of the test problems .....	238
Table 28. Expressions for the noise functions .....	240
Table 29. Total numbers of returned near-optimal solutions being feasible .....	244
Table 30. Optimization results for the pharmaceutical case study .....	255

# I. Introduction

## 1 Introduction to Continuous Pharmaceutical Manufacturing

Over the past decade, there has been a significant growth of research in modernizing manufacturing processes in the pharmaceutical industry. The main goal is to improve the agility, flexibility, and robustness of the manufacturing process, which is important for solving the drug shortage problems [1]. To achieve this goal, the continuous manufacturing (CM) process is considered as a promising approach and has attracted increasing research attention both from the academia and the industry. CM is an integrated flow process in which materials are continuously processed at each step and directly sent to the next step for further processing [2]. Compared to the traditional batch manufacturing process, CM has a great amount of potential in reducing process steps, using smaller equipment size and plant footprint, and having higher process controllability, all of which can result in lower capital and operation costs, fewer risks of defected products, and better final drug products [3].

However, the advantages of a CM pharmaceutical process can only be realized based on an in-depth process understanding. For the manufacturing process of solid-oral dosage forms, which are the majority of the drug products [4], it can be especially difficult since the powder systems are more complex than the fluid-based systems. Extensive experimental studies have been conducted to characterize material properties [5] and unit operation [6], investigate the traceability of materials [7], and develop online measurement tools [8] of critical quality attributes of intermediate and products. On the other hand, process models have also been built to supplement the experimental studies and enhance process understanding. First-principle models, such as discrete-element method (DEM) simulations, have been extensively used to study particle-level phenomena (e.g. segregation, agglomeration) [9]. However, its huge computational costs usually prohibit the use of DEM to model large numbers of particles. An alternative approach is to use the Population Balance Model (PBM), which has been widely used to model granulation [10], blending [11], and milling [12]. Moreover, semi-empirical models, such as residence time distribution (RTD) models [6] and

data-driven models [13] have also been implemented to describe the performance of continuous processes. Recently, integrated flowsheet models have been developed to capture the influence of material properties and process conditions on the final drug products [14].

On the basis of a developed process model, we can apply advanced mathematical tools to systematically analyze and improve the process. In the rest of this chapter, we first give an overview of the models that are developed for a continuous direct compaction process in Section 1.1. These models are used throughout the following chapters as a case study for the developed process analysis and optimization methods. Then, we briefly introduce the process analysis and optimization methods for deterministic systems in Section 1.2. Such methods have been further extended to stochastic systems, which are mentioned in Section 1.3. Section 1.2 and 1.3 can be used as a guide to the work detailed in Chapters 4 to 8.

### 1.1 An Overview of models for the Continuous Direct Compaction Process

The flowsheet model in this work is built based on a continuous DC pilot plant that has been installed and situated at ERC C-SOPS, Rutgers University (flowsheet shown in Figure 5). In this DC process, an active pharmaceutical ingredient (API) and an excipient (substance added along with API to aid the processing of the drug) are first fed to the co-mill, which is used for the de-lumping purpose. The lubricant is added after the co-mill to improve the flowability of the powder mixture. Then the raw materials are mixed in a continuous convective blender, in order to provide a homogeneous mixture of all the component ingredients. Finally, the well-mixed blends transport to a rotary feed frame, which feeds the powder mixture into dies where the press compacts the powders into tablets.

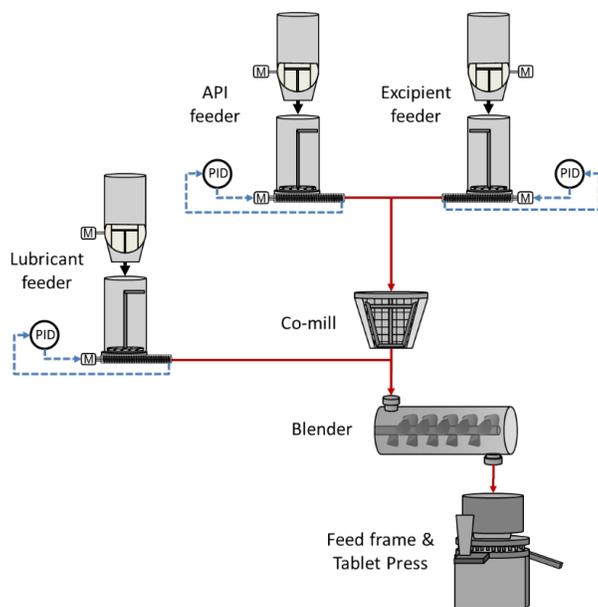


Figure 1. Flowsheet of Direct Compaction

The flowsheet model we used in this work was developed in gPROMS, a dynamic simulation software by Process Systems Enterprise. gPROMS has a variety of built-in model libraries, and also allows users to customize and build their own models. It supports different numerical solvers that can be used for models that include differential algebraic equations (DAEs), and partial differential equations (PDEs), which makes gPROMS a quite useful tool for dynamic simulation. For pharmaceutical processes, gPROMS has been widely used to build both semi-empirical models[15] and population balance models (PBMs)[16]. In this paper, we use semi-empirical models for each unit operation because they require less computational cost (compared to PBMs) when integrated as a flowsheet model. The unit operation models that we used are described as follows.

### ***Loss-in-weight Feeder***

Loss-in-weight (LIW) feeders are used to feed ingredients in the continuous tablet manufacturing process. They usually consist of a hopper, which is used as a receptacle for raw materials to be fed,

a flow aid system which is used to break powder bridges and facilitate material flow, and a conveying system, e.g. a rotating screw, which is used to transport materials out of the feeder. A discharge screen can be used to break up clumps at the exit of the feeder. In order to get constantly stable and accurate feeding flow rate, LIW feeders are usually operated under the gravimetric mode, in which case the flow rate is controlled by automatically adjusting the screw speed according to the loss in weight of the materials in the feeder over time.

Recently, research has been conducted to increase the understanding of LIW feeders. Engisch and Muzzio[17] have developed a method to characterize the LIW feeders, which can be used for the selection of feeding tools. Boukouvala, Muzzio et al.[18] proposed to use data-driven methods to investigate the design space of LIW feeders, and compare the performance of different designs by using optimization programming techniques. Boukouvala, Niotis et al.[15] and Rogers, Inamdar et al. [14] used a first-order delay differential equation to describe the dynamic behavior of a closed-loop LIW feeder. In the current work, we use a semi-empirical model to simulate the feeder behavior, which was initially proposed by Escotet-Espinoza, Jayjock et al.[19]

The actual mass flow rate out of a feeder is modeled with a time-dependent expression (Equation 64):

$$F_{out}(t) = ff(t) \omega(t) \quad (1)$$

where  $\omega(t)$  is the screw speed controlled by a PID controller taken from the gPROMS model library.  $ff(t)$  is the feed factor, defined as the maximum mass of powder fitting in a screw flight.  $ff(t)$  has the unit of mass per screw revolution, and is expressed by Equation 2.

$$ff(t) = \rho_{effective}(t) V_{ScrewPitch} \quad (2)$$

where  $\rho_{effective}(t)$  is the effective density of the materials in the screw pitch, which is a time-dependent material property. This density also accounts for the fill efficiency of the screw pitches.

$V_{ScrewPitch}$  is the volume per screw pitch, which is an equipment-dependent constant. The feed factor was found to be dependent on the amount of materials available in the hopper, indicating that changes in the mass of material in the hopper could change the mass delivered per screw revolution. This phenomenon can be interpreted as the changes in the effective density of the powder entering the screws due to the changes in pressure exerted by the static head of the material above. This relationship was found to be pseudo-first-order in nature and dependent on material properties such as bulk density and flowability.  $ff(t)$  is modeled with Equation 3.

$$\frac{1}{\beta} \frac{d(ff(t))}{dW(t)} + ff(t) = ff_{level}^{sat} \quad (3)$$

$$\therefore ff(W(t)) = ff_{level}^{sat} - \text{Exp}[-\beta W(t)](ff_{level}^{sat} - ff_{level}^{min})$$

In this equation, the feed factor can be computed as a function of the amount of materials in the feeder hopper ( $W(t)$ ). The major coefficients of this equation are the saturated feed factor ( $ff_{level}^{sat}$ ), the minimum feed factor ( $ff_{level}^{min}$ ), and the feed factor exponential decay constant ( $\beta$ ), which are regressed from data. These coefficients are related to material properties and therefore allow for relationship of bulk properties and equipment performance. The saturated and minimum feed factors were found to be directly correlated to the bulk density of the material. The pressure decay constant was found to be positively correlated to the material's Hausner Ratio. (i.e. a flowability related number defined as “tapped bulk density” divided by “aerated bulk density” [20]) The feed factor exponential decay constant is a lumped parameter which incorporates equipment geometry into the system by accounting for hopper area. This value can be used as an indicator of how much pressure the material distributes the compression forces to the walls of the system. Characterizing the feed factor exponential decay constant and the span of the feed factor coefficients is critical for both equipment design and selection.

### ***Co-mill***

A high shear co-mill is used before blending for a de-lumping purpose. It is composed of a cone-shaped screen and a rotating impeller in the center[21]. The impeller rotates at high speed and grounds the materials between the impeller and the screen until the materials have sizes less than the screen holes and exit the co-mill. Experimental work has been done by Vanarase, Osorio et al.[22] to investigate the effects of process parameters and material properties on the flow behaviors of the powder materials. Deng, Scicolone et al.[23] have used discrete element method (DEM) simulation to compute the residence time distribution (RTD) and mean residence time (MRT), and demonstrate the relationship between MRT and process parameters to be consistent with the finding in Vanarase, Osorio et al.[22].

In this work, the co-mill model includes a mass balance equation, which relates the hold-up and flow-rate-in to the flow-rate-out (Equation 4). The model assumes the co-mill reaches a steady state mass accumulation following first-order dynamics based on the mean residence time of the unit. This assumption is based on the fact that the flow-rate-out of powder systems is dependent on the accumulation inside of the unit.

$$\bar{\tau} \frac{dM(t)}{dt} + M(t) = M_{ss}$$

$$M_{ss} = \bar{\tau} F_{in}^{total} \quad (4)$$

$$\frac{dM(t)}{dt} = F_{in}^{total} - F_{out}^{total}$$

$M(t)$  is mass hold-up in the co-mill;  $\bar{\tau}$  is the mean residence time;  $M_{ss}$  is the mass hold-up at steady state;  $F_{in}^{total}$  is the mass flow rate at the inlet of the co-mill, which is equal to the sum of the flow rate of API and Excipient;  $F_{out}^{total}$  is the mass flow rate at the outlet of the co-mill.

The MRT is calculated with an empirical expression, as Equation 5.

$$\bar{\tau} = \tau_{max} \left( 1 - \exp \left( -\frac{a}{\omega_{co-mill} - \omega_{min}} - \frac{b}{F_{in}^{total}} \right) \right) \quad (5)$$

where  $\omega_{co-mill}$  is the impeller speed;  $\omega_{min}$  is the minimum value of impeller speed;  $a$ ,  $b$ , and  $\tau_{max}$  are regressed parameters from experiments.

Mixing of the co-mill has been modeled using a perfect mixing model based on RTD experiments. The short residence time of the system is what dictates the level of dissipation of incoming concentrations. The drug concentration (mass fraction) is calculated with a CSTR-based Equation 6, where  $C_{in}^i$  and  $C_{out}^i$  are the concentration of component  $i$  at inlet and outlet respectively.

$$\frac{d(C_{out}^i M(t))}{dt} = C_{in}^i F_{in}^{total} - C_{out}^i F_{out}^{total} \quad (6)$$

### **Blender**

The continuous blender is used to dampen the high-frequency variations resulted from the feeding system, and is a crucial unit operation in the pharmaceutical industry which has stringent quality constraints on the concentration uniformity in the products[24]. A variety of modeling techniques has been used to simulate the powder dynamics in the blender. DEM simulations have been used to understand the mechanism of mixing in a tubular blender, when considering different equipment geometry, material properties, and operating conditions[25-27]. Population balance models (PBM), combined with DEM studies, have been developed to predict blender dynamics, including concentration, relative standard deviation (RSD) and RTD[16,28]. A discrete element reduced order modeling (DE-ROM) method has been implemented to map the velocity trajectories within a continuous convective blender[29]. In addition, RTD modeling approaches have also been used to characterize the non-ideality in mixing within a continuous blender[30,6].

In this work, the blender is modeled as a two-level semi-empirical model. The first level is based on a CSTR in-series model, which is used to characterize RTD and obtain the mixing parameters.

The CSTR in-series model uses multiple perfect stirred tanks in series to simulate the mixing behaviors along the axial direct of the blender. After incorporating the delay time which represents the time that materials take to convectively travel the length of the blender while being mixed, the probability distribution function (PDF), used for representing the time distribution of particles, can be expressed by Equation 7.

$$E(t) = Unit\ Step[t - \tau_{delay}] \frac{(t - \tau_{delay})^{n-1} \exp\left(-\frac{t - \tau_{delay}}{\bar{\tau}}\right)}{(n-1)! \bar{\tau}^n} \quad (7)$$

where  $\tau_{delay}$  is the delay time;  $n$  is the number of tanks;  $\bar{\tau}$  is the mean residence time of one tank in the CSTR in-series model. Therefore, the blender's mean residence time ( $\tau_{blender}$ ) can be computed using Equation (8)

$$\tau_{blender} = \tau_{delay} + n\bar{\tau} \quad (8)$$

The second level is based on an axial dispersion model, and is used for mixing calculations as a function of time, since the axial dispersion model cannot be implement in gPROMS through the convolution algorithm. The dispersion model is expressed by Equation 9. The coefficients of the axial dispersion equation are calculated from their relationship to the CSTR-in-series model constants.

$$(n\bar{\tau}) \frac{\partial C_{out}^i}{\partial t} = \frac{1}{Pe} \frac{\partial^2 C_{out}^i}{\partial \xi^2} - \frac{\partial C_{out}^i}{\partial \xi}$$

$$Pe = n + \sqrt{8n + n^2}, \text{ if } n < 10$$

$$Pe = 2n - 1, \text{ if } n \geq 10$$

$$(9)$$

$$I.C. \quad C_{out}^i|_{t=0} = 0$$

$$B.C. \quad C_{out}^i|_{\xi=0} = C_{in}^i$$

$$\frac{\partial C_{out}^i}{\partial \xi} \Big|_{\xi=1} = 0$$

where  $C_{out}^i$  and  $C_{in}^i$  are the concentration of component  $i$  at outlet and inlet respectively;  $\xi$  is the dimensionless axial length ranging from 0 to 1;  $Pe$  is Peclet number.

The mass balance is modeled in the similar fashion with that of the co-mill (Equation 10), meaning that the accumulation inside of the blender reaches a steady state value asymptotically. From the accumulation and mass flow rate in, we can then compute a flow rate out of the system. This trend has been proven experimentally.[31]

$$\bar{\tau} \frac{dM(t)}{dt} + M(t) = M_{ss} \quad (10)$$

$$\frac{dM(t)}{dt} = F_{in}^{total} - F_{out}^{total}$$

In this two-level model, the model parameters, including  $\tau_{delay}$ ,  $n$ ,  $\bar{\tau}$ ,  $Pe$ , and  $M_{ss}$ , depend on the inlet flow rate and blade speed, and need to be regressed with experimental data.

### ***Tablet Press***

In tablet press equipment, materials from the previous unit are transferred via a feed frame to a series of dies. The shape of the dies and the fill depth of the materials determine the geometry and weight of the final tablet products. The dies can run continuously, during which time the materials within the dies are compressed by an upper and a lower punch into tablets, which are discharged afterwards[32]. A two-stage strategy is used by most of the tablet processes, namely pre-compression and main-compression[33]. Pre-compression is employed to release the entrapped air and reduce the porosity of the materials. This step will help prevent tablet defects such as cracks and laminations when tablets are made during the main-compression step[34].

Extensive research has been conducted to understand and model the compaction process. DEM simulation has been used to investigate the segregation behaviors during die filling[35] and the bonding interaction within compressed tablets[36,37]. Empirical models have also been developed to correlate the powder material properties (e.g. bulk density) and process parameters (e.g.

compression force, die geometry) with tablet attributes (e.g. weight, hardness). For example, Heckel's model uses a first-order equation to describe the relationship between powder density and compression pressure[38]. Kawakita model predicts the compression force with powder volume and initial porosity[39]. Kuentz and Leuenberger developed a model to predict tablet hardness based on tablet density[40].

In this work, the tablet press model is based on the work by Singh, Gernaey & Gani[41] and Singh, Ierapetritou & Ramachandran[42]. For simplicity, only the key equations are listed below. The readers are referred to the original papers for details.

The pre-compression and main compression pressure are modeled with equations adapted from Kawakita model[39], as in Equation 11 and 12.

$$CP_{pre} = \frac{V_0 - V_{pre}}{b_{pre}(V_0(\varepsilon_0 - 1) + V_{pre})} \quad (11)$$

$$CP_{main} = \frac{V_{pre} - V_{tablet}}{b_{main}(V_{pre}(\varepsilon_{main} - 1) + V_{tablet})} \quad (12)$$

where  $b_{pre}$  and  $b_{main}$  are Kawakita parameters for pre-compression and main compression respectively, with the unit in 1/MPa, and need to be obtained by experiments;  $V_0$  is the initial volume in the die;  $V_{pre}$  is the powder volume after pre-compression;  $V_{tablet}$  is the volume of tablet;  $\varepsilon_0$  and  $\varepsilon_{main}$  are the porosity of the material prior to pre-compression and prior to main-compression respectively.

The tablet hardness is predicted with the model by Kuentz and Leuenberger[40] (Equation 13)

$$H = H_{max}(1 - e^{\rho_r - \rho_{rc} + \lambda_H}) \quad (13)$$

where  $H_{max}$  is the maximum hardness, which represents the tablet hardness at the zero porosity, and need to be obtained from experiments;  $\rho_{rc}$  is the relative critical density obtained from

experiments;  $\rho_r$  is the relative density, which is defined as the ratio of solid volume over tablet volume (Equation 14);  $\lambda_H$  is an intermediate variable defined with Equation 15.

$$\rho_r = \frac{(1 - \varepsilon_0)V_0}{V_{tablet}} \quad (14)$$

$$\lambda_H = \log \left( \frac{1 - \rho_r}{1 - \rho_{rc}} \right) \quad (15)$$

## 1.2 Process analysis and optimization for deterministic systems

Deterministic simulations are widely used as a simplified representation of a real-life systems. On the basis of a well-developed deterministic process model, risk assessment can be conducted by using a variety process analysis techniques. In this work, we mainly focus on two types of approaches: (1) global sensitivity analysis; and (2) feasibility analysis. Global sensitivity analysis investigates the process in a “forward” manner: within certain ranges of the inputs, understand how these inputs can contribute to the variability in the output. Such analysis can be used to prioritize the inputs by their influence on the outputs of interest, and reduce the dimension of the simulation model by fixing the inputs to their nominal values whose effects are negligible on the outputs. On the other hand, feasibility analysis evaluates the process in a “backward” manner: given specified ranges (i.e. process constraints) on the output, find the design space of the inputs within which a process is guaranteed to be feasible. For pharmaceutical processes, these two types of analysis can be used to reduce the problem definition, identify bottlenecks of the process, and facilitate the development of control strategies. A detailed review of the methods for global sensitivity analysis and feasibility is given in Chapter 2. Additionally, to better solve the feasibility analysis problem of computationally expensive simulations, we propose a novel surrogate-based adaptive sampling approach, which is discussed in Chapter 4.

Process models have also been used to facilitate the optimization of a process. For pharmaceutical manufacturing, a variety of optimization techniques have been adopted to improve the product qualities, decrease the environmental impacts, enhance the process robustness, and reduce the operating costs. A review of the applications of optimization in pharmaceutical processes is given in Chapter 3.

For a comprehensive understanding of a pharmaceutical process, we propose a framework consisting of global sensitivity analysis, feasibility analysis, and process optimization. This framework is demonstrated with a direct compaction process, and details are given in Chapter 5.

### 1.3 Process analysis and optimization for stochastic systems.

Process variabilities have always been a critical issue in pharmaceutical processes. From the modeling point of view, a direct way to incorporate variabilities is by using stochastic simulations. A stochastic simulation may give different observed values when it is run multiple times at the same sample point. In such cases, the stochasticity can cause uncertainties to the evaluation of the system, which greatly increases the difficulties of process analysis and optimization.

The difficulties of using stochastic systems are multi-fold. First, the randomness in the system can cause the derivative information of the system to be unreliable or hard to approximate. In this case, traditional derivative-based optimization methods cannot not be directly applied to process optimization problems. Second, when the simulation is computationally expensive to evaluate, it is usually required to conduct process analysis and optimization within a limited sampling budget. Therefore, efficient approaches need to be developed to extract process information. Third, the inherent noise level can vary depending on process inputs (i.e., heteroscedastic noise variances) which need to be considered when performing process analysis and optimization.

To address the difficulties of using stochastic systems, we have developed and compared three surrogate-based feasibility analysis methods, with details mentioned in Chapter 6. A novel “one-stage” optimization algorithm is developed for unconstrained optimization problems of stochastic systems, which is described in Chapter 7. A feasibility-enhanced Kriging-based method is developed to solve stochastically constrained optimization problems, of which details are discussed in Chapter 8.

Finally, the conclusions and suggested future work are given in Chapter 9.

## II. Literature Review

## 2 Global Sensitivity and Feasibility Analysis of Continuous Pharmaceutical Manufacturing Processes

### 2.1 Introduction

Mathematical models have been used to investigate the input-output relationship for pharmaceutical manufacturing processes. With the increasing process knowledge that is obtained both in-line and off-line, the descriptive models also get more complex. Therefore, we need to use simulation-based mathematical methods to systematically and efficiently analyze the process, and use the analysis results to guide the development of control strategies and identify process bottlenecks. Sensitivity analysis and feasibility analysis are two powerful approaches that can facilitate this process analysis.

Sensitivity analysis investigates how variability in the model inputs contribute to the variations in model outputs[43]. It can be used to support the initial risk assessment of the process by estimating the relative impact of variability in process parameters and material properties on the quality attributes[1]. Sensitivity analysis is usually implemented to prioritize the input factors by quantifying the influence on the output and identifying the subset of most important input factors while screening out the less important input factors whose effects can be considered negligible. Therefore, it is usually used to reduce the dimension of the model and simplify it for further feasibility analysis or optimization purpose. Generally, there are two categories of sensitivity analysis: local and global methods. Local methods reflect the sensitivity information around one base case (nominal point). On the other hand, global methods reveal the sensitivity information over the entire input space. In this chapter, we would focus on the global methods since it's mostly used by the pharmaceutical process modeling community [15,14]. There are numerous global

sensitivity analysis methods available, and the choice of a specific method usually depends on the sampling budget that a user can afford.

Feasibility and flexibility analysis is a mathematical approach to quantify and characterize the design space of a process[44]. The concept of feasibility and flexibility analysis was initially proposed by Grossmann and Morari[45] from the process systems engineering community, and has evolved to many different applications over the years[44]. The objective of this analysis is to quantify the maximum limit of variations in the input that a process can tolerate while retaining process robustness. For a pharmaceutical process, feasibility analysis can be used to indicate the feasible region in which the operation conditions need to be controlled, and/or the ranges where material properties can be varying, in order to ensure product qualities and process safety. Based on exhaustive sampling, process models can be directly used to identify the design space[46,47]. However, for computationally expensive models, more efficient methods such as surrogate-based adaptive sampling approaches [48] are more favorable to predict the design space. For process models with many input factors, an additional difficulty for the feasibility analysis is to visualize the design space in the high-dimensional space.

Sensitivity analysis and feasibility analysis are two key aspects of a systems approach to extract process knowledge, and can facilitate the risk assessments and failure mode analysis for both individual unit operations and integrated processes. The results can be used as guidance for manufactures to foresee and address hidden process problems during the development of a CM process. Additionally, they can also help users identify the areas where process models need to be further developed[14]. On the other hand, sensitivity analysis and feasibility analysis are also beneficial to formulate and solve process optimization problems. For example, global sensitivity analysis can be applied to reduce the number of decision variables; surrogate-based feasibility analysis can be seen as a way of addressing black-box process constraints to the optimization

problems[49]. Therefore, it is important to consider sensitivity analysis and feasibility analysis as computer-aided process analysis tools when developing a continuous process.

## 2.2 Global Sensitivity Analysis

Global sensitivity analysis aims to evaluate how sensitive the model outputs are in response to the changes in the model inputs[50]. For pharmaceutical process models, the input factors usually refer to the operation conditions (e.g. flow rate, blender blade speed), raw material properties (e.g. bulk density). The output variables are the responses of the model (e.g. mean residence time, tablet properties) that are determined by the input factors. The results can help identify and prioritize the most influential input factors that contribute most to the output variability, and also find the model input that are insignificant to the output. Those less significant inputs can be fixed at their nominal values for further use because they do not have much effects on the outputs of interest. In addition, the global sensitivity analysis can also reflect if (and which) factors interact with each other, and how these interactions can affect the outputs. Note that before conducting the sensitivity analysis, users need to first specify the distribution and ranges of the inputs. Common distributions include the uniform distribution and the normal distribution. It is important to carefully choose the distributions and ranges of the inputs based on the actual process settings because such choices may affect the results from sensitivity analysis.

Generally, there are four categories of global sensitivity analysis methods, including: (1) screening methods; (2) regression-based methods; (3) variance-based methods; and (4) metamodel-based methods[51]. In the following subsections, we will discuss each method by covering (1) basics of the methods, (2) calculation and interpretation of the sensitivity metrics, and (3) required sampling cost, followed by highlighting the commonly used approaches to visualize the results of this analysis.

## 2.2.1 Methods

### 2.2.1.1 Screening methods

A screening method is effective to find the few most influential factors of a model which may have many inputs, by using a relatively small sampling budget[43]. A mostly used screening method is the Morris method. It is based on one-at-a-time (OAT) designs, in which only one factor changes values between consecutive simulations. Morris method considers wide ranges of variations in the input and uses an average of local measures to provide global sensitivity information. Morris method is especially useful when there are a large number of inputs (e.g. tens of factors), and/or when the model is very costly.

The Morris method depends on the calculation of the elementary effects (EE). For a selected base point, EE of the  $i^{\text{th}}$  factor is defined with Equation 16,

$$EE_i = \frac{Y(x_1, \dots, x_{i-1}, x_i + \Delta_i, x_{i+1}, \dots, x_k) - Y(x_1, \dots, x_k)}{\Delta_i} \quad (16)$$

where  $k$  represents the number of inputs;  $(x_1, \dots, x_k)$  is the selected base point;  $\Delta_i$  is the step change in the  $i^{\text{th}}$  input factor;  $Y(\cdot)$  is the model output.

The original version of the Morris method[43] is performed by sampling  $r$  trajectories, of which the input factors are moved OAT on a grid of levels covering the entire input space. We use a 3-dimensional example to illustrate this trajectory-based method, which is shown in Figure 2 (adapted from Campolongo, Saltelli et al.[52]). Along the trajectory (bold dashed line in Figure 2), each input is changed by the same step size  $\Delta$ . In this example,  $\Delta = 0.25$ . The EE effects are thus calculated sequentially. For example, the EE for the 1<sup>st</sup> input is calculated as  $EE_1 = \frac{Y(x^{(1)}) - Y(x^{(0)})}{\Delta}$ ; the EE for the 2<sup>nd</sup> input is  $EE_2 = \frac{Y(x^{(2)}) - Y(x^{(1)})}{\Delta}$ , etc. In order to more accurately reflect the global

sensitivity information, it is suggested that trajectories should be generated in a way so that their spread in the input space is maximized[43].

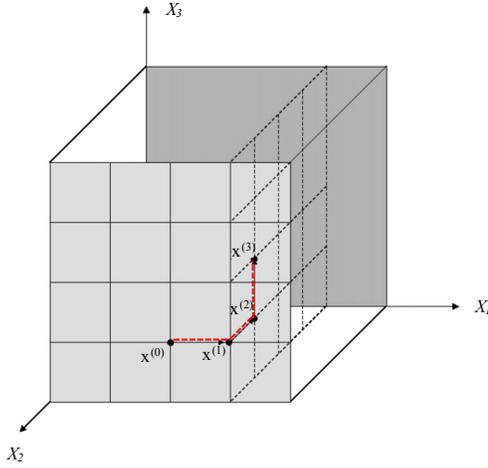


Figure 2. Example of the trajectory in a 3D input space.

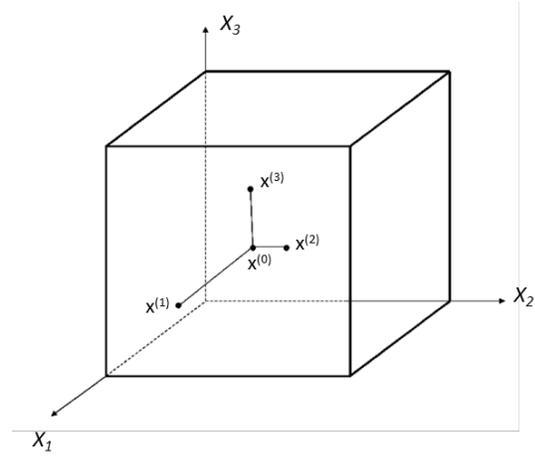


Figure 3. Example of the radial design in a 3D input space

A more efficient and accurate design was later proposed by Campolongo, Saltelli et al.[52] We also use a 3D model as an example, which is shown in Figure 3 (adapted from Campolongo, Saltelli et al.[52]). A “radial design” calculates each  $EE_{i,i \in \{1,2,3\}}$  by using the current visiting point  $(\mathbf{x}^{(i), i \in \{1,2,3\}})$  with the base point  $(\mathbf{x}^{(0)})$ . Sobol’ quasi-random numbers[53] are used to generate radial sample points. Note that for both designs (trajectory-based design and radial design), the value of  $\Delta$  must always represent the sampling step with the range  $[0, 1]$ . For non-uniform distributions (e.g. normal distribution),  $\Delta$  should represent the variation in the quantiles of the input factors[43].

Based on the calculation of EE, three sensitivity metrics can be calculated (Equation 17).  $\mu_i$  is the average of  $EE_i$ , and indicates the individual sensitivity information.  $\sigma_i^2$  is the variance of  $EE_i$ , and

reflects the nonlinearity or the interaction in the  $i^{\text{th}}$  input.  $\mu_i$  and  $\sigma_i^2$  are usually used simultaneously to show the sensitivity information. This is to account for the cases when a significant input has  $EE_i$  of different signs. In such situations, positive and negative values of  $EE_i$  may cancel out each other, and cause  $\mu_i$  to be very small (close to zero). However,  $\sigma_i^2$  will still be large, and can thus be used to identify this input factor as significant. On the other hand, the third metric  $\mu_i^*$  can be more convenient. Since  $\mu_i^*$  is calculated as the average of the absolute values of  $EE_i$ , it is not vulnerable when the EE has both positive and negative values. Saltelli, Ratto et al.[43] suggested use all three metrics to extract the maximum sensitivity information.

$$\begin{aligned}\mu_i &= \frac{1}{r} \sum_{j=1}^r EE_i^j \\ \sigma_i^2 &= \frac{1}{r-1} \sum_{j=1}^r (EE_i^j - \mu_i)^2 \\ \mu_i^* &= \frac{1}{r} \sum_{j=1}^r |EE_i^j|\end{aligned}\tag{17}$$

The total sampling cost of the Morris method is  $r(k+1)$ , where  $r$  equals to the number of trajectories (or number of radial base points). This is because for each trajectory (or base point), the simulation needs to be evaluated at  $(k+1)$  different locations. Iooss and Lemaître[51] recommended that  $r$  should be selected between 2 and 10.

After conducting the Morris method, the input factors with large values of  $\mu_i$  and/or  $\sigma_i^2$ , also large values of  $\mu_i^*$  are considered to be significant. In practice, if a input factor has a sensitivity metric with values smaller than 10% of the largest of this metric, then this input can be treated as not significant[43,54]. Note that Morris method is only considered to be a qualitative approach. This is because it can only rank the input factors in the order of importance, but does not quantify how much an input factor is more important than others[50].

### 2.2.1.2 Regression-based methods

Regression-based methods are based on the analysis of linear regression models which explain the output variable given the values of inputs[50]. A typical method of this category is the Partial Rank Correlation Coefficient (PRCC), which is an extension of the Partial Correlation Coefficient (PCC).

For a model with multiple inputs, PCC is used to determine if an input is strongly linearly correlated to the output when all the linear effects of other variables are removed[55]. PCC can be calculated with Equation 18:

$$\begin{aligned}\hat{x}_i &= c_0 + \sum_{j=1, j \neq i}^k c_j x_j \\ \hat{y} &= b_0 + \sum_{j=1, j \neq i}^k b_j x_j \\ PCC &= CC(x_i - \hat{x}_i, y - \hat{y})\end{aligned}\tag{18}$$

where  $k$  is the number of input factors. In Equation 18, the  $i^{\text{th}}$  input factor is expressed ( $\hat{x}_i$ ) with a linear regression model considering all the other input factors ( $x_{j, j \neq i}$ ). The output variable is also expressed ( $\hat{y}$ ) with a linear model using the same set of input factors ( $x_{j, j \neq i}$ ). Then, PCC is calculated as the correlation coefficient between the two residual terms:  $(x_i - \hat{x}_i)$  and  $(y - \hat{y})$ .

When the relationship between the input and the output is nonlinear but monotonic, it can be transferred to a linear relationship by applying rank transformations to the data[55]. This means that the data values are replaced with their ranks when building the linear models in Equation 18. This procedure gives the values of PRCC. Marino, Hogue et al.[55] commented that PRCC is a reliable sensitivity metric for nonlinear but monotonic relationships between input and output as long as there is little to no correlation among the inputs.

PRCC is a standardized measure whose value is between -1 and 1. If PRCC is close to -1 (or 1), it means the input has a strong negative (or positive) effect on the output. If PRCC is close to 0, it indicates that the input and output are independent. Therefore this input is not significant to the output[51]. Latin Hypercube Sampling (LHS)[56] is usually adopted to sample the data points[55]. The number of sample points should be at least larger than  $k$  for the purpose of constructing the linear regression models. Iooss and Lemaître[51] recommended the sampling budget between  $2k$  to  $100k$ .

### 2.2.1.3 Variance-based methods

Variance-based methods consider the decomposition of the variance of the output (i.e.  $V(\mathbf{y})$ ) into several components including individual inputs and the interactions between inputs[57]. Under the assumption that the input factors are independent, the decomposition of  $V(\mathbf{y})$  can be expressed with Equation 19,

$$V(\mathbf{y}) = \sum_{i=1}^k V_i + \sum_{1 \leq i < j \leq k} V_{i,j} + \dots + V_{1,2,\dots,k} \quad (19)$$

where  $V_i$  is the partial variance term that is solely due to  $x_i$ ;  $V_{i,j}$  is partial variance term due to the interaction between  $x_i$  and  $x_j$ ;  $V_{1,2,\dots,k}$  is the partial variance term due to the interaction of  $x_1, \dots, x_k$ .

Based on such a variance decomposition, different sensitivity measures can be defined. Common measures are shown in Equation 20 to 22.

$$S_i = \frac{V_i}{V(\mathbf{y})} \quad (20)$$

$$S_{ij} = \frac{V_{i,j}}{V(\mathbf{y})} \quad (21)$$

$$S_{Ti} = \frac{V_i + \sum_{j \neq i} V_{i,j} + \dots + V_{1,2,\dots,k}}{V(\mathbf{y})} = 1 - \frac{V_{\sim i}}{V(\mathbf{y})} \quad (22)$$

$S_i$  is called “first-order sensitivity index” for the input factor  $x_i$ , which measures the main effect of  $x_i$  on the output.  $S_{ij}$  is “second-order sensitivity index”, which is the interaction effects of  $x_i$  and  $x_j$  on the model output;  $S_{Ti}$  is the “total effect index”, which accounts for the main effect and all the higher-order interaction effects including  $x_i$ .

### **Sobol' method**

Sobol' method uses Monte Carlo techniques to compute the indices of  $S_i$  and  $S_{Ti}$ , the expressions of these two indices[43] are shown in Equation 23 and 24,

$$S_i = \frac{V_{X_i}(E_{X_{\sim i}}(Y|X_i))}{V(Y)} \quad (23)$$

$$S_{Ti} = 1 - \frac{V_{X_{\sim i}}(E_{X_i}(Y|\mathbf{X}_{\sim i}))}{V(Y)} = \frac{E_{X_{\sim i}}(V_{X_i}(Y|\mathbf{X}_{\sim i}))}{V(Y)} \quad (24)$$

where  $E(\cdot)$  is the expected value;  $\mathbf{X}_{\sim i}$  represents all possible combination of input values with  $X_i$  being fixed. In Equation 24,  $S_{Ti}$  is calculated as the sum of all terms (i.e. “1”) subtracted by the

terms that do not include  $x_i$  (i.e.  $\frac{V_{X_{\sim i}}(E_{X_i}(Y|\mathbf{X}_{\sim i}))}{V(Y)}$ ).

Saltelli, Annoni et al.[58] summarized several approaches to calculating  $S_i$  and  $S_{Ti}$  with Sobol' method, and recommended using the following expressions for efficiency and accuracy.

$$S_i = \frac{\frac{1}{N} \sum_{j=1}^N f(\mathbf{B})_j \left( f(\mathbf{A}_B^{(i)})_j - f(\mathbf{A})_j \right)}{V(Y)} \quad (25)$$

$$S_{Ti} = \frac{\frac{1}{2N} \sum_{j=1}^N \left( f(\mathbf{A})_j - f(\mathbf{A}_B^{(i)})_j \right)^2}{V(Y)} \quad (26)$$

In these expressions,  $\mathbf{A}$  and  $\mathbf{B}$  are two matrices, each having N rows of different sample points.  $\mathbf{A}_B^{(i)}$  is a matrix in which the  $i$ th column of  $\mathbf{A}$  is substituted by the  $i$ th column of  $\mathbf{B}$ .

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1k} \\ a_{21} & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ a_{N1} & a_{N2} & \cdots & a_{Nk} \end{bmatrix}, \mathbf{B} = \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1k} \\ b_{21} & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ b_{N1} & b_{N2} & \cdots & b_{Nk} \end{bmatrix}, \mathbf{A}_B^{(i)} = \begin{bmatrix} a_{11} & b_{1i} & \cdots & a_{1k} \\ a_{21} & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ a_{N1} & b_{Ni} & \cdots & a_{Nk} \end{bmatrix} \quad (27)$$

The sample points in matrices  $\mathbf{A}$  and  $\mathbf{B}$  are generated by Sobol' quasi-random numbers[53]. From Equation 27, we can notice that a total number of  $N(k + 2)$  sample points are required. In order to get accurate estimates of  $S_i$  and  $S_{Ti}$ ,  $N$  should be at least 500[58]. Iooss and Lemaître[51] recommended the total sampling budget between  $100k$  to  $1000k$ .

### ***FAST and eFAST method***

Fourier Amplitude Sensitivity Test (FAST)[59] method is based on the idea of converting the calculation of  $E(y)$  and  $V(y)$  to one-dimensional integrals via the construction of a proper space-filling curve (Equation 28):

$$x_i = G_i(\sin \omega_i s) \quad (28)$$

where  $G_i$  are transformation functions for  $i = 1, \dots, k$ ;  $\omega_i$  are integers;  $s \in (-\pi, \pi)$  is a scalar variable. Then,  $E(y)$  can be approximated with Equation 29,

$$E(Y) \doteq \frac{1}{2\pi} \int_{-\pi}^{\pi} f(s) ds \quad (29)$$

where  $f(s) = f(G_1(\sin \omega_1 s), \dots, G_k(\sin \omega_k s))$ .

Based on Fourier series properties[60],  $V(y)$  can be approximated with Equation 30.

$$V(y) \doteq \frac{1}{2\pi} \int_{-\pi}^{\pi} f^2(s) ds - [E(Y)]^2 \approx 2 \sum_{j=1}^M (A_j^2 + B_j^2) \quad (30)$$

where  $A_j$  and  $B_j$  are defined as follows:

$$A_j = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(s) \cos(js) ds \quad (31)$$

$$B_j = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(s) \sin(js) ds \quad (32)$$

Furthermore,  $V_i$  can be approximated as follows:

$$V_i \doteq 2 \sum_{p=1}^M (A_{p\omega_i}^2 + B_{p\omega_i}^2) \quad (33)$$

Extended FAST (eFAST) method[59] further approximated  $V_{\sim i}$  as follows:

$$V_{\sim i} \doteq 2 \sum_{p=1}^M (A_{p\omega_{\sim i}}^2 + B_{p\omega_{\sim i}}^2) \quad (34)$$

where  $\omega_i$  is the integer (associated with  $G_i$ ) for the factor  $X_i$ ;  $\omega_{\sim i}$  is a set of almost identical integers (different from  $\omega_i$ ) that are assigned to all the remaining factors but  $X_i$ ;  $M$  is the maximum harmonic that's considered, usually taken to be 4 or 6[59].

Therefore, the sensitivity metrics  $S_i$  and  $S_{Ti}$  from Equation 20 and 24 can be approximated with the following expressions:

$$S_i \doteq \frac{\sum_{p=1}^M (A_{p\omega_i}^2 + B_{p\omega_i}^2)}{\sum_{j=1}^M (A_j^2 + B_j^2)} \quad (35)$$

$$S_{Ti} \doteq 1 - \frac{\sum_{p=1}^M (A_{p\omega_{\sim i}}^2 + B_{p\omega_{\sim i}}^2)}{\sum_{j=1}^M (A_j^2 + B_j^2)} \quad (36)$$

The minimum number of sample points should be  $(2M\omega_{max} + 1)$ , where  $\omega_{max}$  is the maximum values in the set of  $\omega_i$ . An automated algorithm of selecting  $\omega_i$  can be found in Section 8.4.4 of Saltelli, Chan et al.[59] Iooss and Lemaître[51] recommended the total sampling budget between  $100k$  to  $1000k$ .

For the variance-based methods,  $S_i$  and  $S_{Ti}$  are within the range of  $[0, 1]$ . Those input factors with sensitivity metrics' values close to 1 are significant to the output variable. The necessary and sufficient condition for an input factor to be non-influential is that  $S_{Ti} = 0$ . Note that we always have  $S_i \leq S_{Ti}$ , and the difference between  $S_i$  and  $S_{Ti}$  indicates the interaction effects on the model output.

#### 2.2.1.4 Metamodel-based methods

The metamodel-based method aims to build a surrogate model to approximate the original simulation. This surrogate should have good prediction accuracy and low computational costs, so that it can be used to conduct sensitivity analysis which may require huge numbers of sample points. In this subsection, we focus on the application of random sampling high dimensional model representation (RS-HDMR) method[61].

The HDMR model maps the relationship between the inputs and output with the following equation:

$$f(x) = f_0 + \sum_{i=1}^k f_i(x_i) + \sum_{1 \leq i < j \leq k} f_{ij}(x_i, x_j) + \dots + f_{12\dots k}(x_1, x_2, \dots, x_k) \quad (37)$$

where  $f_0$  is a constant representing the mean effect (zeroth-order term);  $f_i$  denotes the first-order term, that is the effect of  $x_i$  acting independently on the output;  $f_{ij}$  is the second-order term considering the interaction effects of  $x_i$  and  $x_j$  on the model output. The HDMR model is computationally efficient if higher-order interactions of the inputs are not significant, and can be neglected. In most cases, a HDMR model with up to 2<sup>nd</sup> order terms is sufficient to capture the model behaviors[62].

A RS-HDMR model can be constructed by using  $N$  randomly sampled points over the entire input space. A scaling procedure is conducted for all the input factors so that  $0 \leq x_i \leq 1$  for all  $i$ . Then, the zeroth-term  $f_0$  in Equation 37 is approximated with the average of  $f(x)$  for all sample points:

$$f_0 \approx \frac{1}{N} \sum_{s=1}^N f(x^{(s)}) \quad (38)$$

By using analytical basis functions (such as orthonormal polynomials), the first-order and second-order component functions can be approximated as follows:

$$f_i(x_i) \approx \sum_{r=1}^m \alpha_r^i \varphi_r(x_i) \quad (39)$$

$$f_{ij}(x_i, x_j) \approx \sum_{p=1}^l \sum_{q=1}^{l'} \beta_{pq}^{ij} \varphi_p(x_i) \varphi_q(x_j) \quad (40)$$

where  $m, l, l'$  denote the orders of polynomial expansion, and can be chosen based on the algorithm in Ziehn and Tomlin[61];  $\varphi_r, \varphi_p, \varphi_q$  are orthonormal basis functions;  $\alpha_r^i$  and  $\beta_{pq}^{ij}$  are model coefficients, and can be determined with the following equations:

$$\alpha_r^i \approx \frac{1}{N} \sum_{s=1}^N f(x^{(s)}) \varphi_r(x_i^{(s)}) \quad (41)$$

$$\beta_{pq}^{ij} \approx \frac{1}{N} \sum_{s=1}^N f(x^{(s)}) \varphi_p(x_i^{(s)}) \varphi_q(x_j^{(s)}) \quad (42)$$

With the constructed RS-HDMR model, the partial variances in Equation 19 can be obtained as follows:

$$V(y) = \int f^2(x) dx - f_0 \quad (43)$$

$$V_i = \int_0^1 f_i^2(x_i) dx_i \quad (44)$$

$$V_{i,j} = \int_0^1 \int_0^1 f_{ij}^2(x_i, x_j) dx_i dx_j \quad (45)$$

Therefore, the sensitivity metrics  $S_i$  and  $S_{ij}$  can be calculated by their definitions in Equations 20 and 21. LHS can be used to generate the sample points to build the RS-HDMR[14]. Iooss and Lemaître[51] suggested the number of sample points should be between  $10k$  and  $100k$ .

### 2.3 Feasibility Analysis

Feasibility and flexibility analysis quantifies the capabilities of a process to remain feasible over a range of uncertain parameters[44]. This concept is based on the feasibility function, which

describes whether for a fixed value of uncertainty parameter  $\theta$  a process can meet all constraints  $f_j \leq 0$  by adjusting its control variable  $z$ . This can be evaluated by choosing  $z$  to minimize the maximum value of  $f_j$ . This leads to the definition of feasibility function as follows:

$$\begin{aligned}\psi(d, \theta) &= \min_z \max_{j \in J} \{f_j(d, z, \theta)\} \\ z \in Z &= \{z: z^L \leq z \leq z^U\} \\ \theta \in T &= \{\theta: \theta^L \leq \theta \leq \theta^U\}\end{aligned}\tag{46}$$

where  $\psi(d, \theta)$  denotes the feasibility function;  $d$  represents the design variables, which are usually constant parameters for a fixed process design. If  $\psi(d, \theta) \leq 0$ , then the process is feasible over the whole input space. If  $\psi(d, \theta) > 0$ , then we know there exists at least one constraint that's violated no matter how we adjust the control variable  $z$ .

If we treat all the input variables as uncertainty parameters, (i.e. no control variables  $z$ ), then the feasibility function is simplified to the form in Equation 47.

$$\psi(d, \theta) = \max_{j \in J} \{f_j(d, \theta)\}\tag{47}$$

In the following subsections, we will first overview the methods that are developed to solve the feasibility analysis problems for pharmaceutical processes, then illustrate the common measures that are taken to visualize the design space. Finally, we will discuss some extensions to the traditional feasibility analysis problems.

### 2.3.1 Methods

#### 2.3.1.1 *Traditional simulation-based approach*

When the simulation model is computationally fast, it can be directly used for feasibility analysis. For example, with the use of a response surface model, Huang, Kaul et al established the design space consisting of three parameters (water amount, wet massing time, and lubricant time) in order

to get desired tablet blend flow[63]. Prpich, am Ende et al. used two fundamental models to determine the design space of air temperature and air pressure for a tablet film coating process[47], with the results verified with experimental studies. Brueggemeier, Reiff et al. applied several process models to the characterization of design space for a drug API step[46].

### *2.3.1.2 Surrogate-based adaptive sampling approach*

As the process model gets more complex, it can be computationally infeasible to directly use the model for feasibility analysis. In such cases, we need to seek for more intelligent ways to characterize the design space. Surrogate-based adaptive sampling approaches can be more appealing.

This approach relies on building a surrogate model to efficiently approximate the feasibility function of the computationally costly simulation model. This surrogate model is then updated and gradually gets improved at the adaptive sampling stage, during which the new sample points are selected in a way that they can quickly get close to the feasible region boundary of the original simulation. It is reasonable that if we can correctly identify the feasible region boundary, then we will know where the feasible region is. Therefore, the adaptive sampling is efficient in that, instead of doing exhaustive sampling, it intelligently searches for sample locations that are most promising to provide information on the feasible region boundary. Below, we describe a Kriging-based method that is used in Ref. [64,65,48].

### ***Kriging-based adaptive sampling approach***

Kriging model[66] uses a stochastic process to model an unknown function  $y$  (Equation 48),

$$y(x) = \mu + \varepsilon(x) \tag{48}$$

where  $\mu$  is a constant representing the overall surface mean;  $\varepsilon(x)$  is a realization of a random variable with mean 0.  $\varepsilon(x)$  is assumed to have spatial correlations, which means that if  $x$  is close to  $x'$ , then  $\varepsilon(x)$  and  $\varepsilon(x')$  will tend to be similar. The covariance between  $\varepsilon(x)$  and  $\varepsilon(x')$  can be expressed as  $\Sigma_\varepsilon(x, x') = \tau^2 R_\varepsilon(|x - x'|)$ , where  $\tau^2$  is the variance of  $\varepsilon$  for all  $x$ .  $R_\varepsilon$  is the spatial correlation that can be modeled with different correlation kernel functions, of which the commonly used are listed in Table 1.

Table 1. Correlation kernel functions

Name	Form
Exponential	$\exp(-\theta_j  d_j )$
Gaussian	$\exp(-\theta_j  d_j ^2)$
Linear	$\max\{0, 1 - \theta_j  d_j \}$

With  $n$  sample points, the unbiased Kriging predictor at an unsampled point  $x^*$  can be derived as follows:

$$\hat{y}(x^*) = \mu + \Sigma_\varepsilon(x^*, \mathbf{x})^T [\Sigma_\varepsilon]^{-1} (\mathbf{y} - \mu \mathbf{1}_n) \quad (49)$$

where  $\Sigma_\varepsilon$  is the  $n$ -by- $n$  covariance matrix for the sample points,  $\Sigma_\varepsilon(x^*, \mathbf{x})$  is the  $n$ -by-1 vector of covariance between  $x^*$  and the  $n$  sample points.

A benefit of using Kriging is that it can provide the estimated prediction uncertainty (estimated mean squared error), which can be derived with the following equation:

$$\widehat{s^2}(x^*) = \tau^2 - \Sigma_\varepsilon(x^*, \mathbf{x})^T [\Sigma_\varepsilon]^{-1} \Sigma_\varepsilon(x^*, \mathbf{x}) + \left( \frac{(1 - \mathbf{1}_n^T [\Sigma_\varepsilon]^{-1} \Sigma_\varepsilon(x^*, \mathbf{x}))^2}{\mathbf{1}_n^T [\Sigma_\varepsilon]^{-1} \mathbf{1}_n} \right) \quad (50)$$

The model parameters  $(\mu, \tau^2, \theta_j)$  of Kriging can be obtained by maximizing the log-likelihood function.

Based on the Kriging surrogate model, Boukouvala and Ierapetritou[64] proposed an adaptive sampling method, which is based on maximizing a modified Expected Improvement (EI) function, in order to search for new sample points. The modified EI function is shown in Equation 51

$$EI_{feas}(x) = s \cdot \phi\left(\frac{0-y}{s}\right) = s \cdot \frac{1}{\sqrt{2\pi}} e^{-0.5\left(\frac{y^2}{s^2}\right)} \quad (51)$$

where  $s$  is the standard error of the prediction from Equation 50;  $y$  is the Kriging predictor from Equation 49;  $\phi(\cdot)$  is the standard normal density function. To understand how this function can be beneficial for feasibility analysis, we can take the partial derivative of  $EI_{feas}$  with respect to  $y$  and  $s$ :

$$\frac{\partial EI_{feas}}{\partial y} = -\frac{1}{\sqrt{2\pi}} \left( \frac{e^{-\frac{0.5y^2}{s^2}} y}{s} \right) \quad (52)$$

$$\frac{\partial EI_{feas}}{\partial s} = \frac{1}{\sqrt{2\pi}} \left( e^{-\frac{0.5y^2}{s^2}} + \frac{e^{-\frac{0.5y^2}{s^2}} y^2}{s^2} \right) > 0$$

In Equation 52, for the term  $\partial EI_{feas}/\partial y$ , because  $s$  is non-negative by definition,  $\partial EI_{feas}/\partial y$  will always have opposite signs with  $y$ . Therefore, when  $y$  is positive,  $EI_{feas}$  will increase if  $y$  decreases and gets close to zero. Similarly, when  $y$  is negative,  $EI_{feas}$  will increase if  $y$  increases and gets close to zero. In other words, by maximizing  $EI_{feas}$ , it will lead the search direction to the regions where Kriging predictor is close to zero, which is the feasible region boundary of the surrogate. On the other hand, it can be observed that  $\partial EI_{feas}/\partial s$  is always positive, meaning that  $EI_{feas}$  will increase as prediction uncertainty gets higher. Therefore, by maximizing  $EI_{feas}$ , it is also favored to sample points in the less explored regions. Also, we should notice that since prediction uncertainty  $s$  is always zero at sampled points,  $EI_{feas}$  is also zero for sampled points. Therefore, when maximizing  $EI_{feas}$ , it is less likely that the sample points converge to a certain local area. This

feature is beneficial to explore all the feasible region boundaries when the feasible region has complex landscapes or has multiple discrete sub-regions.

In summary, by maximizing  $EI_{feas}$ , the adaptive sampling strategy will keep a balance between sampling points near the feasible region boundary of the surrogate (local search), and sampling points in the less explored regions (global search). This is similar with the idea of surrogate-based global optimization methods (e.g. the Efficient Global Optimization (EGO) algorithm[67]). Only the objective is different: surrogate-based global optimization algorithms try to find the global optimum, which is usually a single point; while surrogate-based feasibility analysis methods attempt to identify the feasible region boundary, which is usually a complex surface.

The algorithm of the Kriging-based adaptive sampling approach[65] is shown in Figure 4. We first use a space-filling design (e.g. LHS) to find the initial sample points. A rule of thumb is to use at least  $10k$  LHS sample points to build the initial Kriging surrogate, where  $k$  is the dimension of the problem. This is followed by a model selection step which is used to find the correlation kernel function that gives the best surrogate prediction accuracy. Then, the Kriging surrogate model is constructed. By maximizing the  $EI_{feas}$  function, adaptive sampling is performed to find the next sample location, at which the original simulation is called and obtain the new sample point. The surrogate is then updated with the sample point. This adaptive sampling step is performed iteratively until it uses up all the sampling budget. Then, the final Kriging surrogate is used to predict the feasible region of the original simulation.

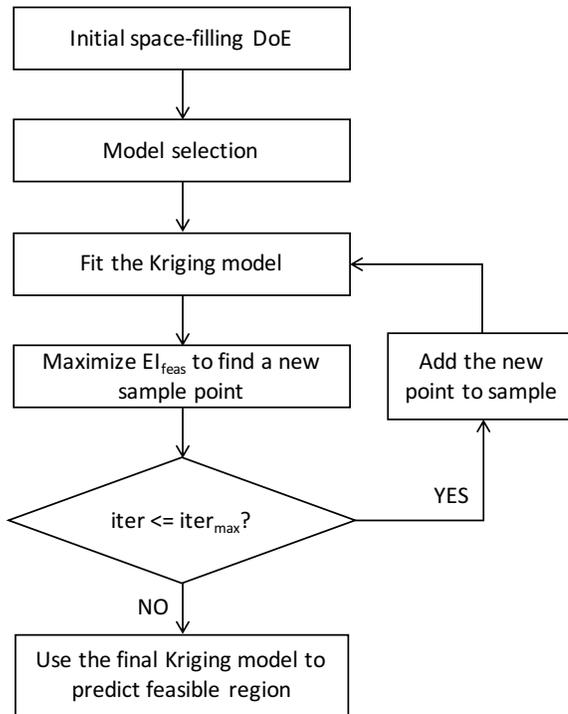


Figure 4. Algorithm of the Kriging-based adaptive sampling approach

### ***Extensions***

Several extensions to the surrogate-based adaptive sampling approaches have been developed to account for more complex cases. For example, Rogers and Ierapetritou[65] proposed dynamic surrogate-based feasibility analysis in order to investigate the change of design space with respect to time. Rogers and Ierapetritou[48] further applied the surrogate-based methods to calculating flexibility index and stochastic flexibility index, which are quantitative measures of the process flexibility when the uncertainties in the have a uniform distribution (using flexibility index) or non-uniform distribution (using stochastic flexibility index). In addition, new flexibility metrics have been developed to quantify the flexibility in complex cases. Lai and Hui[68] defined a “volumetric flexibility index” as the hyper-volume ratio of the feasible region and the overall input uncertainty space. This flexibility index is suitable to quantify flexibility for nonconvex

constraints. Adi and Chang[69] proposed a “temporal flexibility index” to reflect the cumulative effects of temporary disturbances within finite time intervals.

## 2.4 Software

The aforementioned sensitivity analysis and feasibility analysis methods are all based on simulation models. In the research of pharmaceutical process modeling, gPROMS ModelBuilder[70] is a useful platform to develop and run dynamic simulation models; gSOLIDS[71] provides a large model library to support the modeling for solids processes. For sensitivity analysis, Simlab[72] is a free software (based on Windows platform) for global sensitivity analysis. GUI-HDMR[73] is a free Matlab toolbox for users to implement RS-HDMR method. For surrogate-based feasibility analysis, the Kriging surrogate can be built with DACE toolbox[74] in Matlab.

## 2.5 Summary

In this chapter, we’ve reviewed two aspects of process analysis approaches that have been implemented and developed for pharmaceutical process models: (1) global sensitivity analysis; and (2) feasibility and flexibility analysis.

For global sensitivity analysis, Morris method requires the smallest number of sample points and can be used to get a preliminary and qualitative knowledge on sensitivity information. Regression-based methods (e.g. PRCC) can be applied to the cases where the output is monotonically related to the input. Variance-based methods (e.g. Sobol’, FAST and eFAST) evaluates how the variations in the inputs (including main effects and total effects) contribute the variance in the output variable. Metamodel-based methods (e.g. RS-HDMR) builds a surrogate model to approximate the original simulation, and can produce sensitivity indices that are equivalent to those from variance-based methods. This can be suitable for computationally expensive simulations, which cannot afford a

huge sampling budget to perform traditional variance-based sensitivity analysis methods. However, it can be difficult to validate the surrogate model. Note that most of the sensitivity methods reviewed in this chapter are based on the assumption that the input factors are independent. For those cases where this assumption is not satisfied, sensitivity analysis should be conducted on groups of correlated inputs[75].

For feasibility analysis, surrogate-based adaptive sampling approaches were developed recently to facilitate the analysis for integrated flowsheet models, which can usually be computationally expensive. Kriging model has been adopted to approximate the feasibility function of the original simulation. After constructing the initial surrogate models, adaptive sampling can be iteratively performed (with surrogate getting updated at each iteration) to search for sample points with a balanced local search and global search. The objective is to efficiently identify the feasible region by finding the feasible region boundary. This approach has the potential to be efficient and accurate for process models with thousands of equations and constraints. However, the increase in problem dimensionality can be a challenge, known as the “curse of dimensionality” [76]. In addition, the definition and proof of the convergence of the surrogate-based approaches need to be further investigated.

### 3 Applications of Optimization in Continuous Pharmaceutical Manufacturing

#### 3.1 Introduction

There is an increasing number of collaborative efforts in improving the pharmaceutical process development and manufacturing, jointly from the academia, regulatory agencies, and the industry. Such a trend is stimulated by a few factors. From the economic perspective, it has been well acknowledged by the pharmaceutical industry that efficient manufacturing processes need to be developed and adopted in order to produce qualified products and maximize the profits within a

drug's patent life. From the regulatory perspective, it is the vision for Food and Drug Administration (FDA) to have a highly efficient, agile, flexible pharmaceutical sector that can consistently provide drugs of high-quality [77]. To achieve the goal of improving pharmaceutical product quality, FDA had a number of initiatives, including process analytical technology (PAT) [78], Quality by Design (QbD) [79]. Recently, FDA has stressed the importance of emerging technologies (e.g., continuous manufacturing, 3D printing technologies [80], etc.) in the improvement of pharmaceutical product quality [81].

In order to embrace the benefits of emerging technologies in the pharmaceutical manufacturing, it is critical to have an in-depth process knowledge. Process modeling tools have become increasingly important in gaining insights into processes and assisting risk assessment via prediction based on process data [9,82]. For a review on the recent development in the modeling of pharmaceutical processes, the interested readers are referred to [83] and Chapter 2 in [84]. Commonly used modeling approaches include first-principle models (e.g., discrete element models (DEM) [25], finite element models (FEM) [85]), population balance models [86], phenomenological models [84], and reduced-order models (e.g., response surface models [87], artificial neural network [88], latent variable methods [89], etc.). A well-developed process model is a powerful tool to predict process dynamics [90], characterize design space (Chapter 6 in [91]), investigate critical process parameters [14], facilitate process control [92], and perform process optimization [49].

Based on process models, mathematical optimization approaches have long been used in other industries to improve process performance, with a variety of applications covering process design, operations, and control. A general classification of the mathematical optimization problem is given in [93], where the general formulation of an optimization problem is as follows:

$$\begin{aligned}
& \min Z = f(x, y) \\
& s. t. \\
& h(x, y) = 0 \\
& g(x, y) \leq 0 \\
& x \in X, y \in \{0,1\}^m
\end{aligned} \tag{53}$$

where  $f(x, y)$  is the objective function (e.g., cost);  $h(x, y)$  are the equality constraints describing the process systems (e.g., mass balance);  $g(x, y)$  are the inequality constraints defining the process constraints (e.g., specifications on product qualities);  $x$  represents continuous variables;  $y$  denotes discrete variables. Problem (53) corresponds to a mixed-integer problem (MIP). When no discrete variables are existent in the system, Problem (53) is reduced to a nonlinear program (NLP) when any of the functions involves nonlinearities, or a linear program (LP) when all the functions are linear.

In the pharmaceutical process development, mathematical optimization approaches have been implemented to improve product formulations, drug delivery systems, and manufacturing processes. This chapter aims to provide an overview of the applications of optimization in the pharmaceutical process development, and briefly introduce the mathematical tools (e.g., data-driven models, optimization algorithms) that are generally used.

### 3.2 Different optimization objectives in the pharmaceutical process development

In process optimization, the first thing that should to be considered is what needs to be optimized, namely the objective function. Depending on the nature of the study, there are a number of objective functions that have been used when optimizing a pharmaceutical process. In this section, we first introduce some commonly adopted objective functions for single-objective optimization, and then discuss the cases when we need to consider multiple objectives simultaneously.

### 3.2.1 Single-objective optimization

One common choice for the objective function is to optimize the critical quality attribute (CQA) of a drug product. Velásco-Mejía et al. [94] determined the optimal operating conditions for a drug crystallization process that provided the highest crystal density. Monteagudo et al. [95] developed an optimized formulation to obtain the pharmaceutical product with best taste-masking efficiency. Chavez et al. [96] optimized the formulation of a pharmaceutical tablet product to maximize the joint probability for five CQAs to meet a minimum satisfactory level of quality. Pal et al. [87] determined the optimal formulation that can give a desired drug release profile.

The optimization of process performance and efficiency has been investigated in various pharmaceutical processes. Based on a thermodynamic model that can predict phase equilibria of multicomponent systems, Sheikholeslamzadeh et al. [97] investigated the optimal operating conditions that maximized the crystallization yields for a batch cooling-antisolvent crystallization process. Zhang and Huang [98] calculated the optimal operating conditions that led to the largest chemical oxygen demand (COD) of a pharmaceutical wastewater treatment process in which wet peroxide oxidation (WPO) was used. Grom et al. [99] developed a mechanism-based reaction kinetic model and used it to obtain maximal desired product concentration of an API synthesis process for a benzazepine class of heterocyclic compound (a weight-loss drug).

Economic objectives are often used in process optimization since they are critical to decision making. Jolliffe and Gerogiorgis [100] formulated a NLP optimization problem of a conceptual upstream continuous process for the production and purification of ibuprofen, with the objective of minimizing the total cost consisting of capital and time-discounted operating expenditure. Abejón et al. [101] minimized the total costs for a separation process using multistage membrane cascades,

with the operation variables as independent variables and the product specifications as constraints. Boukouvala and Ierapetritou [49] formulated a constrained optimization problem to minimize the total cost of a continuous direct compaction (CDC) process while meeting product quality requirements.

Recently, there is a growing interest in improving the flexibility of operating the pharmaceutical process. Flexibility is a quantitative measure of the capability of a process to remain feasible in the presence of process uncertainties (e.g., variations in the flow rate), which was initially proposed by Grossmann and Morari [102]. Grossmann et al. [44] provided an overview of recent advances in quantifying flexibility of chemical processes. The concept of flexibility has also been introduced to pharmaceutical processes and its applications are closely related to the problem of characterizing the design space. Rogers and Ierapetritou [65] developed a surrogate-based approach to investigate both the steady-state and the dynamically changing design space of a roller compaction process. Rogers and Ierapetritou [48] further evaluated the flexibility of the roller compaction process by computing the stochastic flexibility index, with the uncertainties in process inputs being described by an arbitrary probability distribution. Adi and Laxmidewi [103] computed the volumetric flexibility index to evaluate the operational flexibility of a separation process using membranes.

The analysis of environmental impacts of pharmaceutical processes has gained increasing attention over the past few years. A systematic approach is via Life Cycle Assessment (LCA), which evaluates the environmental impacts throughout a product's life cycle, covering raw material acquisition, production and usage, and waste disposal [104]. When integrating the environmental consideration to the pharmaceutical industry, it can make significant contribution to building a more sustainable and environmental benign pharmaceutical process. Ott et al. [105] presented a holistic life-cycle-based process optimization and intensification for an active pharmaceutical ingredient (API) production process of an anticancer drug, which investigated the main bottlenecks

of the process and made recommendations for strategies of optimization by accounting for a number of partly contradictory environmental effects. Jolliffe and Gerogiorgis [100] evaluated the environment impact of an API process by using the Environmental Factor (E-factor), which was defined as the total mass of waste generated per unit mass of product. Ott et al. [106] performed an environmental assessment of different pathways of the production of rufinamide, based on both simplified metrics (e.g., process mass intensity, cumulative energy demand) and a holistic LCA investigation, and showed the environmental benefits for switching from a multi-step batch process to a flow process.

### 3.2.2 Multi-objective optimization

Many pharmaceutical processes have more than one objective to be optimized. For example, when considering drug qualities, it is usually needed to simultaneously control their mean values (to be as close to targets as possible) and the variance (to be as small as possible). This multi-objective problem has been formulated into a robust design (RD) problem, with applications demonstrated in [107,108]. In addition, it has also been investigated to achieve target product qualities and meanwhile reduce the operational costs for a pharmaceutical granulation process [109]. Ardakani and Wulff [110] gave an overview of a wide range of methods for multi-objective optimization problems. Below, we first show the general form of a multi-objective optimization problem, then provide a review on the mathematical approaches that are widely adopted in pharmaceutical processes. Interested readers are referred to [111] for more mathematical details.

A multi-objective optimization has the following general form

$$\begin{aligned} & \min\{f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_k(\mathbf{x})\} \\ & s. t. \mathbf{x} \in S, \end{aligned} \tag{54}$$

which has  $k$  ( $\geq 2$ ) conflicting objective functions  $f_i: \mathbb{R}^n \rightarrow \mathbb{R}$  that need to be minimized simultaneously. The decision variable  $\mathbf{x}$  belong to a nonempty feasible region  $S \subset \mathbb{R}^n$ . Objective vectors are denoted as  $\mathbf{z} = \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_k(\mathbf{x}))^T$ . In the multi-objective optimization, objective vectors are regarded as optimal (i.e., Pareto optimal) if none of their components can be improved without deterioration to at least one of the other components [111].

The most direct way of solving a multi-objective optimization problem is to find the Pareto curve consisting of multiple Pareto optimal solutions. Abejón et al. [101] used the Pareto curves to interrelate two variables (i.e., the product purity and the process yield) for a continuous organic solvent nanofiltration process. Brunet et al. [112] formulated MIP model for the process design of the production of Penicillin V, which aims to determine the optimal operating conditions of the pharmaceutical plant (continuous variables) and the plant topology (integer variables) that optimize simultaneously the profitability of the process and the associated environmental impact.

Another way to solve a multi-objective optimization problem is by maximizing a desirability function. Derringer [113] calculated the desirability as a weighted geometric mean of transformed objectives  $f_i$  (scaled to the range between 0 and 1). The advantage of using the desirability is that it can use a single measure to characterize the overall performance. However, the resulted optimal solution is highly sensitive to the assigned weights. Uttekar and Chaudhari [114] selected the optimal formulation that maximizes the desirability function in order to achieve the targeted particle size distribution for Budesonide (a drug used for the treatment of asthma), which was produced by using the amphiphilic crystallization process. Sato et al. [115] calculated the optimal conditions of a crystallization process that satisfy both the required amount of residual solvent and the particle size D50 based on the desirability. Chakraborty et al. [116] optimized the formulation of a fast-dissolving pharmaceutical wafer containing Loratadine (a drug to treat allergies), with the

desirability as the objective function which involves four critical quality attributes (CQAs) of the drug. Kermet-Said and Moulai-Mostefa [117] applied electrocoagulation to a pharmaceutical waste water treatment process and investigated the optimal operating conditions to maximize COD removal and turbidity removal.

Goal programming has also been used to solve the multi-objective optimization problem. It requires the decision maker to specify a goal point and finds a feasible solution that is as close to the goal as possible [110]. Nha et al. [107] developed a lexicographical dynamic goal programming to account for the dynamic nature of pharmaceutical quality characteristics and utilized it in testing for *in vitro* bioequivalence (considering two time-dependent responses: gelation kinetics and drug release rates) of a generic drug. Li et al. [118] proposed a priority-based optimization scheme, which incorporates goal programming methods, modified desirability functions, and higher-order response surface models, to address the multi-response pharmaceutical formulation optimization problem.

### 3.3 Applications of data-driven models in optimization

In order to formulate the optimization problem, data-driven models have been widely used for pharmaceutical process optimization. Such models investigate the system input-output relationship only on the basis of system data, without requiring any explicit knowledge of the physical behavior of the system [119]. In some research areas, the data-driven model is also known as “surrogate model” [120], “metamodel” [121], “reduced-order model” [122], or “response surface” [123]. For pharmaceutical processes, data-driven models are a useful tool to enhance the fundamental understanding when first-principle models are not available, due to, for example, a lack of knowledge on the mechanical and physiochemical properties of raw materials [124]. Moreover, even in the cases when first-principle simulations (e.g., DEM, FEM) can be performed,

they are usually too computationally expensive to be directly applied in the process optimization or design settings [121]. In such cases, data-driven models can be used as a computational efficient approximation to the expensive simulation and contribute to the formulation and solving of an optimization problem.

In this section, we first briefly review the sampling plans for data-driven models, and then discuss a variety of modeling techniques that have been adopted in the pharmaceutical process, which is followed by the model validation methods. Finally, we will demonstrate how data-drive models can support the process optimization.

### 3.3.1 Sampling plan

In order to extract the most process information within a limited sampling budget, we need an organized sampling plan to determine how to sample the input space. The selection of a specific sampling depends on what kind of experiments are being conducted – whether it’s a physical experiment or a computational experiment – and on what data-driven modeling technique is being used.

For pharmaceutical process development, it is mainly dependent on the physical experiments conducted to gain process knowledge. In such cases, sampling plans are chosen by using the “Design of Experiment” (DoE) theory. The main idea is to plan experiments in order to minimize the effects of random errors [121]. Singh et al. [125] presented an extensive review on the classical DoE sampling plans. Widely used designs include (fractional) factorial designs [87,126,116], central composite designs [114,98,108], mixture designs [107], Box-Behnken designs [127,128], and Plackett-Burman designs [129,130]. These experimental plans usually focus on sampling around the input space boundaries with a few at the center of the input space.

When computer experiments are to be conducted, since most of simulations are deterministic, the goal of an experimental plan is focused on reducing systematic errors rather than random errors. For this type of experiments, Sacks et al. [131] claimed that a good design should fill out the entire input space rather than only concentrate on the boundaries. Commonly used “space-filling” designs include Latin Hypercube designs (LHD) [56,132], Hammersley sequence sampling [133], orthogonal arrays [134], and uniform designs [135]. An extensive discussion of modern designs for computer experiments is provided in [136].

### 3.3.2 Building a data-driven model

There is a variety of data-driven models that have been developed and used in different engineering fields. Below we discuss four types of models that are widely used in modeling and optimizing pharmaceutical processes. These models can be applied when the following assumptions are valid: the process output to be modeled is continuous and smooth. These assumptions are generally valid for most engineering processes. Forrester and Keane [137] commented that when the continuity assumption is not guaranteed, the process can then be modeled with multi data-drive models which are patched together at discontinuities. Such an exception will not be considered in this chapter.

#### ***Response surface methodology***

Response surface methodology (RSM) was first proposed by Box and Wilson [138] to improve chemical manufacturing processes. A RSM model approximated the process input-output relationship with a low-degree polynomial model [139] in the following form:

$$y = \mathbf{f}'(\mathbf{x})\boldsymbol{\beta} + \varepsilon \quad (55)$$

where  $\mathbf{x} = (x_1, x_2, \dots, x_k)'$ ;  $\mathbf{f}(\mathbf{x})$  if a vector of function consisting of powers and cross-products of powers of  $x_1, x_2, \dots, x_k$  up to a degree  $d (\geq 1)$ ;  $\boldsymbol{\beta}$  is a vector of  $p$  model parameters;  $\varepsilon$  is a

random experimental error with zero mean. Two commonly used RSM models are the first-degree model:

$$y = \beta_0 + \sum_{i=1}^k \beta_i x_i + \varepsilon \quad (56)$$

and second-degree model:

$$y = \beta_0 + \sum_{i=1}^k \beta_i x_i + \sum_{i=1}^k \sum_{i < j} \beta_{ij} x_i x_j + \sum_{i=1}^k \beta_{ii} x_i^2 + \varepsilon \quad (57)$$

The values of  $\beta$  can be estimated using analytical expressions (i.e., ordinary least-squares estimators) [140,139].

In order to evaluate the significance of model parameters of a specific RSM model, ANalysis Of VAriance (ANOVA) can be carried out, together with Student's t-test [125]. It is advised that only retain significant model parameters in the final model. In addition, to choose the best RSM model and prevent the danger of over-fitting, Singh et al. [125] suggested use several metrics to evaluate the model fitting, including  $R^2$ ,  $R_{adj}^2$ , Predicted Residual Sum of Squares (PRESS), and  $Q^2$ .

RSM is regression technique. A RSM model is usually constructed using experimental designs from the DoE theory [139]. The applications of RSM in modeling pharmaceutical processes can be found in a variety of studies [87,107,117,108,12]. Forrester and Keane [137] stated that RSM models are appropriate for problems with low dimensions, uni- or low-modality, where physical experiments were conducted. For deterministic computer experiments, however, Jones [123] demonstrated that RSM models could be insufficient to capture the shape of the function and thus may not be able to identify an optimal solution in optimization settings.

### ***Partial least squares***

Partial least squares (PLS) regression, also termed as Projection to Latent Structures, is a multivariate regression approach [141,142]. Assuming the data have been mean centered and scaled to unit variance, PLS projects the process inputs  $\mathbf{X}$  (of size  $N$ -by- $K$ ) and process outputs  $\mathbf{Y}$  (of size  $N$ -by- $M$ ) to the common latent space of  $A$  latent variables, with the following model structure:

$$\mathbf{X} = \mathbf{TP}' + \mathbf{E}_X \quad (58)$$

$$\mathbf{Y} = \mathbf{TQ}' + \mathbf{E}_Y \quad (59)$$

$$\mathbf{T} = \mathbf{XW} \quad (60)$$

where  $\mathbf{T}$  (of size  $N$ -by- $A$ ) is the score matrix;  $\mathbf{P}$  (of size  $K$ -by- $A$ ) and  $\mathbf{Q}$  (of size  $M$ -by- $A$ ) are loading matrices;  $\mathbf{E}_X$  and  $\mathbf{E}_Y$  represent residuals;  $\mathbf{W}$  (of size  $K$ -by- $A$ ) is the weight matrix.

Different algorithms can be applied to calculate the PLS model, such as Non-linear Iterative Partial Least Squares (NIPALS) algorithm [141], Expectation Maximization (EM) algorithm [143]. The number of latent variables  $A$  is determined by cross validation (CV), which is performed by dividing the data in a number of groups and then developing different models in parallel using reduced data with one of the groups removed [142]. Among different CV methods, Shao [144] suggested not use the leave-one-out approach. Details on CV will be further discussed in Section 3.3.3.

By collecting data based on the DoE theory, the PLS approach is a powerful tool of analyzing data with high-dimensions, noises, and strong collinearities in both  $\mathbf{X}$  and  $\mathbf{Y}$  [142]. This modeling technique aims to well approximate both of the data tables  $\mathbf{X}$  and  $\mathbf{Y}$  as well as maximize the correlation between  $\mathbf{X}$  and  $\mathbf{Y}$  [145]. In cases when there exist strong collinearities in  $\mathbf{X}$ , and/or  $\mathbf{Y}$  is only sensitive to a reduced set of input combinations, PLS can sufficiently reduce the problem dimensions and simplify the problem without losing much information. Recently advances have

been made to extend PLS to the modeling of nonlinear systems [146]. Pharmaceutical process applications involving the PLS techniques can be seen in [109,95,147].

### ***Artificial neural network***

Artificial neural network (ANN) is a biologically inspired modeling technique that simulates the human brain's way of processing information. An ANN model is formed by numbers of single units (known as processing elements (PE) or neurons), which are connected with coefficients (weights) and constitute the neural structure [148]. As the building component, a neuron passes a sum of weighted inputs to a transfer function (e.g., sigmoidal function) and yields the output. The versatility of ANN comes from the various ways that neurons can be connected in a network. A vast number of ANNs have been developed and are surveyed in [149], among which the most widely used type of ANN is the backpropagation (BP) network [150]. A BP network has a multilayer perception architecture: (1) an input layer of nodes representing the process inputs; (2) an output layer of nodes for the process outputs; and (3) one or more hidden layers containing nodes to capture the nonlinear relationship. Practically, the BP network with one hidden layer is sufficient to approximate most functions [151].

For a BP network, the number of nodes in the input and output layers are determined by the dimension of process inputs and outputs, respectively. However, it is challenging to determine the number of nodes in the hidden layer. With the number being too large, it requires a high computational cost to train the model and also raises the risk of overfitting. On the other hand, if there are too few hidden nodes, it may not be able to accurately represent input-output relationship [152]. In most applications, the number of hidden nodes is still determined by trial-and-error, while some practical rules have been applied in [153,152].

The model coefficients (weights) of an ANN model are calculated by using a training algorithm. The mostly used algorithm is the feedforward error-backpropagation (BP) learning algorithm. [154]. Each iteration of the BP algorithm involves two steps. A forward propagation is performed to pass information from the input layer through the hidden layer to the output layer. Then a backpropagation of error stage is used to calculate the error of each layer of nodes sequentially from the output layer to the input layer. Based on the calculation, the weights are adjusted to reduce the error by using the gradient descent method. This BP algorithm is performed iteratively until it reaches a certain pre-specified level of accuracy.

As a useful method of modeling nonlinearities, the ANN model and its different variants have been implemented both as a regression technique and an interpolation technique [155]. In pharmaceutical processes, the applications of ANN are described in [94,156-158].

### ***Kriging***

Kriging is a widely used interpolation method which is named after the South African mining engineer Krige [159]. In different fields, Kriging is also known as stochastic process model [67], Gaussian process model [160]. The ordinary Kriging represents a process output with the following model:

$$y(\mathbf{x}) = \beta + \varepsilon(\mathbf{x}), \quad (61)$$

where  $\beta$  is a model parameter which represents the surface mean,  $\varepsilon(\mathbf{x})$  is the realization of a stationary Gaussian random field ( $\mathbb{R}^d \rightarrow \mathbb{R}$ ):  $\varepsilon(\mathbf{x}) \sim \text{Normal}(0, \sigma^2)$ . It is assumed that  $\varepsilon(\mathbf{x})$  is spatially correlated: if two points  $\mathbf{x}$  and  $\mathbf{x}'$  are spatially close to each other, then  $\varepsilon(\mathbf{x})$  and  $\varepsilon(\mathbf{x}')$  will tend to be similar. The spatial correlation can be modeled with various correlation functions. The mostly used is the Gaussian function:

$$\text{Corr}[\varepsilon(\mathbf{x}), \varepsilon(\mathbf{x}')] = \exp \left[ - \sum_{h=1}^d \theta_h |x_h - x'_h|^2 \right], \quad (\theta_h \geq 0) \quad (62)$$

With the Kriging model, the best linear unbiased predictor at  $\mathbf{x}^*$  can be expressed as:

$$\hat{y}(\mathbf{x}^*) = \beta + \mathbf{r}' \mathbf{R}^{-1}(\mathbf{y} - \mathbf{1}\beta) \quad (63)$$

where  $\mathbf{y} = (y^{(1)}, \dots, y^{(n)})'$  is the n-vector of observed function values;  $\mathbf{R}$  is the n-by-n matrix with the  $(i, j)$  entry being  $\text{Corr}[\varepsilon(\mathbf{x}^{(i)}), \varepsilon(\mathbf{x}^{(j)})]$ ;  $\mathbf{1}$  is an n-vector of ones;  $\mathbf{r}$  is an n-vector with the  $i$ -th entry being  $\text{Corr}[\varepsilon(\mathbf{x}^*), \varepsilon(\mathbf{x}^{(i)})]$ .

In addition, the mean squared error of the predictor can be derived as follows:

$$\hat{s}^2(\mathbf{x}^*) = \sigma^2 \left[ 1 - \mathbf{r}' \mathbf{R}^{-1} \mathbf{r} + \frac{(1 - \mathbf{1}' \mathbf{R}^{-1} \mathbf{r})^2}{\mathbf{1}' \mathbf{R}^{-1} \mathbf{1}} \right] \quad (64)$$

The model parameters  $\beta, \theta_h, \dots, \theta_d, \sigma^2$  are obtained by maximizing a likelihood function [67].

Kriging is mostly used to approximate computer simulations with data sampled from a space-filling design (e.g. LHD). On the other hand, Kriging can be modified by introducing a “nugget” factor in order to approximate a stochastic simulation with a homogeneous noise level [161]. Recently, Ankenman et al. [162] developed a stochastic Kriging to deal with stochastic simulations with heterogeneous noises. For the pharmaceutical processes, the Kriging has been applied to model steady-state pharmaceutical processes with missing data [124], dynamic pharmaceutical processes [13], and a continuous blending process [12].

### 3.3.3 Model validation

As a critical step of the model development, model validation is defined as the process of determining the degree of accuracy to which a model can represent the real world within the intended use of the model [163]. Based on the calculation of quantitative validation measures to

assess the model fidelity, the user can decide whether it is needed to conduct more experiments to increase the model accuracy.

Cross validation (CV) is a widely used approach to estimate the error of a constructed model [164]. Starting with a dataset of  $N$  samples,  $S\{X, Y\}$ , with a  $p$ -fold cross validation, this original dataset is divided (randomly) roughly equally into  $p$  different subsets:  $S\{X, Y\} = S1\{X1, Y1\}, \dots, Sp\{Xp, Yp\}$ . Then, the model is fitted  $p$  times, each time with one of the subset removed from the training set, and this left-out subset is used to calculate the error. As a variation of the  $p$ -fold cross validation, the leave- $k$ -out approach considers  $\binom{N}{k}$  subsets, each with  $k$  elements left out from the training set. The special case when  $k = 1$  is called leave-one-out (LOO) cross validation which can be computed efficiently [165]. Meckesheimer and Booker [164] recommended that LOO cross validation was appropriate to estimate the prediction error for low-order RSM and radial basis function models (a special case of ANN). However, for the Kriging, it was suggested choose  $k = 0.1N$  or  $k = \sqrt{N}$ .

The advantage of CV is that it can provide a nearly unbiased estimation of the generalization error (compared to the “split sample” case where the sample data is divided once into a training set and a test set), since every point is used in a test set once and in a training set  $(k-1)$  times (for the LOO approach) [166]. The disadvantage, however, is that we need to fit the data-driven model multiple times, which can be computationally expensive. Further, Lin [167] stated that the LOO cross validation was not a sufficient metric to evaluate the model accuracy. The LOO method actually measures the insensitivity of a model to information loss at the data points, while an “insensitive” model is not necessarily equivalent to an “accurate” one. Therefore, it is suggested conduct additional experiments for model validation. A vast number of additional validation metrics have been surveyed in [76].

### 3.3.4 Data-driven models in support of optimization needs

In the pharmaceutical process development, a data-driven model is used to support the process optimization mainly in two ways, which are shown in Figure 5.

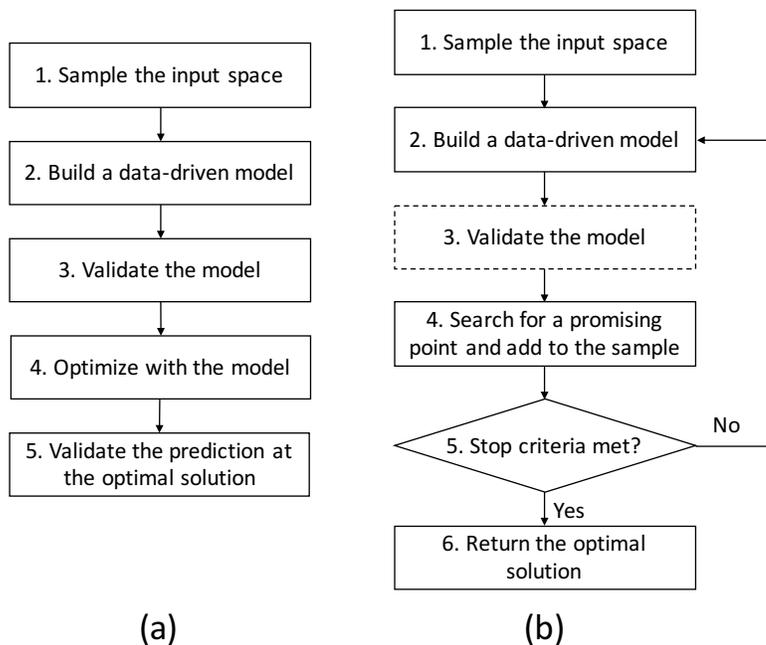


Figure 5. Two approaches that use a data-driven model to support optimization. (a) sequential approach; (b) adaptive approach

The first approach (Figure 5 (a)) is a “sequential approach”. Based on an initial dataset, a data-driven model is constructed, which, after being validated (mostly with a CV approach), is optimized using a mathematical programming approach. Under some circumstances, after the optimal solution is found, a new experiment is performed at this optimal point. The idea is to validate whether the predicted optimal value is consistent with the experiment. Note that this “sequential approach” is mostly adopted when physical experiments are performed. Thus, the initial data points are usually sampled by using a DoE sampling plan. The advantage of this approach is its simplicity. However, the efficacy of this approach depends on the assumption that a sufficiently accurate model can be constructed based on a pre-specified number of sample points. In cases when the model is not

accurate enough (which can be reflected at Step 3 and Step 5), this approach does not give guidance on further sampling directions. In pharmaceutical processes, “sequential approach” is still the dominant method of process optimization, with applications in [87,115,98,109,94,95,117].

The second approach (Figure 5 (b)) is an “adaptive approach” which features an adaptive sampling stage. Similar with “sequential approach”, the “adaptive approach” also starts with an initial data set and an initial data-driven model. In some cases, the model validation is optional after the initial data-driven model is built because the model accuracy can be further improved in the subsequent adaptive sampling stage. The crucial part of “adaptive approach” is the search criteria that are used in Step 4. Instead of directly optimizing the data-driven model, different mathematical approaches have been developed to guide the search direction, with a balance between finding a better optimal solution and reducing the uncertainty in the data-driven model. Once a new sample point is added, the data-driven model also gets updated. This adaptive sampling (i.e. from Step 4 back to Step 2) is iteratively performed until some stop criteria are met. This approach is also called surrogate-based optimization [137], of which details will be further discussed in Section 3.4. Since “adaptive approach” is mostly used when deterministic computer simulations are available, it usually does not require the validation of the optimal value with the experiment (i.e., Step 5 in Figure 5 (a)). Applications of “adaptive approach” in pharmaceutical processes can be found in [64,49].

### 3.4 Optimization methods in pharmaceutical processes

In previous sections, we have discussed different optimization objectives in pharmaceutical processes, and how data-driven models can be used within an optimization setting. One more important aspect that remains to be discussed is how we can find an optimization solution, i.e., the mathematical programming methods. A classification of optimization problems and solution

methods have been presented in [93]. Generally, there are two major approaches to solve an optimization problem: the (1) derivative-based methods; and (2) the derivative-free methods. In this section, we review the most popular algorithms under these two categories and refer interested readers to the applications in pharmaceutical process development.

### 3.4.1 Derivative-based methods

The derivative-based methods require the derivative information (e.g., gradient, Hessian, etc.) to direct the search to an optimal solution. Such methods are appropriate for problems whose derivative information is reliable and easy to obtain (either provided by users or estimated by computational tools). The advantages of this type of approach are the fast rate of convergence and its capability of dealing with large-scale problems.

#### ***Successive quadratic programming***

Successive (or sequential) Quadratic Programming (SQP) [168] is a conceptual method that has evolved into numbers of different specific algorithms for constrained NLP problems. The basic idea is to model the NLP at iteration  $k$  (with an approximate solution  $\mathbf{x}^{(k)}$ ) by a quadratic programming (QP) subproblem, to which the solution is used to direct to the search to the next iteration  $k+1$ . The process is iteratively conducted until it converges to a local optimal solution. There are two main reasons of using a QP subproblem: (1) it is relatively easy to solve; and (2) its objective can reflect the nonlinearities of the original problem. For the generation of QP subproblem, a Hessian matrix of the Lagrangian function needs to be constructed, which can be obtained by either (1) the second derivatives for the objective or constraint functions, or (2) positive definite quasi-Newton approximations [93].

SQP methods can only guarantee the convergence to a local optimal solution [168]. The convergence from poor starting points can be promoted by using a line search or a trust-region method [93]. It is found that SQP solvers generally require fewest function evaluations to solve NLPs [169,170]. In addition, SQP methods do not require feasible points at any stage of the process, which is advantageous because it can be usually difficult to find a feasible point in the existence of nonlinear constraints. However, modifications have been made to ensure SQP always remain feasible through the process [171]. A list of SQP-based solvers was presented in [93]. Aside from the SQP algorithm, a vast number of other gradient-based NLP solvers are also available, which can require more function evaluations than SQP but provide a good performance when interfaced to optimization model platforms (e.g., GAMS [172], AMPL [173]).

Using the SQP method, Sen et al. [174] estimated the parameters of a 2-dimensional PBM of a cooling crystallization process based on experimental data. Acevedo et al. [175] calculated the optimal temperature profile for an unseeded batch cooling crystallization system, with the goal of achieving the desired shape and size distribution of crystals subject to a set of process constraints considering temperature range, product yield, and batch time. Yang and Nagy [176] identified the optimal steady state operating profiles of a continuous mixed suspension, mixed product removal (MSMPR) cascade system by maximizing the crystal mean size, which was constrained on temperature, solvent composition, and residence time. Gagnon et al. [177] computed the optimal control strategy based on a phenomenological state-space model of a fluid bed drying (FBD) process. Compared to traditional open-loop FBD operations, the control approach could improve the process by reaching the target particle moisture content while limiting operation problems including under/over drying and particles over-heating. With the CONOPT solver, Wang and Lakerveld [178] maximized the attainable region of crystal size of a continuous membrane-assisted crystallization (cMAC) process, and demonstrated the advantages of cMAC over conventional crystallization processes. In order to find the optimal reactor design of an API synthesis, Emenike

et al. [179] used the elementary process functions (EPF) methodology, which was formulated into a dynamic optimization problem and solved with the CONOPT solver. Further, in those references that were mentioned in previous sections of this chapter, applications of SQP can be seen in [109,112], while other derivative-based NLP solvers have been used in [101,100].

### 3.4.2 Derivative-free methods

The Derivative-free optimization (DFO) methods find the optimal solution only based on the objective functions values (and constraint values) without any derivative information. These methods are successful in cases where derivative-information is unreliable or impractical to get (e.g., when the model is expensive or noisy). However, for many algorithms, difficulties still remain for the proof of global convergence. A survey of DFO algorithms for bound-constrained was presented in [180]. Traditionally, most of the DFO methods are only suitable for low-dimensional problem. However, recent efforts have been made to adapt some of the methods to high-dimensional problems, of which a review was provided in [76]. Below, we briefly introduced three types of DFO methods that have been applied to pharmaceutical processes.

#### ***Direct search methods***

The Nelder-Mead algorithm [181] involves iteratively building and updating a simplex formed by a set of points (vertices). At each iteration, it aims to replace the worst vertex by a new vertex and then forms a new simplex. This process is conducted by a series of operations considering the centroid of the current simplex, including reflection, expansion, contraction, and shrink. The convergence of the Nelder-Mean algorithm was investigated in [182], while further developments can be found in [183]. Another direct search algorithm is the generalized pattern search (GPS) algorithm [184], which was initially developed for unconstrained problems. It is a generalization of direct search methods including the Hooke and Jeeves method [185] and the coordinate search method [186]. GPS updates the current iterate by sampling at a finite number of points along a

suitable set of search directions, with the goal of finding a decrease in the objective function values. It has been extended to bound-constrained problems [187] and linearly constrained problems [188]. GPS was further generalized as Generating Search Set (GSS) methods by Kolda et al. [189], which were shown to converge to stationary points under mild conditions [184].

Grimard et al. [190] used the Nelder-Mead algorithm to evaluate the parameters for a mathematical model consisting of mass and energy balance equations for a hot-melt extrusion process. Besenhard et al. [191] combined Nelder-Mead with global optimization techniques to estimate the crystal growth model parameters of a PBM for crystallization processes. Based on a DoE, Paul et al. [192] identified and quantified critical process parameters (CPP) (i.e., sodium chloride concentration, pH value for elution) of a multimodal ion exchange step for the purification of biopharmaceuticals. The optimal values of the CPP were calculated to maximize the purity and recovery of this purification process. Zou et al. [193] fitted a Korsmeyer-Peppas (KP) model with dissolution data, which was used to describe the kinetics of an *in vitro* drug release process using nanoparticle formulations. Xi et al. [194] computed the optimal values for three categories of design variables (related to device, particles, and patients) to maximize the efficiency of an electric-guided drug delivery system for the treatment of rhinosinuitis. Moudjari et al. [195] estimated the values of interaction parameters of thermodynamic models for the solubility prediction of pharmaceutical compounds in various solvents. In earlier mentioned references from Section 3.2, direct-search algorithms were used in [117,118].

### ***Genetic algorithms***

Genetic algorithms (GA) [196], sometimes known as evolutionary algorithms, are a family of population-based heuristic search algorithms that mimic the mechanistic of natural selection and reproduction processes. A GA starts with a randomly sampled initial population of chromosomes (i.e., initial generation), which are basically sample points with variables represented by binary

strings. Then, the structures of the chromosomes are evaluated and reproductive opportunities can be allocated in a way so that those with a better solution have higher chances to reproduce. At each iteration, the descendant generation is generated successively via a series of operations including selection, recombination, and mutation. GA can be categorized as a stochastic global search algorithm [180]. It usually requires the objective function to be fast to evaluate [197], and is only suitable for low-dimensional problems [198].

For a continuous crystallization process modeled with the PBM approach, Ridder et al. [199] used a GA technique to calculate the Pareto optimal solutions of a multi-objective optimization problem, which simultaneously maximized the average crystal size and minimized the coefficient of variation. Zaki et al. [200] used the ANN approach to model the fabrication process of Bupropion HCl loaded agar nanospheres which were used for sustained drug release. Based on this model, the GA approach was applied to optimize the process, namely, to minimize the particle size, release efficiency, and maximize loading efficiency, etc. Allmendinger et al. [201] formulated a constrained optimization problem to improve the performance (considering process cost, time, and product waste) of a chromatography purification process. Four types of GA methods were applied to identify the optimal equipment sizing strategies of this biopharmaceutical process. Rostamizadeh et al. [202] identified the optimal process parameters for fabricating a type of nanoparticles (used for oral insulin delivery) in order to achieve its best performance with respect to six performance measures. Kalkhorana et al. [203] applied the GA approach to estimate the parameters of a drug release model, which was developed to predict the drug diffusion rate from a hydrogel-based drug delivery system. Wang et al. [204] used the GA approach to find the optimal formulation of Doxy inclusion complex (a type of broad-spectrum antibiotic drug) that could lead to optimum inclusion efficiency and stability in the aqueous solution. For those references that were earlier mentioned in Section 3.2, case studies of applying GA can be found in [109,94].

### ***Surrogate-based optimization methods***

Surrogate-based optimization (SBO) methods treat the original problem as a black-box process. Accordingly, a surrogate model is built as a fast approximation to this black-box process and then guides the search direction to the next sample point. The general optimization framework is shown in Figure 5 (b), and its description is mentioned in Section 3.3.4. Depending on the choice of surrogate models, there are mainly two SBO algorithms: (1) Kriging-based approach (sometimes known as Bayesian optimization); and (2) Radial Basis Function (RBF)- based approach. The seminal work of the Kriging-based approach is the Efficient Global Optimization (EI) algorithm by Jones [67], which used an Expected Improvement (EI) function as the infill criteria to search for the next sample point. Jones [123] presented a survey of various infill criteria that could be applied with the Kriging-based methods. The classical EGO algorithm can be seen as a greedy search approach. Recently, it has been combined with dynamic programming which can account for the remaining number of evaluations [205,206]. This look-ahead approach finds the optimal strategy by maximizing a long-term reward, and is shown to be more effective when a limited sampling budget is available. In terms of the RBF-based methods, Gutmann [207] proposed a RBF-based approach which used a bumpiness measure to search for the next sample point. This approach has very similar characteristics with the one-stage approach for the Kriging-based method that was discussed in [123]. Regis and Shoemaker [208] developed a RBF-based method for black-box constrained optimization problems. Regis [209] further proposed a RBF-based method for high-dimensional constrained optimization problems.

The number of applications of SBO methods in pharmaceutical processes is relatively small compared to other optimization algorithms. Luna and Martínez [210] used the Bayesian optimization approach to maximize biomass growth based on a hybrid cybernetic model which was used to simulate the animal cell metabolism of bioreactors. Based on a reduced model for a perfusion bioreactor, Mehrian et al. [211] applied the Bayesian optimization approach to find the

optimal medium refreshment regime that would result in the maximized neotissue growth kinetics in the bioreactor. Boukouvala and Ierapetritou [49] developed Kriging-based methods for the optimization of continuous direct compaction pharmaceutical manufacturing processes.

### 3.5 Summary

In this chapter, we have discussed recent developments of optimization in pharmaceutical processes. A review is first provided on various objectives that are mostly considered in different pharmaceutical processes, such as API processes, waste water treatment process, downstream tableting processes, etc. Further, we introduced four types of data-driven models that are commonly used under two optimization frameworks for pharmaceutical processes, including response surface methodology (RSM), partial least squares (PLS) regression, artificial neural network (ANN), and Kriging. We also included an overview on several optimization algorithms that are widely adopted to solve an optimization problem, which may or may not require the derivative information.

Compared to traditional chemical and petrochemical processes, the applications of optimization tools in the pharmaceutical process development is still in a primary stage. Challenges remain to improve the mechanistic understanding and models' predicting capabilities of the pharmaceutical processes. From this perspective, hybrid models (e.g., [212]) which combine both the mechanistic knowledge from first-principle models and the efficiency from reduced order models can be a highly promising tool to be used in the optimization settings. In addition, the flowsheet modeling approach, acting as a representation of the whole integrated process, is also gaining an increasing attention from the research community. However, due to its high model complexity, it also raises the challenge of optimizing for large-scale systems, which can involve black-box model components and may potentially be computationally expensive to compute. A promising approach to address such difficulties is via the use of surrogate-based optimization methods. In addition, for pharmaceutical manufacturing processes, the variability of product qualities is usually a critical

aspect that requires attention. Such variabilities can be modeled by introducing a random error term to the simulation (i.e., a stochastic simulation). To address the challenges of optimizing with black-box stochastic simulations, the simulation optimization (SO) methods can be used.

### III. Process analysis and optimization for deterministic systems

## 4 A Novel Feasibility Analysis Method for Black-Box Processes using a Radial Basis Function Adaptive Sampling Approach

### **Abstract**

Feasibility analysis is used to determine the feasible region of a multivariate process. This can be difficult when the process models include black-box constraints or the simulation is computationally expensive. To address such difficulties, surrogate models can be built as an inexpensive approximation to the original model and help identify the feasible region. An adaptive sampling method is used to efficiently sample new points towards feasible region boundaries and regions where prediction uncertainty is high. In this paper, cubic Radial Basis Function (RBF) is used as the surrogate model. An error indicator for cubic RBF is proposed to indicate the prediction uncertainty and is used in adaptive sampling. In all case studies, the proposed RBF-based method shows better performance than a previously published Kriging-based method.

### 4.1 Introduction

A process is considered to be feasible if it meets all operating, quality and production constraints. In the presence of uncertainty, including variability in inlet materials and operating conditions, etc., it is important to know whether a given process design is guaranteed to be feasible. This problem was initially formulated as a “flexibility test problem” in [213], which determines whether a process design is feasible over the whole uncertain domain by solving a max-min-max optimization problem. Many approaches have been developed to solve the flexibility test problem. The vertex solution method [213] considers the vertices and extreme values of the uncertain parameter sets and can be used under some convexity conditions. The active-set method [214] formulates the flexibility test problem into a mixed integer optimization problem. However, both of these methods require closed-form, differentiable constraints. Banerjee and Ierapetritou [215] used an  $\alpha$ -shape

surface reconstruction method to represent the feasible domain with a polygon. Adi, Laxmidewi et al. [216] applied a random line search algorithm to find the feasible region boundary points. Both methods can be rather accurate in identifying nonconvex and disjoint feasible regions, but they may require large sampling costs. Michalewicz and Schoenauer [217] developed an efficient evolutionary algorithm to search for the boundary of the feasible region, but the success of such a heuristic method depends on the proper choice of algorithm parameters.

Recently surrogate-based feasibility analysis methods have gained increasing attention [44]. These approaches are based on building surrogate models to approximate the original simulation model and use these approximations to find the feasible region. Because such methods are computationally more efficient, they are especially suitable for problems with black-box constraints or computationally expensive simulation models whose derivatives are difficult to calculate. There are two key factors for the surrogate-based feasibility analysis methodology: (1) which surrogate model to use; and (2) how to do sampling to efficiently build and update the surrogate models.

Many different surrogate models have been used in the literature to address the feasibility problem. Goyal and Ierapetritou [218] developed an approach that is based on inner and outer approximation of the feasible region. Banerjee et al. [219] built a high dimensional model representation (HDMR) surrogate over the whole range of uncertain parameters of the original model. Boukouvala and Ierapetritou [220] and Rogers and Ierapetritou [65] used Kriging as the surrogate model and could identify the feasible region boundary with high accuracy.

In this paper, we introduce a new surrogate-based method for black-box feasibility analysis, which is based on Radial Basis Functions (RBFs). RBFs use a weighted sum of radial functions to emulate the original model [155]. The radial functions feature a monotonic behavior with distance from a central point (e.g. sample point). Park and Sandberg [221] showed that RBFs are capable of

universal approximation. RBFs have been widely used in surrogate-based optimization problems. In [207] RBFs are used to solve global optimization problems (GOP). The method using linear, cubic, or thin plate spline radial functions, is proved to converge for any continuous function. This method was used in a commercial solver TOMLAB/rbfSolve [222]. Regis and Shoemaker [208] proposed an approach to use RBFs to solve constrained global optimization problems with black-box functions. Koch, Bagheri et al. [223] further introduced the repair mechanism to the RBF-based algorithm by Regis [209], which resulted in better final results and faster rate of convergence.

Besides selecting the surrogate model, sampling strategy is also an important aspect for surrogate-based feasibility analysis, because it is always desired to use an effective and efficient sampling strategy to build and update surrogate models. “Space filling” designs [224], such as uniform designs, Latin Hypercube Sampling (LHS), etc. can be used to sample points throughout the whole design space. Initial surrogate models are usually built based on space-filling designs. Adaptive sampling methods can then be used to improve surrogate accuracy. This idea of adaptive sampling originates from the optimization literature [67] where the optimum is obtained by balancing exploitation sampling (sample points where surrogate model is minimized) with exploration sampling (sample points where prediction error may be high). In Refs. [220,65], the adaptive sampling method was tailored and developed for surrogate-based feasibility analysis for the purpose of reducing number of samples while maintaining surrogate accuracy.

In this paper, we propose to use cubic RBF as the surrogate model, and apply adaptive sampling by using an error indicator to characterize prediction uncertainty from RBF predictors. This RBF-based method shows better performance than the Kriging-based method proposed in our previous work [220,65]. The rest of the paper is organized as follows. Current advances in feasibility analysis is overviewed in section “Feasibility analysis”. Surrogate models and the adaptive

sampling method are described in section “Surrogate-based feasibility analysis” . Results on test problems are shown in section “Computational study” . A real-world case study of roller compaction is illustrated in section “Case study of a roller compaction process” . The final conclusions and future work are discussed in section “Summary and future work” .

### Feasibility analysis

Flexibility [45] describes the capability of a process to remain feasible over a range of uncertain parameters. A flexibility analysis problem can be mathematically formulated with a flexibility test problem [213], the derivation of which is given as following.

To start with, in order to determine the feasibility of a process at steady state, we need to first define a feasibility function. This feasibility function is used to describe whether for a fixed value of uncertainty parameter  $\theta$ , a process can meet all constraints  $f_j$  by simply adjusting control variable  $z$ . This can be accomplished by selecting  $z$  to minimize the largest value of  $f_j$ . This feasibility function can be defined with Equation (65)

$$\psi(d, \theta) = \min_z \max_{j \in J} \{f_j(d, z, \theta)\} \quad (65)$$

where  $\psi(d, \theta)$  is the feasibility function;  $d$  represents the design variables, e.g. equipment size;  $z$  represents the control variables,  $z \in Z = \{z: z^L \leq z \leq z^U\}$ ;  $\theta$  represents the uncertain parameters,  $\theta \in T = \{\theta: \theta^L \leq \theta \leq \theta^U\}$ ;  $f_j(d, z, \theta)$  represents the problem’s constraints. If  $\psi(d, \theta) \leq 0$ , it means the process design is feasible. If  $\psi(d, \theta) > 0$ , it means one or more of the constraints are violated, no matter how we adjust the control variables  $z$ . If  $\psi(d, \theta) = 0$ , it means we are at the boundary of the feasible region.

Using the feasibility function, we can then define the flexibility test problem, which determines whether the process is feasible over the whole range of uncertain parameters. This is equivalent to

determining whether the maximum value of  $\psi(d, \theta)$  in the whole uncertainty space is less than or equal to zero. Thus the flexibility test problem can be defined with Equation (66)

$$\begin{aligned} \chi(d) &= \max_{\theta} \psi(d, \theta) \\ \text{s. t. } \theta &\in T = \{\theta: \theta^L \leq \theta \leq \theta^U\} \end{aligned} \quad (66)$$

Generally, the flexibility test problem can be represented as a max-min-max formulation in Equation (67)

$$\chi(d) = \max_{\theta} \min_z \max_{j \in J} \{f_j(d, z, \theta)\} \quad (67)$$

In this paper, we will focus our attention on the case when there are no control variables. In this case the feasibility function in (65) can be simplified as (68)

$$\psi(d, \theta) = \max_{j \in J} \{f_j(d, \theta)\} \quad (68)$$

The flexibility test problem in (67) can be simplified and formulated as (69)

$$\begin{aligned} \chi(d) &= \max_{\theta} \psi(d, \theta) \\ \text{s. t. } \psi(d, \theta) &= \max_{j \in J} \{f_j(d, \theta)\}, \quad j \in J \\ \theta &\in T = \{\theta: \theta^L \leq \theta \leq \theta^U\} \end{aligned} \quad (69)$$

In this paper, feasibility analysis is referred to as a procedure to identify the feasible region where  $\psi(d, \theta) \leq 0$ .

## 4.2 Surrogate-based feasibility analysis

Surrogate-based methods are developed to solve feasibility analysis problems when we have black-box constraints in the original model or when the simulation is computationally expensive to run. With a black-box simulation model, we can calculate the output feasibility function values given a set of uncertainty parameter values as input. The surrogate model is thereafter built based on the input and output of feasibility function. To improve the accuracy of the surrogate model, an adaptive sampling method is used to iteratively update the surrogate model. This adaptive sampling

method is considered to be more efficient than space-filling designs because it can automatically search for new points near the feasible region boundary as well as those points in less explored regions. The surrogate model is used for feasibility analysis after it is updated to be sufficiently accurate.

In this work, we propose to use an RBF-based adaptive sampling method for feasibility analysis. The performance of this method is compared with a previously published Kriging-based method. A general overview on these two surrogate models is provided in Section “Surrogate Models”. The adaptive sampling method is described in Section “Radial Basis Function (RBF)”.

## Surrogate Models

### Kriging

The ordinary Kriging model [66] predicts the value at an unknown point from sample data based on a stochastic model of continuous spatial variation. It is based on the assumption that variation is random and spatially dependent, and that the underlying random process has constant mean and variance that only depends on spatial distance. In the literature it is referred to with different names, such as stochastic process model in global optimization [67], Gaussian processes in statistics [225], and Kriging in geostatistics [66]. According to the DACE (Design and Analysis of Computer Experiments) theory [131], a Kriging predictor can be expressed in terms of a regression model and a correlation model, as is shown in Equation (70)

$$\hat{y}(x) = f(x)^T \beta + r(x)^T \gamma \quad (70)$$

where  $f(x)^T$  is a vector of regression model values at  $x$ ;  $r(x)^T$  is a vector of correlation model values at  $x$ . The parameters in the Kriging (including those in the regression and correlation model) can be estimated by maximum likelihood estimation (MLE) [226]. There are many regression models and correlation models that can be used. The models that are considered in this work are

listed in Tables 2 and 3. These models are used because they are shown to perform well in previous literature on surrogate-based feasibility analysis problems [220,65].

Table 2. Regression Models

<b>Model</b>	<b>Form</b>
Constant	$a$
Linear	$a_i x_i + b$
Quadratic	$a_i x_i^2 + b_{ij} x_i x_j + c_i x_i + d$

Table 3. Correlation Models

<b>Model</b>	<b>Form</b>
Exponential	$exp(-\theta_j  d_j )$
Gaussian	$exp(-\theta_j  d_j ^2)$
Linear	$max\{0, 1 - \theta_j  d_j \}$

An important feature of Kriging is that it can provide an estimate of the mean squared error (MSE) of the predictor [131]. This is especially important for adaptive sampling which is discussed in Section “Adaptive sampling”. Note that the MSE estimation from DACE theory is reported to be slightly underestimated with small samples but it appears to have no serious consequences when it is used for global optimization [67].

In this work, the DACE toolbox [226] in Matlab is used to fit the Kriging model. For the Kriging-based method, a model selection phase is implemented as the first step when an initial design of experiment (DOE) is conducted. The goal of this initial model selection phase is to find the combination of regression and correlation model, which leads to lowest prediction error [65]. This

Kriging-based method is used as a comparison to the new RBF-based method described in Section “Radial basis function (RBF)”. Further details on this Kriging-based method can be found in [65].

### Radial basis function (RBF)

Given  $n$  distinctly sampled points  $x_1, x_2, \dots, x_n \in R^d$  with function values known as  $f(x_1), f(x_2), \dots, f(x_n)$ , the RBF surrogate model [222] can be expressed with Equation (71)

$$s_n(x) = \sum_{i=1}^n \lambda_i \phi \left( \|x - x_i\|_2 \right) + b^T x + a \quad (71)$$

where  $\|\cdot\|_2$  represents the Euclidean distance;  $\phi$  is the basis function that can be one of the following forms in (72)

$$\begin{aligned} \phi(r) &= r \quad (\text{linear}) \\ \phi(r) &= r^3 \quad (\text{cubic}) \\ \phi(r) &= r^2 \ln(r) \quad (\text{thin plate spline}) \\ \phi(r) &= e^{-\gamma r^2} \quad (\text{Gaussian}) \\ \phi(r) &= \sqrt{r^2 + \gamma^2} \quad (\text{multiquadric}) \end{aligned} \quad (72)$$

In this work, we choose to use cubic basis function because it can provide relatively accurate approximation without requiring a complex parameter estimation process (compared with Gaussian and multiquadric) [137]. Additionally, cubic RBF is shown to perform better than other basis functions in solving optimization problems in optimization literature [208,227,228].

The model coefficients  $\lambda_i$ ,  $b$ , and  $a$  can be obtained by Equation (73)

$$\begin{pmatrix} \Phi & P \\ P^T & 0 \end{pmatrix} \begin{pmatrix} \lambda \\ c \end{pmatrix} = \begin{pmatrix} F \\ 0 \end{pmatrix} \quad (73)$$

where  $\Phi$  is the  $n$  by  $n$  matrix with  $\Phi_{ij} = \phi \left( \|x - x_i\|_2 \right)$ , and

$$P = \begin{pmatrix} x_1^T & 1 \\ x_2^T & 1 \\ \vdots & \vdots \\ x_n^T & 1 \end{pmatrix}, \lambda = \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_n \end{pmatrix}, c = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_d \\ a \end{pmatrix}, F = \begin{pmatrix} f(x_1) \\ f(x_2) \\ \vdots \\ f(x_n) \end{pmatrix} \quad (74)$$

It is shown that if the rank of  $P$  is  $d + 1$ , where  $d$  is the dimension of the problem, then the matrix  $\begin{pmatrix} \Phi & P \\ P^T & 0 \end{pmatrix}$  is nonsingular [207,222]. Therefore, the coefficients  $\lambda_i$ ,  $b$ , and  $a$  in of  $s_n(x)$  in Equation (71) are uniquely defined with the system (73)(74), which is important for RBF to be used as an interpolation technique.

Unlike Kriging, cubic RBF cannot provide an estimated MSE for the predictor. However, Gutmann [207] introduced a term, namely “ $1/\mu_n(y)$ ”, for cubic RBF that can describe how well the region near an unsampled point has been explored. A short explanation on this term is given below. For a detailed derivation, the authors are referred to the original paper [207].

In [207], RBFs are used to solve global optimization problems by minimizing an objective function  $g_n(y)$ :

$$\min_y g_n(y) = (-1)^{m_0+1} \mu_n(y) [s_n(y) - f_n^*]^2, \quad y \in D \setminus \{x_1, x_2, \dots, x_n\} \quad (75)$$

where  $y$  is an unsampled point;  $\min_y g_n(y)$  can be seen as equivalent to minimizing the “bumpiness” [207] of a RBF surrogate model;  $f_n^*$  is a variable dependent on the surrogate,  $f_n^* \in [-\infty, \min_{y \in D} s_n(y)]$ ;  $m_0$  is 1 for cubic and thin plate spline;  $m_0$  is 0 for linear and multiquadric;  $m_0$  is -1 for Gaussian.  $\mu_n$  is the coefficient of the new term  $\phi(\|x - y\|_2)$  in the surrogate  $s_n(x)$  if a new (unsampled) point  $y$  is added.  $\mu_n$  is calculated as the  $n$ th element of vector  $v$ , and  $v$  can be calculated as:

$$\begin{pmatrix} \Phi_y & P_y \\ P_y^T & 0 \end{pmatrix} v = \begin{pmatrix} 0_n \\ 1 \\ 0_{d+1} \end{pmatrix}, \Phi_y = \begin{pmatrix} \Phi & \phi_y \\ \phi_y^T & 0 \end{pmatrix}, P_y = \begin{pmatrix} P & 1 \end{pmatrix}, (\phi_y)_i = \phi(\|x - x_i\|_2), i = 1, 2, \dots, n.$$

Gutmann [207] showed that we can sample points in unexplored region by choosing  $f_n^* = -\infty$ , which is equivalent to minimizing  $(-1)^{m_0+1}\mu_n(y)$ . In the case of cubic RBF, it is equivalent to minimizing  $\mu_n(y)$  which is also equivalent to maximizing  $1/\mu_n(y)$ . With this finding we can see that  $1/\mu_n(y)$  can be used to indicate how well the region near the unsampled  $y$  has been explored. The larger  $1/\mu_n(y)$  is, the less its nearby region has been explored.

Furthermore, we notice the following interesting properties that have been proved in [207]:

- $1/\mu_n(y)$  is always positive when  $y$  is at an unsampled point
- $1/\mu_n(y)$  is defined to be zero at sampled points where  $y \in \{x_1, x_2, \dots, x_n\}$ .

These properties of  $1/\mu_n(y)$  are also shared by the estimated MSE for Kriging predictor. Therefore, we propose to use  $1/\mu_n(y)$  as an indicator to show how well a region near a point  $y$  has been sampled. In the rest of this paper, we will call  $1/\mu_n(y)$  as an “error-indicator” for cubic RBF. This is used in the adaptive sampling for the RBF-based method for feasibility analysis.

### **Adaptive sampling**

Adaptive sampling can be used to improve the accuracy of the surrogate model without the need to exhaustively sample the whole design space of the original simulation model [65]. Therefore, the method is considered to be more efficient than space-filling designs and is usually used in the optimization literature [220]. From the perspective of surrogate-based optimization, the adaptive sampling should direct search towards the region where surrogate model is minimized (in case of minimization problems) and also towards the region that has not been well explored. The goal of balancing global search and local search is to find the global minimum [67]. Similarly, from the perspective of surrogate-based feasibility analysis, the adaptive sampling should direct search

towards the feasible region boundaries and also towards unexplored regions. This balanced search can be achieved by maximizing a modified expected improvement (EI) function [220,65] as is shown in Equation (76)

$$\max_x EI_{feas}(x) = s * \phi\left(\frac{-y}{s}\right) = s * \frac{1}{\sqrt{2\pi}} e^{-0.5\left(\frac{y^2}{s^2}\right)} \quad (76)$$

where  $EI_{feas}(x)$  is the modified EI function value at  $x$ ;  $y$  is the surrogate model predictor;  $s$  is the standard error of the predictor;  $\phi(\cdot)$  represents the standard normal density function. To see how this modified EI can help balance local search and global search for feasibility analysis, we can take its partial derivatives:

$$\frac{\partial EI_{feas}}{\partial s} = \frac{1}{\sqrt{2\pi}} \left( e^{-\frac{0.5y^2}{s^2}} + \frac{e^{-\frac{0.5y^2}{s^2}} y^2}{s^2} \right) > 0 \quad (77)$$

$$\frac{\partial EI_{feas}}{\partial y} = -\frac{1}{\sqrt{2\pi}} \left( \frac{e^{-\frac{0.5y^2}{s^2}} y}{s} \right) \quad (78)$$

From Equation (77) we can see  $\frac{\partial EI_{feas}}{\partial s}$  is always larger than 0. This indicates that  $EI_{feas}$  increases monotonically with the prediction error  $s$ . Thus maximizing  $EI_{feas}$  is in favor of searching for new points with high uncertainty, which is to direct search in unexplored region (namely global search).

From Equation (78) we notice that  $\frac{\partial EI_{feas}}{\partial y}$  is of opposite sign with the surrogate value  $y$ . When  $y$  is less than zero,  $\frac{\partial EI_{feas}}{\partial y}$  is positive, meaning that  $EI_{feas}$  increases as  $y$  increases and gets close to zero. Similarly, when  $y$  is larger than zero,  $\frac{\partial EI_{feas}}{\partial y}$  is negative, meaning that  $EI_{feas}$  increases as  $y$  decreases and gets close to zero. In other words, Equation (78) indicates that maximizing  $EI_{feas}$  favors to sample points where surrogate predictor is close to zero, which is the feasible region boundary predicted by the surrogate model (namely local search). Thus, we can see that modified EI function (76) is larger where prediction uncertainty is high and also near the feasible region

boundary of the surrogate model. Therefore, the properties of  $El_{feas}$  serve our objective to use this function for surrogate-based feasibility analysis.

In the Kriging based method, the prediction error  $s$  is equal to  $\sqrt{MSE}$ , where  $MSE$  is the estimated prediction variance. For the RBF-based method, even though we do not have the estimation for  $MSE$ , we can use an “error-indicator” ( $1/\mu$ ) as explained in section “Radial basis function (RBF)”. Instead of using  $1/\mu$  directly, at the first step we introduce a scale factor to balance the magnitude of  $\sqrt{1/\mu}$  with that of surrogate value  $y$ . In practice, we find that the RBF-based method works best if we let

$$scale = \frac{\max(1/\mu_0)}{\max(RBF_0)^2 / numIniPts^2} \quad (79)$$

where  $\max(1/\mu_0)$  is the maximum value of  $(1/\mu_0)$  with the initial cubic RBF model;  $\max(RBF_0)$  is the maximum value of the initial cubic RBF model;  $numIniPts$  is the number of initial sample points. With this scale factor, we can substitute prediction error  $s$  with  $\sqrt{\frac{1/\mu}{scale}}$  for the RBF-based method, as is shown in Equation (80).

$$\sqrt{\frac{1/\mu}{scale}} = \sqrt{\frac{1/\mu}{\max(1/\mu_0)} * \frac{\max(RBF_0)}{numIniPts}} \quad (80)$$

In Equation (80),  $\max(1/\mu_0)$  and  $\max(RBF_0)$  are used to balance the order of magnitude for  $\sqrt{1/\mu}$  to be similar to surrogate model value.  $numIniPts$  is introduced because intuitively the prediction uncertainty should be lower with more initial sample points.

The Kriging-based algorithm and the RBF-based algorithm are shown in Figure 6. For both methods, initial sample points are first selected based on a design of experiment (DOE). In this paper, a rectangular grid is used for the initial DOE because it was shown to have good performance for different feasibility analysis test problems [220,65]. After initial points are selected, the

Kriging-based method uses a model selection step to find the combination of regression model and correlation model that provides the smallest prediction variance. The initial Kriging model is then fit with the DACE software package [226] in Matlab. The RBF-based method uses the initial sample points to build an initial cubic RBF model and evaluates the scale factor for the error indicators. Then during the model improvement step, new points are added iteratively by maximizing the modified EI function. At each iteration a local optimization solver, “tomlab/conopt” [229], is used to find the optimum solution. To find a relatively good initial guess point for the local optimization solver, the modified expected EI function is evaluated at 1000 different points sampled over the whole uncertainty space using Latin Hypercube DOE. The point with largest modified EI value is selected as the initial guess for the local optimization solver. The adaptive sampling will terminate if the iteration exceeds an upper bound ( $iter_{max}$ ) defined by users.

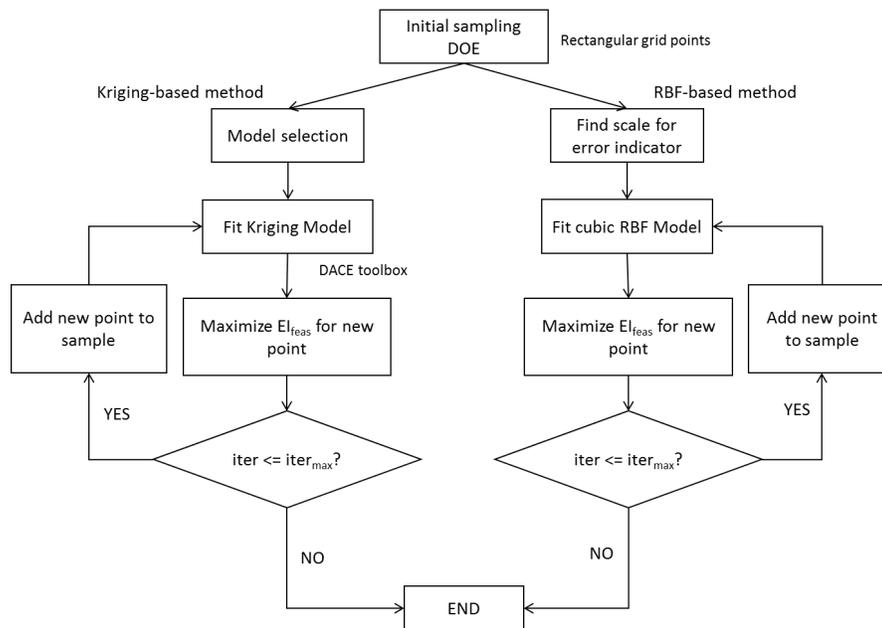


Figure 6. Framework for Kriging-based and RBF-based feasibility analysis algorithm

### 4.3 Surrogate Model Accuracy

To measure how well the surrogate-based methods perform for feasibility analysis, we use three metrics, namely percentage of Correct Feasible region (CF%), percentage of Correct InFeasible

region (CIF%), and percentage of Not Conservative feasible region (NC%). These three metrics are illustrated with Figure 29.

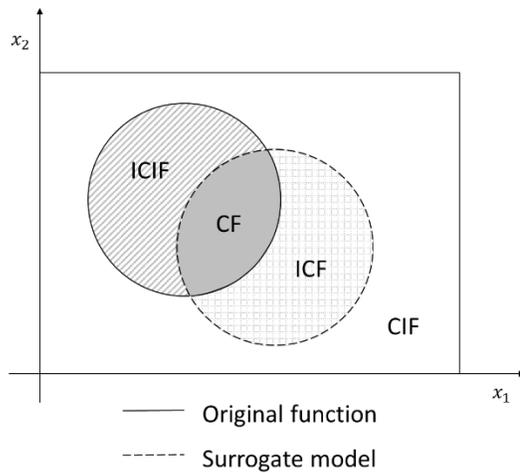


Figure 7. Illustrating figure for the model accuracy metrics

In Figure 7, the rectangle area represents the whole range of uncertain parameters of two dimensions; the solid circle represents the feasible region of the original function; and the dashed circle represents the feasible region predicted by the surrogate model. Ideally, we want these two feasible regions to overlap, meaning that the surrogate is 100% accurate. However, the surrogate model is always an approximation of the original model. For the purpose of calculating accuracy metrics, the whole range of uncertainty parameters is divided into four regions: CF (Correct Feasible region shaded in gray) represents the overlapped feasible regions; CIF (Correct InFeasible region in white) represents the overlapped infeasible region; ICF (Incorrect Feasible region shaded with dotted grids) represents the area where it's feasible in the surrogate model but infeasible in the original function; ICIF (Incorrect InFeasible region shaded with upward diagonal lines) represents the area where it's infeasible in the surrogate model but is feasible in the original function.

With these definitions, we can then define the three metrics for model accuracy in Equation (81)

$$\begin{aligned}
 CF\% &= \frac{CF}{CF + ICIF} * 100 \\
 CIF\% &= \frac{CIF}{CIF + ICF} * 100 \\
 NC\% &= \frac{ICF}{ICF + CF} * 100
 \end{aligned}
 \tag{81}$$

CF%: the percentage of feasible region in original function, which has been correctly discovered by the surrogate model;

CIF%: the percentage of infeasible region in original function, which has been correctly discovered by the surrogate model;

NC%: the percentage of feasible region in surrogate model, which has been overestimated by the surrogate model;

The first two metrics (CF% and CIF%) describe how well the uncertainty parameter space has been correctly explored and classified with respect to feasibility. The third metric (NC%) evaluates the conservativeness of the feasible region predicted by surrogate model. The goal of surrogate-based feasibility analysis is to predict feasible regions thoroughly and conservatively. Therefore, by definition, we can say surrogate-based methods can accurately approximate the feasible region if CF% and CIF% are close to 100%, and NC% is close to 0.

#### 4.4 Computational studies

In the following sections, we first present a thorough analysis of the performance of RBF-based and Kriging-based method with case studies on low dimensional test problems, which consist of four 2D test problems and one 3D test problem. All of the test problems are nonlinear and nonconvex, and are considered to be difficult to accurately identify the feasible region. We're especially interested in low dimensional problems because they can be easily visualized and are more common in real-life cases where feasibility analysis is needed, e.g. pharmaceutical process

unit models [65]. Then we also compare the performance of two surrogate-based methods on high dimensional test problems (5D and 6D) with nonlinear and/or linear constraints. Since these high dimensional problems cannot be easily visualized, only the model accuracy metrics are compared. The idea is to show the potential and limit of the current two surrogate-based approaches in high dimensional cases.

### Low dimensional test problems

#### *Modified Branin function (branincon)*

“branincon” is a 2D test problem [65]. It is defined by a single constraint shown in Equation (82). From the feasibility function value contour plot in Figure 8, we can see there are three disjoint feasible regions, which are shaded with dots.

$$\begin{aligned}
 & a(x_2 - bx_1^2 - cx_1 - d)^2 + h(1 - ff) \cos x_1 - 5 + h \leq 0 \\
 & -5 \leq x_1 \leq 10 \\
 & 0 \leq x_2 \leq 15
 \end{aligned} \tag{82}$$

$$a = 1; b = \frac{5.1}{4\pi^2}; c = \frac{5}{\pi}; d = 6; h = 10; ff = \frac{1}{8\pi}$$

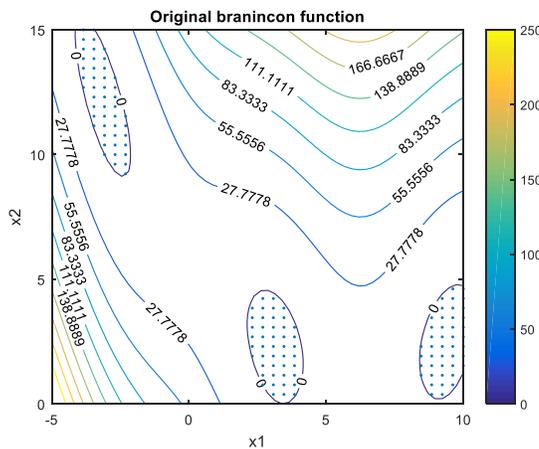


Figure 8. Contour of Feasibility function for branincon. (Feasible regions are shaded with dots.)

#### *Nonlinear nonconvex function (ex3)*

“ex3” is a 2D test problem [65] with one linear constraint and two nonlinear constraints which are shown in Equation (83). It has one nonconvex feasible region, which is shaded with dots in Figure 29.

$$\begin{aligned}
 -2x_1 + x_2 - 15 &\leq 0 \\
 \frac{x_1^2}{2} + 4x_1 - x_2 - 5 &\leq 0 \\
 \frac{-(x_1 - 4)^2}{5} - \frac{x_2^2}{0.5} + 10 &\leq 0 \\
 -10 \leq x_1 \leq 5 \\
 -15 \leq x_2 \leq 15
 \end{aligned} \tag{83}$$

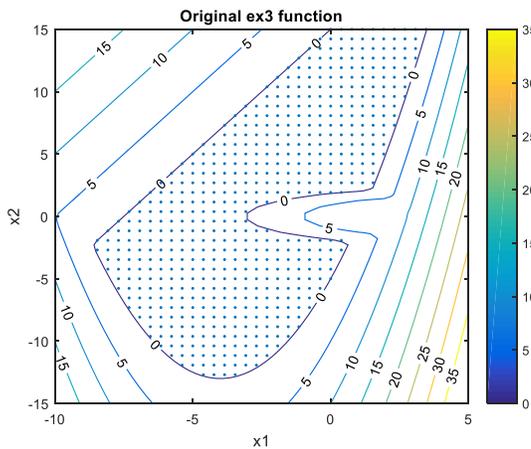


Figure 9. Contour of Feasibility function for ex3. (Feasible region is shaded with dots.)

#### *Constrained Sasena function (sasenacon)*

“sasenacon” is a 2D test problem taken from [230]. Only the constraints are considered when it is used as a feasibility analysis test problem. The test problem is shown in Equation (84), and the contour of feasibility function is shown in Figure 29. For “sasenacon”, there are two disjoint feasible regions which are shaded with dots.

$$(x_1 - 3)^2 + (x_2 + 2)^2 * \exp(-x_2^7) - 12 \leq 0 \tag{84}$$

$$10x_1 + x_2 - 7 \leq 0$$

$$(x_1 - 0.5)^2 + (x_2 - 0.5)^2 - 0.2 \leq 0$$

$$0 \leq x_1 \leq 1$$

$$0 \leq x_2 \leq 1$$

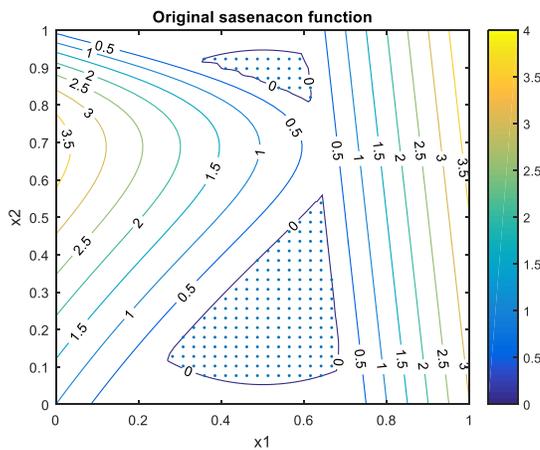


Figure 10. Contour of feasibility function for sasenacon. (Feasible regions are shaded with dots.)

#### *Modified Camelback function (camelback)*

“camelback” is a 2D test problem from [231]. The original objective function is modified into a nonlinear constraint for the purpose of feasibility analysis. The test problem is shown in Equation (85), and the contour plot of feasibility function is shown in Figure 29. There are two large feasible regions and two small feasible regions which are shaded with dots.

$$\left(4 - 2.1x_1^2 + \frac{x_1^4}{3}\right) * x_1^2 + x_1x_2 + (-4 + 4x_2^2)x_2^2 \leq 0$$

$$-3 \leq x_1 \leq 3$$

$$-2 \leq x_2 \leq 2$$
(85)

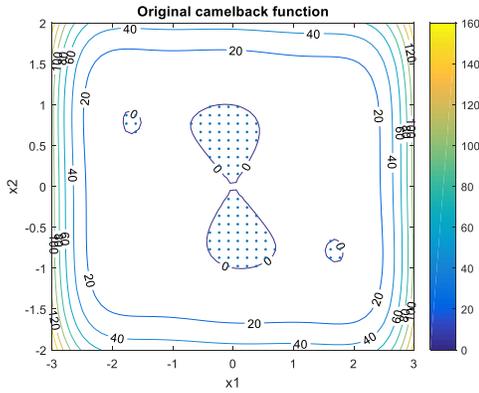


Figure 11. Contour of feasibility function for camelback. (Feasible regions are shaded with dots.)

*Modified quadratic constrained function (qcp4con)*

“qcp4con” is a 3D test problem from [64]. Only the constraints are considered for the purpose of feasibility analysis. The test problem is shown in Equation (86), and the 3D plot of feasible region boundary is shown in Figure 29. The feasible region is the area within the feasible region boundary hull, which is shaded with dots.

$$x_1 + x_2 + x_3 - 4 \leq 0$$

$$3x_2 + x_3 - 6 \leq 0$$

$$-\left(x' A' A x - 2y' A x + \|y\|^2 - 0.25\|b - z\|^2\right) \leq 0$$

$$x^{low} = [0, 0, 0]$$

$$x^{up} = [2, 3, 3]$$

where

$$A = [0, 0, 1; 0, -1, 0; -2, 1, -1]$$

$$b = [3; 0; -4],$$

$$y = [1.5; -0.5; -5]$$

$$z = [0; -1; -6]$$

(86)

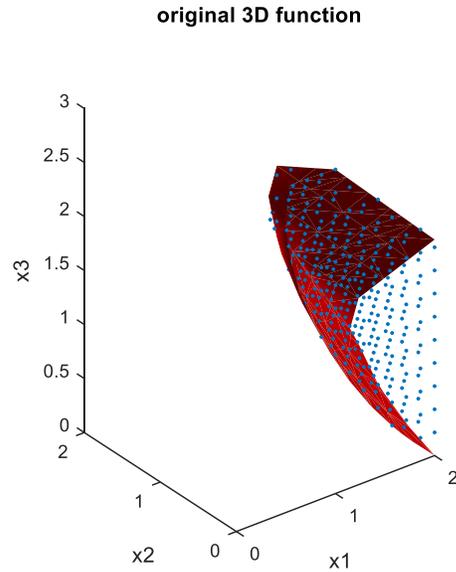


Figure 12. 3D plot of feasible region boundary for qcp4con. (Feasible regions are shaded with dots.)

### ***Results on the performance of two surrogate-based methods***

The performance of the Kriging-based and RBF-based methods on test problems are compared and the results are shown in Table 4 and Figure 13 to Figure 17. For both methods, the rectangular grid sampling plan is used to build the initial surrogate model. With respect to the number of initial sample points, 49 ( $=7^2$ ) initial points are used for two-dimensional problem, which is in accordance with that in [65]. Considering the increase in dimension, the 3D test problem uses 64 ( $=4^3$ ) initial points, which is slightly more than that for two-dimensional problems. After the initial surrogate is built based on initial points, the adaptive sampling is then used to iteratively sample new points and improve the surrogate accuracy. The model accuracy metrics are calculated after 100 iterations of adaptive sampling.

Table 4. Results on 2D and 3D test problems

test problem	surrogate	initial surrogate accuracy			surrogate accuracy after 100 iterations		
		CF%	CIF%	NC%	CF%	CIF%	NC%
branincon	RBF	57.89	99.79	3.75	99.88	100.00	0.00
	Kriging	26.10	100.00	0.00	98.22	99.87	1.43
ex3	RBF	93.53	98.55	2.63	99.31	99.76	0.41
	Kriging	91.53	98.25	3.22	98.57	99.43	0.99
sasenacon	RBF	75.83	98.83	9.09	98.20	99.91	0.61
	Kriging	58.03	98.06	17.85	88.74	99.55	3.19
camelback	RBF	57.06	84.10	79.98	99.69	99.93	1.07
	Kriging	99.39	76.21	77.44	94.79	99.30	9.52
qcp4con	RBF	81.52	98.94	12.78	97.94	99.81	2.09
	Kriging	84.80	96.74	30.76	96.34	99.60	4.53

From the results on “branincon” and “ex3”, after 100 iterations of adaptive sampling, we notice that both the Kriging-based and RBF-based methods can correctly identify the feasible and infeasible regions with high accuracy (over 98% of the feasible regions and 99% of the infeasible regions are correctly discovered). They also exhibit a low percentage of overestimated feasible regions (less than 2% of the feasible regions are incorrectly predicted). It should be noticed that the RBF-based method is slightly more accurate than the Kriging-based approach in these two test problems. The final surrogate models after adaptive sampling are shown in Figure 13 and Figure 14. We can observe that the adaptive sampled points (noted as circle points) are mostly located near the feasible region boundaries and with a small portion located in some other less explored regions. This is in accord with the analysis on the properties of modified expected improvement function in Section “Adaptive sampling”.

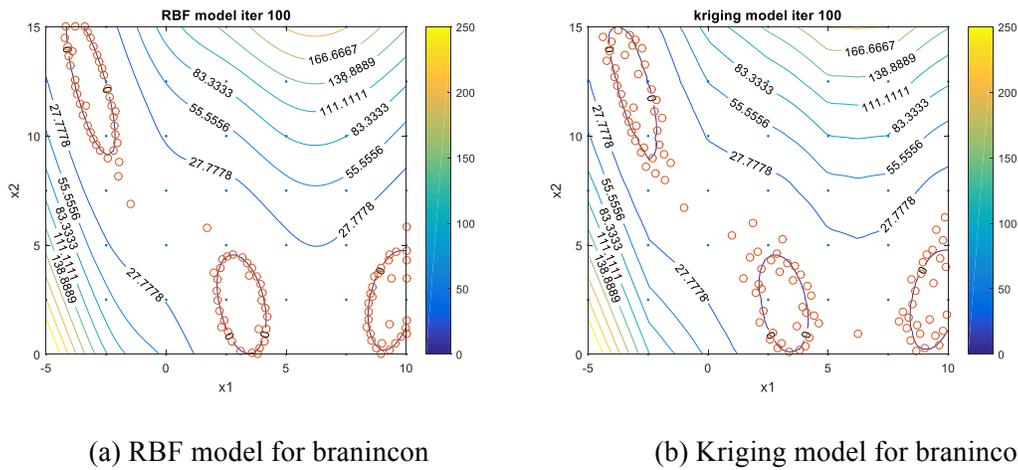


Figure 13. Contour of surrogate models for branincon. (a) RBF model for branincon; (b) Kriging model for branincon. (Initial sample points are noted with dots. Adaptive sampling points are noted with circle points.)

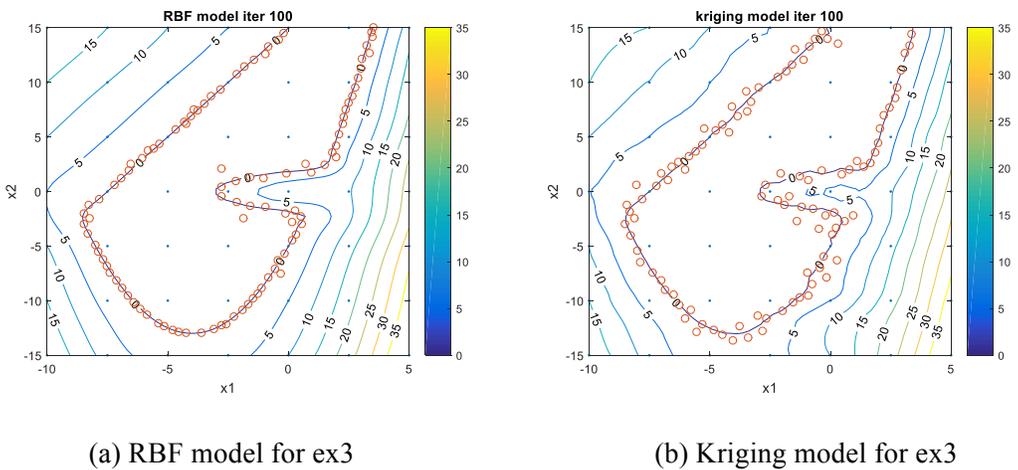
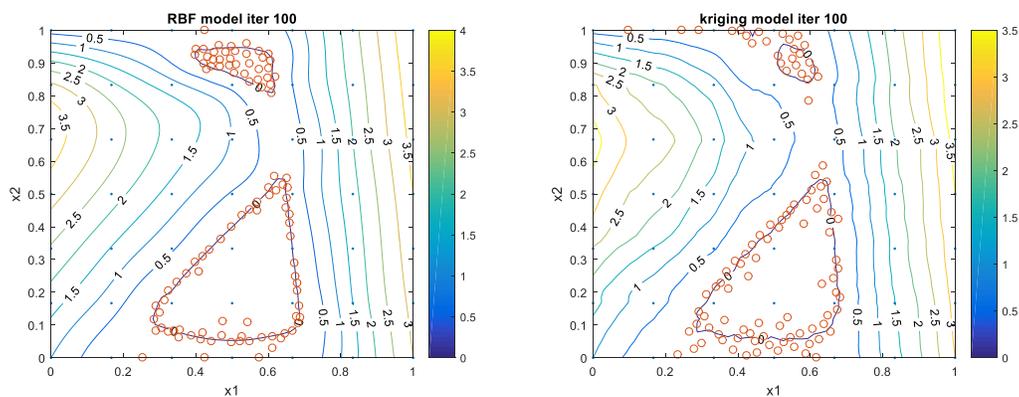


Figure 14. Contour of surrogate models for ex3. (a) RBF model for ex3; (b) Kriging model for ex3. (Initial sample points are noted with dots. Adaptive sampling points are noted with circle points.)

From the results of the other two case studies on “sasenacon” and “camelback”, the RBF-based method shows significantly better performance than the Kriging-based method. For “

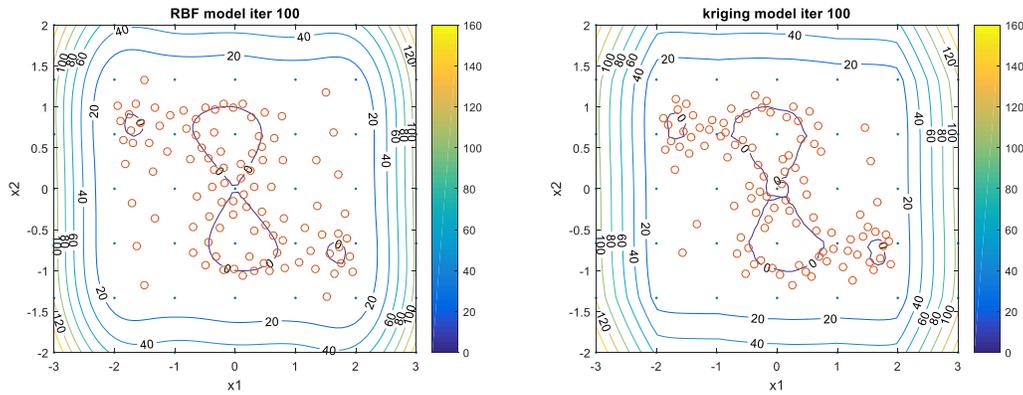
“sasenacon” , the RBF-based method (98.20% as CF%) is much more accurate in predicting the feasible region than the Kriging-based method (88.74% as CF%) by over 9%. Also the RBF-based method is more conservative in the prediction of feasible regions. For “camelback” , the advantage of the RBF-based method over the Kriging-based method is also very obvious, where the RBF-based method can correctly identify 99.69% of the feasible region compared to 94.79% from the Kriging-based method. In addition, the RBF-based method is also much more conservative (1.07% as NC%) than the Kriging-based method (9.52% as NC%). The final surrogate models after adaptive sampling are shown in Figure 15 and Figure 16. From Figure 15, it can be seen that the relatively small feasible region is not well explored by the Kriging-based method with 100 adaptive sampling points. A large portion of adaptive sampled points are located near the boundary of the relatively large feasible regions, while only a small portion is located near the small feasible region boundary. This explains why Kriging-based method cannot give accurate prediction within 100 iterations of adaptive sampling. While for the RBF-based method, there appears to be a better balanced sampling in exploring both the larger and the smaller feasible regions. This can be attributed to the use of scale factor in RBF-based method.



(a) RBF model for sasenacon

(b) Kriging model for sasenacon

Figure 15. Contour of surrogate models for sasenacon. (a) RBF model for sasenacon; (b) Kriging model for sasenacon. (Initial sample points are noted with dots. Adaptive sampling points are noted with circle points.)

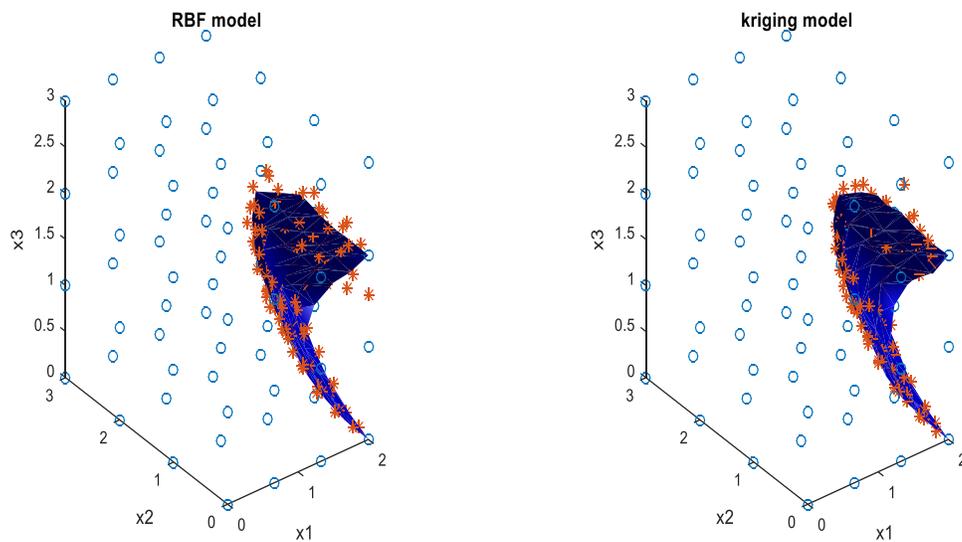


(a) RBF model for camelback

(b) Kriging model for camelback

Figure 16. Contour of Surrogate models for camelback. (a) RBF model for camelback; (b) Kriging model for camelback. (Initial sample points are noted with dots. Adaptive sampling points are noted with circle points.)

For the 3D test problem “qcp4con”, the RBF-based still shows higher accuracy in identifying feasible region boundaries and better conservativeness in predicting feasible regions. The final surrogate models after adaptive sampling are shown in Figure 17. In the figure, the circle points represent the initial 64 sampling points, and the asterisk points are the adaptive sampling points.

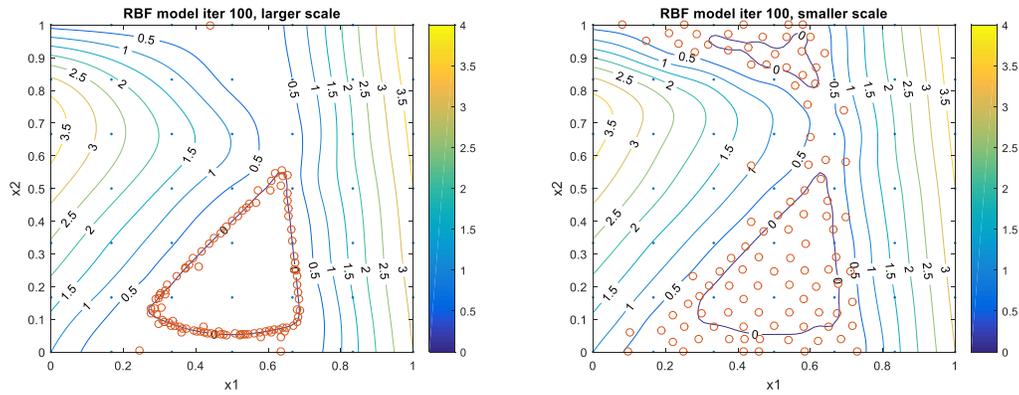


(a) RBF model for qcp4con

(b) Kriging model for qcp4con

Figure 17. 3D plot of feasible region boundaries from surrogate models for qcp4con. (a) RBF model for qcp4con; (b) Kriging model for qcp4con. (Initial sample points are noted with circle points. Adaptive sampling points are noted with asterisk points.)

To illustrate the influence of the scale factor on the balance between local search and global search during adaptive sampling of the RBF-based method, we use the “sasenacon” function as an example. Based on the initial surrogate RBF model constructed with 49 grid sample points, in the first case, we try a larger scale factor value, which is 100 times the proposed one; and in the second case, we try a smaller scale factor value, which is 0.01 times the proposed one. With such changes, the values of “ $s$ ” in the modified EI function will be smaller, namely 0.1 times the original value, in the first case; and “ $s$ ” will be larger, namely 10 times the original value, in the second case. After 100 iterations of adaptive sampling, the final surrogate RBF models are shown in Figure 18. From Figure 18 (a), we notice that the adaptive sampling points are closely located near the bottom feasible region, but fails to explore the top feasible region. This indicates that with a larger scale factor, local search may be preferred over global search, and thus we face the risk of over trusting the surrogate model and failing to identify all the disjoint feasible regions. On the other, we notice from Figure 18 (b) that although both feasible regions are discovered, adaptive sampling points span a rather extensive area of the input space, and fails to exploit the feasible boundaries by sampling points near the boundaries. This indicates that with a smaller scale factor, global search may be preferred over local search, and thus we may face the risk of over exploring the input space and waste some sampling points in unnecessary regions. Such influences of the scale factor are also found in other test problems, and thus further similar results are not included here for conciseness.



(a) RBF model using a larger scale factor

(b) RBF model using a smaller scale factor

factor

Figure 18. Contour of surrogate models for sasenacon using different scale factors. (a) RBF model using a larger scale factor; (b) RBF model using a smaller scale factor (Initial sample points are noted with dots. Adaptive sampling points are noted with circle points.)

To better understand the trends of surrogate accuracy for feasible region boundaries during adaptive sampling, and also to compare the performance of two surrogate-based methods by iteration of adaptive sampling, the accuracy metrics during adaptive sampling are checked every ten iterations in previous case studies. The results are shown in Figure 19 to Figure 23. From these comparisons, we can see the general trend of surrogate accuracy is to increase with more iterations of adaptive sampling. Namely CF% increases by iteration; CIF% increases by iteration; and NC% decreases by iteration. But there may exist some oscillations during the adaptive sampling process. The oscillations are because during the first few tens of iterations, the surrogate models are not ready to be trusted to predict the feasible region boundaries. In other words, as adaptive sampling proceeds, the newly explored regions may not have been exploited enough to provide accurate predictions on the feasible region boundaries. For example, for the case study of Kriging-based method on “branincon” in Figure 19, comparing the results at iteration 10 with those at iteration 0, more regions get explored and more feasible regions are discovered (CF% increases in Figure 19 a).

However, among the newly discovered feasible regions, a small portion is actually overestimated (NC% increases in Figure 19 c), which is why the percentage of correctly discovered infeasible regions decreased (Figure 19 b). As the surrogate model gets updated with more adaptive sampling points, the accuracy of predicting feasible region boundaries increases, and this causes the oscillations to dampen.

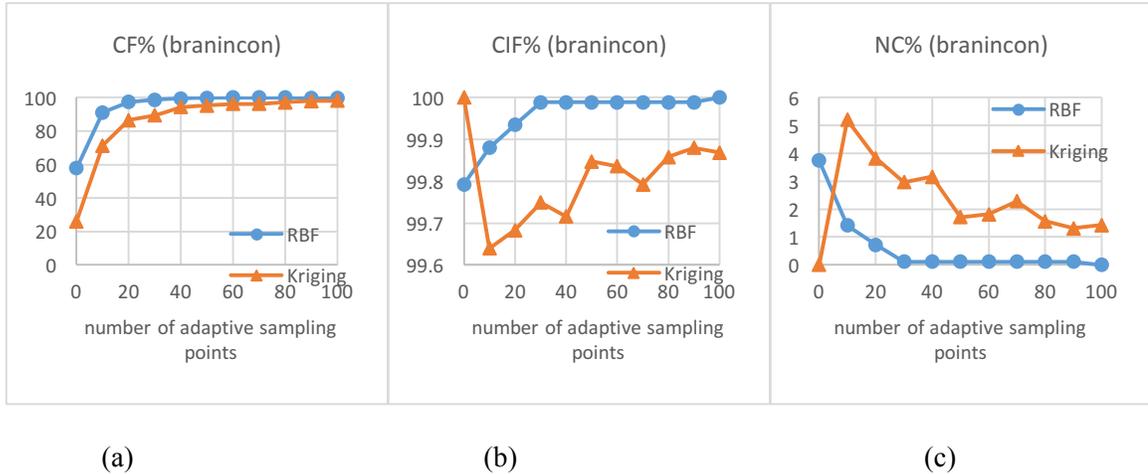


Figure 19. Comparison of surrogate accuracy by iteration of adaptive sampling for branincon. (a) CF% (b)CIF (c) NC%

In terms of the comparison of accuracy for the RBF-based method and the Kriging-based method, the RBF-based method is generally more accurate than the Kriging-based method during the whole process of adaptive sampling (Figure 20, Figure 21, Figure 23), or after a few number of iterations of adaptive sampling (Figure 19, Figure 22). This indicates that the RBF-based method can be more efficient in that it requires fewer numbers of adaptive sampling points to improve the surrogate accuracy to a relatively high level. This is especially obvious for the case study of “sasenacon” in Figure 22, where the CF% of Kriging-based method at iteration 100 is merely approximately equal to that of RBF-based method at iteration 50, while with similar values of CIF% and NC%. The demonstrated efficiency of RBF-based method is quite important for surrogate-based feasibility analysis methods, especially when the black-box function is computationally expensive.

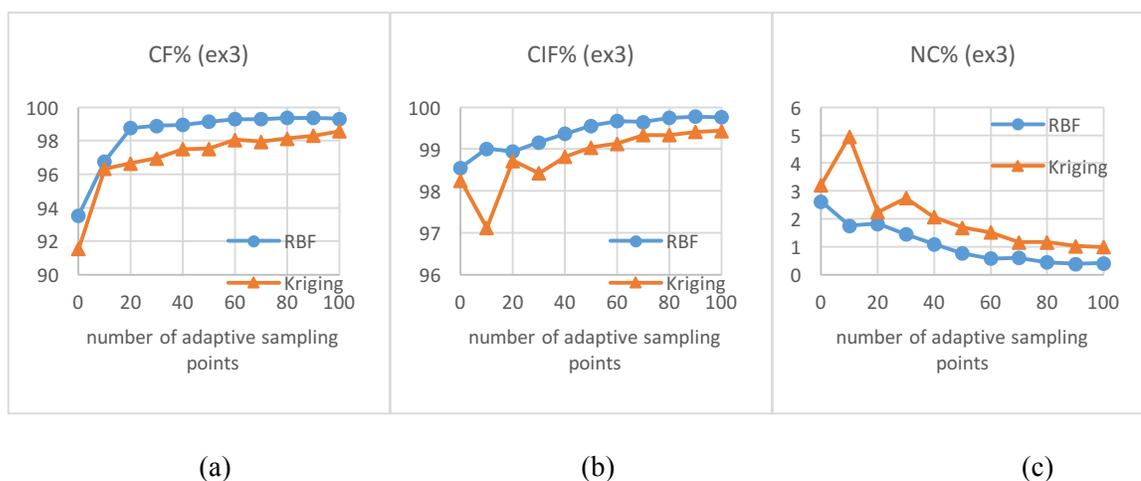


Figure 20. Comparison of surrogate accuracy by iteration of adaptive sampling for ex3. (a) CF% (b)CIF (c) NC%

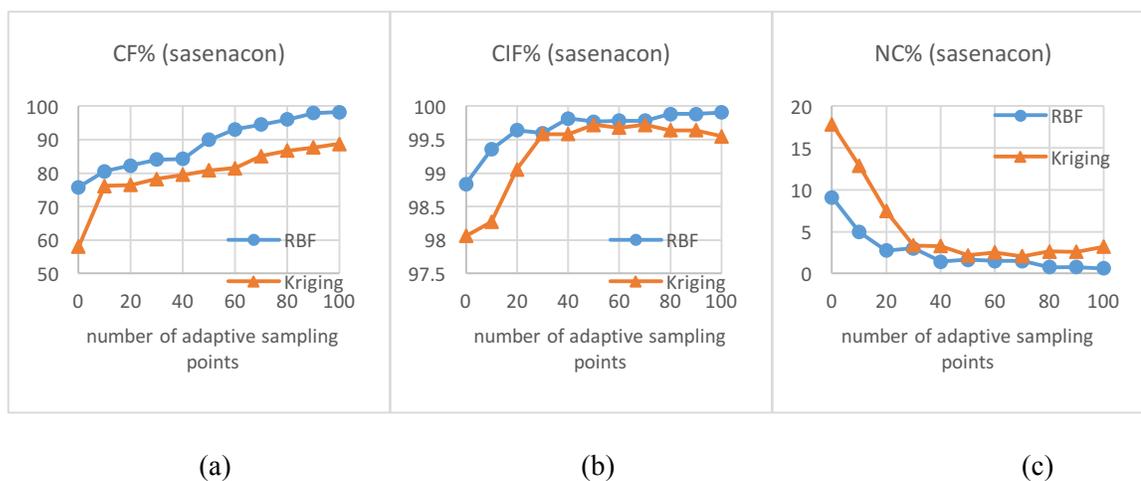


Figure 21. Comparison of surrogate accuracy by iteration of adaptive sampling for sasenacon. (a) CF% (b)CIF (c) NC%

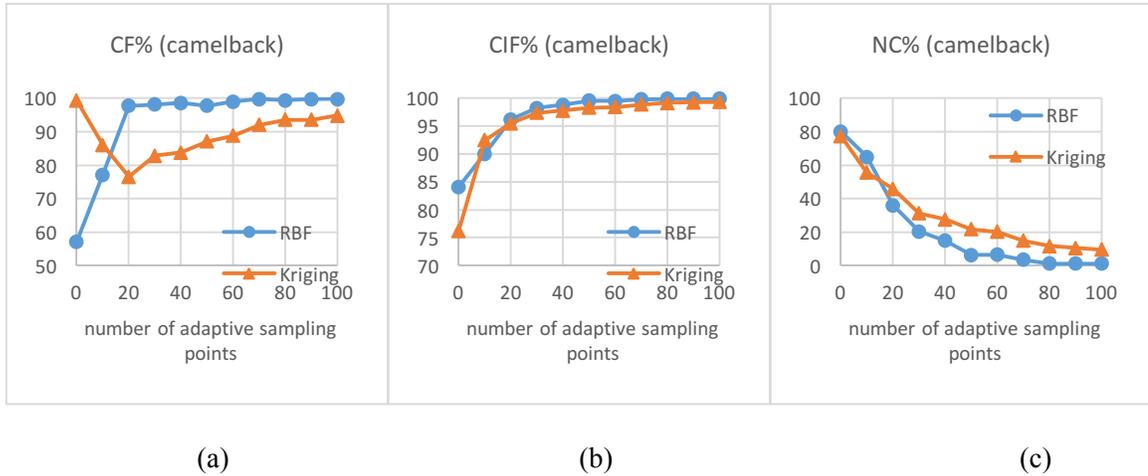


Figure 22. Comparison of surrogate accuracy by iteration of adaptive sampling for camelback. (a) CF% (b)CIF (c) NC%

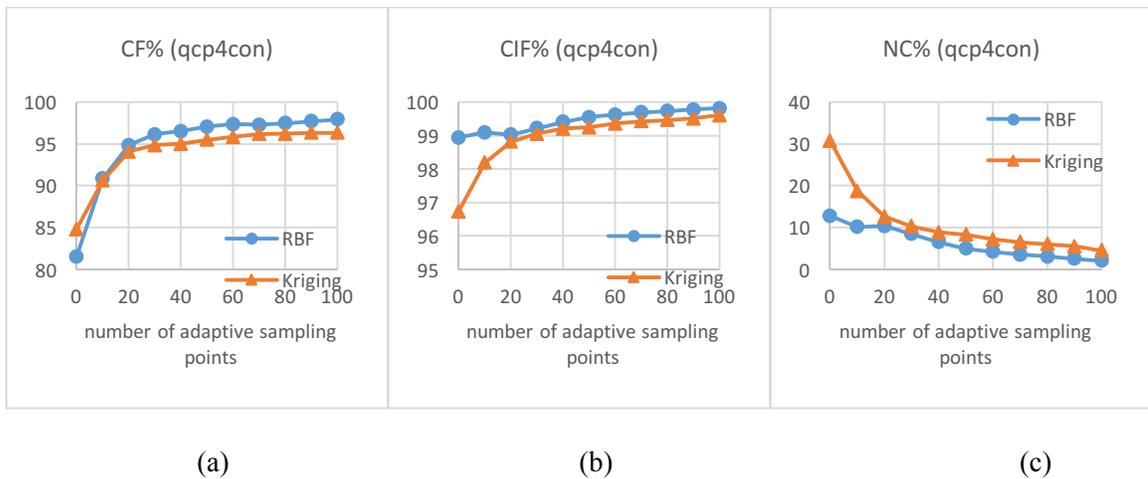


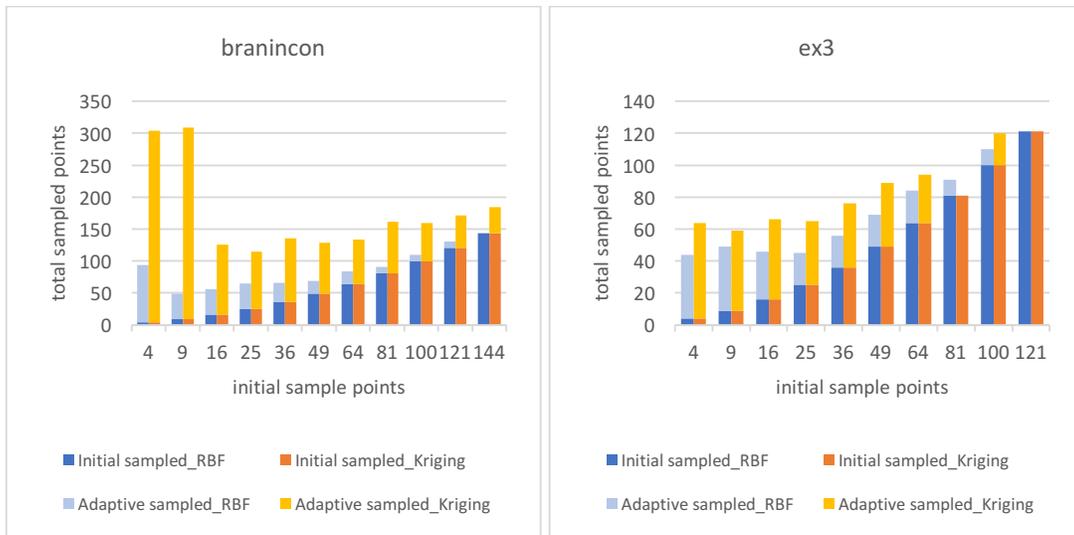
Figure 23. Comparison of surrogate accuracy by iteration of adaptive sampling for qcp4con. (a) CF% (b)CIF (c) NC%

Another important aspect about surrogate-based adaptive sampling method is its dependence on the number of initial sample points. This can be seen as to answer two questions:

- 1) If fewer initial sample points are used, will the adaptive sampling method still be able to correctly identify the feasible regions?

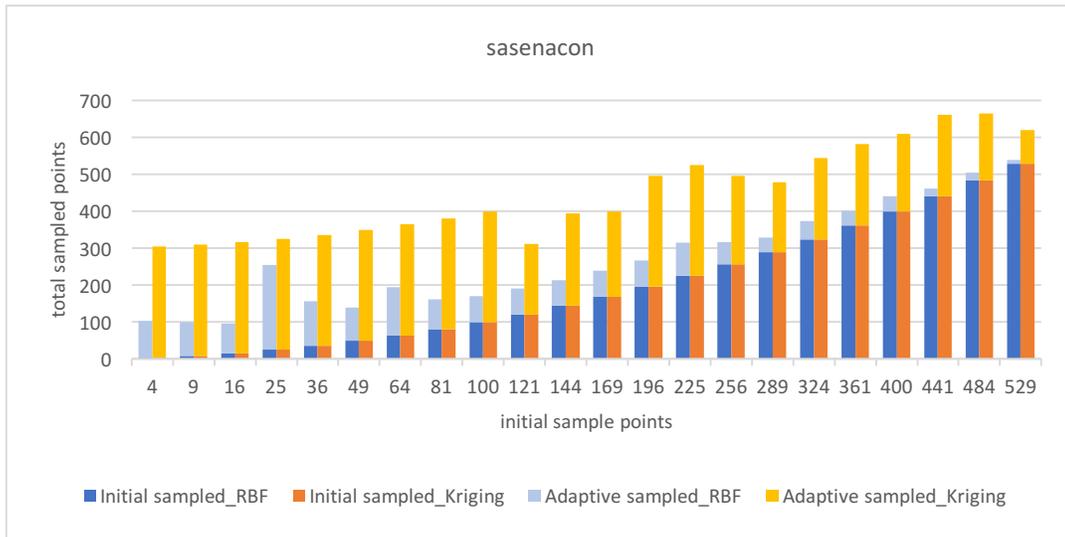
- 2) If more initial sample points are used, will it reduce the number of adaptive sampling points required for accurate prediction and thus reduce the total number of sampling points?

To understand this, both the RBF-based method and the Kriging-based method are tested with a range of various numbers of initial sample points. Starting from different initial surrogate models, it is desired to know how many adaptive sampling points are needed to improve the different initial surrogate models' accuracy to a similarly high level. For comparison purpose, a common stopping criterion for this model accuracy level is set as “CF% > 97% && CIF% > 97% && NC% < 3%”. This relatively mild model accuracy criterion is chosen to decrease computational cost. Meanwhile, it is also sufficient enough to guarantee that for each low dimensional test problem in this paper all feasible regions can be discovered thoroughly and conservatively. In practice, during the process of adaptive sampling, the surrogate model accuracy is only checked every 10 iterations due to its relatively high computational cost. The adaptive sampling will terminate if it is found that the surrogate model accuracy meets the accuracy criterion or the number of iterations of adaptive sampling exceeds the upper bound “ $iter_{max}$ ”. Here “ $iter_{max}$ ” is set as 300 to reduce the computational cost. The results are shown in Figure 24.

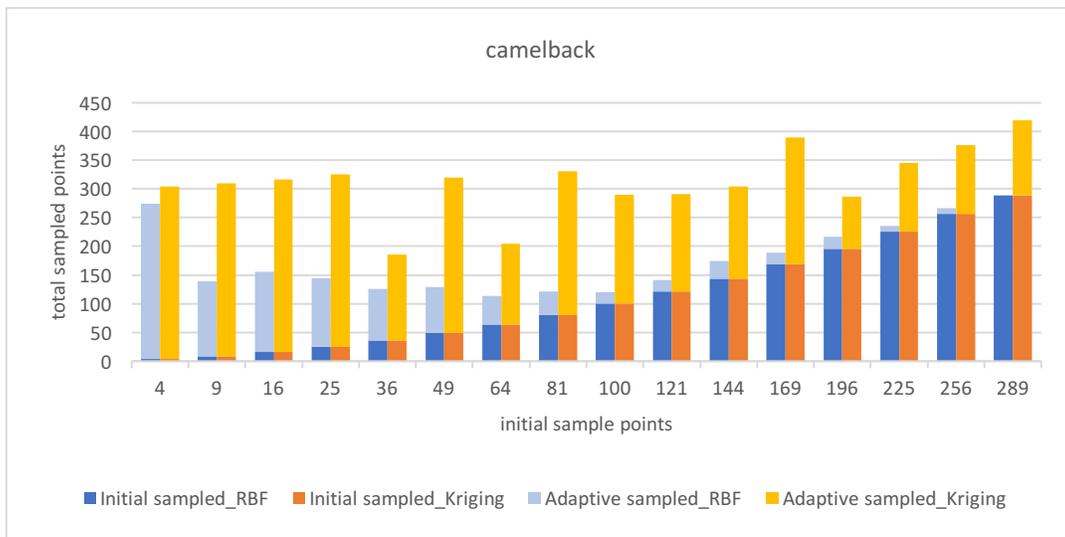


(a)

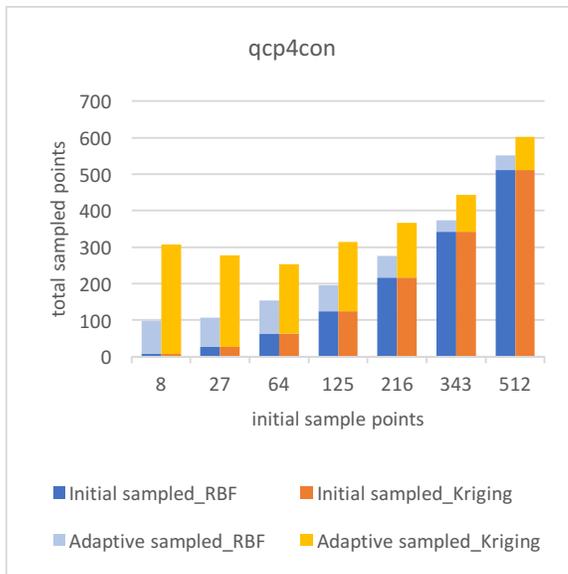
(b)



(c)



(d)



(e)

Figure 24. Number of adaptive sampling points v.s. number of initial sample points using the RBF-based and the Kriging-based method. (a) branincon; (b) ex3; (c) sasenacon; (d) camelback; (e) qcp4con

From the results, it can be noticed that there is a general trend that the number of adaptive sampling points decreases as initial sample points increases. However, there exist a few exceptions. These exceptions are more obvious in Kriging-based method, especially for “sasenacon” and “camelback” test problems. One possible explanation for such exceptions is the poor choice of the initial sample points. Because the test problems are all highly nonlinear and nonconvex, poor selection of the initial sample points’ location may cause the initial surrogate model to misrepresent the original function and mislead adaptive sampling, especially when there are multiple disjoint feasible regions in the test problem. This indicates the necessity of trying different space-filling sampling plans, such as Latin Hypercube Sampling (LHS), when selecting initial sample points.

Additionally, it is most obvious that the RBF-based method is much more efficient than Kriging-based in that the former method requires much fewer adaptive sampling points than the latter method in almost all cases. Note that in all cases, the RBF-based method can meet the accuracy criterion ( $CF\% > 97\%$  &&  $CIF\% > 97\%$  &&  $NC\% < 3\%$ ) within 300 iterations. However, the Kriging-based method cannot achieve this in several cases, especially when there are fewer initial sample points. There are two possible explanations for this. First, RBF can usually give a smoother surface than Kriging, as can be noticed from Figure 13 to Figure 16. This indicates RBF may be a more suitable surrogate than Kriging when used for feasibility analysis on problems with relatively smooth landscapes. Second, when there are multiple disjoint feasible regions, the RBF-based method is less likely to be “trapped” in exploiting a certain feasible region than the Kriging-based method, especially when there are few initial sample points, and the initial surrogate models are not good enough to make any predictions on feasible regions. For example, in test problem “branincon”, there are three feasible regions of similar sizes. In the case starting with 4 ( $=2^2$ ) initial sample points, the initial surrogate models from the RBF-based method and the Kriging-based method are similarly bad in accuracy (Figure 25). After adaptive sampling, the final surrogate models for the RBF-based method and the Kriging-based method are shown in Figure 26. After 90 iterations, the RBF-based method can accurately identify all three feasible regions, however the Kriging-based method is trapped in exploiting in only one of the feasible region. This is also the reason why “sasenacon” and “camelback” are so difficult for the Kriging-based method to solve. Each of these two problems has multiple feasible regions of different sizes. During the adaptive sampling process, the Kriging-based method is more likely to be trapped in exploiting the larger feasible regions than the RBF-based method. This indicates the importance of introducing scale factor to the RBF-based method for the purpose of better balancing global search and local search during adaptive sampling.

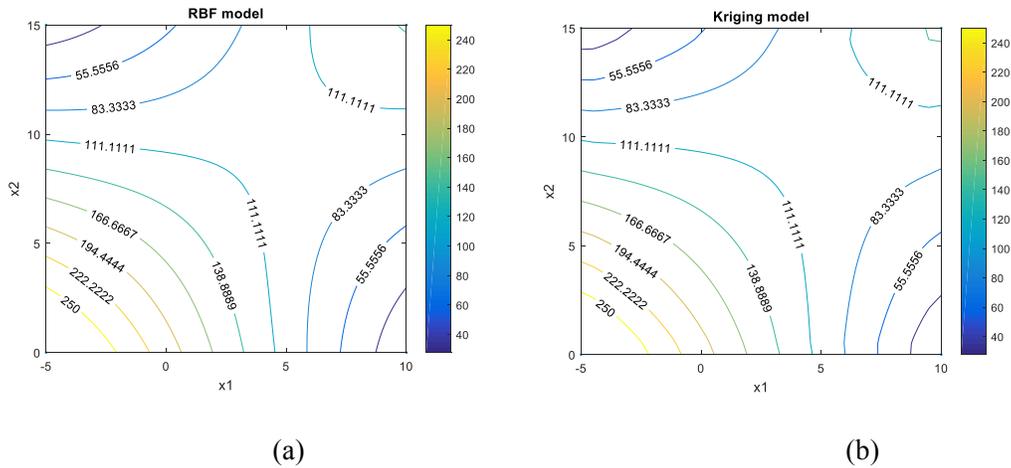


Figure 25. Initial surrogate models for “branincon” with 4 ( $=2^2$ ) initial sample points. (a) Initial RBF model. (b) Initial Kriging model

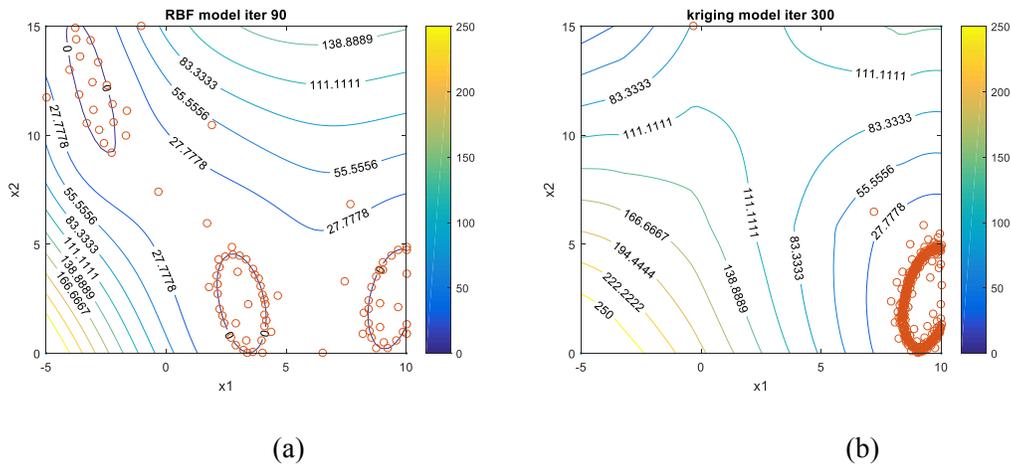


Figure 26. Final surrogate models for “branincon” after adaptive sampling. (a) Final RBF model after 90 iterations. (b) Final Kriging model after 300 iterations. Circle points represent adaptive sampling points.

The data on the minimum number of total sampling points for each test problem using two surrogate-based methods are shown in Table 5. The results show that the RBF-based method is generally the most efficient when adaptive sampling starts with small numbers of initial sample

points, while Kriging-based method needs to start with a relatively larger number of initial sample points to guarantee efficiency. Because the purpose of using adaptive sampling is to improve sampling efficiency when limited information is provided about the system, the advantage of RBF-based adaptive sampling method shows itself practically more useful than the Kriging-based adaptive sampling method. However, for both methods, the number of initial sample points for highest efficiency (namely the total number of sampling points being smallest) is actually a case-by-case problem. For test problems with one feasible region, both methods need relatively fewer initial sample points (e.g. `branincon`, `qcp4con`). For test problems with multiple disjoint feasible regions, especially those of different sizes, slightly more initial sample points are generally needed. In summary, it still requires further study in order to determine how many initial sample to use for both of these surrogate-based adaptive sampling methods for black-box feasibility analysis.

Table 5. Minimum number of total samples

<b>test problems</b>	<b>surrogate</b>	<b>total samples</b>	<b>initial samples</b>	<b>adaptive samples</b>
<b>branincon</b>	RBF	49	9	40
	Kriging	115	25	90
<b>ex3</b>	RBF	44	4	40
	Kriging	59	9	50
<b>sasenacon</b>	RBF	96	16	80
	Kriging	311	121	190
<b>camelback</b>	RBF	114	64	50
	Kriging	180	36	150
<b>qcp4con</b>	RBF	98	8	90
	Kriging	254	64	190

### High dimensional test problems

#### 5D test problem (g4con)

“g4con” is taken from [232] and only the constraints are considered for feasibility purpose. This test problem contains 6 nonlinear constraints which are shown in Equation (87).

$$\begin{aligned}
 0 &\leq 85.334407 + 0.0056858x_2x_5 + 0.0006262x_1x_4 - 0.0022053x_3x_5 \leq 92 \\
 90 &\leq 80.51249 + 0.0071317x_2x_5 + 0.0029955x_1x_2 + 0.0021813x_3^2 \leq 110 \\
 20 &\leq 9.300961 + 0.0047026x_3x_5 + 0.0012547x_1x_3 + 0.0019085x_3x_4 \leq 25 \\
 78 &\leq x_1 \leq 102, 33 \leq x_2 \leq 45, 27 \leq x_i \leq 45 \text{ for } i = 3, 4, 5.
 \end{aligned} \tag{87}$$

#### 6D test problem (t3con)

“t3con” is taken from [233] and only the constraints are considered for feasibility purpose. This test problem contains 2 nonlinear constraints and 4 linear constraints which are shown in Equation (88).

$$\begin{aligned}
 (x_3 - 3)^2 + x_4 &\geq 4 \\
 (x_5 - 3)^2 + x_6 &\geq 4 \\
 x_1 - 3x_2 &\leq 2 \\
 -x_1 + x_2 &\leq 2 \\
 x_1 + x_2 &\leq 6 \\
 x_1 + x_2 &\geq 2 \\
 0 \leq x_1 &\leq 5, \\
 0 \leq x_2 &\leq 5, \\
 1 \leq x_3 &\leq 5, \\
 0 \leq x_4 &\leq 6, \\
 1 \leq x_5 &\leq 5, \\
 0 \leq x_6 &\leq 10
 \end{aligned} \tag{88}$$

***Results on the performance of two surrogate-based methods***

Considering that the number of sample points from rectangular grid sampling plan grows exponentially with dimensions, we only use 2-level grid points to build the initial surrogate models. For “g4con”, we use 32 ( $= 2^5$ ) initial points; and for “t3con”, we use 64 ( $= 2^6$ ) initial points. After the initial surrogate models are built, 200 iterations of adaptive sampling are used to improve the surrogate accuracy. The values of surrogate accuracy metrics are shown in Table 6. From the initial surrogate accuracy, we can observe that CF% are very low, which indicates that very little information on the feasible regions can be obtained from initial surrogate models based on the 2-level grid sampling. Such initial models do not seem to affect the performance of two surrogate-based methods for the “g4con” test problem, because both methods can still well predict the feasible regions with relatively high accuracy (CF% around 95%, and CIF% around 98%) and good conservativeness (NC% around 5%) after 200 iterations of adaptive sampling, with the RBF-based method being slightly better than the Kriging-based method. However, for “t3con”, both surrogate-based approaches fail to give satisfactory predictions of feasible regions with the final surrogate models. Only around 60% of feasible regions have been correctly predicted, and with over 10% of the predicted feasible regions being non-conservative.

Table 6. Results on 5D and 6D test problems with grid sampling plan for initial points

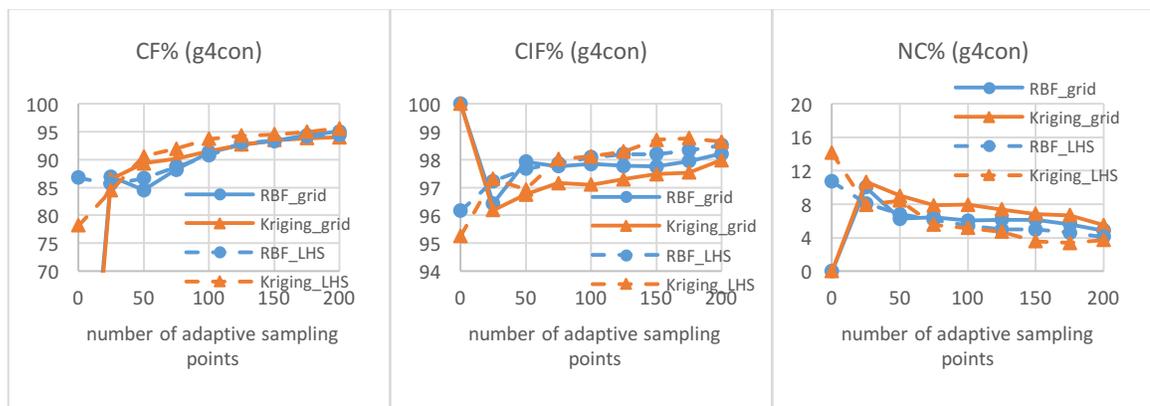
test problem	Surrogate	initial surrogate accuracy			surrogate accuracy after 200 iterations		
		CF%	CIF%	NC%	CF%	CIF%	NC%
g4con	RBF	20.47	100.00	0.00	95.03	98.21	4.87
	Kriging	16.43	100.00	0.00	94.01	97.98	5.52
t3con	RBF	0.00	100.00	NA	62.06	98.41	11.79
	Kriging	0.00	100.00	NA	59.21	98.37	12.58

Given that the use of grid sampling plan can be rather limited in high dimensional cases, and also to test whether a different initial sampling plan with better space-filling properties can improve the results of “t3con”, we then use LHS as the initial sampling plan. For comparison purposes, we again use 32 LHS initial points for “g4con” and 64 LHS initial points for “t3con”. The results after 200 iterations of adaptive sampling are shown in Table 7. From the initial surrogate accuracy, we notice that significantly higher percentage of the feasible region is discovered, though not very conservative, compared with that from grid sampling plan. Such changes do not affect much the final results for “g4con”, as we can observe that both surrogate-based approaches result in similarly good accuracy metrics compared with that in Table 6. However, for “t3con”, both surrogate-based approaches can benefit from LHS initial sampling plan in that more feasible regions are correctly discovered after 200 iterations, though the conservativeness is slightly worse, than that in Table 6, and the Kriging-based method even outperforms the RBF-based method with a higher CF%. However, it must be noted that the performance of both surrogate-based methods is still not satisfactory because they still fail to sufficiently predict all of the feasible regions in “t3con” correctly and conservatively.

Table 7. Results on 5D and 6D test problems with LHS plan for initial points

test problem	Surrogate	initial surrogate accuracy			surrogate accuracy after 200 iterations		
		CF%	CIF%	NC%	CF%	CIF%	NC%
g4con	RBF	86.76	96.17	10.70	94.44	98.52	4.09
	Kriging	78.19	95.26	14.12	95.59	98.64	3.72
t3con	RBF	62.89	93.41	35.44	65.01	98.07	13.45
	Kriging	54.75	95.77	28.78	72.73	97.83	13.50

To better understand the limitations of the approaches when dealing with high dimensional problems, we also checked the surrogate accuracy every 25 iterations, with results shown in Figure 27 and Figure 28. From Figure 27, we find that for “g4con”, starting from 25 iterations when a large portion of the feasible region has been correctly discovered ( $CF\% > 85\%$ ), the surrogate accuracy improves very slowly for both surrogate-based approaches. This is expected because with dimensions increasing, more sample points are needed to capture the landscape of the original model, which is known as “curse-of-dimensionality” [76]. From Figure 28, for “t3con”, in the cases where grid sample points are used, after 100 iterations, both methods tend to be stagnant with 50% to 60% feasible region correctly identified. From the analysis of 2D test problems, we can infer that a highly possible cause for this is there exist disjoint feasible regions in the high dimensional input space. Compared to low dimensional problems, high dimensional problems may require more adaptive sampling points to exploit the discovered feasible regions due to “curse-of-dimensionality”. The difficulty of being trapped in exploiting a discovered feasible region may be slightly mitigated by using LHS, as can be seen from Figure 28 where  $CF\%$  for both methods using LHS points are higher than those using grid points. This is because with better space-filling initial points, there is a higher chance that more disjoint feasible regions can be discovered by the initial surrogate models, though also with a risk of non-conservativeness in the prediction.

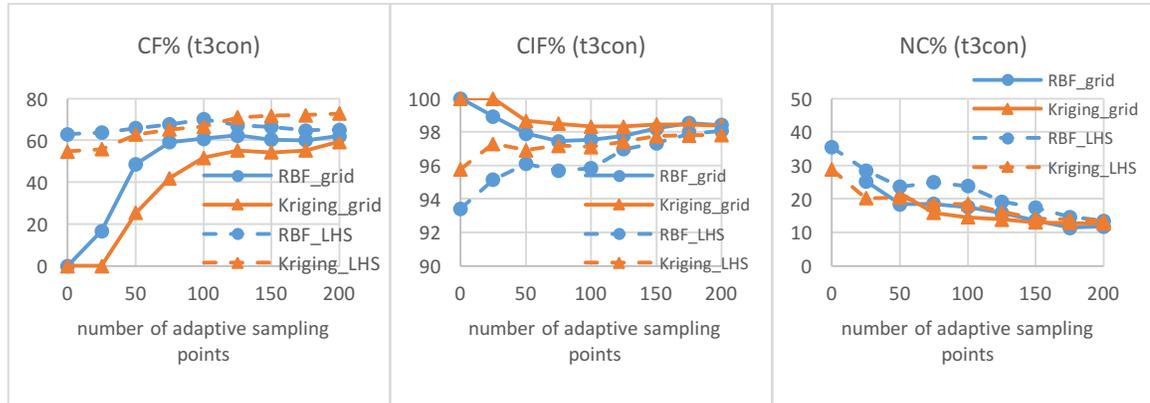


(a)

(b)

(c)

Figure 27. Comparison of surrogate accuracy by iteration of adaptive sampling for g4con. (a) CF% (b)CIF (c) NC%



(a)

(b)

(c)

Figure 28. Comparison of surrogate accuracy by iteration of adaptive sampling for t3con. (a) CF% (b)CIF (c) NC%

#### 4.5 Case study of a roller compaction process

We applied the two surrogate-based feasibility analysis approaches to a real-world case study of a roller compactor simulation model from [234] in pharmaceutical manufacturing processes. The model is shown in Equation (89). This model can be used to predict ribbon properties (ribbon density  $\rho_{exit}$ , ribbon thickness  $h_0$ ) given operating conditions (hydraulic pressure  $P_h$ , rotating roll speed  $\omega$ , feed speed  $u_{in}$ ) and inlet material properties (inlet angle  $\theta_{in}$ , powder bulk density  $\rho_{in}$ ). The product quality constraints are shown in Equation (90). The parameter values, operating conditions, and bounds used in this work are listed in Table 8, which are based on the process settings in [234] and consistent with the application in [65].

Table 8. Parameter values and boundaries in the roller compactor case study

Parameter	Symbol	Value/Range	Units
<b>Equipment parameters and model coefficients</b>			
Roll radius	$R$	0.125	m
Roll width	$W$	0.05	m
Compression parameter	$K$	4.97	
Compression parameter	$C_I$	$7.5 \times 10^{-8}$	$\text{Pa}/(\text{kg}/\text{m}^3)^{4.97}$
Compact surface area	$A$	0.01	$\text{m}^2$
Effective angle of friction	$\delta$	0.7069	rad
Nip angle	$\alpha$	0.173	rad
Angular position	$\theta$	$NA$	rad
<b>Operating conditions</b>			
Hydraulic pressure set point (roll pressure)	$P_h$	0.9	MPa
Rotating roll speed set point	$\omega$	5	rpm
Powder feed speed	$u_{in}$	$3.27 \times 10^{-4}$	m/s
<b>Bounds</b>			
Inlet powder density	$\rho_{in}$	200 - 400	$\text{kg}/\text{m}^3$
Inlet angle	$\theta_{in}$	0.3 - 0.5	rad
Ribbon thickness	$h_0$	$1.7 \times 10^{-3}$ - $1.9 \times 10^{-3}$	m
Ribbon density	$\rho_{exit}$	850 - 950	$\text{kg}/\text{m}^3$

$$\frac{d}{dt} \left( \frac{h_0}{R} \right) = \frac{\omega \left[ \rho_{in} \cos \theta_{in} (1 + h_0/R - \cos \theta_{in}) (u_{in}/\omega R) - \rho_{exit} (h_0/R) \right]}{\int_0^{\theta_{in}} \rho(\theta) \cos(\theta) d\theta}$$

$$P_h = \frac{W}{A} \frac{\sigma_{exit} R}{1 + \sin \delta} \int_0^\alpha \left[ \frac{h_0/R}{(1 + h_0/R - \cos \theta) \cos \theta} \right]^K \cos \theta d\theta \quad (89)$$

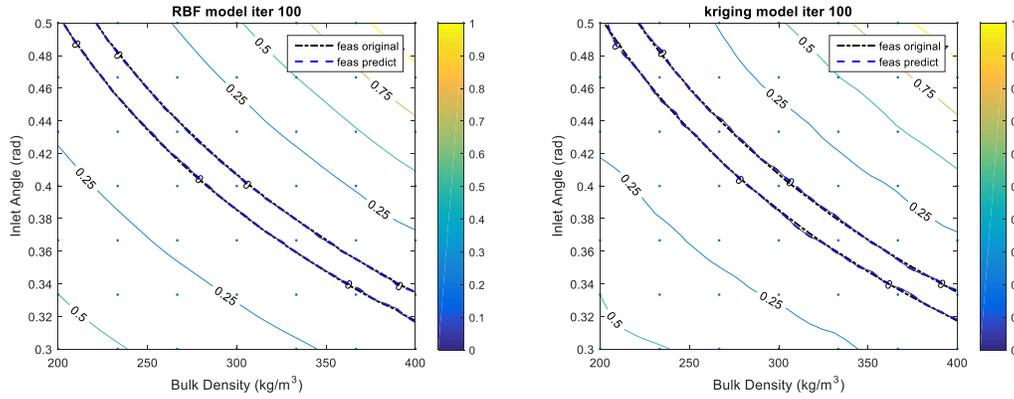
$$\sigma_{exit} = C_1 \rho_{exit}^K$$

$$\rho_{exit}^L \leq \rho_{exit} \leq \rho_{exit}^U \quad (90)$$

$$h_0^L \leq h_0 \leq h_0^U$$

The objective of the steady state feasibility analysis is to identify the 2D feasible region of  $\theta_{in}$  and  $\rho_{in}$  so that we can guarantee the products meet the quality constraints ( $\rho_{exit}$  and  $h_0$ ) in Equation (90) under a certain set of operating conditions ( $P_h$ ,  $\omega$ , and  $u_{in}$ ). The simulation model can be computationally expensive due to integral terms, and it requires reformulation to explicitly express the output ribbon properties ( $\rho_{exit}$  and  $h_0$ ). Thus we can apply the surrogate-based methods to solve the feasibility analysis problem. Note here the product quality variables  $h_0$  and  $\rho_{exit}$  have very different magnitudes. So the constraint violations of  $h_0$  and  $\rho_{exit}$  are scaled with their corresponding upper bound values, namely  $1.9 \times 10^{-3}$  m and  $950$  kg/m<sup>3</sup>. This is to make sure the quality constraints can contribute to  $EI_{feas}$  on similar magnitude during the adaptive sampling stage.

When applying the two surrogate-based approaches to this 2D feasibility analysis problem, we use 49 grid sample points to build the initial surrogate models, and use 100 iterations of adaptive sampling to improve the surrogate models. The contours of predicted feasibility function values are shown in Figure 29 and the surrogate accuracy values are listed in Table 9.



(a) RBF model for roller compaction (b) Kriging model for roller compaction

Figure 29. Contour of surrogate models for feasibility function values of roller compaction model.

(a) RBF model for roller compaction; (b) Kriging model for roller compaction. (Black dash-dot line represents the feasible region boundary of the original model; blue dashed line represents the feasible region boundary predicted by the surrogate model.)

Table 9. Comparison of accuracy between RBF-based method and Kriging-based method

test problem	Surrogate	initial surrogate accuracy			surrogate accuracy after 100 iterations		
		CF%	CIF%	NC%	CF%	CIF%	NC%
RC	RBF	77.69	98.57	13.68	99.52	99.89	0.96
	Kriging	29.62	99.91	2.53	98.56	99.19	6.65

From the results, we can see that both the surrogate-based methods can sufficiently predict the feasible region after 100 iterations of adaptive sampling, with the RBF-based method being more accurate and conservative than Kriging, which is consistent with the finding from 2D test problems. The feasible region (feasibility function values  $\leq 0$ ) reflects the ranges of variations in the inlet parameters ( $\theta_{in}$  and  $\rho_{in}$ ) that the process can tolerate in order to obtain qualified ribbon products ( $\rho_{exit}$  and  $h_o$ ) under the operating conditions. In addition, the feasible region also indicates that

when  $\theta_{in}$  is small,  $\rho_{in}$  must be relatively large to guarantee qualified ribbon qualities and vice versa. This finding can help choosing the right inlet materials in order to get desired ribbon properties under a certain set of operating conditions.

#### 4.6 Summary and future work

In this work, we propose a new RBF-based adaptive sampling method for feasibility analysis with black box constraints. We use an error-indicator for cubic RBF to indicate how well the nearby region of an unsampled point has been explored. The error-indicator is used with a scale factor in the adaptive sampling which can balance global search and local search to find the feasible region boundaries. For all case studies on 2D and 3D benchmark problems, the new RBF-based method shows better performance than a Kriging-based method in several aspects. First, the RBF-based method can generally be more accurate than the Kriging-based method with the same limited numbers of adaptive sampling points. Second, the RBF-based method is more efficient than the Kriging-based method in that it can provide accurate predictions with fewer adaptive sampling points. Third, the RBF-based method can be practically more useful since it is more efficient when starting with fewer initial sample points than the Kriging-based method. For 5D and 6D benchmark problems, both the RBF-based and the Kriging-based show their limitations due to “curse of dimensionality” .

In our future work, more research is needed to improve the performance of surrogate-based feasibility analysis approaches in high dimensional test problems. Moreover, we are also interested in applying different initial sampling plans, and see their effects on the performance of adaptive sampling.

## 5 Process Analysis and Optimization of Continuous Pharmaceutical Manufacturing using Flowsheet Models

### **Abstract**

Continuous manufacturing has attracted increasing research attention in the pharmaceutical industry within the last decade. Based on the extensive experimental studies, numerous modeling and computational approaches have been developed to capture the process information and make predictions. Moreover, flowsheet models have been built to simulate the dynamic behaviors of a plant-wide manufacturing process with respect to different process input factors. However, there still lacks a systematic way to make the best use of flowsheet models in pharmaceutical processes. In this work, we propose a framework of process analysis and optimization for the continuous pharmaceutical manufacturing process where flowsheet models are available. Specifically, sensitivity analysis is conducted to identify the input factors that are most influential on the output; feasibility analysis is then implemented to characterize the design space in the high-dimensional space. Finally, process optimization is performed to find the optimal operation conditions that result in the minimum cost.

### 5.1 Introduction

The pharmaceutical industry has put increasing efforts to improve process understanding over the past few years, with the main focus on improving product quality and reducing costs in product development and manufacturing[32]. Such a research trend to improve manufacturing processes is incited by various factors, including the urge to maximize profits within patent life[15], the increased difficulty in developing new competitive drugs[15], and the support from regulatory agencies[235,79], such as U.S. Food and Drug Administration (FDA). Among the recent advances in pharmaceutical manufacturing processes, a major change is a transition from a traditional batch mode to a continuous mode[32].

There are several benefits of applying continuous manufacturing (CM) to the pharmaceutical industry. First, CM can use the same equipment for large quantities of production, which eliminates the scale-up problem[236], and shortens the time to release products to the market[237]. Second, CM equipment usually has a much smaller footprint, which can potentially reduce the capital cost and increase the controllability[238]. Third, CM eliminates the need to store intermediates between steps[2], and reduces human factors in handling raw materials[15]. This can directly diminish the risks associated with material degradation and improve product quality. Therefore, with such benefits, CM is considered promising to improve agility, flexibility, and robustness in the manufacture of pharmaceuticals[2].

However, there are also some challenges in implementing continuous pharmaceutical manufacturing. In this paper, we only consider the continuous manufacture of solid oral-dosage products, since this is the main form of drug products[4]. Because the powder materials flow continuously between unit operations, the first challenge that needs to be considered is understanding the material attributes and how they can affect the flowability and process dynamics. This would require a systematic method for material characterization and prediction of the intermediate material properties[239]. Moreover, the residence time of the materials needs to be monitored and controlled in order to ensure product quality. This needs advanced online measurement tools and control strategies that are specifically designed for the continuous equipment and processes[32,8]. Finally, the process design and optimization problems need to be solved with a systems approach, which considers the effect of material properties, equipment, and process parameters on the product qualities[32]. To address such difficulties, we can develop mathematical models, based on experimental studies, and use computer-aided tools to enhance the understanding of a continuous process[32].

Numerous modeling techniques have been applied to simulate the particle-level or bulk behaviors of the material flow in the equipment of continuous processes. Discrete-element method (DEM) has been employed to study the dynamics of the material flow in a blender[240,29]. However, its high computational cost limits its use in integrated-process modeling applications. In comparison, population balance modeling (PBM), with a comparatively lower computational cost, has been extensively used to model blending[12,241] and granulation processes[242]. Furthermore, data-driven models[13] and semi-empirical models based on residence-time distribution (RTD) theories[6,243] have also been developed for different powder processing unit operations. Based on the development of modeling individual continuous equipment, recent research has been focused on the integrated flowsheet modeling approach[15,14], which is an approximation of the actual plant-wide operation process. A flowsheet model can be used to predict the process dynamics affected by material properties and operating conditions of different component unit operations. More importantly, we can make a systematic process analysis of the flowsheet model, which gives insight on the characteristics and bottlenecks of the process and thus facilitates the development of control strategies[14]. In this paper, we focus on two aspects of the process analysis: sensitivity analysis and feasibility analysis.

Sensitivity analysis is the study of how uncertainty in the output of a model can be apportioned to different sources of uncertainty in the model input[43]. Given a complex model consisting of numbers of input factors, it is usually the case that only a few factors have a large influence on the output variables of interest, while the rest majority only have a negligible contribution[43]. Thus, sensitivity analysis can be used to prioritize the input factors by influence and reduce the dimension of the model by screening out the less important input factors. In the pharmaceutical manufacturing processes, sensitivity analysis has been used to quantify the individual effects and the interaction effects of the input factors, including critical design parameters, material properties, and operating conditions, for unit operations[244,245] and integrated processes[15,14]. The results of sensitivity

analysis can also be used to guide where the models need to be further detailed, and to aid the development of control strategies for qualified products.

Feasibility analysis is used to identify the feasible region (i.e. design space) within which all the process and quality constraints are met, including equipment capacity, production rate, and product quality[246]. Such analysis gives a systematic view, together with sensitivity analysis, on the relationship between materials, processes, and products properties, which can benefit the manufacturers with an understanding of the maximum limit of a process to remain robustness while conserving product quality. In the pharmaceutical industry, with the development of various models, mathematical approaches can be used to define and solve the feasibility analysis problems. For computational inexpensive models, the original models can be directly used to determine the feasible region. Examples include the design space identification for drug substance[46] and drug product[47]. When the models are computationally expensive, surrogate-based approaches have been developed to approximate the original models and predict the feasible region. Examples include the feasibility analysis for the mixer[18], the feeder[18], and the roller compactor[65,246]. However, little research has focused on the feasibility analysis problem of an integrated process. This is because the integrated process usually has a large number of input factors, which may include material properties and process parameters. The high dimensionality will make the feasibility analysis problem very complicated to solve. The difficulties may include high computational cost due to increased numbers of input factors and constraints, and the visualization problem of the design space in high dimensions.

In this paper, we propose a systematic approach for process analysis, consisting of sensitivity analysis and feasibility analysis, and process optimization for the continuous pharmaceutical processes. More specifically, we aim to apply this approach to the integrated process which is computationally expensive and include large numbers of model inputs and constraints. The

approach first uses sensitivity analysis to rank the input factors by influence and reduce the dimension of the model. Then a surrogate-based feasibility analysis approach is used to depict the design space with efficiency and accuracy, after which the high-dimensional design space is visualized with a matrix of contour plots of feasibility. Finally, process optimization is conducted to find the optimal operation conditions to minimize the total operation costs. The rest of this article is organized as follows. The sensitivity analysis and feasibility analysis methods are explained in Section 5.2 and 5.3 respectively. The process analysis results are discussed in Section 5.4, and optimization results discussed in Section 5.5. Then, the conclusion and future work is discussed in Section 5.6.

## 5.2 Sensitivity Analysis

Given a complex flowsheet model, sensitivity analysis can be used to identify the subset of most influential input factors (e.g. material properties, process parameters), which need special attention when developing control strategies. On the other hand, it also makes it easier for the subsequent feasibility analysis and process optimization problems to concentrate on the most important parameters.

Many sensitivity analysis methods have been developed for different uses by statisticians. Generally, these methods can be classified into two categories: (1) local methods, which focus on the effects of uncertain inputs around one single point (base case); and (2) global methods, which determine the influences of uncertain input factors over the whole input space[43]. Since we are usually interested in the overall sensitivity behaviors of the input factors within the whole uncertain region, global sensitivity analysis methods should be adopted for our flowsheet models.

Under the category of global sensitivity analysis, there are numerous methods available, of which the following three are implemented in our work: (1) screening Method (e.g. Morris method); (2)

regression-based method (e.g. PRCC method); and (3) variance-based method (e.g. Sobol's method). Various methods are adopted because they can altogether provide sensitivity analysis from a qualitative as well as a quantitative perspective. In practice, they have been shown effective in analyzing the pharmaceutical manufacturing processes[15,14,49].

### 5.2.1 Morris Method

Morris method is an efficient way of screening a few important input factors with small sampling cost, and is usually used when the model has a large number of input factors, and/or the model is computationally expensive[52]. To use the Morris method, we need to first define the elementary effect (EE) with Equation 16.

$$EE_i = \frac{Y(x_1, \dots, x_{i-1}, x_i + \Delta_i, x_{i+1}, \dots, x_k) - Y(x_1, \dots, x_k)}{\Delta_i} \quad (91)$$

where  $k$  is the number of input factors;  $(x_1, \dots, x_k)$  is a selected base point;  $\Delta_i$  is the step change in the  $i$ th input factor;  $Y(*)$  is the output variable.

In order to get the global sensitivity information, multiple base points need to be randomly chosen so that they can cover the whole input space. The Morris sensitivity metrics are thus calculated as a function of  $EE$ . In specific, three sensitivity metrics have been proposed[43], which are shown in Equation 92:

$$\begin{aligned} \mu_i &= \frac{1}{r} \sum_{j=1}^r EE_i^j \\ \sigma_i^2 &= \frac{1}{r-1} \sum_{j=1}^r (EE_i^j - \mu)^2 \\ \mu_i^* &= \frac{1}{r} \sum_{j=1}^r |EE_i^j| \end{aligned} \quad (92)$$

where  $r$  equals to the number of base points. The mean  $\mu_i$  reflects the individual sensitivity information; the standard deviation  $\sigma_i$  reflects the nonlinearity or interaction with other factors. Usually  $\mu_i$  is used simultaneously with  $\sigma_i$  for a reliable sensitivity analysis. This is because if a significant input factor has elementary effects of both positive and negative signs,  $\mu_i$  can be small and close to zero (e.g.  $EE_i^j$  of different signs cancel out each other), but  $\sigma_i$  would still be large. In such cases,  $\mu_i^*$  can be more convenient, which is calculated as the average of the absolute values of the elementary effects. Generally, it is recommended[43] that all of these three metrics should be used together to get a thorough analysis of the sensitivity behaviors of the whole system. To calculate the sensitivity metrics, the sample points are chosen by following a “radial design” [52], which proves to provide most accurate results for Morris method. Sobol’s quasi-random numbers[247,53] are used to generate the sample points, and the total sampling cost is  $r(k + 1)$ . According to Iooss and Lemaître[51],  $r$  is suggested be selected between 2 and 10. In this work,  $r$  is chosen as 10.

To interpret the Morris results, for the input factors with large values of  $\mu$  and/or  $\sigma$ , or the input factors with large values of  $\mu^*$ , they are considered as most influential. In application, if an input factor has a sensitivity metric value smaller than 10% of the largest value of all the metric’s values, that input factor is considered to be non-influential[43,54].

### 5.2.2 PRCC Method

Partial Rank Correlation Coefficient (PRCC) is an extension of Partial Correlation Coefficient (PCC) for sensitivity analysis. In statistics, PCC is a measure of the strength of the linear relationship between two variables when all linear effects of other variables are removed[55]. To determine whether an input (e.g.  $X_i$ ) is strongly linearly correlated with the output (e.g.  $Y$ ), first

build two multivariate linear regression models for both the input  $X_i$  and  $Y$ , and then PCC is calculated as the correlation of the residual of  $X_i$  and  $Y$ , which is shown in Equation 93.

$$\begin{aligned}\widehat{X}_i &= c_0 + \sum_{j=1, j \neq i}^k c_j X_j \\ \widehat{Y} &= b_0 + \sum_{j=1, j \neq i}^k b_j X_j \\ PCC &= CC(X_i - \widehat{X}_i, Y - \widehat{Y})\end{aligned}\tag{93}$$

Because PCC is based on the assumption that there exists a linear relationship between the input and output, it is not suitable for nonlinear models. However, if the relationship between input and output is nonlinear but monotonic, we can use rank transformed data of the input and output (e.g. replacing the raw data with their ranks) to transform the nonlinear relationship into a linear relationship[248]. With the rank transformed data, we can build the multivariate linear regression models (described in Equation 93) and calculate partial rank correlation coefficient (PRCC) to measure the sensitivity information. PRCC is suitable for both linear models and nonlinear but monotonic models.

PRCC is a standardized sensitivity measurement with values between -1 and 1. For those with positive values, the output will increase as the input increases. While for those with negative values, the output will decrease as the input increases. Latin Hypercube Sampling (LHS)[249] is used to sample the data points. To get accurate sensitivity analysis results, the sampling cost should be larger than  $50k$ , where  $k$  is the number of input factors[51]. Those input factors with PRCC values close to 1 or -1 are identified as influential.

### 5.2.3 Sobol' Method

Sobol' method is a variance-based global sensitivity analysis method, which describes how the variance of output can be decomposed into terms depending on input factors and their interactions[51]. Sobol' method generally uses two metrics to measure the sensitivity information, which are  $S_i$  (first-order effect) and  $S_{Ti}$  (total effect), as are defined by Equation 94:

$$S_i = \frac{V_{X_i}(E_{\mathbf{X}_{\sim i}}(Y|X_i))}{V(Y)} \quad (94)$$

$$S_{Ti} = 1 - \frac{V_{\mathbf{X}_{\sim i}}(E_{X_i}(Y|\mathbf{X}_{\sim i}))}{V(Y)} = \frac{E_{\mathbf{X}_{\sim i}}(V_{X_i}(Y|\mathbf{X}_{\sim i}))}{V(Y)}$$

where  $V(*)$  represents the variance;  $E(*)$  represents the expected value;  $X_i$  is the  $i$ th input factor;  $Y$  is the output variable;  $\mathbf{X}_{\sim i}$  represents all possible combination of input values with  $X_i$  being fixed.  $S_i$  indicates the main effect of  $X_i$  in the variance of the output.  $S_{Ti}$  is called total effect, which considers all the effect terms (first-order and higher-order interactions) that include the input factor  $X_i$ . It is derived as the difference between 1 (i.e. sum of all possible sensitivity terms) and  $\frac{V_{\mathbf{X}_{\sim i}}(E_{X_i}(Y|\mathbf{X}_{\sim i}))}{V(Y)}$  (i.e. all terms of any order that do not include  $X_i$ ).

In this work, a more efficient strategy is used to calculate these two metrics[58], which is shown in Equation (95):

$$S_i = \frac{\frac{1}{N} \sum_{j=1}^N f(\mathbf{B})_j \left( f(\mathbf{A}_B^{(i)})_j - f(\mathbf{A})_j \right)}{V(Y)} \quad (95)$$

$$S_{Ti} = \frac{\frac{1}{2N} \sum_{j=1}^N \left( f(\mathbf{A})_j - f(\mathbf{A}_B^{(i)})_j \right)^2}{V(Y)}$$

where  $\mathbf{A}$  and  $\mathbf{B}$  are two matrices (Equation 27), each consisting of  $N$  rows of different sample points,  $\mathbf{A}_B^{(i)}$  is a matrix in which the  $i$ th column of  $\mathbf{A}$  is substituted by the  $i$ th column of  $\mathbf{B}$ .

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1k} \\ a_{21} & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ a_{N1} & a_{N2} & \cdots & a_{Nk} \end{bmatrix}, \mathbf{B} = \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1k} \\ b_{21} & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ b_{N1} & b_{N2} & \cdots & b_{Nk} \end{bmatrix}, \mathbf{A}_B^{(i)} = \begin{bmatrix} a_{11} & b_{1i} & \cdots & a_{1k} \\ a_{21} & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ a_{N1} & b_{Ni} & \cdots & a_{Nk} \end{bmatrix} \quad (96)$$

Sobol's quasi-random number[247,53] is used to generate the sampling points, and the total sampling cost should be larger than  $500(k + 2)$  for accurate estimate of  $S_i$  and  $S_{T_i}$ . [43]

To interpret Sobol's results, as both metrics are within the range of  $[0, 1]$ , the larger the value is, the more influential the input factor is. The condition  $S_{T_i} = 0$  is necessary and sufficient for  $X_i$  to be a noninfluential factor[43]. As  $S_i \leq S_{T_i}$ , the difference between  $S_i$  and  $S_{T_i}$  reflects the influence of the interaction effects on the variance of the output.

### 5.3 Feasibility Analysis

After applying sensitivity analysis to identify the subset of most influential input factors, feasibility analysis can be conducted on this subset of input to characterize the design space of the process. Mathematically, the feasibility of a process can be defined in the following manner. Given a process with  $J$  constraints, we define that the  $j$ th constraint  $f_j(d, \theta)$  is met if  $f_j(d, \theta) \leq 0$ , where  $d$  represents the design variables (e.g. equipment geometry) which are usually constant parameters after a process is determined;  $\theta$  represents all the uncertain variables (e.g. blade speed, flow rate, etc.) A feasible process needs to meet all the constraints, meaning that  $f_j(d, \theta) \leq 0$  for all  $j \in J$ . To check whether there is a violation of any constraints, we simply need to check the maximum value of all the constraint function values, which can be defined with a "feasibility function" [44] in Equation 97.

$$\psi(d, \theta) = \max_{j \in J} \{f_j(d, \theta)\} \quad (97)$$

The objective of feasibility analysis is to identify the feasible region where  $\psi(d, \theta) \leq 0$ . Feasibility analysis problems have been well investigated by the process systems engineering community, and a good review of the recent advances on this topic can be found in Grossmann, Calfa et al.[44].

### 5.3.1 Surrogate-based method

The feasibility analysis problem can be rather difficult when the simulation is computationally expensive, which is usually the case for an integrated flowsheet model. To address such difficulties, the surrogate-based adaptive sampling approach can be used to efficiently explore the uncertain parameter space and identify the feasible region with high accuracy. This approach builds a surrogate as a computationally cheap approximation to the original flowsheet model, and use this surrogate to predict the feasible region. To make the best use of the sampling budget, the adaptive sampling strategy is used to iteratively sample new points in promising regions and update the surrogate. In this work, we applied a recently published RBF-based adaptive sampling method to solve the feasibility analysis problem for the integrated flowsheet model, which was shown to outperform the other surrogate-based approaches[246]. The RBF-based adaptive sampling method is briefly described below. The readers are referred to the original work by Wang and Ierapetritou[246] for the derivation and further details of this approach.

#### 5.3.1.1 Radial basis function (RBF) surrogate model

Radial Basis Function (RBF)[222] uses a weighted sum of radial functions to predict the function value at an unsampled point (Equation 98).

$$s_n(x) = \sum_{i=1}^n \lambda_i \phi(\|x - x_i\|_2) + b^T x + a \quad (98)$$

where  $x_1, x_2, \dots, x_n \in R^d$  are  $n$  distinct sample points with known function values  $f(x_1), f(x_2), \dots, f(x_n)$ ;  $\| * \|_2$  represents the Euclidean distance;  $\phi$  is the basis function. In this work, we choose to use cubic basis function with the form in Equation 99.

$$\phi(r) = r^3 \quad (99)$$

The model coefficients  $\lambda_i$ ,  $b$ , and  $a$  can be obtained by Equation 100.

$$\begin{pmatrix} \Phi & P \\ P^T & 0 \end{pmatrix} \begin{pmatrix} \lambda \\ c \end{pmatrix} = \begin{pmatrix} F \\ 0 \end{pmatrix} \quad (100)$$

where  $\Phi$  is the  $n$  by  $n$  matrix with  $\Phi_{ij} = \phi(\|x_i - x_j\|_2)$ , and the rest terms are expressed as follows.

$$P = \begin{pmatrix} x_1^T & 1 \\ x_2^T & 1 \\ \vdots & \vdots \\ x_n^T & 1 \end{pmatrix}_{n \times (d+1)}, \lambda = \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_n \end{pmatrix}_{n \times 1}, c = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_d \\ a \end{pmatrix}_{(d+1) \times 1}, F = \begin{pmatrix} f(x_1) \\ f(x_2) \\ \vdots \\ f(x_n) \end{pmatrix}_{n \times 1}$$

For an unsampled point  $y$ , an error indicator “ $1/\mu_n(y)$ ” [246] is used to reflect the uncertainty of prediction, where  $\mu_n$  is the coefficient of the new term  $\phi(\|x - y\|_2)$  in the surrogate  $s_n(x)$  if a new (unsampled) point  $y$  is added.  $\mu_n$  is calculated as the  $n$ th element of vector  $v$ , and  $v$  can be calculated using Equation 101.

$$\begin{pmatrix} \Phi_y & P_y \\ P_y^T & 0 \end{pmatrix} v = \begin{pmatrix} 0_n \\ 1 \\ 0_{d+1} \end{pmatrix}, \Phi_y = \begin{pmatrix} \Phi & \phi_y \\ \phi_y^T & 0 \end{pmatrix}, P_y = \begin{pmatrix} y^T P \\ 1 \end{pmatrix}, \quad (101)$$

$$(\phi_y)_i = \phi(\|y - x_i\|_2), i = 1, 2, \dots, n.$$

### 5.3.1.2 Adaptive sampling

Adaptive sampling can be used to improve the accuracy of the surrogate model without the need to exhaustively sample the whole input space of the original simulation model[65]. New sample points are chosen by maximizing a modified expected improvement function (Equation 102).

$$\max_x EI_{feas}(x) = s \times \phi\left(\frac{-y}{s}\right) = s \times \frac{1}{\sqrt{2\pi}} e^{-0.5\left(\frac{y^2}{s^2}\right)} \quad (102)$$

where  $EI_{feas}(x)$  is the modified EI function value at  $x$ ;  $y$  is the surrogate model predictor;  $s$  is the standard error of the predictor;  $\phi(*)$  represents the standard normal density function. For the RBF-based method,  $s$  is replaced with  $\sqrt{\frac{1/\mu}{\text{scale}}}$ , which is calculated with Equation 103.

$$\sqrt{\frac{1/\mu}{\text{scale}}} = \sqrt{\frac{1/\mu}{\max(1/\mu_0)} * \frac{\max(RBF_0)}{\text{numIniPts}}} \quad (103)$$

$$\text{scale} = \frac{\max(1/\mu_0)}{\max(RBF_0)^2 / \text{numIniPts}^2}$$

where  $\max(1/\mu_0)$  is the maximum value of  $(1/\mu_0)$  with the initial RBF model;  $\max(RBF_0)$  is the maximum value of RBF predictor with the initial RBF model;  $\text{numIniPts}$  is the number of initial sample points.

If we take the derivative of  $El_{feas}$ , we would notice that its value is large when the surrogate predictor  $y$  is close to zero, and also when prediction uncertainty  $s$  is high. Therefore, by solving the optimization problem (102) iteratively, we can sample new points near the feasible region boundary of the surrogate (e.g.  $y$  being 0) as well as sample new points in the highly uncertain regions (e.g.  $s$  being large)[246]. In other words, the adaptive sampling will keep a balance between local search and global search for the feasible region boundary, which is the key to the efficiency and accuracy of the surrogate-based feasibility analysis strategy.

The algorithm of conducting the RBF-based adaptive sampling approach for feasibility analysis is shown in Figure 30, which mainly consists of three steps. Step #1: build an initial surrogate model with sample points from a space-filling design of experiment (DOE), e.g. a rectangular grid sampling plan[246,65], so that the sample points are selected to cover the whole input space as uniformly as possible[250]. And use the initial surrogate to calculate the scale factor. Step #2: improve the surrogate model by adding new points iteratively based on the “adaptive sampling” strategy. The adaptive sampling will terminate if the iteration exceeds an upper bound defined by users. Step #3: use the improved surrogate model to predict the feasible region of the original model.

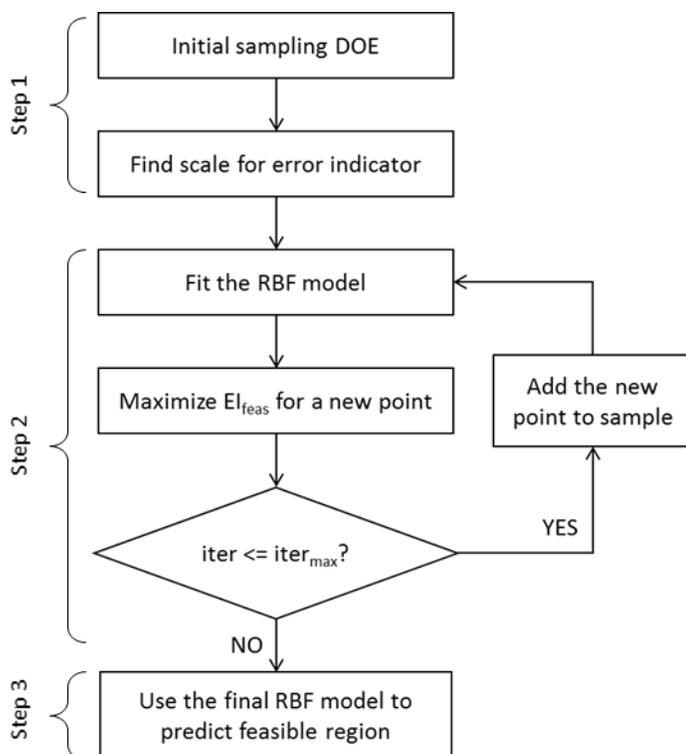


Figure 30. Algorithm of the RBF-based adaptive sampling approach for feasibility analysis

## 5.4 Analysis of the Direct Compaction line for production of pharmaceutical tablets

### 5.4.1 Sensitivity Analysis Results

Sensitivity analysis is conducted when the simulated process is at steady state. With the integrated flowsheet model (Figure. 5), we take into account 22 input factors, which include material properties (e.g. bulk density, true density, particle size distribution), operation conditions (e.g. flow rate set point, blade speed, tablet die fill depth), and tablet geometry (e.g. tablet thickness). Note that “tablet die fill depth” is the height of powder in the die before compression, which determines the total weight of each tablet. In comparison, “tablet thickness” is the actual thickness of each tablet after the compression, which is a geometry parameter. These input factors are chosen because they are the main factors that can determine the output variables of interest. The input factors are considered to vary uniformly within the ranges listed in Table 1. In this work, the ranges are selected as +/- 5% of the nominal values, which are set depending on the actual

production requirements. For example, the flow rate set points (30 kg/hr in total) in the table are chosen based on the production capacity of the plant in the lab, which leads to a production rate of  $3.7379 \times 10^6$  tablets per day. The co-mill blade speed is set to a relatively low value to prevent over shearing. Accordingly, the sensitivity analysis investigates the input factors' influence on 20 output variables, which include blends' material properties (e.g. mean bulk density, mean true density, mean particle size distribution), operation safety (e.g. mass hold up at steady state, tablet compression pressure), mixing characterization (e.g. mean residence time), and tablet product qualities (e.g. API concentration, weight, hardness). The output variables are listed in Table 11.

Table 10. 22 input factors for sensitivity analysis

<b>Input</b>	<b>Nominal</b>	<b>LB</b>	<b>UB</b>	<b>Unit</b>
API flow rate	3	2.85	3.15	kg/hr
API bulk density	250	237.5	262.5	kg/m <sup>3</sup>
API true density	2100	1995	2205	kg/m <sup>3</sup>
API d10	38	36.1	39.9	μ m
API d50	75	71.25	78.75	μ m
API d90	125	118.75	131.25	μ m
EXP flow rate	26.7	25.365	28.035	kg/hr
EXP bulk density	400	380	420	kg/m <sup>3</sup>
EXP true density	2500	2375	2625	kg/m <sup>3</sup>
EXP d10	30	28.5	31.5	μ m
EXP d50	120	114	126	μ m
EXP d90	250	237.5	262.5	μ m
LUB flow rate	0.3	0.285	0.315	kg/hr

LUB bulk density	200	190	210	kg/m <sup>3</sup>
LUB true density	1900	1805	1995	kg/m <sup>3</sup>
LUB d10	70	66.5	73.5	μ m
LUB d50	150	142.5	157.5	μ m
LUB d90	270	256.5	283.5	μ m
Co-mill blade speed	1120	1064	1176	RPM
Blender blade speed	250	237.5	262.5	RPM
Tablet fill depth	0.01	0.0095	0.0105	m
Tablet thickness	0.0025	0.002375	0.002625	m

Table 11. 20 output variables for sensitivity analysis

	<b>Output</b>	<b>Unit</b>
Blender	Mean residence time	s
	Delay time	s
	Mass holdup SS	kg
	Mean true density	kg/m <sup>3</sup>
	Mean bulk density	kg/m <sup>3</sup>
	MeanD10	μ m
	MeanD50	μ m
	MeanD90	μ m
	Co-mill	MeanD10
MeanD50		μ m
MeanD90		μ m

	Mean bulk density	kg/m <sup>3</sup>
	Mean true density	kg/m <sup>3</sup>
	Mean residence time	s
	Mass holdup SS	kg
	Concentration	%
Tablet Press	Weight	kg
	Hardness	kp
	Main compression pressure	MPa
	Pre-compression pressure	MPa

#### 5.4.1.1 Results of Morris' and PRCC methods

We first implement Morris' method using 230 ( $=10*(22+1)$ ) sample points, which are sampled by using Sobol's quasi-random numbers. The results are shown in the supplementary material [*URL will be inserted by publisher*], which shows that 14 input factors are identified as influential. 8 material properties include: API bulk density, API true density, API d50, Excipient bulk density, Excipient true density, Excipient d10, Excipient d50, and Excipient d90. 6 operation conditions include: API flow rate, Excipient flow rate, Co-mill blade speed, Blender blade speed, Tablet fill depth, and Tablet thickness.

From the results, we notice that the material properties and flow rate set points of API and Excipient have a major influence on the output variables. This finding is expected because these two components account for 99% of the entire mixed flow (approximately 89% Excipient and 10% API) in the simulation. Co-mill blade speed and blender blade speed are influential because they directly affect the residence time distribution and mass-hold up within each equipment. In addition,

tablet fill depth determines the materials that are fed to produce each tablet, which directly affects the tablet weight. Together with the influence of tablet thickness, these two parameters can greatly impact the tablet hardness and operation safety factors (e.g. compression pressure).

We then implement the PRCC method using 1100 (=55\*22) LHS sample points, with results visualized using an intensity plot (Figure. 31). In this intensity plot, the blue color indicates a negative effect of input on the output, and the red color indicates a positive effect. A darker color suggests there exists a stronger correlation between the input and the output. From the results, we can notice that PRCC identifies almost the same subset of most influential input factors as the results from Morris' method, with an addition of “lubricant d50” which may have a strong effect on the “mean d50” of the blends out of the blender.

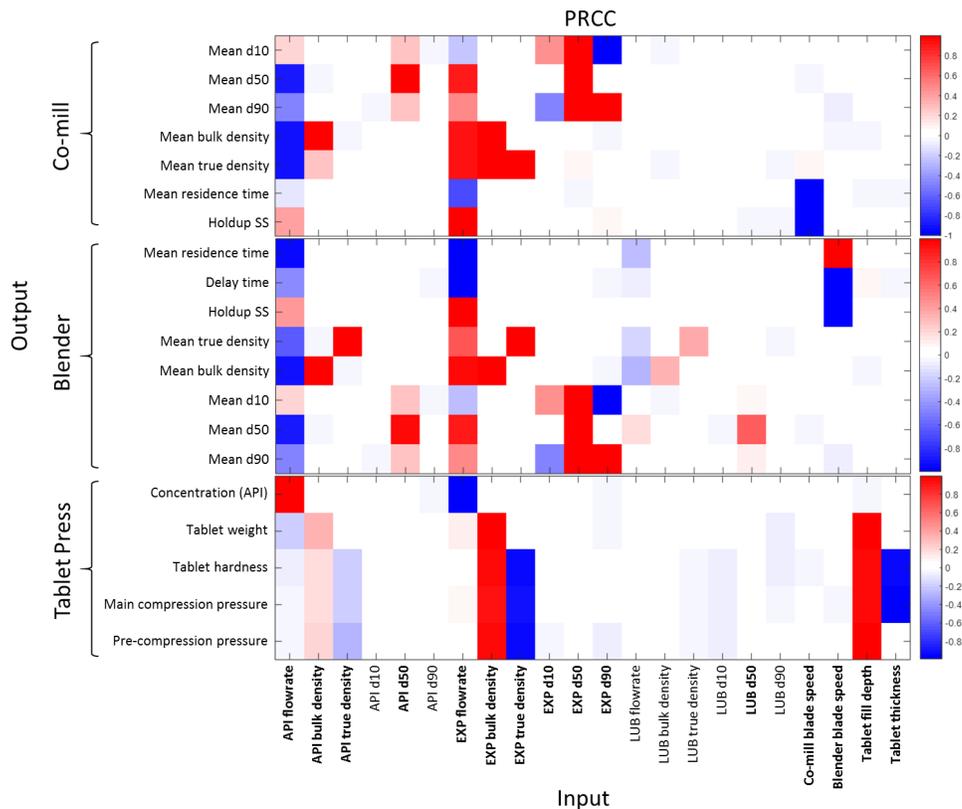


Figure 31. PRCC results on the DC line.

#### 5.4.1.1.1 Results of Sobol's method

After using Morris method and PRCC method to identify the subset of most influential input factors, practically we can implement Sobol's method on the subset of input factors identified as influential to get a quantitative sensitivity analysis. However, for the purpose of verifying the results of the Morris' and PRCC methods, we implement Sobol's method on all of the 22 input factors for sensitivity analysis. The Sobol's sensitivity analysis results are shown in Figure 32. Here only the values of total effect sensitivity index  $S_{Ti}$  are shown, since the values of the first-order effect  $S_i$  are found to be very close to  $S_{Ti}$ , which indicates that the interaction effects between each input is so weak, compared to the first-order effect, that can be ignored in this case study.

The Sobol's method identified 10 input factors to be most influential. 4 material properties include: Excipient bulk density, Excipient true density, Excipient d50, and Excipient d90. 6 operation conditions include: API flow rate, Excipient flow rate, Co-mill blade speed, Blender blade speed, Tablet fill depth, and Tablet thickness.

All these 10 input factors have been pre-identified as influential by Morris and PRCC methods, which suggests both of these methods are reliable for screening purposes in this case study. It should be noted that the mean residence time of the co-mill is more sensitive to blade speed than to flow rate. However, this is not the case for the mean residence time of the blender, in which flow rate plays a major role. One possible explanation for this is the different ranges of the blade speeds. For the co-mill, the blade speed is so fast (over 1000 rpm) that it can always quickly transfer the powder materials to the subsequent unit, without being much affected by the varying flow rate within its ranges. On the other hand, the blade speed of the blender (around 250 rpm) is not so fast,

and in this case the flow rate contributes mostly to the variance in the mean residence time of the blender.

Note that the sensitivity analysis results are highly affected by the ranges that are set by users[43]. With the same flowsheet model, a change in the input ranges can result in a completely different set of sensitivity indices values. Therefore, when applied to real-life cases, the ranges of input factors must be carefully chosen according to the actual production conditions. In terms of computation costs, the calculation of sensitivity metrics is fast for all the three mentioned methods, which can usually be completed within seconds of CPU time. It is running the simulations that takes the majority of the overall computational time. Therefore, when the simulation is computationally costly, it can be unaffordable to directly use variance-based method (Sobol's method) on the full set of input factors, which requires a large number of sample points. An alternative is to first implement the screening methods (Morris and/or PRCC) with a small number of sample points to get the preliminary sensitivity information and find the subset of most influential input factors. Then, use the Sobol's method on the subset of influential input in order to get a more detailed and quantitative sensitivity information. Note that for screening purpose, Morris and PRCC can usually be used together so as to extensively extract the sensitivity information. However, if it is unknown whether the relationship between input and output is monotonic, it is suggested only apply Morris method.

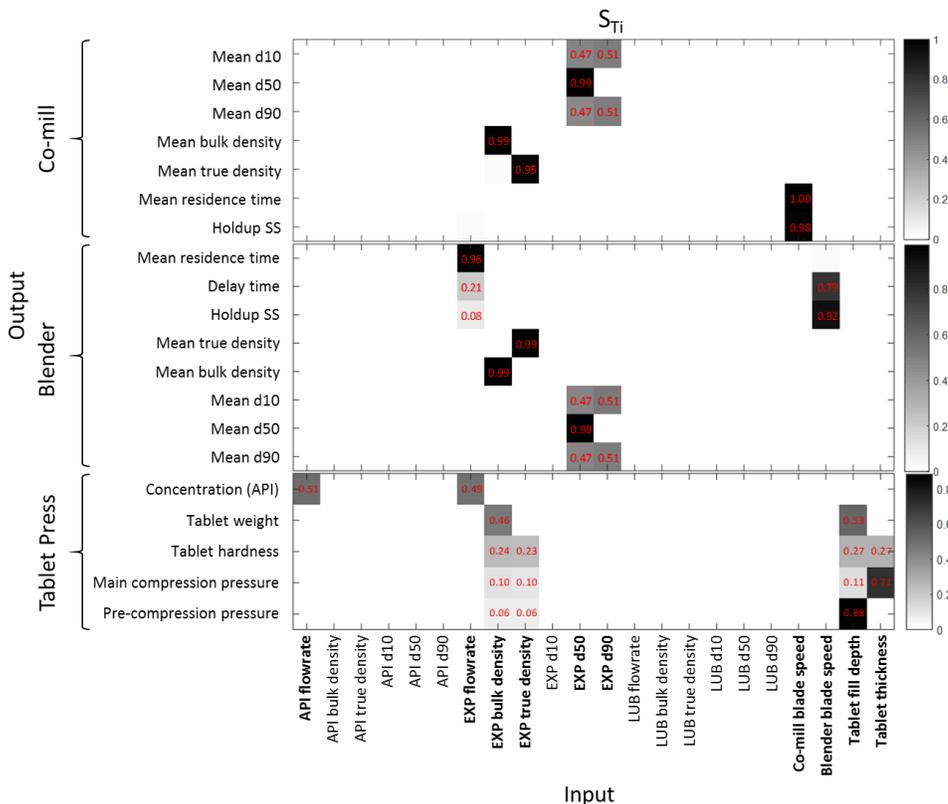


Figure 32. Sobol' total effect sensitivity index ( $S_{Ti}$ ) on the DC line.

### 5.4.2 Surrogate-based Feasibility Analysis Results

Following the sensitivity analysis results, 10 most influential factors should be taken into account for the feasibility analysis. Considering that for a process with specified component materials the material properties have constant values, when conducting the feasibility, we only investigate the 6 operating conditions and fix the remaining 4 material properties at their nominal values (Table 12). The constraints on the output variables are set as +/-5% of the nominal values, which are shown in Table 13. This gives us a 6D feasibility analysis problem with 40 inequality constraints based on the flowsheet model. To visualize the feasible region in high dimensions, we use a matrix of contour plots to describe feasibility, which is in the same manner by Forrester, Sobester et al.[251] The matrix consists of 15 contour plots, each of which depicts the feasibility of two distinct input

factors, with the rest four input fixed at the base line values. The base line can be any base case of interest, and may usually be chosen as the center point of the input space. We apply the surrogate-based feasibility analysis strategy to each pair of the input factors. In each of the 15 cases, 49 grid sample points are used to build the initial surrogate model, with an additional 100 adaptive sample points to improve the surrogate accuracy. This sampling budget is used because it is shown to provide accurate prediction even in highly nonconvex case studies[246].

Table 12. Ranges for 6 uncertain variables and nominal values for the remaining 4 material properties

<b>Input</b>	<b>LB</b>	<b>UB</b>	<b>Unit</b>
API flow rate	2.85	3.15	kg/hr
EXP flow rate	25.365	28.035	kg/hr
co-mill blade speed	1064	1176	RPM
blender blade speed	237.5	262.5	RPM
TP fill depth	0.0095	0.0105	m
TP thickness	0.002375	0.002625	m
<b>Fixed material properties</b>	<b>Nominal values</b>		<b>Unit</b>
EXP bulk density	400		kg/m <sup>3</sup>
EXP true density	2500		kg/m <sup>3</sup>
EXP d50	120		μm
EXP d90	250		μm

Table 13. Constraints for 20 output variables

<b>Output</b>	<b>LB</b>	<b>UB</b>	<b>Unit</b>
---------------	-----------	-----------	-------------

Blender	Mean residence time	15.609	17.252	s
	Delay time	26.797	29.618	s
	Mass holdup SS	0.402	0.444	kg
	Mean true density	2331.300	2576.700	kg/m <sup>3</sup>
	Mean bulk density	363.850	402.150	kg/m <sup>3</sup>
	Mean d10	16.908	18.688	μm
	Mean d50	110.010	121.590	μm
	Mean d90	203.112	224.492	μm
Co-mill	Mean d10	15.645	17.291	μm
	Mean d50	109.682	121.227	μm
	Mean d90	203.719	225.163	μm
	Mean bulk density	365.606	404.091	kg/m <sup>3</sup>
	Mean true density	2500.707	2763.939	kg/m <sup>3</sup>
	Mean residence time	21.922	24.229	s
	Mass holdup SS	0.181	0.200	kg
Tablet Press	Concentration	9.5	10.5	%
	Weight	1.8289e-04	2.0214e-04	kg
	Hardness	4.465	4.935	kp
	Main compression pressure	151.711	167.681	MPa
	Pre-compression pressure	46.735	51.655	MPa

The results of the surrogate-based feasibility analysis are shown in Figure 33. From the results, we can notice that the input factors “API flow rate ( $FR_{API}$ )”, “Excipient flow rate ( $FR_{Exp}$ )”, “Blender blade speed ( $RPM_{blender}$ )” are feasible almost over their whole ranges. The pair “Blender

blade speed ( $\text{RPM}_{\text{blender}}$ )” - “API flow rate ( $\text{FR}_{\text{API}}$ )” is feasible everywhere in the input ranges (contour in row 3, column 1). The other two pairs, namely “Excipient flow rate ( $\text{FR}_{\text{Exp}}$ )” - “API flow rate ( $\text{FR}_{\text{API}}$ )” (contour in row 1, column 1) and “Blender blade speed ( $\text{RPM}_{\text{blender}}$ )” - “Excipient flow rate ( $\text{FR}_{\text{Exp}}$ )” (contour in row 3, column 2), also have feasible regions covering the majority of the space. This indicates that the process can tolerate most of the variations in these three input factors while remaining process robustness. On the other hand, the remaining three input factors “Co-mill blade speed ( $\text{RPM}_{\text{co-mill}}$ )”, “tablet fill depth (FillDepth)”, and “tablet thickness (Thickness)” only have very narrow feasible regions. Especially for the two pairs, “Tablet fill depth (FillDepth)” - “Co-mill blade speed ( $\text{RPM}_{\text{co-mill}}$ )” (contour in row 4, column 3) and “Tablet thickness (Thickness)” - “Co-mill blade speed ( $\text{RPM}_{\text{co-mill}}$ )” (contour row 5, column 3), the feasible regions only take up small center areas. The co-mill blade speed has narrow feasible regions because it can drastically affect the mean residence time when it’s operated close to the minimum blade speed (in this case,  $\omega_{\text{min}} = 999$  RPM). On the other hand, the variations in tablet fill depth and thickness set point can cause large deviations from nominal values of tablet properties, such as tablet weight and hardness, and compression force. Therefore, the variations in these three input factors can most likely cause potential violations of the process constraints, such as process safety issues or product quality defects.

The matrix of feasibility contour plots gives a straightforward view on the extent to which the process is capable of tolerating the process variations while maintaining feasibility. Additionally, it also indicates where control strategies are needed to ensure better process robustness and product quality. The feasible regions can be used to guide the choice of target values of control variables when the process is experiencing deviations from normal production conditions.

To summarize the usage of sensitivity analysis and feasibility, sensitivity analysis tells us whether there's strong relationship between a certain input and output (screening methods), and the percentage that a certain input can contribute to the output variations (variance-based methods). Meanwhile, feasibility analysis indicates how large the variations in the output can be when changing the input factors, which in turn gives us the maximum ranges of the input (design space) that the process can tolerate while maintaining process robustness. While sensitivity and feasibility investigates the process from different angles, the combined analysis results give us a thorough understanding of the whole manufacturing process.

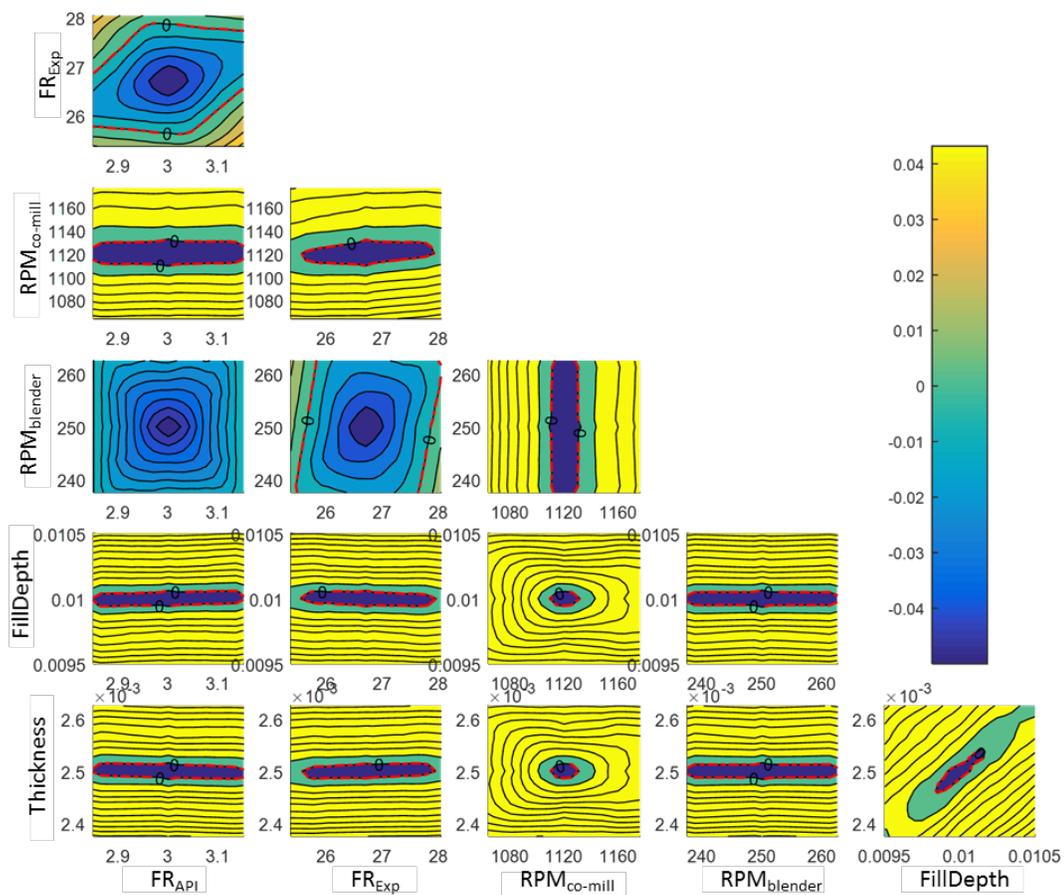


Figure 33. Contour plots of the feasibility of the DC line flowsheet model. (The red dotted line is the feasible region boundary. The area within the red dotted is the feasible region with feasibility function values less than or equal to zero.)

### 5.5 Optimization of the Direct Compaction line using flowsheet models

After conducting the process analysis, our final goal is to find the optimal operation conditions that result in minimum total operation cost. In order to correctly define the optimization problem, we need to look back at the actual pharmaceutical manufacturing process. To operate the process in a continuous manner, the feeders need to be refilled manually at a certain frequency. This procedure is unavoidable and can lead to temporal variations in the feeder outlet flow rate, which will propagate along the whole process[49]. Even though the variations are only temporal and will not affect the steady state feasibility, there is a possibility that such variations may cause temporal unqualified products, which need to be disposed and increase the waste costs. Therefore, the refill strategy must be considered when defining the optimization problem.

We define the refill strategy (RS) as the fill level of the feeder, below which the refill starts to take place. The refill will automatically stop once the fill level is over 90%. From the results of sensitivity analysis, we have identified the flow rate of API and Excipient as influential. Therefore, we add two more decision variables,  $RS_{API}$  and  $RS_{Exp}$ , when formulating the optimization problem.

With this adjustment, the optimization problem can be defined as follows:

$$\begin{aligned}
 & \min_{\theta, RS_{API}, RS_{Exp}} \quad cost_{total} = cost_{material} + cost_{utility} + cost_{waste} \\
 & s. t. \\
 & \psi(\theta) \leq 0 \\
 & TabletProductionRate(\theta, RS_{API}, RS_{Exp}) \geq DesiredProductionRate \\
 & \theta^{lb} \leq \theta \leq \theta^{ub} \\
 & RS_{API}^{lb} \leq RS_{API} \leq RS_{API}^{ub} \\
 & RS_{Exp}^{lb} \leq RS_{Exp} \leq RS_{Exp}^{ub} \\
 & \theta = \{FR_{API}, FR_{Exp}, RPM_{co-mill}, RPM_{blender}, Thickness, FillDepth\}
 \end{aligned} \tag{104}$$

The objective total cost considers the operation cost of a 24-hour continuous manufacturing time. Specifically this includes the following three components: the material cost calculated based on the market price of the component materials; the utility cost and waste cost calculated according to Schaber, Gerogiorgis et al.[238], which suggests utility cost as \$1.50/kg material input, and waste cost as \$10.36/kg material input (adapted for the materials in this work). The inequality constraints mainly include the steady-state feasibility and the production rate of qualified tablet products. In this optimization problem, the generation of waste is implicitly affected by the decision variables and is very difficult to be expressed with an explicit model equation. This causes the two terms, namely “ $cost_{waste}$ ” and “ $TabletProductionRate$ ”, to be black-box terms, for which we do not have easy access to the exact model expressions. The bounds of the decision variables are listed in Table 14, and the desired tablet production rate is  $0.95 \times 3.7379 \times 10^6$  tablets per day.

Table 14. Bounds of the decision variables

<b>Input</b>	<b>LB</b>	<b>UB</b>	<b>Unit</b>
$FR_{API}$	2.85	3.15	kg/hr
$FR_{Exp}$	25.365	28.035	kg/hr
$RPM_{co-mill}$	1064	1176	RPM
$RPM_{blender}$	237.5	262.5	RPM
$FillDepth$	0.0095	0.0105	m
$Thickness$	0.002375	0.002625	m
$RS_{API}$	10	70	%
$RS_{Exp}$	10	70	%

This optimization problem is solved in Matlab 2015a with tomlab/conopt solver[252], using the mid-point (nominal values) as the initial guess. The interface “gO: MATLAB” [253] is used to

automatically communicate between Matlab and gPROMS at each function call during the optimization process. The flowsheet model in gPROMS consists of 5315 model equations (not including equations for initial conditions). In terms of the computational cost, each function call of gPROMS simulation takes approximately 30 to 60 seconds (cpu time) depending on the values of simulation input. With the formulated optimization problem (104), the feasible optimal solution is found after 39 iterations (1315 function evaluations), which is listed in Table 15. The minimum total cost is \$ 94515.9 per day.

Table 15. Optimal operation conditions

<b>Input</b>	<b>Optimal Values (lb - ub)</b>	<b>Unit</b>
$FR_{API}$	2.9119 (2.85 - 3.15)	kg/hr
$FR_{Exp}$	25.5759 (25.365 - 28.035)	kg/hr
$RPM_{co-mill}$	1112.6 (1064 - 1176)	RPM
$RPM_{blender}$	240.7924 (237.5 - 262.5)	RPM
$FillDepth$	0.01 (0.0095 - 0.0105)	m
$Thickness$	0.0025 (0.002375 - 0.002625)	m
$RS_{API}$	40 (10 - 70)	%
$RS_{Exp}$	39.88 (10 - 70)	%

It should be noticed that the flow rate of API and Excipient are both lower than their nominal values. This is because by using small flow rate the material and utility cost are reduced. Accordingly, the co-mill blade speed and blender blade speed slightly deviate from the nominal values, which will help ensure feasibility in terms of steady mass hold-up and mean residence time. On the other hand, the fill depth and thickness of the tablet press are strictly at their nominal values. From the feasibility analysis results, we know these two variables have very small feasible regions,

and thus need to stay near the nominal values in order to keep feasibility. Finally, in terms of the refill strategy, we notice that the optimal solution suggests both API and Excipient feeders start to refill around the level of 40%. Even though a more frequent refill strategy can generate even smaller variations in the flow rate[254], this won't be necessary for the current process, because the variations can be dampened by the blender. This can be seen from Figure 34, which shows the concentration at the inlet (blue line) and the outlet (red line) of the blender at the current optimal operating conditions. For simplicity, we only show the first 1000 seconds of simulation. While we can notice very large variations at the inlet, they are dampened into much smaller ones after the materials transfer to the outlet of the blender.

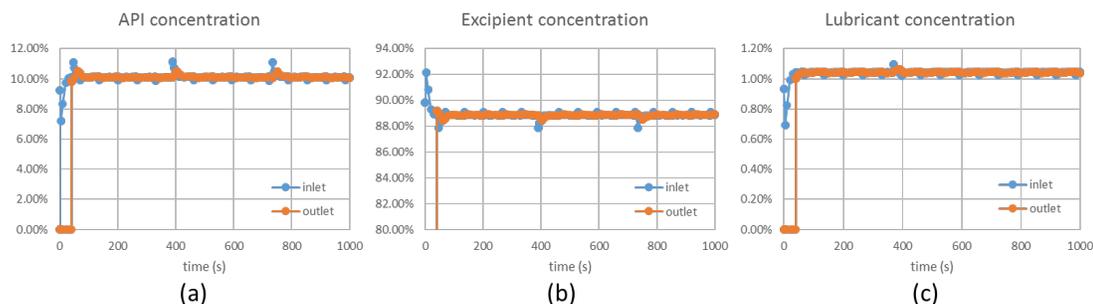


Figure 34. Dynamic simulation results of the concentration at inlet and outlet of the blender. (a) API concentration; (b) Excipient concentration; (c) Lubricant concentration.

At the optimal operation conditions, the dynamic simulation results of the tablet product properties within the first 1000 seconds of manufacture are shown in Figure 35. (a), (b) and (c). The solid lines are the simulation results, and the dashed lines are the upper and lower bounds of the product qualities. (d) shows the index of the waste. When any of the three quality constraints is violated, the waste index is assigned the value of 1, meaning that such products are not qualified and need to be disposed. From the figure we can see that at the optimal operation condition, the only waste products are produced during the starting stage (40s to 100s), when the tablet weights are lower than the target values. As soon as the process arrives at its steady state, the process does not generate

any more waste products. Even though there exist variations resulted from feeder refill, they are within the bounds of the tablet product qualities. Because the refills occur periodically, we can anticipate that the process will not generate any more unqualified products after the 1000 seconds.

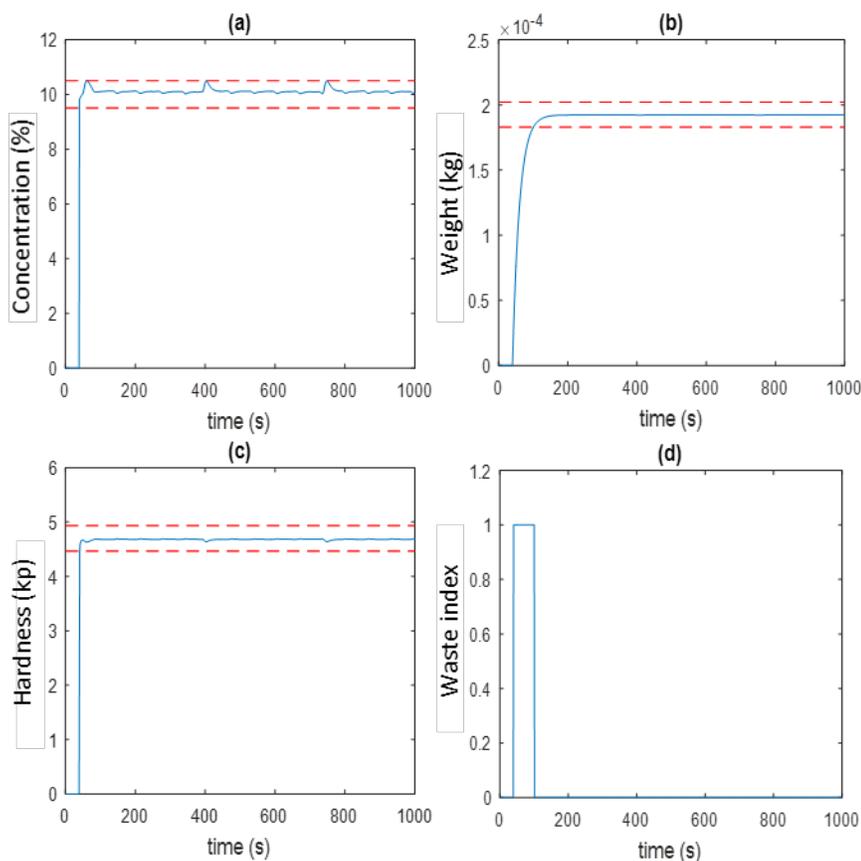


Figure 35. Dynamic simulation results of tablet properties at the optimal operation conditions. (a) API concentration; (b) tablet weight; (c) tablet hardness; (d) waste index. (Solid line: simulation results. Dashed lines: upper and lower bounds of the quality constraints).

## 5.6 Conclusion

This work proposed a systematic way to conduct process analysis and optimization using a flowsheet model of a continuous DC pharmaceutical manufacturing process. Global sensitivity analysis is first implemented to analyze the effects of the 22 input factors on the 20 outputs, and

narrow down to a subset of 10 input factors which contribute most to the variations in the output. To reduce the overall sampling cost, the combination of screening methods (Morris and/or PRCC) and a variance-based method (Sobol's) can be used to obtain the sensitivity information. Then, feasibility analysis is conducted to identify the feasible region (design space) of the process. A surrogate-based adaptive sampling approach is used to improve the efficiency and accuracy of the feasibility analysis, which is especially suitable for computationally expensive models. The feasibility analysis results provide a general view on the process capabilities of conserving robustness when facing variations in the input, and can direct the research in developing control strategies. Finally, process optimization is conducted to find the optimal operation conditions and refill strategy, in order to minimize the total operation costs. Future work includes improving the unit models, and developing strategies for process analysis to consider noises in the output.

## IV. Process analysis and optimization for stochastic systems

## 6 Surrogate-based feasibility analysis for black-box stochastic simulations with heteroscedastic noise variances

### **Abstract**

Feasibility analysis has been developed to evaluate and quantify the capability that a process can remain feasible under uncertainty of model inputs and parameters. It can be conducted during the design stage when the objective is to get a robust design which can tolerate a certain amount of variations in the process conditions. Also, it can be used after a design is fixed when the objective is to characterize its feasible region. In this work, we have extended the usage of feasibility analysis to the cases in which inherent stochasticity is existent in the model outputs. With a surrogate-based adaptive sampling framework, we have developed and compared three algorithms that are promising to make accurate predictions on the feasible regions with a limited sampling budget. Both the advantages and limitations are discussed based on the results from five benchmark problems. Finally, we apply such methods to a pharmaceutical manufacturing process and demonstrate its potential application in characterizing the design space of the process.

### 6.1 Introduction

Most realistic optimization problems are characterized by a degree of uncertainty. For example, in chemical processes, uncertainties can exist in different sources, such as the variations of reaction constants, material physical properties, and fluctuations of the flow rate of inlet streams. Such uncertainties can propagate along the process and affect the overall process behaviors and final product qualities. Therefore, it is essential to account for their influence on the optimal solution. In the process systems engineering community, process feasibility and flexibility analysis tools have been developed to evaluate and quantify the ability of a system to be operated feasibly in the

presence of uncertainties. The concepts and formulations for feasibility and flexibility analysis are reviewed as follows.

Halemane and Grossmann [213] mathematically defined the feasibility with a feasibility function  $\psi(D, \theta)$  in Equation (105):

$$\psi(D, \theta) = \min_z \max_{j \in J} \{f_j(D, z, \theta)\}, \quad (105)$$

where  $D \in \mathbb{R}^{n_D}$  represents the design variables such as equipment sizes, which are constants for a fixed process design;  $\theta \in \mathbb{R}^{n_\theta}$  is the uncertain parameters;  $z \in \mathbb{R}^{n_z}$  denotes the control variables that can be adjusted during operations (e.g., flow rate).  $f_j, j \in J = \{1, 2, \dots, n_f\}$ , are a set of  $n_f$  inequality constraints ( $f_j \leq 0$ ) that need to be met. With this definition,  $\psi(D, \theta)$  is used to determine whether, for a fixed parameter  $\theta$ , the process can be feasible by adjusting the control variable  $z$ , which is simply accomplished by selecting  $z$  in order to minimize the largest value of  $f_j$ . In the cases when there are no control variables,  $\psi(D, \theta)$  is reduced to Equation (106):

$$\psi(D, \theta) = \max_{j \in J} \{f_j(D, \theta)\}, \quad (106)$$

where the feasibility function can be seen as the maximum violation of all the constraints. A process is only feasible in regions where  $\psi(D, \theta) \leq 0$ .

Based on the definition of feasibility function, traditional flexibility analysis can be formulated as a “flexibility test problem” [213]:

$$\chi(D) = \max_{\theta \in T} \min_z \max_{j \in J} \{f_j(D, z, \theta)\}. \quad (107)$$

If  $\chi(D) \leq 0$ , then the process is feasible over the whole range of parameter space  $T$ . If  $\chi(D) > 0$ , it means that there exist some regions where the feasible operation cannot be achieved.

The limitation of “flexibility test problem” is that it only reflects whether a design can be feasibly operated over the whole parameter range. In order to quantitatively measure the flexibility for a process design, Swaney and Grossmann [255,256] further proposed the formulation of “flexibility index problem” :

$$\begin{aligned}
 F &= \max \delta \\
 \text{s. t. } \chi(D) &= \max_{\theta \in T} \min_z \max_{j \in J} \{f_j(D, z, \theta)\} \\
 T(\delta) &= \{\theta: \theta^N - \delta \Delta \theta^- \leq \theta \leq \theta^N + \delta \Delta \theta^+\} \\
 \delta &\geq 0,
 \end{aligned} \tag{108}$$

where  $\theta^N$  is the nominal point for the uncertain parameters,  $\Delta \theta^+$  and  $\Delta \theta^-$  are the expected deviations of the uncertain parameters in the positive and negative directions;  $\delta$  is a nonnegative scalar variable.  $F$  is called “flexibility index”. Geometrically,  $T(F)$  can be interpreted as the largest hyper-rectangle inscribed within the operation region where the process is guaranteed to be feasible. Different approaches were proposed to solve the “flexibility test problem” and “flexibility index problem”, including vertex enumeration method [213], active-set method [214], and global optimization strategy [257]. Based on the “flexibility index problem”, several extensions have been made. Pistikopoulos and Mazzuchi [258] defined the “stochastic flexibility index” for problems where the uncertain parameters  $\theta$  are described by a joint probability distribution function rather than by lower and upper bounds. Dimitriadis and Pistikopoulos [259] further extended flexibility analysis to dynamic systems which contain time-dependent uncertain parameters. Adi and Chang [69] formulated a temporal flexibility analysis problem to consider the cumulative effects of temporary disturbances on process feasibility.

The drawback for “flexibility index problem” is that  $T(F)$  (i.e., the hyper-rectangle feasible subdomain) is not an accurate reflection of the true feasible region, and it can largely underestimate

a design's feasibility in cases when  $F < 1$ . In order to describe the true feasible region (referred to as the "feasibility analysis problem"), Ierapetritou [260] developed a "Quickhull algorithm" which constructs a convex hull of the feasible region and proposed a new metric "feasible convex hull ratio" to compare designs' feasibility. The key to feasibility analysis is to have an accurate representation of the true feasible region. To achieve this, Goyal and Ierapetritou [218] described a simplicial approximation method for convex cases. Banerjee and Ierapetritou [215] presented a  $\alpha$ -shape surface reconstruction method which can handle both convex and non-convex feasible regions. Adi et al. [216] used a random line search algorithm to identify feasible region boundary points. Michalewicz and Schoenauer [217] utilized an evolutionary algorithm to search the boundary of the feasible region.

Surrogate-based approaches have been adopted recently for feasibility analysis. Such methods build a surrogate model as an efficient approximation of the original feasibility function  $\psi(D, \theta)$ . The advantages of surrogate-based methods are multifold. First, they do not require closed-form constraints to conduct feasibility analysis. Instead, they treat the whole process as a black-box, and only inputs (i.e., uncertain parameter values) and outputs (feasibility function values) are needed to build a surrogate. Second, they usually only need a few data points to build the surrogate that can give a high accuracy in approximating the original process. Therefore, they are quite suitable for computationally expensive simulations. Finally, when combined with adaptive sampling methods, the surrogate-based approaches can be even more efficient in sampling while maintaining prediction accuracy. Surrogate-based methods have been mentioned in numerous papers on feasibility analysis. Banerjee et al. [219] demonstrated a high dimensional model representation (HDMR) approach to approximate the black-box process and find the feasible region in the original problem dimension. Boukouvala et al. [18] compared the accuracy of different surrogate models in predicting the design space of pharmaceutical processes. Boukouvala and Ierapetritou developed a

Kriging-based adaptive sampling approach for feasibility analysis. Rogers and Ierapetritou [65,48] extended the Kriging-based approach to solving dynamic feasibility analysis and stochastic flexibility analysis problems. Wang and Ierapetritou [261] presented a Radial Basis Function (RBF-based) adaptive sampling method that outperformed the Kriging-based approach for low-dimensional problems.

While the surrogate modeling techniques have been used in a variety of problems, there are a few characteristics of surrogate-based feasibility analysis that distinguish itself from other types of problems. Below, we list the similarities and differences between the surrogate-based feasibility analysis and four types of problems:

(1) *Global metamodeling* problems use a surrogate model (metamodel) to “map” the process response surface (output variable) as a function of decision variables (input variables) and thus approximate the behavior of a complex computer simulation [262].

Similarities

- Both require sampling strategies to improve the approximation accuracy

Differences

- Surrogate-based feasibility analysis only focuses on improving modeling accuracy at (or near) the feasibility boundaries.
- Global metamodeling aims to achieve high predictive accuracy across the entire input space

(2) *Surrogate-based global optimization* problems rely on a surrogate model to search for the global optimum [67].

Similarities

- Both require global search strategies to explore the unknown space, combined with local search strategies to exploit the regions of interest

#### Differences

- Surrogate-based feasibility analysis needs to predict the feasibility boundaries, which can be lines (of surfaces) that consist of infinite numbers of points
- Surrogate-based global optimization attempts to find the global optimum, which is usually a single point (or a limited number of points)

(3) *Feasibility determination* is to determine whether the performance of each alternative design, which can only be estimated via a stochastic simulation, exceeds a known threshold [263].

#### Similarities

- Both aim to distinguish between feasible designs (or regions) and infeasible designs (or regions).

#### Differences

- Surrogate-based feasibility usually considers deterministic simulations with continuous uncertain parameters  $\theta \in \mathbb{R}^{n_\theta}$ .
- Feasibility determination problems mostly consider stochastic simulations with discrete events.

(4) *System reliability analysis* is aimed to analyze the probabilities of system success considering various system performances (process constraints) [264,265].

#### Similarities

- Both consider continuous uncertain parameters.

#### Differences

- Surrogate-based feasibility analysis usually considers uncertain parameters specified by a fixed parameter set.
- System reliability analysis considers uncertain parameters described by a joint probability density function.

With the evolving concepts and formulations for flexibility and feasibility analysis, we have seen its applications in a variety of areas, ranging from chemical processes [44,266], pharmaceutical manufacturing processes [267,268], to energy systems [269], supply chain management [270,271]. Zhang and Grossmann [272] demonstrated the close relation between flexibility analysis and robust optimization for linear systems. Boukouvala and Ierapetritou [64] used feasibility analysis as a way to deal with black-box constraints and integrate it into a derivative-free optimization framework.

Traditional feasibility analysis has been developed solely for deterministic models (or simulations). However, to the best of our knowledge, no research has been conducted to apply feasibility analysis to stochastic systems. In this paper, the term “stochastic” is used to describe the case where with replicated simulation runs for the same sample point, we will get different observed feasibility values due to random errors inherent to the simulation. Since feasibility analysis is closely related to optimization, and the optimization strategies for stochastic systems (generally categorized as “Simulation Optimization” [273]) has wide applications covering different areas from operations, manufacturing to medicine, engineering, etc. [273], it is important that we develop efficient methods to extend feasibility analysis to stochastic systems.

In this work, we only consider stochastic simulations with continuous parameters  $\theta$ . We assume that the feasibility function  $\psi(D, \theta)$  in Equation (106) cannot be directly observed, and can only be

estimated with the stochastic simulation. Thus, the user only has access to noisy observations of the feasibility  $\tilde{\psi}_i(D, \theta)$ :

$$\tilde{\psi}_i(D, \theta) = \psi(D, \theta) + \omega_i(D, \theta), \quad (10)$$

where  $\tilde{\psi}_i(D, \theta)$  is the observed feasibility function value on the  $i^{th}$  replication of simulation evaluated at the parameter value  $\theta$ ;  $\omega_i(D, \theta)$  is the observed noise term. In this paper, we make the following assumptions on the noise term, which were used in a simulation optimization literature [274]: the observed noises are normally distributed, centered at zero mean, and independent from different simulation runs:

$$\omega_i(D, \theta) \sim \text{Norm}(0, \xi^2(D, \theta)), \quad (110)$$

where  $\xi^2(D, \theta)$  is the variance of the noise term, which depends on the parameter value  $\theta$ , and the function form is assumed to be unknown. In such cases, the simulation has heteroscedastic noise inherently. The objective is to identify the feasible region  $\{\theta: \psi(D, \theta) \leq 0\}$ , without any closed-form expressions (i.e., no knowledge on  $\psi(D, \theta)$  or  $\xi^2(D, \theta)$ ) and the only accessible information is the noisy observations  $\tilde{\psi}_i(D, \theta)$  at parameter  $\theta$ .

In this paper, we propose a surrogate-based adaptive sampling framework for the feasibility analysis of black-box stochastic systems. In this framework, stochastic Kriging [162] is used as the surrogate, and three adaptive sampling methods are developed to sequentially find the next sample point and improve the prediction accuracy of the true feasible region.

## 6.2 Surrogate-based approaches for feasibility analysis

### 6.2.1 Stochastic Kriging

The Kriging model is a popular approach to approximating deterministic models and has been widely used in various science and engineering fields [275-277]. Kriging considers a response

surface as a realization of a Gaussian random field which has spatial correlations. Recently, Stochastic Kriging (SK) [162] was proposed as an extension to Kriging by accounting for the intrinsic noise associated with stochastic simulations. Aside from being used as a meta-modeling approach for stochastic systems, SK model also shows promising results in optimization algorithms for simulation optimization problems [278]. In this section, we provide a brief review on the basic formula for SK model. More details can be found in the original paper by Ankenman et al. [162]

SK model represents the simulation output on  $j^{th}$  replication at a sample point  $\mathbf{x}$  ( $\mathbf{x} \in \mathbb{R}^d$ ) as:

$$y_j(\mathbf{x}) = \beta_0 + M(\mathbf{x}) + \varepsilon_j(\mathbf{x}), \quad (11)$$

where  $\beta_0$  is a model parameter representing a constant trend;  $M$  is a realized random field with a zero mean, which can be considered as  $M$  being randomly sampled from a space of functions mapping  $\mathbb{R}^d \rightarrow \mathbb{R}$ . It is assumed that the functions in this space exhibit spatial correlations:  $M(\mathbf{x})$  and  $M(\mathbf{x}')$  will tend to be similar if  $\mathbf{x}$  is close to  $\mathbf{x}'$ . The correlation between  $M(\mathbf{x})$  and  $M(\mathbf{x}')$  can be modeled with different forms of a correlation function. In this work, we apply the cubic correlation function:

$$\text{Corr}(\mathbf{h}|\boldsymbol{\zeta}) = \prod_{i=1}^d \text{Corr}(h_i|\zeta_i), \quad (112)$$

where

$$\text{Corr}(h_i|\vartheta_i) = \begin{cases} 1 - 6(|h_i|/\vartheta_i)^2 + 6(|h_i|/\vartheta_i)^3, & \text{if } |h_i| \leq \vartheta_i/2 \\ 2(1 - |h_i|/\vartheta_i)^3, & \text{if } \vartheta_i/2 \leq |h_i| \leq \vartheta_i \\ 0, & \text{if } |h_i| \geq \vartheta_i \end{cases} \quad (113)$$

$$\mathbf{h} = \mathbf{x} - \mathbf{x}', \boldsymbol{\zeta} = (\vartheta_1, \dots, \vartheta_d), \vartheta_i > 0, \text{ and } \text{Corr}(h_i|0) \equiv 0.$$

The covariance between  $M(\mathbf{x})$  and  $M(\mathbf{x}')$  can thus be expressed as  $\text{Cov}(\mathbf{h}|\boldsymbol{\zeta}) = \tau^2 \text{Corr}(\mathbf{h}|\boldsymbol{\zeta})$ , where  $\tau^2$  can be interpreted as the variance of  $M(\mathbf{x})$  for all  $\mathbf{x}$ .  $\varepsilon_j(\mathbf{x})$  in Equation (111) represents

the realized noise on replication  $j$ . In [162],  $M$  is termed as “extrinsic uncertainty” as it is introduced to aid in developing the surrogate;  $\varepsilon$  is referred to as “intrinsic uncertainty” because it originates from the nature of stochastic simulations.

For the SK model, Ankenman et al. [162] made the following assumptions: “*the random field  $M$  is a stationary Gaussian random field, and  $\varepsilon_1(\mathbf{x}_i)$ ,  $\varepsilon_2(\mathbf{x}_i)$ ,  $\dots$  are independent and identically distributed with a normal distribution:  $\text{Norm}(0, V(\mathbf{x}_i))$ , independent of  $\varepsilon_j(\mathbf{x}_h)$  for all  $j$  and  $h \neq i$ , and independent of  $M$ ” . It is proved in [162] that such assumptions imply that for any set of  $k$  different sample points  $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_k$ , the random vector  $[M(\mathbf{x}_1), M(\mathbf{x}_2), \dots, M(\mathbf{x}_k)]^T$  has a multivariate normal distribution with marginal mean 0, positive variance  $\tau^2$ , and positive definite correlation matrix  $\mathbf{R}_M$ .*

The intrinsic variance at a sample point  $\mathbf{x}_i$  can be estimated with  $n_i$  replications:

$$\hat{V}(\mathbf{x}_i) = \sum_{j=1}^{n_i} (y_j(\mathbf{x}_i) - \bar{y}(\mathbf{x}_i))^2 / (n_i - 1), \quad (114)$$

where  $\bar{y}(\mathbf{x}_i)$  is the estimated mean at  $\mathbf{x}_i$ ,  $\bar{y}(\mathbf{x}_i) = \sum_{j=1}^{n_i} y_j(\mathbf{x}_i) / n_i$ . Ankenman et al. [162] showed that the surrogate accuracy will not be sacrificed with the estimated  $\hat{V}$ , as long as  $n_i$  is not too small.

Aside from estimating  $\hat{V}$ , the model parameters (i.e.,  $\beta_0, \vartheta_1, \dots, \vartheta_d, \tau^2$ ) are obtained by maximizing the log-likelihood (derived in Ref. [162]). Based on the assumptions for SK that were mentioned above, the unbiased predictor is derived as follows:

$$\hat{y}(\mathbf{x}) = \beta_0 + \boldsymbol{\Sigma}_M(\mathbf{x}, \cdot)^T [\boldsymbol{\Sigma}_M + \widehat{\boldsymbol{\Sigma}}_\varepsilon]^{-1} (\bar{\mathbf{y}} - \beta_0 \mathbf{1}_k), \quad (115)$$

where  $\boldsymbol{\Sigma}_M$  is the  $k$ -by- $k$  covariance matrix for all the  $k$  sample points  $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_k$ ;  $\boldsymbol{\Sigma}_M(\mathbf{x}, \cdot)$  is  $k$ -by-1 covariance vector with the  $i^{th}$  element being  $\text{Cov}(\mathbf{x}, \mathbf{x}_i)$ ;  $\widehat{\boldsymbol{\Sigma}}_\varepsilon$  is the estimated covariance

diagonal matrix:  $\widehat{\Sigma}_{\varepsilon} = \text{Diag}\left\{\frac{\widehat{v}(x_1)}{n_1}, \frac{\widehat{v}(x_2)}{n_2}, \dots, \frac{\widehat{v}(x_k)}{n_k}\right\}$ ;  $\bar{\mathbf{y}}$  is the  $k$ -by-1 vector with the  $i^{th}$  element being  $\bar{y}(x_i)$ .

SK model can also provide estimated prediction variance:

$$\hat{s}^2(\mathbf{x}) = \tau^2 - \Sigma_{\mathbf{M}}(\mathbf{x}, \cdot)^T [\Sigma_{\mathbf{M}} + \widehat{\Sigma}_{\varepsilon}]^{-1} \Sigma_{\mathbf{M}}(\mathbf{x}, \cdot) + \boldsymbol{\delta}^T \boldsymbol{\delta} \left( \mathbf{1}_k^T [\Sigma_{\mathbf{M}} + \widehat{\Sigma}_{\varepsilon}]^{-1} \mathbf{1}_k \right)^{-1}, \quad (116)$$

where  $\boldsymbol{\delta} = \mathbf{1} - \mathbf{1}_k^T [\Sigma_{\mathbf{M}} + \widehat{\Sigma}_{\varepsilon}]^{-1} \Sigma_{\mathbf{M}}(\mathbf{x}, \cdot)$ . The matrix  $[\Sigma_{\mathbf{M}} + \widehat{\Sigma}_{\varepsilon}]$  is always positive definite based on the assumptions made on SK model, which have been mentioned above.

## 6.2.2 Adaptive sampling methods

The purpose of adaptive sampling is to identify the next sample point (i.e., infill point) in the input space, and guide the search direction towards the promising regions. With the sequentially added infill points based on some infill criteria, the surrogate also gets updated. For feasibility analysis, the key to identifying feasible region is to accurately identify the boundary between feasibility and infeasibility (i.e., boundaries where  $\psi(D, \theta) = 0$ ). In this section, we present three infill criteria that can be used with SK model for feasibility analysis.

### 6.2.2.1 Expected improvement for feasibility ( $EI_{feas}$ )

The  $EI_{feas}$  method was first proposed by Boukouvala and Ierapetritou [64], which was initially designed for Kriging-based feasibility analysis of deterministic models. This method determines the next sample point by maximizing the following objective function:

$$EI_{feas}(\mathbf{x}) = \hat{s} \cdot \phi\left(\frac{f^t - \hat{\mu}}{\hat{s}}\right), \quad (117)$$

where  $\hat{\mu}$  is the Kriging predictor of the black-box feasibility function;  $\hat{s}$  is the prediction standard deviation;  $\phi$  is the probability density function for a standard normal distribution.  $f^t$  is equal to 0

for feasibility analysis. They called it as an “expected improvement based” (EI-based) criterion because it is actually the second term of the classical EI function used in the “efficient global optimization” (EGO) algorithm [67], with the only difference being that  $f^t$  is set to 0. (Note: in the rest of this paper,  $f^t$  is equal to 0 by default unless otherwise noted.)

To understand how this infill criterion works, we can analyze the physical meanings of the two components of the  $EI_{feas}$  function:  $\hat{s}$  and  $\phi\left(\frac{f^t - \hat{\mu}}{\hat{s}}\right)$ .  $\hat{s}$  is an indication of the prediction uncertainty; the probability density function  $\phi\left(\frac{f^t - \hat{\mu}}{\hat{s}}\right)$  provides a *relative likelihood* of the random variable being equal to  $\left(\frac{f^t - \hat{\mu}}{\hat{s}}\right)$ .  $\phi\left(\frac{f^t - \hat{\mu}}{\hat{s}}\right)$  is largest when  $\left(\frac{f^t - \hat{\mu}}{\hat{s}}\right) = 0$  (i.e.,  $f^t = \hat{\mu}$ ). Therefore, when maximizing  $EI_{feas}$ , it favors the area where  $\hat{s}$  is large, that is, where prediction uncertainty is high. This can be seen as a “global search” as it tends to sample in highly unexplored areas. On the other hand, maximizing  $EI_{feas}$  also favors areas where  $\phi\left(\frac{f^t - \hat{\mu}}{\hat{s}}\right)$  is larger, that is when  $\hat{\mu}$  is close to (or equal to)  $f^t$ . This can be seen as a “local search” because it tends to sample near (or at) the feasibility boundary predicted by the current surrogate (i.e.,  $\hat{\mu} = 0$ ). Similar conclusions can also be drawn by analyzing the partial derivatives:  $\partial EI_{feas} / \partial \hat{\mu}$  and  $\partial EI_{feas} / \partial \hat{s}$ . Details can be found in Ref. [261].

In this paper, we derive  $EI_{feas}$  by formally defining a new term “improvement for feasibility” and calculating its expected value. This procedure is closely related to the use of “expected improvement (EI)” [67] in global optimization. In various publications, the EI criterion is also termed as “average improvement” [279] or “Bayesian algorithm” [280,281], of which the theoretical substantiation and applications have been reviewed by Žilinskas and Zhigljavsky [279].

For feasibility analysis, there are quite a few differences in the definition of “improvement” and the calculation of the expected value compared to the global optimization approaches.

Suppose we have a Kriging model built with a set of sample points. At any unsampled point  $\mathbf{x}$ , the value of  $y(\mathbf{x})$  is not known. Such an uncertainty can be described using a Kriging model, which treats  $y(\mathbf{x})$  as the realization of a normally distributed random variable  $Y$  with mean  $\hat{\mu}$ , and variance  $\hat{s}^2$ . This is demonstrated with the following one-dimensional problem (Figure 36). If we treat the function’s value at  $\mathbf{x}^*$  as a realization of the random variable  $Y$  with the density function shown in Figure 36, then there is probability that the function’s value will be smaller or larger than  $\hat{\mu}$ . For example, if the realized value is  $y_1$ , ( $y_1 < \hat{\mu}$ ), then the deviation from the Kriging predictor is  $\Delta_1 = (\hat{\mu} - y_1)$ ; if the realized value is  $y_2$ , ( $y_2 > \hat{\mu}$ ), then the deviation is  $\Delta_2 = (y_2 - \hat{\mu})$ . The deviation is in fact a measure of the “inaccuracy” of the Kriging predictor compared to a possible realized function value.

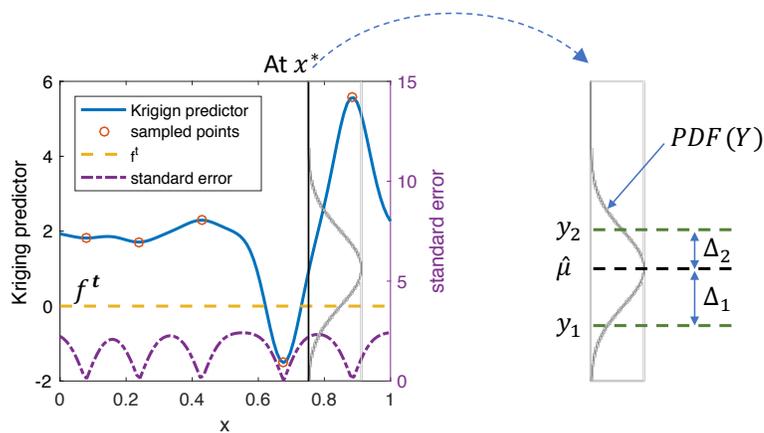


Figure 36. Demonstration of a one-dimensional Kriging model.

The deviation has a wide range of possible values from 0 to  $\infty$ . Different deviations are associated with different density values. However, for feasibility analysis, we are not interested in all possible values. Instead, we are more interested in the range  $[|\hat{\mu} - f^t|, \infty)$ , where  $|\cdot|$  denotes the absolute

value. The expected value of deviations within this range evaluates the average deviations that can be larger than  $|\hat{\mu} - f^t|$ , which is a measure of the average uncertainty of  $Y$  to deviate further than  $f^t$ . If the value is large, then by sampling at  $\mathbf{x}$ , we can achieve a large improvement on reducing the uncertainty of whether the function value is  $f^t$  at the point  $\mathbf{x}$ , which is translated to whether it can be the feasibility boundary at the point  $\mathbf{x}$ .

Based on the previous analysis, we give the formal definition of “improvement for feasibility ( $I_{feas}$ )” as follows:

$$I_{feas} = \begin{cases} \hat{\mu} - Y, & \text{if } Y \leq f^t \leq \hat{\mu} \\ 0, & \text{else} \\ Y - \hat{\mu}, & \text{if } Y \geq f^t > \hat{\mu}. \end{cases} \quad (118)$$

As we have mentioned above,  $I_{feas}$  is in fact the possible realized deviation (i.e., “inaccuracy” of the Kriging predictor) that is defined only in the range  $\{Y: 0 < f^t - \hat{\mu} \leq Y - \hat{\mu} < \infty \cup 0 \leq \hat{\mu} - f^t \leq \hat{\mu} - Y < \infty\}$ . There are several things we should note for this definition: (1) In Equation (118),  $Y$  is a random variable which models the uncertainty around the black-box function’s value at  $\mathbf{x}$  (remember that with Kriging model,  $Y$  is normally distributed:  $Y \sim \text{Norm}(\hat{\mu}, \hat{s}^2)$ ); therefore,  $I_{feas}$  is also a random variable; and (2)  $I_{feas}$  always has a non-negative value by definition.

The expression for the expected value of  $I_{feas}$  (i.e.,  $E[I_{feas}(Y)]$ ) can be obtained following the similar procedures that were considered in Ref. [67,282]. After derivation, we can get the same expression that is shown in Equation (117). Details on the derivation can be found in Appendix A of this paper.

The geometric interpretation of defining  $I_{feas}$  with Equation (118) is shown in Figure 29. Here, we only discuss the case when  $f^t \leq \hat{\mu}$ . The other case of  $f^t > \hat{\mu}$  can be explained in a similar way by

symmetry. We know by definition that  $E[I_{feas}(Y)] = \int_{-\infty}^{f^t} I_{feas}(Y)PDF(Y)dY$ . The probability density function of  $Y$  and  $I_{feas}$  is plotted in Figure 29 (a). We can see that the closer  $f^t$  is to  $\hat{\mu}$ , the more shaded area under the probability density function is integrated. Therefore, maximizing  $E[I_{feas}]$  favors the sample point at which  $f^t$  is close to (or equal to)  $\hat{\mu}$  (i.e., local search). Figure 29 (b) compares two values of  $\hat{s}$ . When  $\hat{s}$  is larger, more shaded area under the probability density function is integrated with larger values of  $I_{feas}(Y)$ , which results in a larger integration value than that of a smaller  $\hat{s}$ . Thus, maximizing  $E[I_{feas}]$  also favors the sample point where  $\hat{s}$  is larger (i.e., global search).

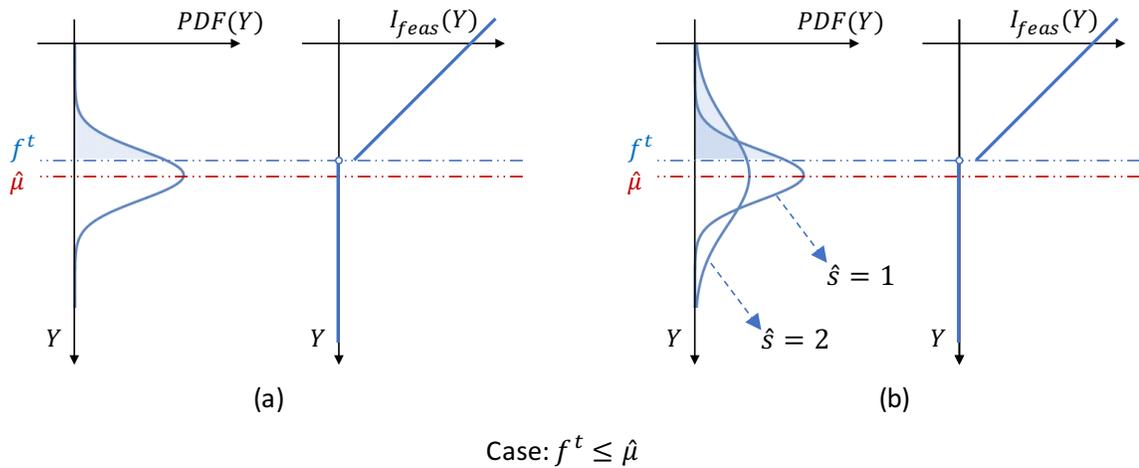


Figure 37. Illustration of the geometric meaning of  $EI_{feas}$ .

From the previous analysis, we have demonstrated that the advantage of the  $EI_{feas}$  method is to keep a balance between global search and local search for the feasible region boundaries during the adaptive sampling stage. The method can easily be adapted to SK model by replacing  $\hat{\mu}$  and  $\hat{s}$  with the SK predictor and prediction standard deviation. The algorithm of the  $EI_{feas}$  is shown in Figure 38. First, an initial SK model is built with sample points from a space-filling design. Latin Hypercube Sampling (LHS) is commonly used to obtain these points for Kriging-based surrogate modeling [283,278]. A rule of thumb is to have  $10d$  initial sample points [67]. At each point,  $m$

replications of the stochastic simulation runs are made. Then, the adaptive sampling stage begins. The next point  $\mathbf{x}'$  is identified by maximizing  $El_{feas}$ . After  $\mathbf{x}'$  is found, it needs to be checked whether this point has already been visited. If  $\mathbf{x}'$  is a new sample point, then  $m$  replicated simulation runs are made at  $\mathbf{x}'$ , and the estimated  $\bar{y}(\mathbf{x}')$  and  $\hat{V}(\mathbf{x}')$  are added to the dataset. Otherwise, with  $m$  replicated simulations,  $\bar{y}(\mathbf{x}')$  and  $\hat{V}(\mathbf{x}')$  in the original dataset are updated with all the replicated simulation runs at the previously visited  $\mathbf{x}'$ . The SK model is then updated with the new dataset. The adaptive sampling stage is conducted sequentially until the total sampling budget  $N_{max}$  is used up. After the algorithm terminates, the feasible region is predicted with the surrogate as  $\{\mathbf{x}: \hat{y}(\mathbf{x}) \leq 0\}$ .

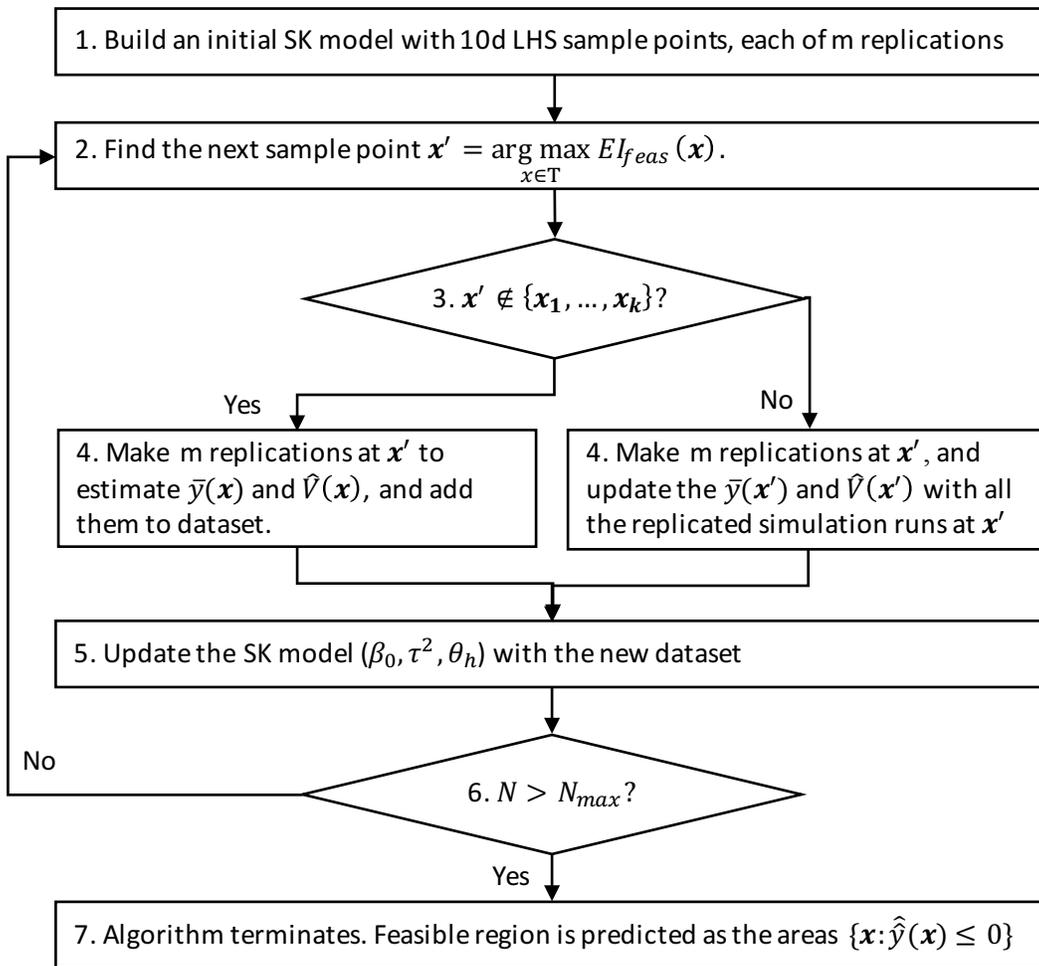


Figure 38. Algorithm of the  $El_{feas}$  method

To demonstrate how the  $EI_{feas}$  method works, we use a 2-dimensional (2D) test problem “branin” as an example (see Figure 39). The model equations can be found in Section 6.4. In Figure 39 (a), the thick-dashed line represents the true feasible region boundary, and the filled contour plot indicates the standard deviation of the noise (i.e.,  $\xi$  in Equation (110)). In this case study, the noise is set to increase linearly with  $x_2$ . Figure 39 (b) shows the predicted feasible region boundary (thick-solid line) and the  $EI_{feas}$  (filled contour) after an initial SK model is constructed (using 50 sample points, each with 50 replications). We can see that the prediction (thick-solid line) is far from accurate with the initial surrogate. However, the  $EI_{feas}$  shows several promising regions in the neighborhood of the thick-solid line. After 50 infill points (circle points in Figure 39 (c)) are added during the adaptive sampling stage, the predicted feasible region boundary is shown in Figure 39 (c) with the thick-solid line, and the  $EI_{feas}$  is depicted with the filled contour. We can see that the prediction is quite close to that of the true function in Figure 39 (a). Therefore, with this example, we can see that the  $EI_{feas}$  algorithm can be used for feasibility analysis of stochastic systems.

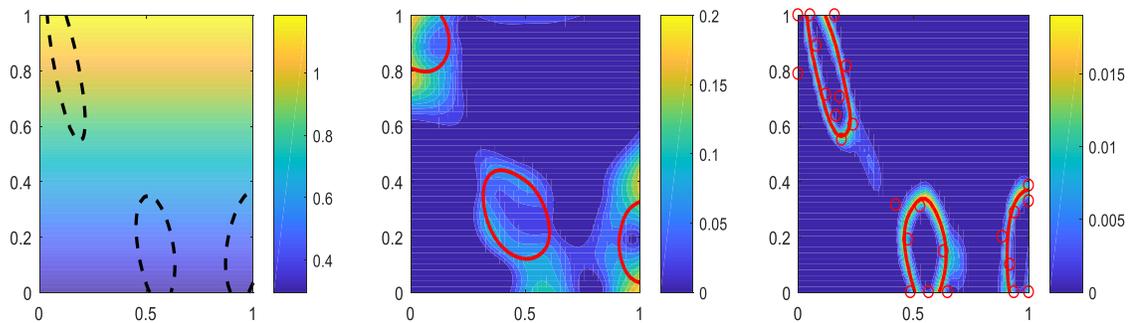


Figure 39. A case study of “branin” using the  $EI_{feas}$  approach for feasibility analysis. (a) feasible region boundary (thick-dashed line) of the true function, and heteroscedastic noise (filled contour). (b) predicted feasible region boundary (thick-solid line) with the initial SK model, and  $EI_{feas}$  (filled

contour). (c) predicted feasible region boundary (thick-solid) line with the updated SK model, and  $EI_{feas}$  (filled contour).

### 6.2.2.2 Augmented expected improvement for feasibility ( $AEI_{feas}$ )

The  $EI_{feas}$  function has been shown to be effective for deterministic models [261]. However, when applied to noisy systems, a drawback is that it does not account for the noise of the future observation. For global optimization problems of stochastic systems, Huang et, al. [161] proposed a modified EI function, which introduced a multiplicative factor to incorporate the effects of noise. Adapting this to the  $EI_{feas}$  function, we have an augmented  $EI_{feas}$  function ( $AEI_{feas}$ ) as follows:

$$AEI_{feas}(\mathbf{x}) = \hat{s} \cdot \phi\left(\frac{f^t - \hat{y}}{\hat{s}}\right) \cdot \left(1 - \frac{\xi/\sqrt{m}}{\sqrt{\hat{s}^2 + \xi^2/m}}\right), \quad (119)$$

$\hat{y}$  is the SK predictor;  $\hat{s}^2$  is the SK prediction variance;  $\xi$  is the standard deviation of the noise term. Compared to the  $EI_{feas}$  function, the  $AEI_{feas}$  function penalizes sample points where prediction variance  $\hat{s}^2$  is small relative to the noise variance  $\xi^2$ . For stochastic simulations with heteroscedastic noise, with limited replications at the design locations, it is known that the SK prediction variance  $\hat{s}^2$  can be more “inflated” where the noise variance  $\xi^2$  is larger [162]. In such cases,  $AEI_{feas}$  can prevent over-exploiting the local promising regions where noise variance  $\xi^2$  is large. Thus, during the adaptive sampling stage, the new sample points are selected more evenly in the promising regions of the input space. From this perspective, we can say the  $AEI_{feas}$  function enhances the global search compared to  $EI_{feas}$ .

To implement the  $AEI_{feas}$  method, because the expression of  $\xi^2(\mathbf{x})$  is unknown, it is estimated with  $\hat{V}(\mathbf{x})$ . A Kriging model can be built to predict  $\hat{V}(\mathbf{x})$  at an unsampled location. In order to guarantee the predicted  $\hat{V}(\mathbf{x})$  is always non-negative, the Kriging model is built for  $\log \hat{V}(\mathbf{x})$ . Such a method of estimating  $\hat{V}(\mathbf{x})$  has also been used by Chen and Zhou [262]. The algorithm of  $AEI_{feas}$

follows the same framework as  $EI_{feas}$ , which was shown in Figure 38, with the following two modifications: (1) In step 1 and 5, a separate Kriging needs to be built (or updated) for the  $\log \hat{V}(\mathbf{x})$ , its predictions are used in the multiplicative factor of  $AEI_{feas}$ ; (2) In step 2, the next sample point is determined by maximizing  $AEI_{feas}$ .

To demonstrate the difference between  $EI_{feas}$  and  $AEI_{feas}$ , we apply the  $AEI_{feas}$  to the same test problem that was used above. The adaptive sampling is conducted using the same total sampling budget (i.e., 20 initial points and 50 infill points, each with 50 replications). The results from the previous study using  $EI_{feas}$  and the results of  $AEI_{feas}$  are shown in Figure 40, where thick-solid lines represent the predicted feasible region boundary, circle points are the added infill points. In Figure 40 (a) where  $EI_{feas}$  is used, the sample points are much denser in the regions where the noise level is high (i.e., feasible region boundaries near the top), yet very scarce sample points are placed in the less noisy areas (i.e., feasible region boundaries near the bottom). In contrast, in Figure 40 (b) where  $AEI_{feas}$  is used, sample points are assigned more evenly in the promising regions throughout the input space. The detailed comparison of the performance of different algorithms is discussed in Section 6.5.

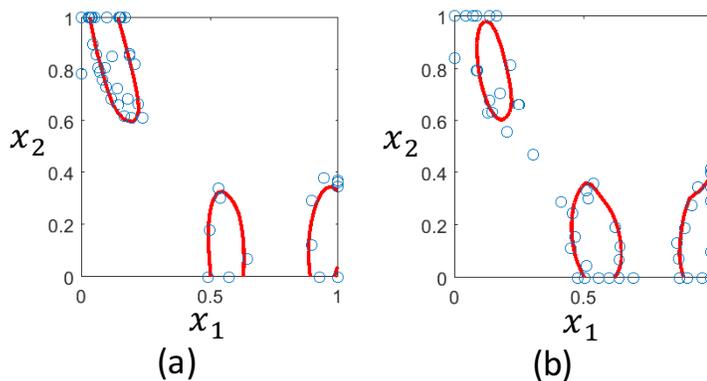


Figure 40. A case study of "branin" using  $EI_{feas}$  and  $AEI_{feas}$ . (a) Results of  $EI_{feas}$ ; (b) Results of  $AEI_{feas}$

### 6.2.2.3 Expected quantile improvement for feasibility ( $EQI_{feas}$ )

A different way to account for the effects of noise on the feasibility improvement is by seeking to the quantile of the SK model. A  $\beta$ -quantile given by the SK model for a specified level  $\beta \in (0,1)$  is expressed with the following expression:

$$q_\beta = \hat{y} + \Phi^{-1}(\beta) \cdot \hat{s}, \quad (120)$$

where  $\Phi^{-1}(\beta)$  is the inverse of the cumulative distribution function (CDF) for a normal distribution at the level of  $\beta$ . Picheny et al. [274] showed that the  $q_\beta$  is also subject to a normal distribution. For an SK model, the mean and variance of  $q_\beta$  can be expressed as follows:

$$\hat{\mu}_Q = \hat{y} + \Phi^{-1}(\beta) \cdot \sqrt{\frac{\tau^2 \hat{s}^2}{\tau^2 + \hat{s}^2}} \quad (121)$$

$$\hat{s}_Q^2 = \frac{(\hat{s}^2)^2}{\tau^2 + \hat{s}^2},$$

where  $\hat{y}$  and  $\hat{s}^2$  are SK predictor and prediction variance respectively;  $\tau^2 = \xi^2/m$ . Based on the  $\beta$ -quantile, Picheny et al. [274] proposed an expected quantile improvement method for global optimization problems of stochastic systems, which showed promising results with a proper choice of  $\beta$ . In this work, we adapt this approach to feasibility analysis by defining a quantile improvement for feasibility. This can be achieved by replacing “ $Y$ ” and “ $\hat{\mu}$ ” in (118) with “ $q_\beta$ ” and “ $\hat{\mu}_Q$ ”. Following similar steps of derivation, the expected quantile improvement ( $EQI_{feas}$ ) function can be expressed as follows:

$$EQI_{feas,\beta}(\mathbf{x}) = \hat{s}_Q \cdot \phi\left(\frac{q_\beta^t - \hat{\mu}_Q}{\hat{s}_Q}\right), \quad (122)$$

where  $q_\beta^t = f^t + \Phi^{-1}(\beta) \cdot \hat{s}$ . Following the analysis that we made on  $EI_{feas}$ , maximizing  $EQI_{feas}$  favors to pick sample points where  $\hat{s}_Q$  is large (i.e., global search), and where  $q_\beta^t$  is close to  $\hat{\mu}_Q$  (i.e., local search). After derivation, we can see that for  $EQI_{feas}$ ,  $q_\beta^t$  being close to  $\hat{\mu}_Q$  is equivalent to

say that  $\hat{\mu}$  is close to  $f^t + \hat{s} \cdot \Phi^{-1}(\beta) \cdot (1 - \tau/\sqrt{\hat{s}^2 + \tau^2})$ . Since  $(1 - \tau/\sqrt{\hat{s}^2 + \tau^2})$  is always positive and  $\hat{s}$  is non-negative, when  $\Phi^{-1}(\beta) < 0$ , maximizing  $EQI_{feas,\beta}$  favors areas where  $\hat{\mu}$  is close to and slightly less than  $f^t$ ; otherwise when  $\Phi^{-1}(\beta) > 0$ , maximizing  $EQI_{feas,\beta}$  favors areas where  $\hat{\mu}$  is close to and slightly larger than  $f^t$ . With the increasing number of infill points,  $\hat{s}$  will decrease, and the algorithm will tend more to sample where  $\hat{\mu}$  is close to  $f^t$ .

To demonstrate the effects of  $\beta$  on the search direction, we use the “branin” test problem that was used in the previous analysis as an example. The  $EQI_{feas,\beta}$  function for two different  $\beta$  (i.e., 0.1, 0.9) are shown in Figure 41. The thick-solid line is the predicted feasible region boundary from the initial surrogate; filled contour is the  $EQI_{feas,\beta}$  function. Figure 41 (a) shows the case when  $\beta = 0.1$ ,  $\Phi^{-1}(0.1) = -1.2816$ . In this case, we can see that the promising regions are mostly from inside of the predicted feasible regions where  $\hat{\mu} \leq f^t$ . On the other hand, when  $\beta = 0.9$ ,  $\Phi^{-1}(0.9) = 1.2816$  (Figure 41 (b)), the promising regions are mostly from outside of the predicted feasible regions where  $\hat{\mu} \geq f^t$ . For feasibility analysis, we would like the search direction from both sides of the feasible region boundary. Therefore, in this work, we choose  $EQI_{feas}$  as the maximum value of  $EQI_{feas,0.1}$  and  $EQI_{feas,0.9}$ :

$$EQI_{feas}(\mathbf{x}) = \max\{EQI_{feas,0.1}(\mathbf{x}), EQI_{feas,0.9}(\mathbf{x})\}. \quad (123)$$

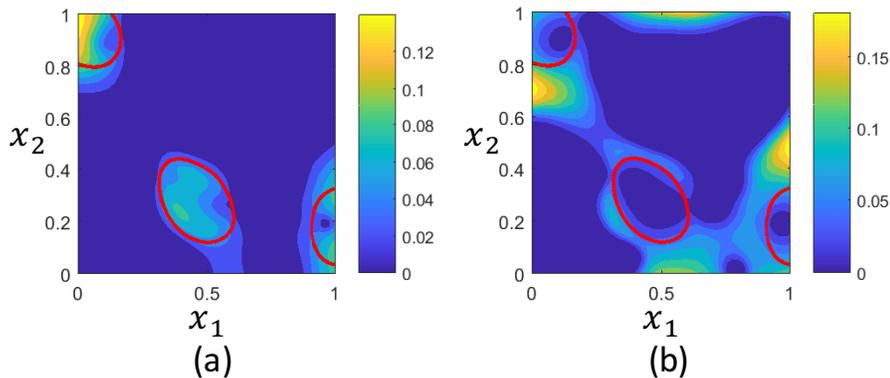


Figure 41. The difference between  $EQI_{feas,0.1}$  and  $EQI_{feas,0.9}$ . (1) Results of  $EQI_{feas,0.1}$ ; (2) Results of  $EQI_{feas,0.9}$ .

To implement the  $EQI_{feas}$  method, as we did for the  $AEI_{feas}$  method,  $\xi^2(\mathbf{x})$  is estimated with  $\hat{V}(\mathbf{x})$ , and  $\log \hat{V}(\mathbf{x})$  is modeled using Kriging. The algorithm of  $EQI_{feas}$  also follows the same framework as  $EI_{feas}$  shown in Figure 38, with the following two modifications: (1) In step 1 and 5, a separate Kriging needs to be built (or updated) for the  $\log \hat{V}(\mathbf{x})$ , its predictions are used in  $EQI_{feas,0.1}$  and  $EQI_{feas,0.9}$ ; (2) In step 2, the next sample point is determined by maximizing  $EQI_{feas}(\mathbf{x})$ .

### 6.3 Convergence

We first discuss the convergence of  $EI_{feas}$  when Kriging model is used for feasibility analysis of deterministic simulations, which is based on the following theorem:

*Theorem 1.* Based on an ordinary Kriging model and the  $EI_{feas}$  function defined in (117),

$$\lim_{n \rightarrow \infty} \sup_{\mathbf{x} \in X} EI_{feas}(\mathbf{x}) = 0.$$

*Proof.* The proof of this theorem uses a similar argument to that was given by Ranjan et al. [284], in which it was proved that the Kriging prediction variance  $\hat{s}^2(\mathbf{x}) \rightarrow 0$  as the number of sample points  $k \rightarrow \infty$ . This implies that  $\sup_{\mathbf{x} \in X} \hat{s}(\mathbf{x}) \rightarrow 0$ . From the expression of  $EI_{feas}(\mathbf{x})$  in (117), we know that  $0 \leq EI_{feas}(\mathbf{x}) \leq \hat{s}(\mathbf{x})$ . Then,  $\sup_{\mathbf{x} \in X} EI_{feas}(\mathbf{x}) \leq \sup_{\mathbf{x} \in X} \hat{s}(\mathbf{x})$ . Therefore, we can see that  $\lim_{n \rightarrow \infty} \sup_{\mathbf{x} \in X} EI_{feas}(\mathbf{x}) = 0$ . ■

Ranjan et al. [284] claimed that, with Theorem 1, it is equivalent to say that, as  $k \rightarrow \infty$ , the feasibility boundary (a contour of the feasibility function) is known perfectly, and we cannot further improve our knowledge.

Then, we proceed to the discussion on the convergence of  $EI_{feas}$ ,  $AEI_{feas}$ , and  $EQI_{feas}$  when stochastic Kriging model is used for feasibility analysis of stochastic simulations. Note that when the number of replications  $m \rightarrow \infty$ , the matrix term  $\widehat{\Sigma}_{\epsilon}$  will vanish from the stochastic Kriging predictor  $\widehat{y}(\mathbf{x})$  (115) and prediction variance  $\widehat{s}^2(\mathbf{x})$  (116). Thus, (115) and (116) will reduce to the standard Kriging predictor and prediction variance that match the data  $\bar{y}$  at design points. In addition, based on the law of large numbers, as  $m \rightarrow \infty$ ,  $\bar{y}$  converges in probability to the mean of the stochastic simulation output.

Based on these findings, as  $m \rightarrow \infty$ , the expression for  $EI_{feas}$  is reduced to the deterministic case where standard Kriging predictor and prediction variance is used;  $AEI_{feas}$  reduces to  $EI_{feas}$  because the multiplicative term in (119):  $\left(1 - \frac{\xi/\sqrt{m}}{\sqrt{\widehat{s}^2 + \xi^2/m}}\right) \rightarrow 1$  as  $\xi/\sqrt{m} \rightarrow 0$ ;  $EQI_{feas}$  also reduces to  $EI_{feas}$  because the quantile predictor  $\hat{\mu}_Q$  and prediction variance  $\widehat{s}_Q^2$  in (121) reduce to the standard Kriging predictor and prediction variance, respectively, as  $\tau^2 \rightarrow 0$ . Therefore, as  $m \rightarrow \infty$ , the convergence of  $EI_{feas}$ ,  $AEI_{feas}$ , and  $EQI_{feas}$  directly follows on from the analysis for the deterministic case.

## 6.4 Computational experiments

### 6.4.1 Test problems and noise scenarios

Four 2D test problems and one 5D test problem are used to test the performance of the three algorithms. Such problems have also been used in feasibility analysis for deterministic models [261]. The deterministic constraints for each test problem are listed in Table 16.

Table 16. Test problems

Test problems	Description
“Branin” (2D)	$\frac{1}{51.95} \left[ \left( \bar{x}_2 - \frac{5.1\bar{x}_1^2}{4\pi^2} + \frac{5\bar{x}_1}{\pi} - 6 \right)^2 + \left( 10 - \frac{10}{8\pi} \right) \cos(\bar{x}_1) - 44.81 \right] + 0.9 \leq 0$ <p>with <math>\bar{x}_1 = 15x_1 - 5, \bar{x}_2 = 15x_2</math></p> <p><math>0 \leq x_i \leq 1, \text{ for } i = 1, 2</math></p> <p><i>Scale</i> = 5.7762, <i>Rf</i> = 5.9236</p>
“Camelback” (2D)	$\left( 4 - 2.1x_1^2 + \frac{x_1^4}{3} \right) x_1^2 + x_1x_2 + (4x_2^2 - 4)x_2^2 \leq 0$ <p><math>-2 \leq x_1 \leq 2; -1 \leq x_2 \leq 1</math></p> <p><i>Scale</i> = 5.7333, <i>Rf</i> = 6.7649</p>
“Example3” (2D)	$-2x_1 + x_2 - 15 \leq 0$ $\frac{x_1^2}{2} + 4x_1 - x_2 - 5 \leq 0$ $\frac{-(x_1-4)^2}{5} - 2x_2^2 + 10 \leq 0$ <p><math>-10 \leq x_1 \leq 5; -15 \leq x_2 \leq 15</math></p> <p><i>Scale</i> = 4.25, <i>Rf</i> = 5.3119</p>
“Sasena” (2D)	$(x_1 - 3)^2 + (x_2 + 2)^2 \text{Exp}(-x_2^7) - 12 \leq 0$ $10x_1 + x_2 - 7 \leq 0$ $(x_1 - 0.5)^2 + (x_2 - 0.5)^2 - 0.2 \leq 0$

---


$$0 \leq x_i \leq 1, \text{ for } i = 1, 2$$

$$\text{Scale} = 4, Rf = 4.1907$$


---

“g4con” (5D)  $0 \leq 85.334407 + 0.0056858x_2x_5 + 0.0006262x_1x_4 -$

$$0.0022053x_3x_5 \leq 92$$

$$90 \leq 80.51249 + 0.0071317x_2x_5 + 0.0029955x_1x_2 +$$

$$0.0021813x_3^2 \leq 110$$

$$20 \leq 9.300961 + 0.0047026x_3x_5 + 0.0012547x_1x_3 +$$

$$0.0019085x_3x_4 \leq 25$$

$$78 \leq x_1 \leq 102, 33 \leq x_2 \leq 45, 27 \leq x_i \leq 45 \text{ for } i = 3, 4, 5.$$

$$\text{Scale} = 5.0430, Rf = 7.5430$$


---

In this work, a random variable  $\omega$  (i.e.,  $\omega \sim \text{Norm}(0, \xi^2(\mathbf{x}))$ ) is added to the feasibility function  $\psi(\mathbf{x})$  of each test problem. Two extreme noise scenarios are considered. In the first noise scenario,  $\xi$  increases linearly as the absolute value of  $\psi(\mathbf{x})$  increases. We call this case as the "easy" noise scenario because the noise is smallest at the feasible region boundary. In the second noise scenario,  $\xi$  decreases linearly with the absolute value of  $\psi(\mathbf{x})$  increases. This case is denoted as "hard" noise scenario since the noise is largest at the feasible region boundary. The range of the noise is linked to  $Rf$ , which is the range of  $\psi(\mathbf{x})$  within the input space:

$$Rf = \max \psi(\mathbf{x}) - \min \psi(\mathbf{x}). \quad (124)$$

In this work, the minimum value of  $\xi(\mathbf{x})$  is set as  $0.05Rf$ , and the maximum value is set as  $0.2Rf$ . Such a range is selected according to the problem settings in a simulation optimization paper [278]. However, we use a relatively smaller range than that in [278] to reduce the difficulty of the problem. This is because the task of approximating feasibility boundaries can be more sampling-costly than approximating a single (or a few) optimal point(s). The expressions for  $\xi(\mathbf{x})$  in the two noise

scenarios are shown in Table 17. The values for  $Rf$  and  $scale$  for each test problem have been listed in Table 16.

Table 17. Noise functions

Noise scenarios	Noise functions
“easy” noise	$\xi(x) = \frac{ \psi(x) }{scale} (0.2 - 0.05)Rf + 0.05Rf$
“hard” noise	$\xi(x) = \frac{ \psi(x) }{scale} (0.05 - 2)Rf + 0.2Rf$

\*  $scale = \max_{x \in X} |\psi(x)|$ .

The filled-contour of the two noise scenarios and the feasible region boundaries for each of the 2D test problems are shown in Appendix B of this paper.

#### 6.4.2 Performance measures

The goal of feasibility analysis is to identify all of the feasible regions, yet without “over-predicting” (i.e., falsely treat an infeasible point to be feasible). To evaluate the accuracy of the feasibility analysis, the following three measures can be used [261].

$$\begin{aligned}
 CF\% &= \frac{|x: \psi(x) \leq 0 \cap \hat{y}(x) \leq 0|}{|x: \psi(x) \leq 0|} \times 100 \\
 CIF\% &= \frac{|x: \psi(x) > 0 \cap \hat{y}(x) > 0|}{|x: \psi(x) > 0|} \times 100 \\
 NC\% &= \frac{|x: \psi(x) > 0 \cap \hat{y}(x) \leq 0|}{|x: \hat{y}(x) \leq 0|} \times 100
 \end{aligned} \tag{125}$$

CF% is calculated as the percentage of true feasible regions that have been correctly discovered by the surrogate. On the other hand, CIF% reflects the percentage of the true infeasible regions correctly identified by the surrogate. A good prediction accuracy requires both of these measures

to be close to 100%. Additionally, NC% shows the percentage of “over-predicting” in the predicted feasible regions by the surrogate. For an accurate prediction, NC% needs to be small and close to 0.

### 6.4.3 Implementation details

The algorithms are implemented in Matlab 2017a. The SK model is built with the codes provided by Ankenman et al. [162]. Kriging model is built with DACE toolbox [285]. The optimization problems in the adaptive sampling stage are solved with `fmincon` in Matlab using the Sequential Quadratic Programming (SQP) method. A multi-start strategy is used to increase the chance of finding the global optimum for the infill criteria. Specifically, 10d sample points (sampled with LHS strategy) are used as the initial points when  $EI_{feas}$  ( $AEI_{feas}$  or  $EQI_{feas}$ ) is maximized. For 2D test problems, the total sampling budget is 11000 simulation runs; for the 5D test problem, the total sampling budget is 12500. For each test problem, the sampling budget is allocated with two plans: one with 50 replications at each newly sampled point; the other with 100 replications. The details of the two plans are shown in Table 18.

Table 18. Summary of allocating the sampling budget

Test problems	Plans of using the sampling budget	Sampling budget
2D	Plan A: 50 replications $\times$ (20 initial points + 200 iterations of adaptive sampling)	11000
	Plan B: 100 replications $\times$ (20 initial points + 90 iterations of adaptive sampling)	
5D	Plan A: 50 replications $\times$ (50 initial points + 200 iterations of adaptive sampling)	12500

---

Plan B: 100 replications  $\times$  (50 initial points + 75 iterations of adaptive sampling)

---

To compare the performance of an algorithm for a given noise scenario and a specified sampling allocation plan for a certain test problem, we use 30 macro-replications: we make 30 runs of the algorithm, each with a different set of initial LHS points. For a given macro-replication, all the three algorithms start with the same set of initial points, and, thus, the same initial SK model. Similar designs of computational experiments have also been used in comparing different algorithms for simulation optimization problems [278,283]. The distribution of the three performance measures (i.e., CF%, CIF%, NC%) is visualized using boxplots for each algorithm. The results are shown in Section 6.5.

## 6.5 Results

### ***Plan A: 50 replications at each newly sampled point***

We first show the results on the sampling allocation plan using 50 replications for each newly sampled point. The boxplots for each test problem are shown after 200 iterations of adaptive sampling. In each figure, CF%, CIF%, and NC% are shown in (a), (b), and (c) respectively. For each of the performance measure, the "easy" noise scenario is plotted on the left, and "hard" noise scenario on the right, each with a group of three boxplots, representing the results from  $EI_{feas}$ ,  $AEI_{feas}$ , and  $EQI_{feas}$  methods.

The results of "example3" and "g4con" are shown in Figure 42 and Figure 43. We notice that these two test problems are the easiest among the five tested. For the "example3" test function, in the "easy" noise scenario, with all the three algorithms, very high values for CF% (mostly > 98%) and CIF% (mostly > 99%) are achieved, together with low values for NC% (mostly < 2%).

Even for the “hard” noise scenario, the degradation of accuracy is still small. On the other hand, for the 5D test function "g4con", the accuracy is slightly worse than that of "example3". This is expected because more sample points could be required for an accurate surrogate as the problem dimension increases. In the "easy" noise scenario, while the values for CIF% are still mostly over 98%, CF% are mostly between 95% and 97%, and NC% mostly between 3% and 5%. In the "hard" noise scenario, the decrease in CF% and the increase in NC% is approximately by 2% compared to “easy” noise scenario. In summary, for these two test problems, all the three algorithms are able to give relatively accurate predictions, and little difference is found between the three algorithms from the tested cases.

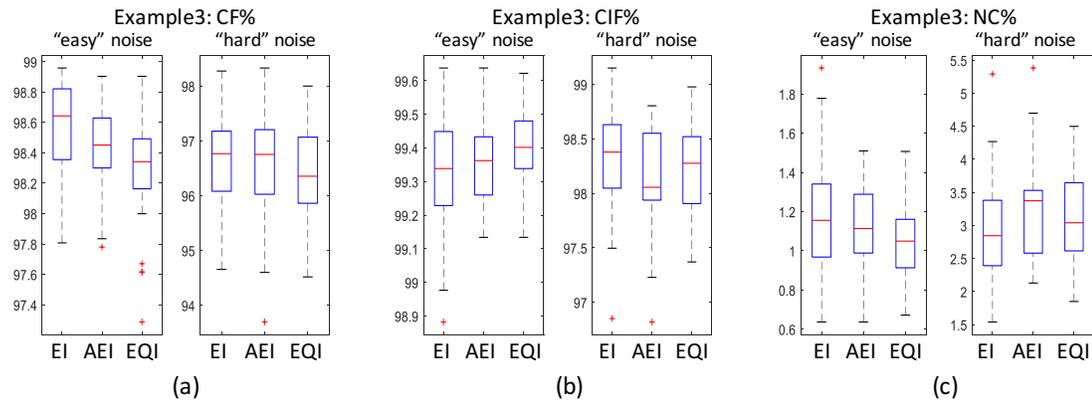


Figure 42. Results of “example3” (plan A for sampling allocation). (a) CF%; (b) CIF%; (c) NC%.

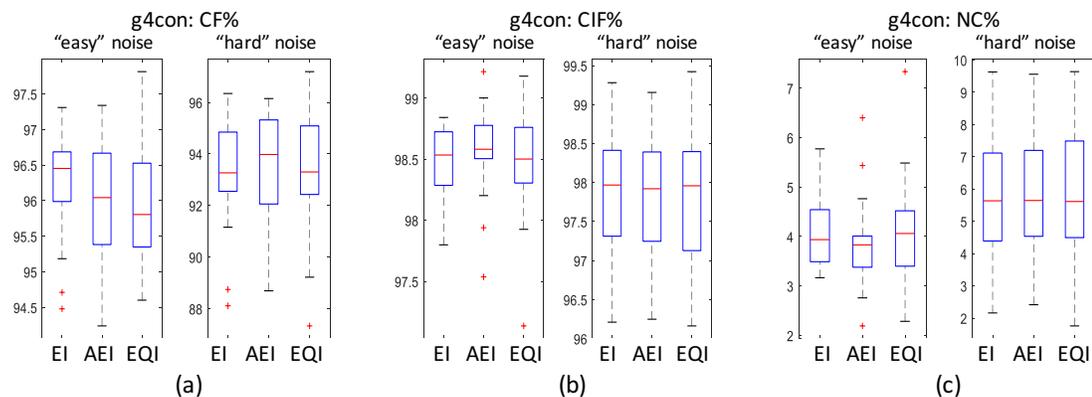


Figure 43. Results of “g4con” (plan A for sampling allocation). (a) CF%; (b) CIF%; (c) NC%

We then examine the results for “branin” and “sasena”, which are shown in Figure 44 and Figure 45. Both of these two test functions have disjoint feasible regions (see **Figure 71, Figure 74** in Appendix B). For “branin”, in the “easy” noise scenario, with all the three algorithms, we have high accuracy with large CF% (mostly between 96% and 98%) and CIF% (mostly >98%), and small NC% (mostly between 2% and 4%). However, with “hard” noise scenario, although CIF% does not drop much, CF% drops significantly, with most values between 85% and 95%, and the NC% also drastically increases, with most values between 6% and 14%. For “sasena”, we are faced with similar situations. In the “easy” noise case, with large CF% (mostly between 93% to 96%), CIF% (>99%), and small NC% (mostly between 3% and 6%), all the three algorithms are still sufficient to make accurate predictions. However, in the “hard” noise case, CF% significantly drops, with most values between 82% to 90%, and NC% increases, with most values between 8% and 16%. For these two test problems, the difference between the three algorithms is small.

The difficulty of the two test problems is mainly attributed to the landscapes of the test problems. For “branin” and “sasena”, the feasible region boundaries (or part of them) are located in a relatively “shallow” and “flat” region. To show the difference, the minimum value of the feasibility function,  $\min \psi$ , for “branin” is -0.1474;  $\min \psi$  for “sasena” is -0.1907; while  $\min \psi$  for “example3” is -1.0619. With test problems that are similar with “branin” and “sasena”, in the relatively flat neighborhood of the feasible region boundaries, the difference in the SK predictor  $\hat{y}$  is small. In such cases, the SK prediction uncertainty  $\hat{s}$  plays a more important role in the infill criteria. This causes the algorithms to be slow in making progress on “converging” to the true feasible region boundaries. Similar difficulties are also faced in solving

optimization problems. When a global optimum is located in a flat area, it is difficult to converge to the optimal solution. (See for example the “Rosenbrock” function tested in a simulation optimization framework in Ref. [283]).

In addition, the “hard” noise structure further increases the difficulty of the problem. With the “hard” noise function defined in Table 17, when feasibility boundary is located in a flat region, large noise is observed in a broad neighborhood of such a boundary (see the filled contours in **Figure 71**, **Figure 74** in Appendix). In such cases, it is more difficult to have an accurate estimate of  $\hat{V}$  and  $\bar{y}$  with a limited number of replications at sample points in promising regions. Hence, the SK model can be less accurate in approximating the neighborhood of the true feasible region boundaries, which can further mislead the infill criteria to searching towards less promising regions. Therefore, comparing to the “easy” noise scenarios, the prediction accuracy in the “hard” noise scenarios degrades drastically. Moreover, we can observe that the boxplots for the performance measures cover wider ranges for the “hard” noise case than the “easy” one, which indicates that the accuracy of each algorithm can be much different from one macro-replication to another. This is due to the increased uncertainty in building the surrogate when large noise is observed at the sample points.

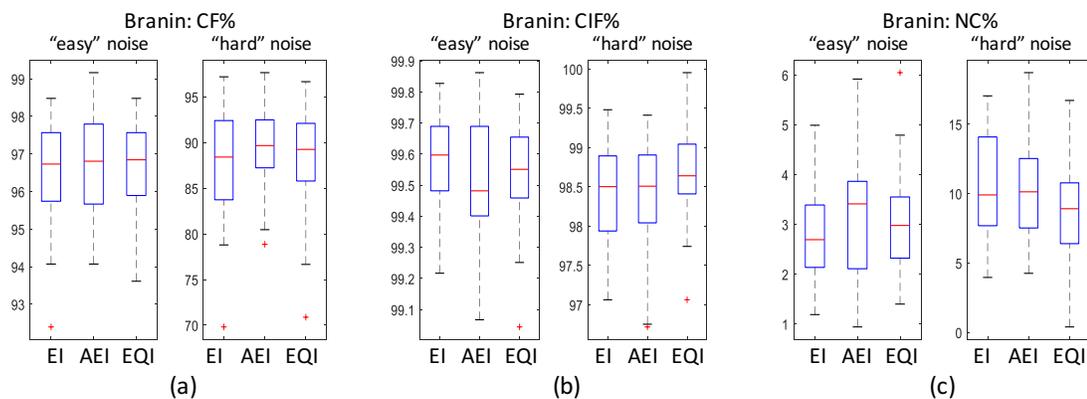


Figure 44. Results of “branin” (plan A for sampling allocation). (a) CF%; (b) CIF%; (c) NC%

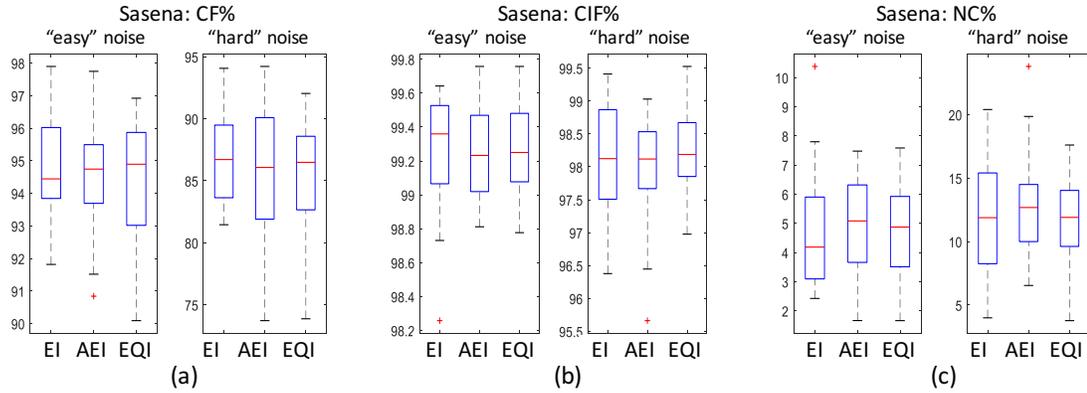


Figure 45. Results of “sasena” (plan A for sampling allocation). (a) CF%; (b) CIF%; (c) NC%

Finally, we show the results of “camelback” in Figure 46. This test function has four disjoint feasible regions: two larger ones in top-center and bottom-center, and two smaller ones in the top-left and bottom-right corners (see **Figure 72** in Appendix). For the “easy” noise scenario, with similar CIF% (over 99%) and NC% (mostly between 1% and 2.5%) for the three algorithms,  $EI_{feas}$  has CF% values most between 97% and 98.5%, with a few outliers below 95%, while  $AEI_{feas}$  and  $EQI_{feas}$  both have a slightly higher CF% value (mostly between 98% to 99%). For the “hard” noise scenario, a significant difference is found between the three algorithms. CF% for  $EI_{feas}$  drops drastically, with values mostly between 89% and 93%. However,  $AEI_{feas}$  and  $EQI_{feas}$  has a much smaller drop, with most values between 93% to 96%. This indicates that, for  $EI_{feas}$ , the capability of exploring all the feasible regions in the “hard” noise scenario is much worse than that for the  $AEI_{feas}$  and  $EQI_{feas}$  methods. On the other hand,  $EI_{feas}$  appears to have a better NC% (mostly between 2.5% to 4%) compared to that of  $AEI_{feas}$  and  $EQI_{feas}$  (NC% mostly between 4% to 6.5% for both methods). This indicates that  $AEI_{feas}$  and  $EQI_{feas}$  have a sacrifice in the conservativeness in the prediction, compared to the  $EI_{feas}$  method.

After checking all the figures of the final surrogate from the 30 macro-replications of each algorithm (not included in this paper, due to limited space), we found that the  $EI_{feas}$  method is more likely to miss the two smaller feasible regions than  $AEI_{feas}$  and  $EQI_{feas}$  methods. For each algorithm, we list in Table 19 the fractions of the overall 30 runs that can discover all of the four feasible regions. From this table, we can find that  $AEI_{feas}$  and  $EQI_{feas}$  are significantly better than  $EI_{feas}$  in discovering all the feasible regions. This is because both of them have a slightly enhanced global search, which causes them to be less likely to get trapped in a local neighborhood of promising areas. However, this also increases the risks of being less conservative in the identified feasible regions for  $AEI_{feas}$  and  $EQI_{feas}$ : sample points are scattered near all the feasible region boundaries rather than only being focused on part of the boundaries.

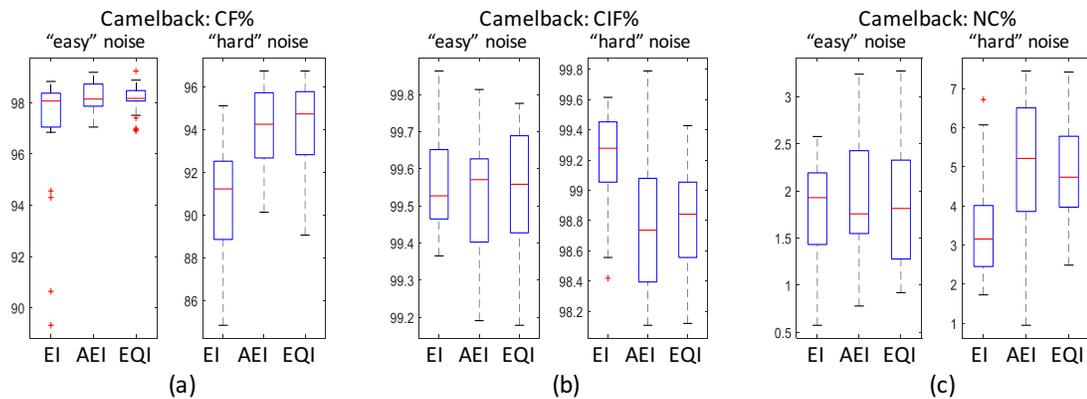


Figure 46. Results of “camelback” (plan A for sampling allocation). (a) CF%; (b) CIF%; (c) NC%

Table 19. Number fraction of 30 runs identifying all feasible regions for “camelback”

	“easy”	“hard”
	noise	noise
$EI_{feas}$	26/30	11/30

$AEI_{feas}$	30/30	29/30
$EQI_{feas}$	30/30	27/30

**Plan B: 100 replications at each newly sampled point**

In order to investigate the effects of sampling allocation strategies on the algorithms' performance, we further conducted computational studies using 100 replications at each newly sampled point (i.e., plan B in Table 18), and accordingly with fewer iterations of adaptive sampling so that the total sampling budget remains the same. For comparison purposes, the boxplots for both sampling strategies are plotted in parallel in each figure, with column "A" representing the strategy with 50 replications; column "B" for the strategy with 100 replications.

We first show the results for "example3" (Figure 47) and "g4con" (Figure 48). In the "easy" noise scenarios, for all three algorithms, plan "A" gives a consistent and noteworthy better accuracy: higher values in CF% and CIF% and smaller values in NC% are achieved by using plan "A". In the "hard" noise scenarios, the difference between plan "A" and "B" is less significant compared to the "easy" noise cases.

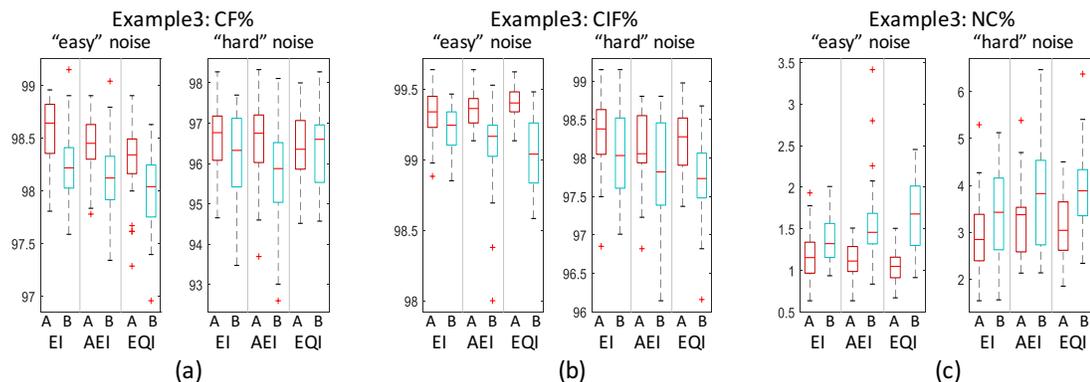


Figure 47. Results of "example3". (a) CF%; (b) CIF%; (c) NC%

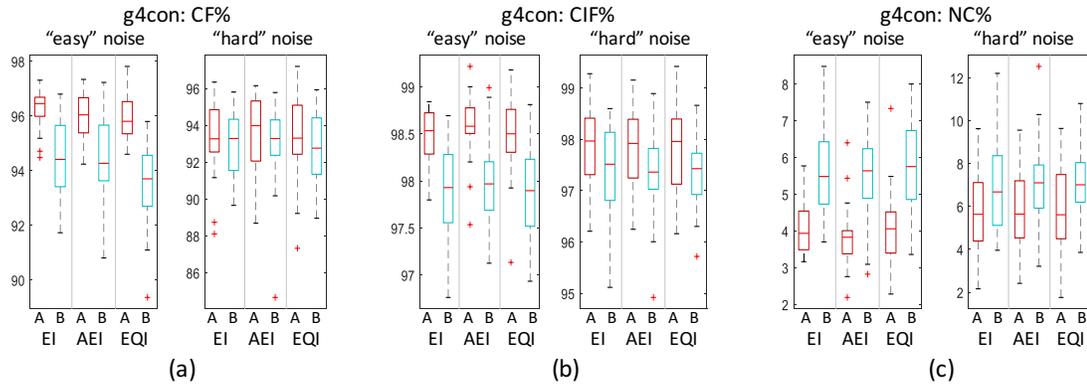


Figure 48. Results of “g4con” . (a) CF%; (b) CIF%; (c) NC%

The results for “branin” and “sasena” are shown in Figure 49 and Figure 50. For “branin” , no significant difference is found between plan “A” and “B” in the tested noise scenarios for the three algorithms. However, the “sasena” function is better solved with plan “A” in the easy noise case: consistently higher accuracy is achieved by all the three algorithms using plan “A” than plan "B", which is similar to what we found for the previous "example3" and "g4con" tested cases.

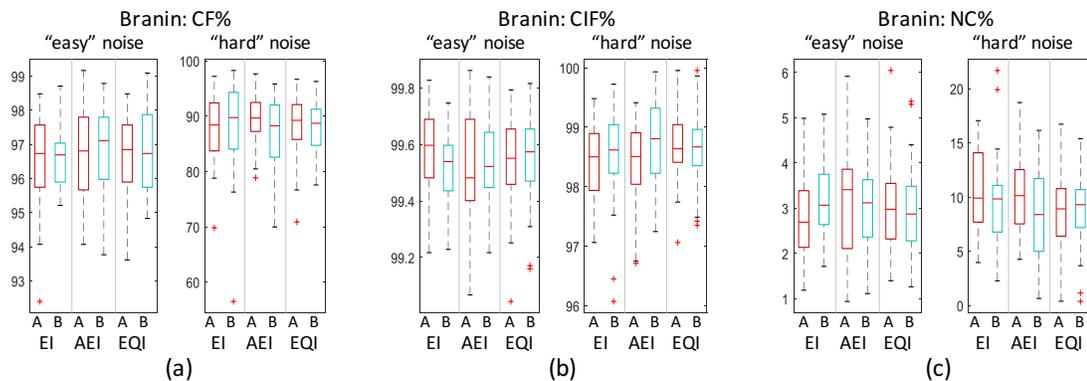


Figure 49. Results of “branin” . (a) CF%; (b) CIF%; (c) NC%

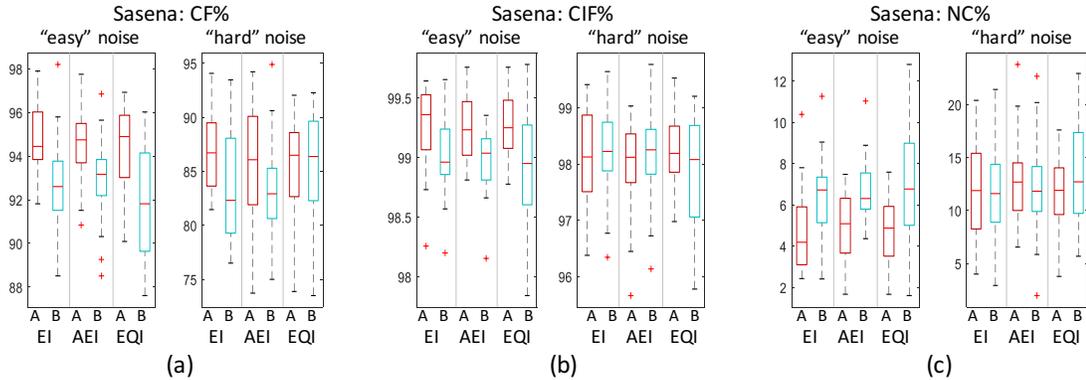


Figure 50. Results of “sasena” . (a) CF%; (b) CIF%; (c) NC%

The results for “camelback” is shown in Figure 51. In the “easy” noise, plan “A” only appears to give higher values in CF% that plan “B” , while the difference in CIF% and NC% is almost indistinguishable. Also, no obvious difference is observed in the “hard” noise scenario.

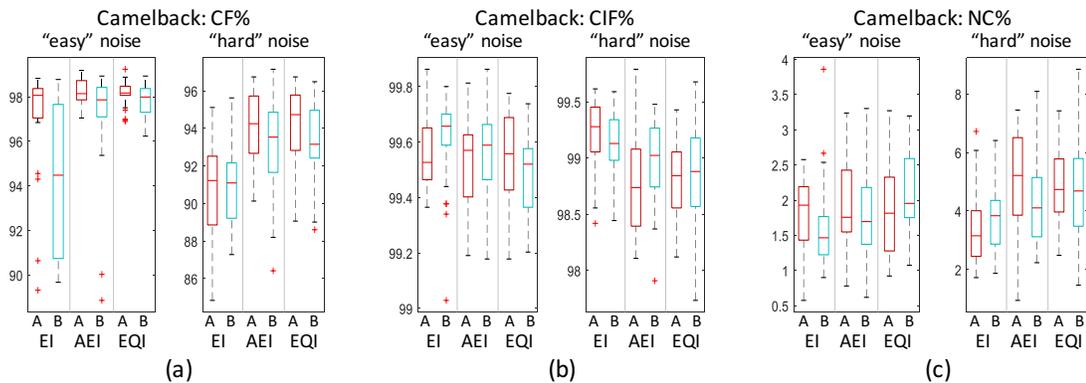


Figure 51. Results of “camelback” . (a) CF%; (b) CIF%; (c) NC%

To summarize the results of the comparison of two sampling allocation strategies: consistent and better accuracy is achieved by plan “A” for the functions “example3” , “g4con” , and “sasena” , especially in the “easy” noise scenarios. Such findings indicate that for "easy" noise scenarios when a small number of replications can already give good estimates of  $\hat{V}$  and  $\bar{y}$ , a relatively large number of sample locations are crucial to obtain a higher accuracy in finding the

feasible region boundaries. However, for the “hard” noise scenarios, without an accurate enough estimate of  $\hat{V}$  and  $\bar{y}$ , the advantage of having more sample points will vanish. In other words, when the noise is larger in the promising regions (i.e., the neighborhood of true feasible region boundaries), it is of top priority to first have a relatively accurate estimate of  $\hat{V}$  and  $\bar{y}$ , (e.g., using a larger number of replications at each sample locations). Otherwise, simply by increasing the number of sample points will not be efficient or effective in improving the overall prediction accuracy of the feasible regions.

### 6.6 Application to a pharmaceutical manufacturing process model

The three algorithms have been applied to the feasibility analysis for the operation of a roller compaction (RC) process in the pharmaceutical manufacturing process. RC is a dry granulation process in which solid-based raw materials are fed continuously and compacted under high pressure exerted by two rotating rolls, in order to produce compacted ribbon products [286]. Hsu et al. [234] developed a mathematical model to describe this unit operation. This model accounts for the operation conditions (i.e., hydraulic pressure  $P_h$ , rotating roll speed  $\omega$ , feed speed  $u_{in}$ ) and powder material physical properties (i.e., inlet angle  $\theta_{in}$ , powder bulk density  $\rho_{in}$ ) as model inputs, and can be used to predict ribbon properties (i.e., ribbon density  $\rho_{exit}$ , ribbon thickness  $h_0$ ). The major model equations are shown in (126).

$$\begin{aligned} \frac{d}{dt} \left( \frac{h_0}{R} \right) &= \frac{\omega \left[ \rho_{in} \cos \theta_{in} (1 + h_0/R - \cos \theta_{in}) (u_{in}/\omega R) - \rho_{exit} (h_0/R) \right]}{\int_0^{\theta_{in}} \rho(\theta) \cos(\theta) d\theta} \\ P_h &= \frac{W}{A} \frac{\sigma_{exit} R}{1 + \sin \delta} \int_0^\alpha \left[ \frac{h_0/R}{(1 + h_0/R - \cos \theta) \cos \theta} \right]^K \cos \theta d\theta \\ \sigma_{exit} &= C_1 \rho_{exit}^K \end{aligned} \tag{126}$$

To account for the variations in the actual ribbon product qualities, we add a normally distributed noise term:  $\omega_{feas} \sim \text{Norm}(0, \xi_{feas}^2)$  on the feasibility of ribbon properties involving  $\rho_{exit}$  and  $h_0$ .

It has been experimentally verified that the hydraulic pressure,  $P_h$ , has a strong influence on the ribbon strength, and can significantly affect the variance of its observed values [286]. Since, for a general compaction process, the strength of a compacted product is strongly correlated to its density and thickness [42], in this work, we make the assumption that  $\xi_{feas}$  is a function of  $P_h$ . According to the experimental data in [286], we make a rough estimate that  $\xi_{feas}$  grows linearly as  $P_h$  increases, ranging from 1% to 4% of the true feasibility function value. Hence, we have a stochastic simulation for the operation of roller compaction to represent the realistic production process.

The feasibility analysis is then formulated as follows: identify the feasible region in the space of  $\{P_h, u_{in}\}$  in which we can obtain qualified ribbon products having  $\rho_{exit}$  and  $h_0$  within the following box constraints:

$$\begin{aligned} \rho_{exit}^L &\leq \rho_{exit} \leq \rho_{exit}^U \\ h_0^L &\leq h_0 \leq h_0^U \end{aligned} \quad (127)$$

The values and ranges for the model parameters and variables are listed in Table 20.

Table 20. Parameters and bounds for the roller compaction model

Parameter	Symbol	Value/Range	Units
<b>Equipment parameters and model coefficients</b>			
Roll radius	$R$	0.125	m
Roll width	$W$	0.05	m
Compression parameter	$K$	4.97	
Compression parameter	$C_1$	$7.5 \times 10^{-8}$	$\text{Pa}/(\text{kg}/\text{m}^3)^{4.97}$
Compact surface area	$A$	0.01	$\text{m}^2$
Effective angle of friction	$\delta$	0.7069	rad

Nip angle	$\alpha$	0.173	rad
Angular position	$\theta$	<i>NA</i>	rad
Inlet powder density	$\rho_{in}$	300	kg/m <sup>3</sup>
Inlet angle	$\theta_{in}$	0.4	rad
<b>Operating conditions</b>			
Hydraulic pressure set point (roll pressure)	$P_h$	0.8 - 1	MPa
Rotating roll speed set point	$\omega$	5	rpm
Powder feed speed	$u_{in}$	$2 \times 10^{-4}$ - $4 \times 10^{-4}$	m/s
<b>Product constraints</b>			
Ribbon thickness	$h_0$	$1.7 \times 10^{-3}$ - $1.9 \times 10^{-3}$	m
Ribbon density	$\rho_{exit}$	850 - 950	kg/m <sup>3</sup>

The three algorithms are implemented by using 100 replications at each newly sampled point. 20 initial LHS points are used to build the initial SK model, with additional 100 iterations of adaptive sampling to improve the surrogate accuracy. The performance measures are listed in Table 21. From this table, we can find that all the three proposed algorithms improve the CF% from 71.06% to over 90%, while maintaining the prediction conservativeness.

Table 21. Performance measures for the RC test problem

Initial Accuracy	Final Accuracy		
	$EI_{feas}$	$AEI_{feas}$	$EQI_{feas}$

CF%	71.06	90.50	90.80	90.35
CIF%	100	100	100	99.98
NC%	0	0	0	0.16

The feasible region is plotted in Figure 52. Since the predicted feasible region is almost identical from the three algorithms, only the results from  $EI_{feas}$  is shown. The true feasible region boundary is denoted with thick-dashed lines, and the predicted feasible region with thick-solid lines. The feasible region is the area in between the two lines. We can see from the figure that the prediction is quite close to the original function. Such feasibility analysis results can be used to guide the process operations. As can be noted, the process is feasible almost over the whole range of  $P_h$ , while only within a small range of  $u_{in}$ . This means that, under the current process settings, the material feed speed must be carefully controlled within the characterized ranges in order to guarantee the desired product properties.

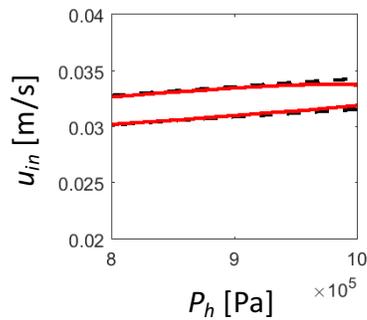


Figure 52. Feasible region for the RC process

## 6.7 Summary and future work

We have presented a surrogate-based adaptive sampling framework for feasibility analysis of stochastic systems. While classical feasibility analysis was initially formulated to account for the uncertain in the model parameters and inputs, by using the framework in this work, it is the first time that we can extend feasibility analysis to the cases in which the stochasticity of model outputs

can be considered. In this framework, an SK model is used to approximate the black-box stochastic feasibility function, which uses replicated function calls to estimate the mean and variance at each sample point. This surrogate model can be used to model stochastic processes with a heteroscedastic noise. Three adaptive sampling approaches, namely  $EI_{feas}$ ,  $AEI_{feas}$ , and  $EQI_{feas}$ , have been used to search for the next infill point. These three algorithms all have the advantage of keeping a balance between global search and local search for the feasible region boundaries. With five benchmark functions, we have found that all the three algorithms can be largely affected by the noise structure. With a larger noise existent in the neighborhood of feasible region boundaries, it is more difficult to have a high prediction accuracy, compared to the cases with a smaller noise. As such, a larger number of replications at each sample location is required in the first place, because only with a highly-accurate surrogate can the adaptive sampling method direct the search towards the actual promising regions. Otherwise, the adaptive sampling stage will only make little progress even with the increased number of sample points. The comparison between the three algorithms indicates that  $AEI_{feas}$  and  $EQI_{feas}$  can be more favored especially when there exist smaller feasible regions that are hard to be explored. This is because they both have a slightly more enhanced global search compared to the  $EI_{feas}$  method, and thus, they are less likely to be trapped in local regions.

For future work, it requires further studies on the average convergence rate for the proposed algorithms. Additional studies are also needed to investigate whether different correlation functions can affect the performance of the proposed feasibility analysis algorithm. In terms of implementation, it still remains a challenge of properly choosing the number of replications when building the SK model. Intelligent methods need to be investigated to determine the number of replications based on the variance (closed-form function assumed to be unknown) at each sample point. The Optimal Computing Budget Allocation [287] approach from simulation optimization

areas seems to be a promising choice. Finally, it should be noted that the calculation of  $EI_{feas}$  (and also  $AEI_{feas}$ ,  $EQI_{feas}$ ) involves inverting a correlation matrix, which can be computationally expensive as the number of sample points increases. This limits the use of the proposed algorithms to be only appropriate for lower-dimensional problems with a relative small noise level. Rullière [288] proposed a global optimization algorithm for stochastic simulations, which applied the branch and bound framework with the partition of the search domain. This approach is promising in reducing the computational burden, and it would be interesting if we could combine similar techniques to our proposed algorithms and make them applicable to a broader range of problems.

## 7 A novel surrogate-based optimization method for black-box stochastic simulations with heteroscedastic noises

### **Abstract**

Simulation optimization (SO) problems can be difficult to solve due to the lack of knowledge of the algebraic model equations and the unknown structure of the noise inherent to the simulation. It is important to investigate approaches capable of handling noise in order to achieve optimal solution with efficiency. In recent years, surrogate-based methods for SO problems have gained increasing attention from different research communities. In this work, we adapted a one-stage adaptive sampling approach to a Kriging-based optimization framework for simulations with heteroscedastic noise. We compared its performance with another Kriging-based approach using expected improvement as the infill criterion. Based on the results of several test problems, each with various noise scenarios, we discussed the benefits and limitations of both algorithms. Finally, we show the application of both algorithms to finding the optimal operation conditions of a continuous pharmaceutical manufacturing simulation model.

## 7.1 Introduction

Simulation models have always been a powerful tool to help investigate and predict phenomena and system behaviors for a variety of research areas including engineering, science, finance, etc. Such simulation models have also been used for optimization purposes. For deterministic optimization problems, there are numerous well-established methods to find the optimal solution of the simulation (e.g., LP, NLP, derivative-free optimization methods), which may or may not require explicit model equations [289,180]. For stochastic optimization problems whose model equations are available, a number of approaches have been developed following stochastic programming mathematical techniques [290]. However, for black-box stochastic simulations with unknown noise structure, challenges remain. Such stochastic optimization problems are categorized as Simulation Optimization (SO) problems in a recent review paper by Amaran, Sahinidis et al. [291]

In this article, we only consider single-objective SO problems with continuous inputs and box constraints, of which the general form can be expressed with Problem 128

$$\begin{aligned}
 & \min_x f(x) \\
 & \text{s. t. } x_l \leq x \leq x_u \\
 & x \in \mathbb{R}^d
 \end{aligned} \tag{128}$$

Under the paradigm of SO, the objective function  $f(x)$  cannot be directly observed, and can only be estimated with a stochastic simulation. The user only has access to noisy observations of the model output  $\tilde{f}_i(x)$ :

$$\tilde{f}_i(x) = f(x) + \omega_i(x) \tag{129}$$

where  $\tilde{f}_i(x)$  is the observed objective value at the  $i^{th}$  replication of simulation evaluated at a specific input  $x$ , and  $\omega_i(x)$  is the observed noise term. In this article, we also make the same

assumptions for the noise term as were used by Picheny, Ginsbourger et al. [274] : the observed noises are normally distributed, centered, and independent from one run to another (Equation 130).

$$\omega_i(x) \sim N(0, \xi^2(x)) \quad (130)$$

We consider the case where the noise variance depends on  $x$  (i.e.,  $\xi(x)^2$ ), and thus we have heteroscedastic noise inherent to the simulation.

The difficulties associated with solving such SO problems are as follows. First, due to the lack of available algebraic equations, one cannot directly apply traditional mathematical programming algorithms (e.g., LP, NLP) or stochastic programming algorithms. Second, the noise in the simulation outputs also makes it harder to estimate the derivative information. Derivative-free optimization (DFO) methods are mostly designed for deterministic models, and may not all be suitable for the black-box stochastic simulations. Third, the stochastic behavior also adds to the challenge of proving convergence [291]. Finally, compared to other optimization problems, there still lack sufficient test problems and an accepted standard to compare the performance of different SO algorithms [291].

A variety of algorithms have been developed to solve SO problems. Deng and Ferris [292] adapted the DIRECT (DIviding RECTangles) algorithm to stochastic simulations, which is originated from Lipschitzian optimization methods. Chang [293] proposed a Stochastic Nelder-Mead simplex method, which used an effective sampling scheme to control the noise and a global-local search framework to ensure solution quality. Xu, Nelson et al. [294] developed an Industrial Strength COMPASS (ISE) algorithm to solve simulation optimization problems with integer-ordered decision variables and linear-integer constraints. In addition, Optimal Computing Budget Allocation (OCBA) [295,296] have been integrated with search algorithms to intelligently allocate simulation budget during the optimization process. Stochastic approximation methods, such as FDSA (Finite-Difference Stochastic Approximation) and SPSA (Simultaneous Perturbation

Stochastic Approximation) [297,298] have also been used for optimization problems with noisy observations of objective function.

The algorithms mentioned above are attractive due to their effectiveness, but a common drawback is that they usually require huge sampling costs, and may not be suitable for computationally expensive simulations. In contrast, the surrogate-based optimization methods can be more appealing. Such methods approximate the simulation with a surrogate model that is used to direct the search. Considering that the detailed surrogate-based approaches may vary depending on the noise structure of the simulations, in the following paragraphs, we first give a literature review on surrogate-based optimization approaches for simulations with homoscedastic noise, which is then followed by recent work on optimization for simulations with heteroscedastic noise.

In terms of the optimization algorithms for homoscedastic-noise simulations, Jakobsson, Patriksson et al. [299] proposed the “qualSolve” algorithm, which constructed a RBF-based approximation surrogate and chose the next point by evaluating the overall decrease in weighted uncertainty. Huang, Allen et al. [161] developed a Sequential Kriging Optimization (SKO) method, which accounted for the noise by adding one more model parameter ( “nugget” factor) to Kriging model, and used a Augmented Expected Improvement (AEI) function to search for new sample points. This is probably the first attempt to extend Jones’ efficient global optimization (EGO) algorithm [67] to stochastic simulations. Vazquez, Villemonteix et al. [300] adapted the Informational Approach to Global Optimization (IAGO) method, which utilized Kriging as the surrogate and evaluated the expected decrease of the entropy to find the next sample point. J. Forrester, Keane et al. [301] demonstrated a re-interpolation approach to refine the Kriging surrogate and used EI as the infill criterion. Picheny, Wagner et al. [283] made a thorough comparison of multiple Kriging-based optimization approaches using different infill criteria for homoscedastic-noise simulations.

When applied to stochastic simulations with heteroscedastic noise, the aforementioned surrogate-based optimization algorithms can be inappropriate due to two aspects of reasons. On the one hand, the surrogate models utilized in these algorithms are usually constructed based on the assumption that the noise variance is constant. This assumption is no longer valid for the heteroscedastic-noise simulations, and thus the surrogate models have the risks of being unable to capture the behavior of such stochastic simulations [302,303]. Hence, the estimated optimum can deviate far from the true optimum due to an inadequate fit of the surrogate [303]. On the other hand, some of the well-performed infill criteria from these algorithms cannot properly handle the noise with non-constant variances. For example, Quan et al. [303] demonstrated this with a test problem and showed that the AEI function from SKO algorithm [161] could be trapped in the local area with low variability, and miss the global minimum with high variability. Therefore, it is necessary to develop different surrogate-based optimization algorithms for stochastic simulations with heteroscedastic noise.

The heteroscedastic noise may affect the surrogate modeling techniques, which leads to some changes in the surrogate-based optimization algorithms compared to those for homoscedastic-noise simulations. Therefore, it is beneficial to first take a look on the different surrogate models designed for simulations with heteroscedastic noise. Ji and Kim [304] proposed a regularized radial basis function model (R-RBF), and derived both the model predictor and prediction error (estimated mean squared error). For the Kriging-based surrogate, there are mainly two branches of strategies to consider noise, which are developed by different research communities. One leading approach is based on Stochastic Kriging (SK), initially proposed by Ankenman, Nelson et al. [162]. It considers the noise component as intrinsic uncertainty of the simulation, and makes replications at each sample location to estimate the mean and variance, which is used to construct the surrogate. Yin, Ng et al. [302] formed a Kriging model with modified nugget-effect, which has equivalent mathematical expressions to SK. Chen and Zhou [305] established a sequential design framework

to improve the surrogate accuracy of SK. Kleijnen and Mehdad [306] formulated a more accurate estimator of the predictor variance using bootstrapping. The other branch of kriging-based approaches follows the Heteroscedastic Gaussian Process Regression (HGPR), which is mostly investigated by the machine learning community [307]. Kersting, Plagemann et al. [308] presented a most likely HGPR (MLHGPR), which used two GPRs to model the mean and log-noise distributions, respectively. Unlike SK, the MLHGPR method does not require replications to build the surrogate. Boukouvalas and Cornford [309] later extended MLHGPR framework to the cases where replicated observations are available, and showed that using replications can increase the accuracy of MLHGPR. Schneider and Ertel [310] introduced a local GPR that clusters the input space into multiple sub-regions and trains a local HGPR for each set of data points. Muñoz-González, Lázaro-Gredilla et al. [311] presented an Expectation Propagation (EP) GPR model, which used EP to approximate the posterior distribution for the log-noise. Lázaro-gredilla and Titsias [312] proposed a Variational Heteroscedastic Gaussian Process Regression (VHGPR), which used a variational approximation for the marginal log-likelihood of the HGPR. This VHGPR method is practically faster and more accurate compared to MLHGPR in all the datasets tested in Ref. [312]. It should be noted that these papers cited above usually make the general assumption that the noise term is subject to a normal distribution with zero mean, independent and identically distributed from one run to another, which is the same assumption that we make in this work. There is only one exception, that is, Yin, Ng et al. [302] did not specify a distribution form for the noise term of stochastic simulations in their studies.

Based on recent advances of surrogate models for heteroscedastic-noise simulations, the surrogate-based optimization techniques are developed accordingly. Picheny, Ginsbourger et al. [274] utilized a Kriging-based surrogate and expected quantile improvement (EQI) for the infill criterion. However, this method requires the noise variance function (i.e.,  $\xi^2(x)$ ) to be known. Quan, Yin et al. [303] proposed a two-stage sequential framework using SK with EI, and incorporated a

computing budget allocation technique to determine the replications among sampled locations. This algorithm does not require known noise function. Mehdad and Kleijnen [313] applied stochastic intrinsic Kriging (SIK) with EI function to find the optimal solution, and introduced a new allocation rule. Jalali, Van Nieuwenhuysse et al. [314] compared the performance of six algorithms based on a SK model using different infill criteria for SO problems with heteroscedastic noise. Kuindersma, Grupen et al. [315] applied VHGPR with expected risk improvement (ERI) to the optimization framework.

Based on the literature review, we can notice that the majority of work that uses Kriging-based approaches adopts expected improvement (and its extensions) as the infill criterion. This category of algorithm can be seen as a two-stage approach: in the first stage, the surrogate is built with the sample points; in the second stage, EI is evaluated based on the constructed surrogate. Although the two-stage approach works well in most cases for deterministic simulations, it should be noted that the performance of such algorithm can be misled by the initial samples and deceptively positioned optimum [137], which is more likely to happen with sparse sample points. In the optimization for stochastic simulations, we would expect to encounter more of such cases. This is because the noise further increases the complexity and uncertainty in building the surrogate, especially during the starting stage of the algorithm when the sample points are less sufficient. To address this issue, we revisited the one-stage (OS) approach [123], which “uses the minimum to find the surrogate”, rather than “use the surrogate to find the minimum” [137]. We are interested to see whether the OS algorithm can be more effective and robust compared to EI-based methods. Therefore, in this paper, we apply SK with OS approach to the optimization of simulation-based models with heteroscedastic noise, and compare its performance with the two-stage (EI-based) algorithm. Both of these two algorithms (i.e., SK-OS and SK-EI) in this work do not require the variance function (i.e.,  $\xi^2(x)$ ) of the noise term to be known. The rest of this article is organized

as follows. Section 7.2 introduces the stochastic Kriging surrogate model. Section 7.3 describes the two infill criteria, EI-based approach and OS-based approach, whereas the computational results are discussed in Section 7.4. Section 7.6 is used for the application of the investigated approaches to a pharmaceutical manufacturing case study.

## 7.2 Stochastic Kriging Model

In this section, we briefly introduce the formulas for SK model that we use as the surrogate in the optimization framework. SK model represents a stochastic simulation's output on replication  $j$  at design point  $x$  with the following model expression (Equation 131)

$$y_j(x) = \beta_0 + M(x) + \varepsilon_j(x) \quad (131)$$

where  $\beta_0$  is a constant term representing the overall surface mean;  $M$  is a realization of a random field with mean 0; that is,  $M$  is considered to be randomly sampled from a space of functions mapping  $R^d \rightarrow R$ . The functions in this space are assumed to exhibit spatial correlations, which indicates that value of  $M(x)$  and  $M(x')$  will tend to be similar if the distance between  $x$  and  $x'$  is small. In Ref. [162],  $M$  is referred to as “extrinsic uncertainty” since it is introduced to aid the development of the surrogate model. The first two terms ( $\beta_0$  and  $M$ ) in Equation 131 share the same characteristics as those in ordinary Kriging [275] for deterministic simulations.  $\varepsilon$  is another random field (over the space of mapping  $R^d \rightarrow R$ ) with mean 0. It is called “intrinsic uncertainty” in Ref. [162] and it represents the sampling variability inherent to a stochastic simulation. The variance of  $\varepsilon$  would depend on  $x$  if the simulations have heteroscedastic noise in the output.

The stochastic Kriging is based on the following assumption [162]: the random field  $M$  is a stationary Gaussian random field, and  $\varepsilon_1(x_i)$ ,  $\varepsilon_2(x_i)$ ,  $\dots$  are i.i.d.  $N(0, V(x_i))$ , independent of  $\varepsilon_j(x_h)$  for all  $j$  and  $h \neq i$ , and independent of  $M$ . According to Ref. [162], this assumption implies that for any set of design points  $x_1, x_2, \dots, x_k$  the random vector  $[M(x_1), M(x_2), \dots, M(x_k)]^T$  has a

multivariate normal distribution with marginal mean 0, positive variance  $\tau^2$ , and positive definite correlation matrix  $\mathbf{R}_M$ . These properties are important for modeling the correlation of  $M$ , estimating intrinsic variance  $V(x_i)$ , and deriving the model prediction, which are discussed as follows.

Several correlation kernel functions can be used to model the spatial correlations of  $M$ . In this article, we follow the approach of Ref. [162] and use the Gaussian correlation kernel (Equation 132):

$$\text{Corr}(M(x), M(x')) = \exp\left(\sum_{h=1}^d -\theta_h |x^h - x'^h|^2\right) \quad (132)$$

where  $d$  is the number of dimensions in the input space;  $\theta_h > 0$  is the model parameter that controls how fast the correlation decays with distance in the  $h^{\text{th}}$  dimension. The covariance between  $x$  and  $x'$  can thus be expressed as:  $\text{Cov}(x, x') = \tau^2 \text{Corr}(x, x')$ , where  $\tau^2$  is the model parameter representing the variance of  $M$ .

In Ref. [162], the intrinsic variance at the design point  $x_i$  is estimated with  $n_i$  replications using (Equation 133):

$$\hat{V}(x_i) = \frac{1}{n_i - 1} \sum_{j=1}^{n_i} (y_j(x_i) - \bar{y}(x_i))^2, \quad (133)$$

where  $\bar{y}(x_i)$  is the estimated mean at  $x_i$ ,  $\bar{y}(x_i) = \frac{1}{n_i} \sum_{j=1}^{n_i} y_j(x_i)$ . It is shown in Ref. [162] that the surrogate accuracy will not be sacrificed by estimating the intrinsic variance by  $\hat{V}$ , as long as  $n_i$  is not too small. In this work, we choose  $n_i = 10$  as is suggested in Ref. [162]. This number of replications was also used as a start point of sampling budget in Ref. [313].

Using these equations, the model parameters can be obtained by maximizing the log-likelihood (derived in the supplement material of Ref. [162]). Based on the assumption on the independence

properties of  $M$  and  $\varepsilon$  [162] which we mentioned earlier in this section, the unbiased predictor at any point  $x$  can be derived as follows (Equation 134):

$$\hat{y}(x) = \beta_0 + \boldsymbol{\Sigma}_M(x, \cdot)^T [\boldsymbol{\Sigma}_M + \widehat{\boldsymbol{\Sigma}}_\varepsilon]^{-1} (\bar{\mathbf{y}} - \beta_0 \mathbf{1}_k) \quad (134)$$

where  $\boldsymbol{\Sigma}_M$  is the  $k$ -by- $k$  covariance matrix across all design points  $x_1, x_2, \dots, x_k$ ;  $\boldsymbol{\Sigma}_M(x, \cdot)$  is  $k$ -by-1 covariance vector with the  $i^{th}$  element as  $\text{Cov}(x, x_i)$ ;  $\widehat{\boldsymbol{\Sigma}}_\varepsilon$  is the estimated covariance diagonal matrix:  $\widehat{\boldsymbol{\Sigma}}_\varepsilon = \text{Diag}\{\frac{\hat{V}(x_1)}{n_1}, \frac{\hat{V}(x_2)}{n_2}, \dots, \frac{\hat{V}(x_k)}{n_k}\}$ ;  $\bar{\mathbf{y}}$  is the  $k$ -by-1 vector with the  $i^{th}$  element as  $\bar{y}(x_i)$

The prediction uncertainty at any point  $x$  can also be derived as follows (Equation 135):

$$\hat{s}^2(x) = \tau^2 - \boldsymbol{\Sigma}_M(x, \cdot)^T [\boldsymbol{\Sigma}_M + \widehat{\boldsymbol{\Sigma}}_\varepsilon]^{-1} \boldsymbol{\Sigma}_M(x, \cdot) + \boldsymbol{\delta}^T \boldsymbol{\delta} (\mathbf{1}_k^T [\boldsymbol{\Sigma}_M + \widehat{\boldsymbol{\Sigma}}_\varepsilon]^{-1} \mathbf{1}_k)^{-1} \quad (135)$$

where  $\boldsymbol{\delta} = \mathbf{1} - \mathbf{1}_k^T [\boldsymbol{\Sigma}_M + \widehat{\boldsymbol{\Sigma}}_\varepsilon]^{-1} \boldsymbol{\Sigma}_M(x, \cdot)$ . Note that the matrix  $[\boldsymbol{\Sigma}_M + \widehat{\boldsymbol{\Sigma}}_\varepsilon]$  is positive definite because  $\boldsymbol{\Sigma}_M$  is a positive definite covariance matrix (based on the assumption for stochastic Kriging which was mentioned earlier in this section), and  $\widehat{\boldsymbol{\Sigma}}_\varepsilon$  is also positive because it is a diagonal matrix with positive entry (i.e.,  $\hat{V}(x_i) > 0$  for any  $x_i$ ).

### 7.3 Infill criteria

An infill criterion is the standard way used to search for new sample locations where the black-box simulation needs to be evaluated. The step to search for infill points is also called adaptive sampling. For the purpose of global optimization, it is necessary that an infill criterion keeps a balance between local search (exploitation) and global search (exploration), which is important for asymptotic convergence and practical performance [316,137]. In this section, we first review the EI approach that is widely adopted by different research communities. Then we introduce the basic theory of OS approach. The modifications of these approaches are also discussed when considering the heteroscedastic-noise simulations.

### 7.3.1 Expected Improvement approach

For deterministic models, Jones' EGO algorithm [67] used the EI function as the infill criterion.

The improvement function can be defined with Equation 136:

$$I(x) = \max(f_{min} - \hat{y}, 0) \quad (136)$$

where  $f_{min}$  is the minimum objective value of the already visited sample points, and  $\hat{y}$  is the Kriging predictor at design point  $x$ . Equation 136 computes the improvement (decrease from  $f_{min}$ ) we can achieve if we sample at  $x$ . The expected value of the improvement (EI function) can be derived as shown in Equation (137).

$$E[I(x)] = (f_{min} - \hat{y}) \Phi\left(\frac{f_{min} - \hat{y}}{s}\right) + s \phi\left(\frac{f_{min} - \hat{y}}{s}\right) \quad (137)$$

where  $\Phi$  is the normal cumulative distribution function, and  $\phi$  is the normal probability density function. If we evaluate the derivatives of the EI function with respect to  $\hat{y}$  and  $s$ , we would notice that the EI value is larger with lower  $\hat{y}$  and higher  $s$ . Therefore, when maximizing the EI function, it will tend to find sample locations by a particular trade-off between local search (lower  $\hat{y}$ ) and global search (higher  $s$ ). After the algorithm ends, the returned approximately optimal solution is the sample point with minimum observed objective function value.

When applying EI to stochastic simulations, it is straightforward to substitute  $\hat{y}$  and  $s$  with SK prediction and square root of estimated prediction variance. However, there are several additional difficulties that we need to address. First, the true value (deterministic values) of  $f_{min}$  is not exactly known when we only have access to noisy observations from the stochastic simulation. It is not plausible to choose  $f_{min}$  as the minimum of the estimated mean of the sample points (i.e.,  $f_{min} = \min\{\bar{y}(x_1), \bar{y}(x_2), \dots, \bar{y}(x_k)\}$ ) because for stochastic Kriging the predicted surface does not necessarily go through every data point  $(x_i, \bar{y}(x_i))$ , especially when the noise variance is high at  $x_i$  [162]. Alternatively, in many papers on Kriging-based approaches for optimization with stochastic simulations (e.g., Ref. [161,283,303,314,300]),  $f_{min}$  is calculated by seeking to the surrogate

prediction (i.e.,  $\hat{y}(x_i)$ ) at the sample points. In this work, we adopt the strategy by Vazquez, Villemonteix et al. [300] and choose  $f_{min}$  as the minimum of the SK prediction at all the sampled locations, as is shown in (Equation 138).

$$f_{min} = \min \{\hat{y}(x_1), \hat{y}(x_2), \dots, \hat{y}(x_k)\} \quad (138)$$

Vazquez, Villemonteix et al. [300] claimed that by choosing  $f_{min}$  in this way we were evaluating the expected improvement upon the minimum of the surrogate prediction when maximizing the EI function (Equation 137). A second issue we need to consider is the choice of final returned optimal solution after the algorithm ends. This is a nontrivial problem because for stochastic simulations, unlike deterministic models, the observed objective value is no longer equivalent to its true value. Here we choose the returned optimal solution (denoted with  $\tilde{x}$ ) as the sample point with the minimum SK prediction value, and the returned optimal objective value (denoted with  $\tilde{y}$ ) as the SK prediction at this returned optimal solution (Equation 139):

$$\begin{aligned} \tilde{x} &= \arg \min_{x \in \{x_1, \dots, x_k\}} \hat{y}(x) \\ \tilde{y} &= \hat{y}(\tilde{x}). \end{aligned} \quad (139)$$

Jalali, Van Nieuwenhuysse et al. [314] mentioned that this identification strategy is promising to show better performance than others in finding the optimal solution.

The algorithm for this SK-EI approach is shown in Figure 53. We first build an initial SK surrogate using a space-filling design. A common choice is to use a Latin Hypercube Sampling (LHS) [249] strategy consisting of  $10d$  sample locations (see Ref. [67,283]), where  $d$  is the dimension of the problem. At each sample location, we make  $m$  replications ( $m = 10$ ) in order to estimate the mean and variance. In this work, we did not include the allocation strategy (e.g., Ref. [303]) to allocate different number of replications at different sample locations, because we would like to compare the performance of two algorithms under the same sampling and replication settings. Then, during the adaptive sampling stage, SK-EI searches from the unvisited locations (i.e.,  $x \in \Theta \setminus$

$\{x_1, \dots, x_k\}$ ) and find the new sample location (i.e.,  $x^{new}$ ) with maximized EI function value. At the  $x^{new}$ , we make  $m$  replications ( $m = 10$ ) of the stochastic simulation and update the surrogate with respect to the model parameters  $(\beta_0, \tau^2, \theta_h)$ . Note that we do not allow to revisit the sampled locations so that it is guaranteed that at each sample location we have the same number of replications of the simulation. This step of adaptive sampling continues repeatedly and terminates when the number of iterations exceeds the user-defined upper bound  $N_{max}$  (i.e. the total sampling budget being depleted). Finally, after the algorithm terminates, the identified optimal solution (and its corresponding SK predictor value) is returned as the sample location at which the SK prediction is minimum among all the sample points.

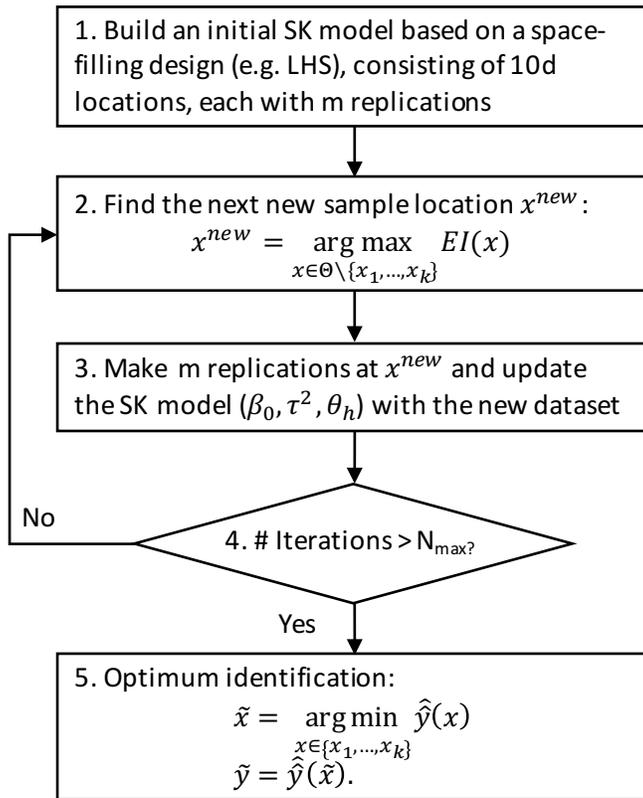


Figure 53. SK-EI algorithm

### 7.3.2 One stage approach

The OS approach was initially proposed by Jones [123] in which it was used with Kriging for optimization problems with deterministic models. This approach assumes that we have a goal value  $f^*$  for the global optimum. Instead of searching for expectations of improvement, we assess the likelihood that the goal value  $f^*$  could exist at a given point. For example, we make a hypothesis that  $f^*$  is achieved at a point  $x$ . To evaluate this hypothesis, we compute the likelihood of the sample data conditional upon the assumption that the surface goes through the point  $(x, f^*)$ . The expression of this conditional likelihood for Kriging model can be found in Ref. [123]

Based on the assumption for SK model that was mentioned Section 7.2, Ankenman, Nelson et al. [162] proved that the vector  $(Y(x_0), \bar{y}(x_1), \bar{y}(x_2), \dots, \bar{y}(x_k))^T$  follows a multivariate normal distribution. Therefore, based on the properties of multivariate normality [317], the conditional likelihood with SK can be derived in a similar fashion with that for Kriging. The expression for conditional log likelihood using SK is shown in Equation 140.

$$\begin{aligned}
 l_{cond}(x, \beta_0, \tau^2, \theta_h) &= -\ln[(2\pi)^{0.5k}] - 0.5\ln[|\mathbf{C}|] - 0.5(\bar{\mathbf{y}} - \mathbf{m})^T[\mathbf{C}]^{-1}(\bar{\mathbf{y}} - \mathbf{m}) \\
 &\text{where} \\
 \mathbf{C} &= \boldsymbol{\Sigma}_M + \widehat{\boldsymbol{\Sigma}}_\varepsilon - \boldsymbol{\Sigma}_M(x, \cdot)\boldsymbol{\Sigma}_M(x, \cdot)^T/\tau^2 \\
 \mathbf{m} &= \beta_0 \mathbf{1}_k + \boldsymbol{\Sigma}_M(x, \cdot)(f^* - \beta_0)/\tau^2
 \end{aligned} \tag{140}$$

Hence, the next sample point is found by maximizing  $l_{cond}$ . An advantage of using  $l_{cond}$  is that it optimizes the sample location  $x$  together with the model parameters  $\beta_0, \tau^2, \theta_h$ , which prevents the risks of finding  $x$  based on a pre-determined, possibly misleading, set of model parameters [123]. However, compared to EI approach, OS approach has a higher algorithmic complexity from several aspects. First, the value of  $f^*$  is usually unknown and needs to be chosen in a certain way, and the method of this is presented in the next paragraph. Additionally,  $l_{cond}$  function has a higher dimension than EI function due to the incorporation of  $(d + 2)$  model parameters, which makes it

a more difficult problem of maximizing  $ll_{cond}$  than maximizing the EI function. Finally, when maximizing  $ll_{cond}$ , it needs to be evaluated at many different  $x$ . At each  $x$ , we need to calculate a different matrix inverse (i.e.,  $[\mathbf{C}]^{-1}$ ), which requires  $O(k^3)$  operations. In comparison, maximizing the EI function at different  $x$  does not need to calculate a different matrix inverse because the calculation of  $\hat{y}$  (Equation 134) and  $s$  (Equation 135) only involves the same matrix inverse (i.e.,  $[\boldsymbol{\Sigma}_M + \widehat{\boldsymbol{\Sigma}}_\varepsilon]^{-1}$ ) which is not dependent on  $x$ .

For the OS approach, the strategy of choosing  $f^*$  is important for the trade-off between global search and local search. An overly optimistic goal (e.g.,  $f^* = -\infty$ ) will result in too much global search, while a much too pessimistic goal (e.g.,  $f^* = \min\{\hat{y}(x_1), \hat{y}(x_2), \dots, \hat{y}(x_k)\}$ ) will lead to purely local search [137]. A practically effective approach is to try a set of different  $f^*$  values, ranging from small to large, so as to keep a balanced search between exploration and exploitation. This method is shown to be very effective by Gutmann [207] when a radial basis function surrogate model is used. In this article, we adapted a similar method developed in Ref. [222] and applied it to the SK model, which is explained as follows.

A cycle of  $N+1$  types of  $f^*$  values are used, where  $N = 5$ . At iteration  $n$ ,  $f_n^*$  is calculated with Equation 141.

$$f_n^* = \min_{x \in \Omega} T(x) - W_n \cdot (\max_i \hat{y}(x_i) - \min_{x \in \Omega} T(x)) \quad (141)$$

with

$$\begin{aligned} W_n &= [1 - (n \bmod (N + 1))/N]^2 \\ T(x) &= (\hat{y}(x) - \hat{s}(x)) \\ \Omega &= \{x_1, \dots, x_k\} \end{aligned} \quad (142)$$

where  $\max_i \hat{y}(x_i)$  varies depending on the cycle iteration. At the first step of the cycle,  $\max_i \hat{y}(x_i)$  is calculated using all the sample locations. In the subsequent steps, the maximum is evaluated after the removal of the  $(n + n_{ini} - n_{max})$  points with largest SK prediction values, where  $n_{max}$  is calculated using Equation 143. In this way, the value of  $\max_i \hat{y}(x_i)$  decreases until the cycle ends, after which all points are considered again and the cycle starts from the beginning.

$$n_{max} = \max \{2, n_{ini} + n - \lfloor n/N \rfloor\} \quad (143)$$

In Equation 141, the parameter  $W_n$  changes its values from 1 (resulting in small  $f_n^*$ ) to 0 (resulting in large  $f_n^*$ ) during each cycle, which enables a balanced search between exploration (when  $f_n^*$  is small) and exploitation (when  $f_n^*$  is large). Note that, unlike the method in Ref. [222], we use a “statistical lower bound”  $T(x)$  (Equation 142) instead of  $\hat{y}(x)$ . This change is made to account for the effect of noise on building the SK model. Ankenman, Nelson et al. [162] demonstrated that for a SK model the prediction uncertainty  $\hat{s}(x)$  (Equation 135) at the sampled locations can be higher than zero when the noise variance is high. Therefore, when  $f_n^*$  cycles its values to local search (i.e.,  $W_n = 0$ ), by choosing  $f_n^* = \min_{x \in \Omega} T(x) = \min_{x \in \Omega} (\hat{y}(x) - \hat{s}(x))$  instead of  $f_n^* = \min_{x \in \Omega} \hat{y}(x)$ , we can prevent over trusting the SK prediction  $\hat{y}(x)$  at the sampled locations. In fact, the use of  $T(x)$  not just affects the case of  $W_n = 0$ , but also affects all the remaining cases of the cycle. Generally, this strategy of choosing  $f_n^*$  can slightly enhance the global search and prevent the sampling from being trapped in a narrow local area during the adaptive sampling stage. Later in Section 7.4, we can see from the results that it is indeed beneficial for the optimization with stochastic simulations.

The SK-OS algorithm follows similar steps with SK-EI (presented in Figure 53). The only difference is Step 2, where SK-OS algorithm first calculates the value  $f_n^*$ , and then, from the unvisited sample locations, it determines the next new sample location (i.e.  $x^{new}$ ) by maximizing  $U_{cond}(x^*, \beta_0, \tau^2, \theta_h)$ . For the remaining steps, including building initial surrogate, updating

surrogate, and identifying optimal solution and optimum objective value, SK-OS and SK-EI follow the same procedures. The reader can refer to Figure 53 for details.

## 7.4 Computational studies

### 7.4.1 Noise scenarios

The performance of the SK-EI and SK-OS algorithms can be affected by different noise structures (i.e., different heteroscedasticity). In this work, we consider two aspects of the noise structure: noise magnitude and noise pattern, which is similar with that in Ref.[314]. In terms of noise magnitude, we relate the square root of the noise variance (i.e.,  $\xi(x)$ ) to  $R_f$ , which is defined [314] as the range of the objective value within the input space  $\Theta$  (Equation 144),

$$R_f = \max_{x \in \Theta} f(x) - \min_{x \in \Theta} f(x). \quad (144)$$

(Note that by the definition of  $R_f$ , it is always a positive value for each test problem.) For “small” noise magnitude,  $\xi(x)$  changes in a narrower range of smaller values (i.e. varying between  $0.15R_f$  and  $0.6R_f$ ); whereas for “large” noise magnitude,  $\xi(x)$  changes in a wider range of larger values (i.e. varying between  $0.3R_f$  and  $1.2R_f$ ). In terms of noise pattern, we assume that  $\xi(x)$  changes linearly with the objective value. For “easy” pattern,  $\xi(x)$  increases linearly as the objective value increases, in which case the noise is smallest at the global minimum; whereas for “hard” pattern,  $\xi(x)$  decreases linearly as the objective value increases, in which case the noise is largest at the global minimum. Based on such considerations, a total number of 4 noise scenarios are used for each test problem. The expressions of  $\xi(x)$  for these 4 noise scenarios are shown with Equation 145. Note that with these expressions  $\xi(x)$  is always positive by its definition.

Small noise magnitude

$$- \text{ Easy pattern: } \xi(x) = 0.45 \left( f - \min_{x \in \Theta} f(x) \right) + 0.15R_f \quad (145)$$

- Hard pattern:  $\xi(x) = -0.45 \left( f - \min_{x \in \Theta} f(x) \right) + 0.6R_f$

Large noise magnitude

- Easy pattern:  $\xi(x) = 0.9 \left( f - \min_{x \in \Theta} f(x) \right) + 0.3R_f$
- Hard pattern:  $\xi(x) = -0.9 \left( f - \min_{x \in \Theta} f(x) \right) + 1.2R_f$

These scenarios are illustrated with a D1 test problem in Figure 54. In this figure, the solid line is the objective function (e.g.,  $f(x)$  in Equation 128), and the dashed lines represent  $f(x) \pm 2\xi(x)$ .

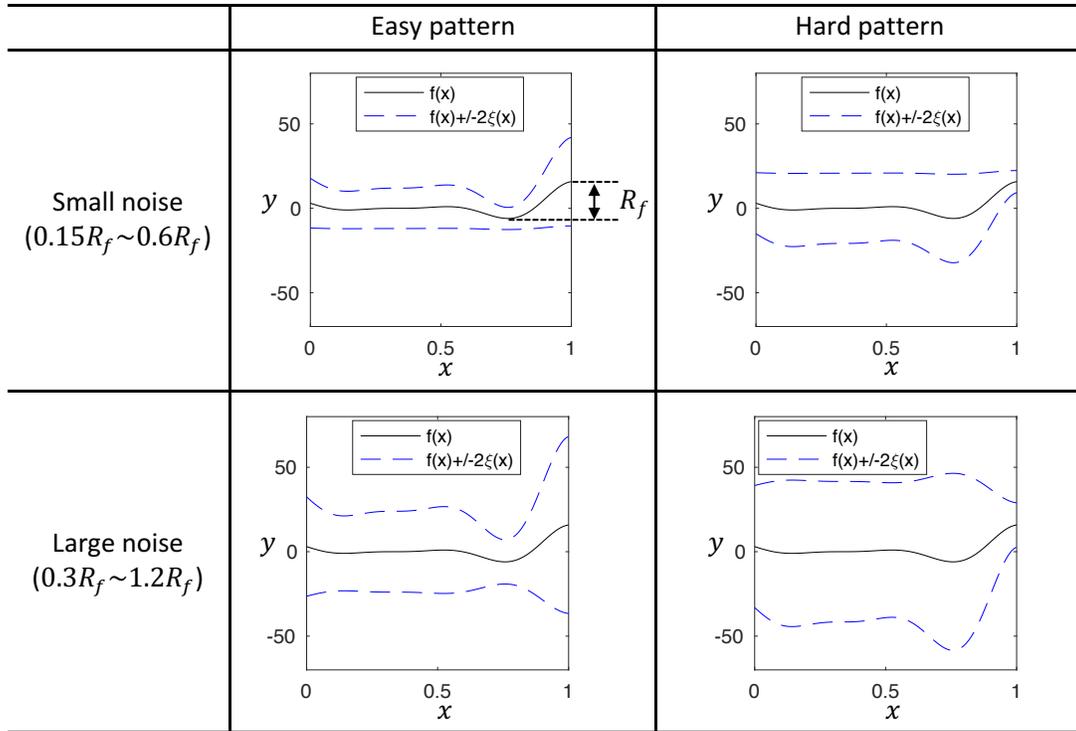


Figure 54. Illustration of 4 noise scenarios with D1 test problem. Solid line:  $f(x)$ ; Dashed line:  $f(x) \pm 2\xi(x)$ .

#### 7.4.2 Performance measures and data visualization

Considering the randomness of the initial LHS design points and the stochastic simulations, both algorithms are run 30 times for all the noise scenarios of each test problem. For comparison

purpose, both algorithms share the same 30 sets of initial sample points (i.e., same noisy observations at the same sample locations) for each noise scenario. As is discussed in Section 7.3, the performance of the algorithm depends on the accuracy of locating the optimal solution ( $x^*$ ) and the accuracy of predicting the optimal objective value ( $y^*$ ). Therefore, in this article, we consider two performance measures, which are illustrated with a D1 test problem in Figure 55. In this figure, the solid line is the SK prediction  $\hat{y}(x)$ ; the dot-dashed line is the original function  $f(x)$ ; the cross point represents the identified optimal solution ( $\tilde{x}$ ) returned from the algorithm.  $x^*$  represents the true optimal solution of the test problem;  $y^*$  represents the true optimal objective value of the test problem.

The first performance measure is “ $y_{orig}(\tilde{x})$ ”, which is defined as the original function (i.e.,  $f(x)$ ) value evaluated at the identified optimal location  $\tilde{x}$ . The algorithm’s performance is good if  $y_{orig}(\tilde{x})$  is close to  $y^*$ . This index indicates the algorithm’s capability of locating the optimal solution.

The second performance measure is “ $gap(\tilde{x})$ ”, which is defined as the absolute value of the difference between the SK prediction and the original function evaluated at  $\tilde{x}$  (e.g.,  $gap(\tilde{x}) = abs(\hat{y}(\tilde{x}) - y_{orig}(\tilde{x}))$ ). The algorithm’s performance is good if  $gap(\tilde{x})$  is close to zero. This index reflects the accuracy of the prediction of optimal objective value.

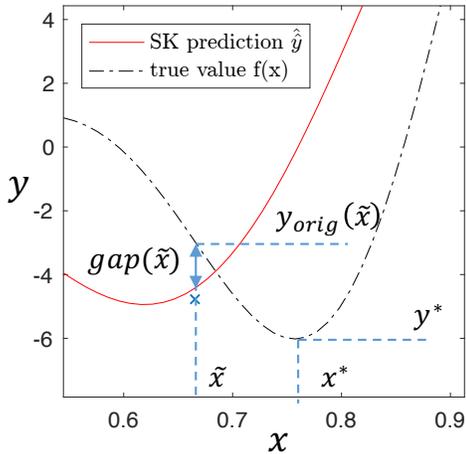


Figure 55. Illustration of two performance measures with the D1 test problem. Black dot-dashed line: objective function  $f(x)$ ; red solid line: SK prediction  $\hat{y}(x)$ .

To visualize the data, we use the boxplot for  $y_{orig}(\tilde{x})$  and for  $gap(\tilde{x})$ , respectively at the last iteration of adaptive sampling. This shows the distribution of the algorithm performance (30 runs) when it terminates. Also, we use the data profiles [318] to show the fraction of problems solved at each iteration. For data profiles, two measures  $\eta_{y95}$  and  $\eta_{gap0.2}$  are used (associated to “ $y_{orig}(\tilde{x})$ ” and “ $gap(\tilde{x})$ ”, respectively), whose calculations are shown in Equation 146.  $\eta_{y95}$  calculates the percentage of runs whose  $y_{orig}(\tilde{x})$  values are within the range  $[0.95y^*, 1.05y^*]$ .  $\eta_{gap0.2}$  measures the percentage of runs which have  $gap(\tilde{x})$  values within the range  $[-0.2, 0.2]$ .

$$\eta_{y95} = \frac{|y_{orig}(\tilde{x}): abs(y_{orig}(\tilde{x}) - y^*) \leq abs((1 - 0.95)y^*)|}{30} \times 100$$

$$\eta_{gap0.2} = \frac{|gap(\tilde{x}): gap(\tilde{x}) \leq 0.2|}{30} \times 100$$
(146)

In practice, we have also evaluated other data profiles, for example,  $y_{orig}(\tilde{x})$  varying in the ranges  $[0.8y^*, 1.2y^*]$ ,  $[0.9y^*, 1.1y^*]$ , and  $[0.99y^*, 1.01y^*]$ ;  $gap(\tilde{x})$  varying in the ranges  $[-0.1, 0.1]$ ,  $[-0.5, 0.5]$ , and  $[-1, 1]$ . While for each test problem the data profiles show similar trends,  $\eta_{y95}$  and

$\eta_{gap0.2}$  provide a relatively better visualization on the comparison of algorithms' performance. Therefore, in this paper we only show the figures of  $\eta_{\gamma95}$  and  $\eta_{gap0.2}$ .

#### 7.4.3 Test problems and implementation details

We use 7 standard test problems to compare the performance of SK-EI and SK-OS. A summary of these test problems is provided in Table 22. For detailed functions, the readers can refer to the supporting information of this article. These test problems have different dimensions ranging from 1D to 4D, and different multi-modality properties. They are chosen because they have been used to test the optimization algorithms for stochastic simulations [283] and/or to compare the EI and OS approach for optimization on deterministic models [319].

Table 22. Summary of 7 test problems

test problems	Dimension	Multi-modal	# global min	Rf
D1	1	Yes	1	21.8504
Branin	2	Yes	3	5.9236
Camelback	2	Yes	2	6.7649
Michalewicz	2	Yes	1	1.8013
Goldstein-Price	2	Yes	1	5.2461
Hartmann3	3	Yes	1	3.8628
Rosenbrock	4	No	1	7.2510

To build the SK model, we use the code provided on <http://stochasticriging.net/>. For all the 4 noise scenarios of each test problem, we make 30 runs with SK-EI and SK-OS, using 100 iterations of adaptive sampling (10 replications at each sample location). The optimization problems (i.e., maximizing  $EI$  in Equation 137 and maximizing  $l_{cond}$  in Equation 140) during the adaptive

sampling stages are solved with “fmincon” in Matlab 2016a. Because  $EI$  and  $ll_{cond}$  can have multiple local optimum, a multi-start strategy should be used when maximizing these two functions. In practice, at each iteration when  $EI$  (or  $ll_{cond}$ ) is maximized, we used a number of  $10d$  random sample points (sampled with the LHS strategy) as the initial points. Then, the returned optimal point with the best value of  $EI$  (or  $ll_{cond}$ ) is taken as the next sample location where the expensive stochastic simulation (i.e., the test problem) is called. The computational results are provided in the following section.

## 7.5 Results

Due to the space limitations, we only show the results on “D1” (Figure 56) and “Rosenbrock” (Figure 57) to support the analysis in the following sub-sections. We choose to show these two test problems because they are two typical representatives that can show the advantages and limitations of the both the SK-EI and SK-OS algorithms. As for the remaining test problems, the result figures are included in the supporting information of this article.

### 7.5.1 Interpretation on the data figures

To explain how to read the computational results, we use the “D1” test problem as an example (Figure. 56). The left column includes the boxplot of  $y_{orig}(\tilde{x})$  and data profiles of  $\eta_{y95}$ , which compare the algorithms’ capability to locate the minimum. The right column includes the boxplot of  $gap(\tilde{x})$  and the data profiles of  $\eta_{0.2}$ , which compare the accuracy of predicting the minimum objective value. In the first row of boxplots, SK-OS result is shown with red boxes; SK-EI result shown in blue boxes. The dashed lines indicate the target values that we are trying to achieve (i.e. the dashed line in (a) is the global minimum objective value -6.0207; the dashed line in (b) is 0, which is the ideal value of  $gap(\tilde{x})$ ). The second row of data profiles, (c) and (d), compares the algorithm’s performance in the two *easy* noise scenarios ( “small-easy” and “large-easy” ). The

third row of data profiles, (e) and (f), compares the algorithm's performance in the two *hard* noise scenarios ( "small-hard" and "large-hard" ). In the data profile figures, SK-OS is noted with solid lines, SK-EI with dot-dashed lines. Detailed annotation on different data point marks is shown in the figure legend, where "S.E." stands for "small-easy" noise scenario; "L.E." -> "large-easy" ; "S.H." -> "small-hard" ; "L.H." -> "large-hard" .

To interpret the results, with boxplots, the performance of the algorithm is considered to be good if the boxplot is close to the target (dashed line) with smaller box spans, which indicates a larger fraction of runs can successfully locate the global minimum (using  $y_{orig}(\tilde{x})$ ), or accurately predict the optimal objective value (using  $gap(\tilde{x})$ ). On the other hand, with data profiles, the algorithm has a good performance if the data increase faster and end up with higher percentage values, which suggest the algorithm can more quickly locate the global minimum (using  $\eta_{y95}$ ) or improve the prediction accuracy of objective value (using  $\eta_{gap0.2}$ ) within fewer iterations. It should be noted that, unlike the cases when we have deterministic test problems, data profiles of the ratios (i.e.,  $\eta_{y95}$  and  $\eta_{gap0.2}$ ) for stochastic test problems are no longer monotone with respect to iterations. This is because the values of SK predictions at sampled locations can change when the surrogate gets updated at each iteration.

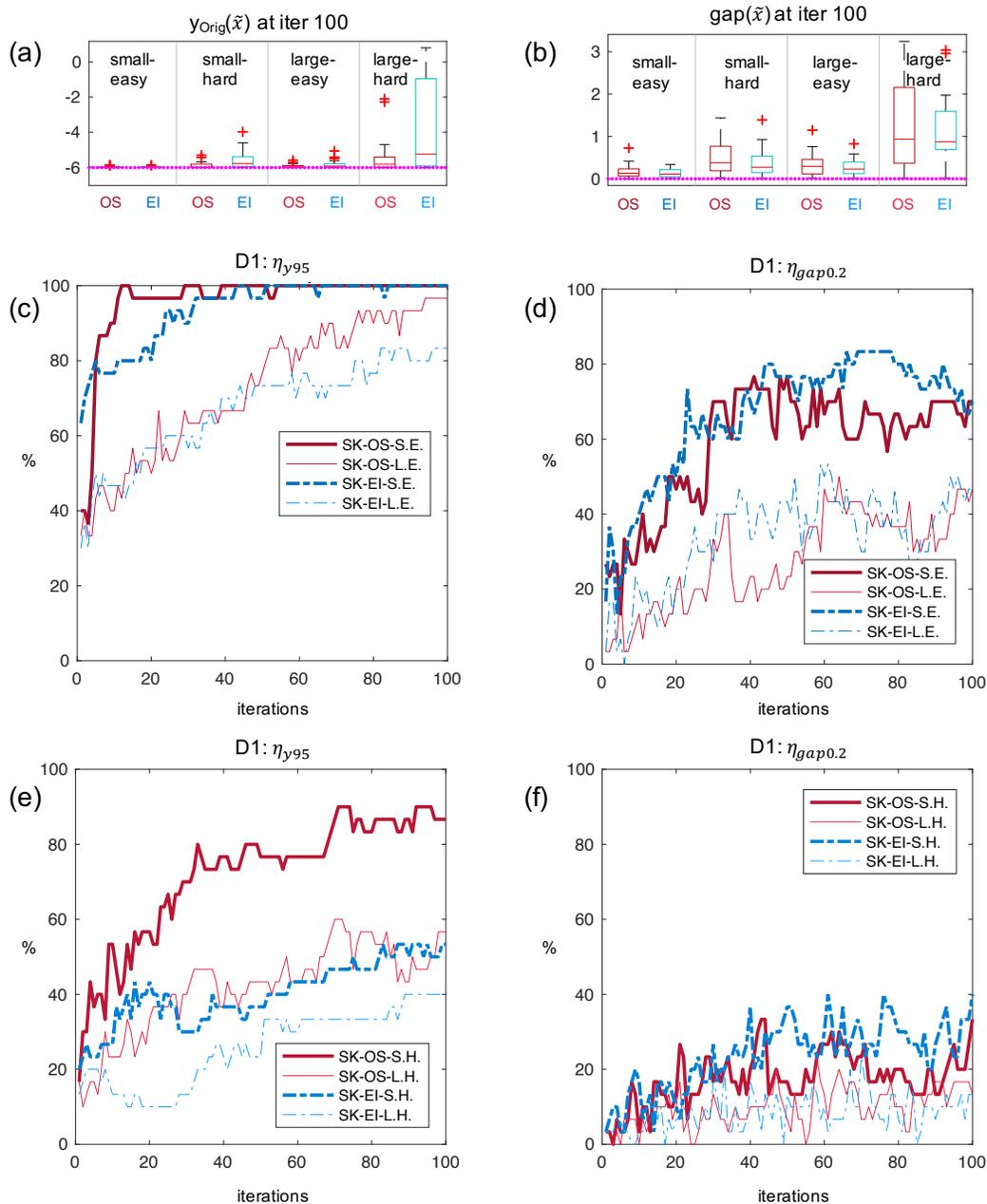


Figure 56. Boxplot and data profiles of “D1” test problem. (a) boxplot of  $y_{orig}(\tilde{x})$  after 100 iterations of adaptive sampling; (b) boxplot of  $gap(\tilde{x})$  after 100 iterations of adaptive sampling; (c) data profiles of  $\eta_{y95}$  for easy noise scenarios; (d) data profiles of  $\eta_{gap0.2}$  for easy noise scenarios; (e) data profiles of  $\eta_{y95}$  for hard noise scenarios; (f) data profiles of  $\eta_{gap0.2}$  for hard noise scenarios.

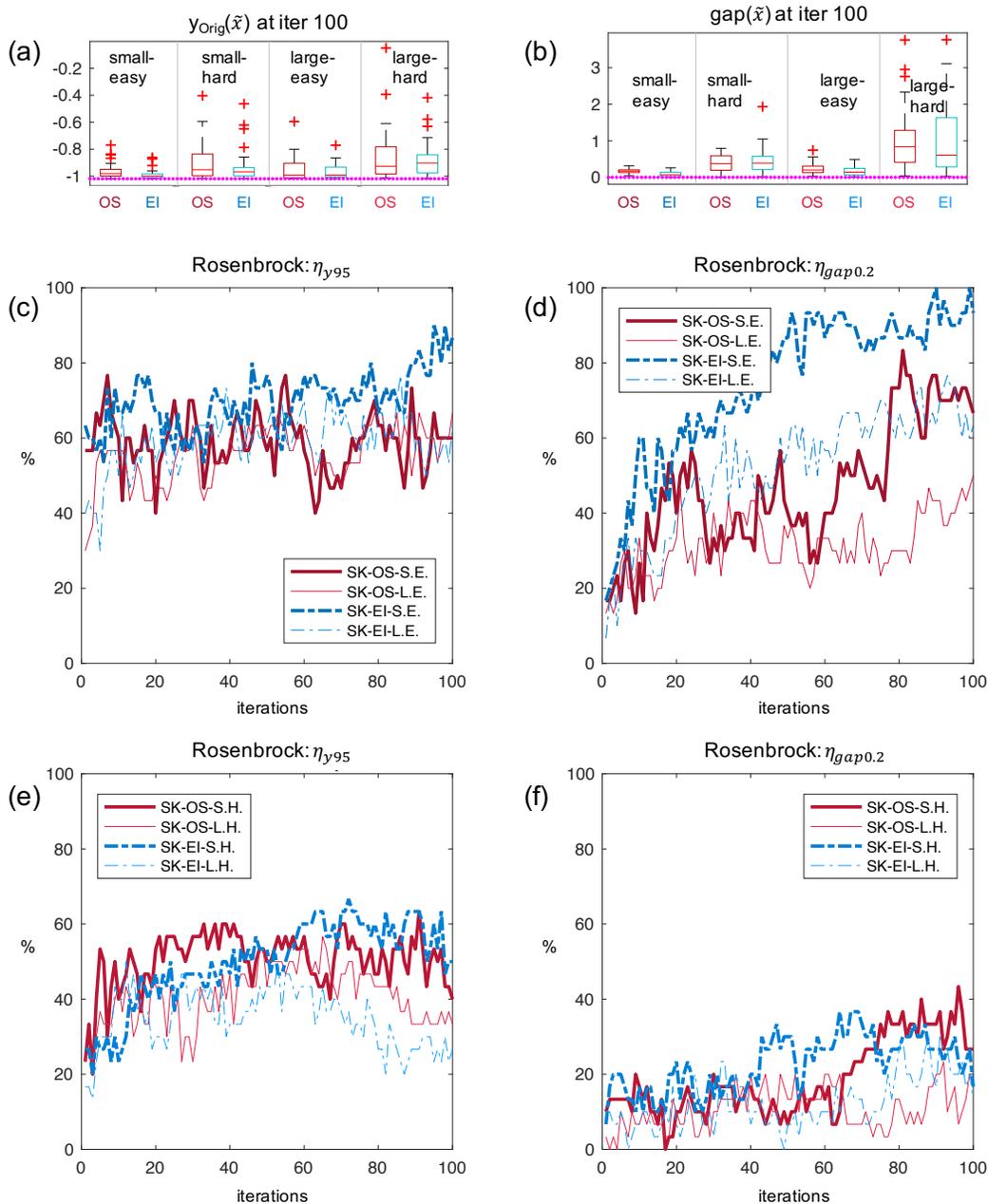


Figure 57. Boxplot and data profiles of “Rosenbrock” test problem. (a) boxplot of  $y_{orig}(\tilde{x})$  after 100 iterations of adaptive sampling; (b) boxplot of  $gap(\tilde{x})$  after 100 iterations of adaptive sampling; (c) data profiles of  $\eta_{y95}$  for easy noise scenarios; (d) data profiles of  $\eta_{gap0.2}$  for easy noise scenarios; (e) data profiles of  $\eta_{y95}$  for hard noise scenarios; (f) data profiles of  $\eta_{gap0.2}$  for hard noise scenarios.

### 7.5.2 Influence of noise structure on algorithms' performance

From the data obtained we notice some common features of both algorithms, that is, they can both be heavily affected by the noise structure in all the test problems. In terms of the noise magnitude, the “large” noise magnitude can severely degrade the performance of both algorithms, as we can observe that, for each algorithm, the boxplots of “large” noise magnitude have larger spans and deviate farther from the target line compared to those of small noise magnitude. The trend in data profiles also reflects this effect. For “large” noise scenarios, the trend is that the data profiles of “ $\eta_{y95}$ ” and “ $\eta_{gap0.2}$ ” increase slower with iteration and end up with smaller percentage values compared with those for “small” noise scenarios. On the other hand, considering the noise pattern, the “hard” pattern usually leads to the algorithm performing worse compared to the “easy pattern”. Such findings are consistent with those provided by Ref. [314]. This is expected because the accuracy of the surrogate models is expected to decline with the increase in the noise magnitude and complexity. To alleviate this negative influence, Jalali, Van Nieuwenhuysse et al [314] claimed that increasing the number of replications can be effective in some cases.

#### 7.5.2.1 Comparison of performance between SK-OS and SK-EI

##### **Boxplots of $y_{orig}(\tilde{x})$ and data profiles of $\eta_{y95}$**

With boxplots of  $y_{orig}(\tilde{x})$  and data profiles of  $\eta_{y95}$ , the comparison on the two algorithms' performance for each test problem is summarized in Table 23. In this table, a solid circle mark “•” indicates that SK-OS method is better than SK-EI in a specific noise scenario of a test problem; an empty circle mark “○” means that SK-EI performs better than SK-OS; a short hyphen mark “-” represents that not much difference is observed between the two algorithms.

Table 23. Summary of comparison between SK-EI and SK-OS with  $y_{orig}(\tilde{x})$  and  $\eta_{y95}$

	S.E.		S.H.		L.E.		L.H.	
	EI	OS	EI	OS	EI	OS	EI	OS
D1		•		•		•		•
Branin	-	-		•	-	-		○
Camelback		•		•		•	-	-
Michalewicz	-	-		•		•		•
Goldstein-Price		•		•	-	-	-	-
Hartmann3		•		•		•		•
Rosenbrock	○		○		○		○	

From Table 23, we can see that SK-OS outperforms SK-EI in locating the minimum in most of the scenarios of the tested problem (i.e. 18 out of 28 cases). For the test problems “D1” and “Hartmann3”, SK-OS has better performance than SK-EI in all the noise scenarios. From the data profiles of these two test problems, we can notice that the difference is especially profound for the hard noise scenarios. As for the 2D test problems “Branin”, “Camelback”, “Michalewicz”, and “Goldstein-Price”, the advantage of using SK-OS varies depending on different noise scenarios, with one exception (i.e., large-hard noise scenario of “branin”), where SK-OS is noticeably worse than SK-EI. In the remaining scenarios, there is no obvious difference between SK-OS and SK-EI. However, with the test problem “Rosenbrock”, SK-OS performs worse than SK-EI in all four noise scenarios.

After taking a careful look on the test problems’ characteristics, we notice that, with respect to locating the global minimum, SK-OS can be more favorable than SK-EI mostly in the multimodal cases where the global minimum is located in a wide-deep basin (e.g., “D1”, “Hartmann3”,

“Camelback” ). This is because the SK-OS has more focus in the global search, which makes it less likely to be trapped in exhaustively searching some unpromising local regions when a test problem has multiple local minimums. In comparison, SK-EI has a higher chance of getting stuck in unpromising regions because its search direction can more easily be misled, due to either an inaccurate estimated  $f_{min}$  or surrogate model parameters, with the existence of simulation noise.

However, SK-OS can be less competent than SK-EI when the global minimum is located in a narrow-flat valley (e.g., “Branin” , “Rosenbrock” ). In such cases, although the valley of the global minimum is easy to find, the convergence to the global minimum is difficult [283]. This can be seen in the data profile of  $\eta_{y95}$  for “Rosenbrock” (Figure 57 c and e), where both algorithms make very slow progress in locating the minimum during the 100 iterations of adaptive sampling. For these test problems, the SK-EI (with more focus on local search than SK-OS) is more likely to make meaningful search within the identified valley with the global optimum.

### ***Boxplots of $gap(\tilde{x})$ and data profiles of $\eta_{gap0.2}$***

The comparison of SK-EI and SK-OS with  $gap(\tilde{x})$  and  $\eta_{gap0.2}$  is summarized in Table 24. In almost half of the total tested scenarios (13 out of 28), no obvious difference is observed between SK-EI and SK-OS. However, there are also 13 cases where SK-EI performs better than SK-OS in the prediction accuracy at the returned optimal location. There are only two scenarios (i.e., S.H. and L.H. for “Hartmann3” ) where SK-EI is worse than SK-OS. Such results indicate that, although SK-OS can be more advantageous than SK-EI in locating the minimum, it is sacrificing the prediction accuracy in some cases. This is because SK-OS may lack sufficient exploitation around the neighborhood of the global minimum that is needed in order to make a more accurate prediction of the optimum objective value.

Table 24. Summary of comparison between SK-EI and SK-OS with  $gap(\tilde{x})$  and  $\eta_{gap0.2}$

	S.E.		S.H.		L.E.		L.H.	
	EI	OS	EI	OS	EI	OS	EI	OS
D1	-	-	-	-	-	-	-	-
Branin	○		○		○		-	-
Camelback	-	-	○		○		-	-
Michalewicz	○		○		-	-	○	
Goldstein-Price	-	-	○		○		-	-
Hartmann3	-	-		●	○			●
Rosenbrock	○		-	-	○		-	-

### 7.6 Case study of a pharmaceutical manufacturing process

We apply both algorithms to a Continuous Direct Compaction (CDC) flowsheet simulation model in order to find the optimal operating conditions that result in minimum total cost. This flowsheet model (Figure 58) was developed in gPROMS software ([www.psenterprise.com](http://www.psenterprise.com)), and is used to simulate the manufacturing process of producing an oral solid drug product. Two components, namely the active pharmaceutical ingredient (API) and Excipient are first fed to a co-mill for de-lumping purposes. After that, lubricant is added to the powder mixture to improve the powder's flowability. The three component materials are then mixed in a blender, and transferred to the tablet press where drug tablets are compressed. For a detailed description of the model equations, readers may refer to Ref. [320]

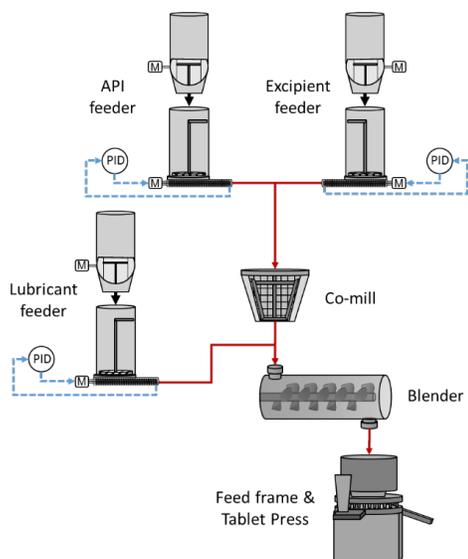


Figure 58. Flowsheet of DC line

The optimization problem is formulated as shown in problem (104). Based on a previous sensitivity analysis, two operation conditions are selected as the decision variables, namely the API flow rate ( $FR_{API}$ ) and the API refill strategy ( $RS_{API}$ ).  $FR_{API}$  is the flow rate set point in the API feeder;  $RS_{API}$  is the fill level of API feeder, above which the refill begins. Those two variables are selected because they have huge influence (relative to other operation conditions) on the final product properties (e.g., API concentration, tablet uniformity). Also, the generation of out-of-spec products is most sensitive to these two factors. The total cost consists of three terms: material cost, utility cost, and waste cost. A detailed calculation of these three terms is shown in Ref. [320]. In this article, we add a noise term  $\varepsilon_{cost_{waste}}$  to the objective in order to reflect the uncertainty associated to the estimation of waste cost. We assume that  $\varepsilon_{cost_{waste}}$  is subject to a normal distribution:  $\varepsilon_{cost_{waste}} \sim N(0, \sigma^2(FR_{API}, RS_{API}))$ , where  $\sigma$  is inherently related to  $FR_{API}$  and  $RS_{API}$ . The assumption we made on  $\varepsilon_{cost_{waste}}$  is a plausible one because the cost waste is linearly dependent on the actual observed flow rate [238], and the observed flow rate is found to be subject to a normal distribution [17]. However, it will need to be further verified with experimental studies in the future.

$$\begin{aligned}
& \min_{FR_{API}, RS_{API}} \text{cost}_{total} = \text{cost}_{material} + \text{cost}_{utility} + \text{cost}_{waste} + \varepsilon_{\text{cost}_{waste}} \\
& \text{s. t.} \\
& FR_{API}^{lb} \leq FR_{API} \leq FR_{API}^{ub} \\
& RS_{API}^{lb} \leq RS_{API} \leq RS_{API}^{ub}
\end{aligned} \tag{147}$$

We use the gPROMS object “gO:MATLAB” to communicate between gPROMS and Matlab.

Figure 59 depicts the landscape of the simulation without noise.

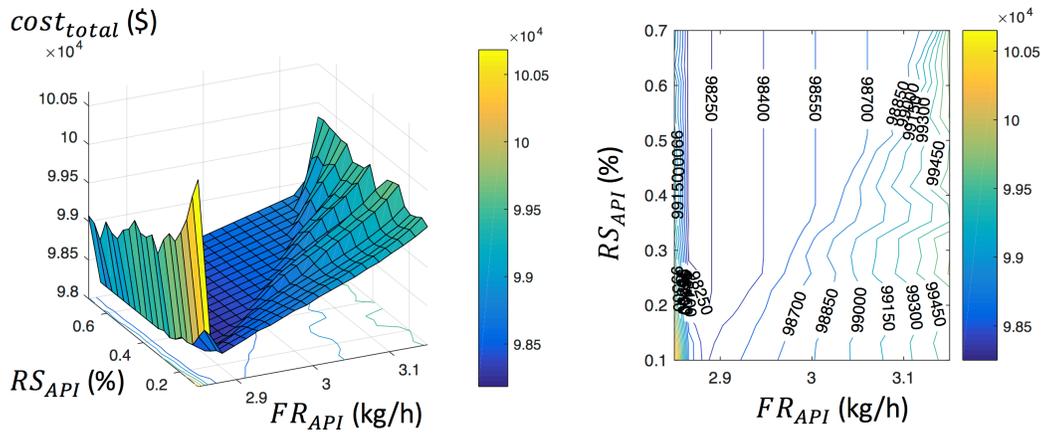


Figure 59. Noise-free landscape of objective function. Left: 3D surface. Right: contour plot

For a comparison purpose, we first solve the noise-free optimization problem (i.e., without considering  $\varepsilon_{\text{cost}_{waste}}$ ) using “fmincon” in Matlab, the results of which are shown in Table 25.

After 48 function calls, a local minimum is returned with the minimum total cost of \$98,181 per day. The optimal solution suggests that the API flow rate should be kept close to its lower bound, with a mid-range value of refill strategy, which is consistent with the findings in Ref. [320]. Then, we consider the noisy case (i.e. with  $\varepsilon_{\text{cost}_{waste}}$ ) and apply SK-OS and SK-EI to the optimization problem with stochastic simulation. The optimization results (after 100 iterations of adaptive sampling) are listed in Table 25. In terms of the optimal solution,  $FR_{API}$  values from both SK-OS

and SK-EI are almost identical to that from the noise-free case;  $RS_{API}$  values from SK-OS and SK-EI are larger. Such results suggest a higher frequency of refilling, which is beneficial to alleviate the overall variations (and reduce the waste cost of subsequent out-of-spec products) due to refill operations [254]. On the other hand, considering the predicted optimal objective value, the  $y_{orig}(\tilde{x})$  values at the optimal solution are \$ 98,180 (SK-OS) and \$ 98,179 (SK-EI) respectively, indicating that both SK-OS and SK-EI can successfully locate the minimum. In addition, SK predictions  $\hat{y}(\tilde{x})$  at the returned optimal solution also show a high accuracy, as we can see the values of  $\hat{y}(\tilde{x})$  are almost the same with  $y_{orig}(\tilde{x})$  for both SK-EI and SK-OS. In summary, both SK-EI and SK-OS can be used to find the optimal solution with this stochastic simulation, and the quality of the solution is competent to that from the optimization with the noise-free simulation.

Table 25. Summary of optimization results

			Noise-free	Noisy	
			fmincon	SK-EI	SK-OS
$x$	$FR_{API}$	[kg/h]	2.86	2.86	2.86
	$RS_{API}$	[%]	32.11	70.00	40.14
$cost_{total}$	$y^*$	[\$]	98,181	-	-
	$y_{orig}(\tilde{x})$	[\$]	-	98,180	98,179
	$\hat{y}(\tilde{x})$	[\$]	-	98,180	98,179

To have a detailed look at this problem, we plot the two performance measures  $y_{orig}(\tilde{x})$ ,  $gap(\tilde{x})$ , and the square root of estimated variance  $\sqrt{\hat{V}(\tilde{x})}$  by iteration (Figure 60), where the solid represents SK-EI and dot-dashed line for SK-OS. From Figure 60 (a)  $y_{orig}(\tilde{x})$  and (b)  $gap(\tilde{x})$ , we can see that both SK-EI and SK-OS make progress mainly in the first 50 to 60 iterations. From (c)  $\sqrt{\hat{V}(\tilde{x})}$ , we can notice that the estimated variance is actually very small (compared to the objective value)

at returned optimal solution  $\tilde{x}$ , which is the main reason that both SK-EI and SK-OS can achieve high prediction accuracy at the returned optimal solution.

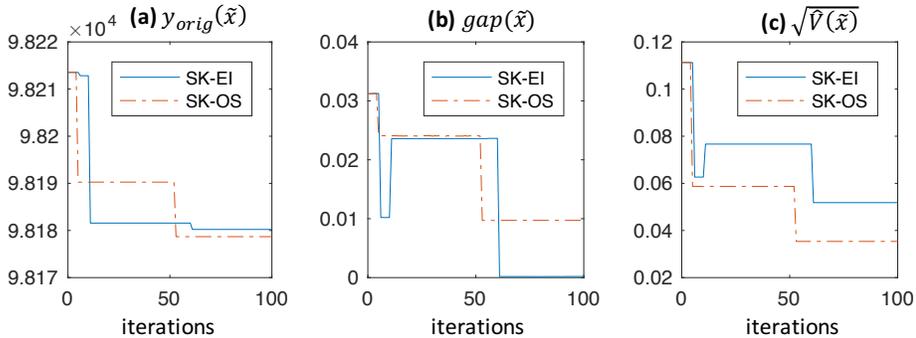


Figure 60. Performance measures and estimated variance at returned optimal solution by iteration.

## 7.7 Summary and future work

We presented and compared the performance of two surrogate-based optimization approaches to solve optimization problems using simulation with heteroscedastic noise in the output. Specifically, we used stochastic Kriging as the surrogate, and adopted the one-stage approach as the infill criterion (SK-OS). Its performance is compared with the popular expected improvement approach (SK-EI). With the use of two performance measures,  $y_{orig}(\tilde{x})$  and  $gap(\tilde{x})$ , we make a comparison between SK-OS and SK-EI using 7 test problems, each with 4 different heteroscedastic noise scenarios. The results show that both algorithms are heavily affected by the noise structure. In most of the test cases, SK-OS is better than SK-EI in locating the global minimum, however it may sacrifice some accuracy in predicting the optimal objective value. The algorithms were finally applied to solve an optimization problem in the pharmaceutical manufacturing process with a black-box stochastic simulation with artificial noise. In this case both SK-OS and SK-EI can find the optimum with a limited sampling budget, and the quality of the solution is competent to that from optimization with the noise-free simulation. For the future work, we would need to improve the efficiency of the surrogate-based optimization approaches. For example, it can be beneficial to

develop an intelligent allocation strategy to determine how to choose the number of replications of simulations at each sample location during adaptive sampling in order to make the best use of the overall sampling budget. Also, more research is needed on the convergence analysis and algorithm stopping criteria. Another interesting area is to investigate how different kernel functions for SK can affect the algorithm performance.

## 8 A Kriging-based method with feasibility enhancements for constrained optimization of black-box stochastic systems

### **Abstract**

Stochastically constrained simulation optimization problems are difficult to solve because the inherent noise terms to a black-box system can bring uncertainties to both the objective and the constraint function. To address such difficulties, we propose a Kriging-based optimization framework, which uses stochastic Kriging models to approximate the objective and the constraint functions and an adaptive sampling approach to sequentially search for the next point that is promising to have a better objective. The main contribution of this work is to incorporate a feasibility-enhanced term to the infill sampling criterion, which improves the Kriging-based algorithm's capabilities of returning a truly feasible near-optimal solution for stochastic systems. The efficacy of the Kriging-based algorithm is demonstrated with eight benchmark problems and a case study on a pharmaceutical process optimization problem.

### 8.1 Introduction

Simulation optimization (SO) refers to the optimization of the performance of a system, of which the objective and constraints can only be estimated by stochastic simulations. Unlike algebraic model-based mathematical programming, SO usually involves black-box simulations; that is, closed-form simulation expressions are not available to decision makers. In this paper, we consider

single-objective SO problems with an inequality constraint and continuous input variables. The general form of this problem is shown in Problem (148).

$$\begin{aligned}
 & \min_{\mathbf{x}} f(\mathbf{x}) \\
 & \text{s. t.} \\
 & g(\mathbf{x}) \leq 0 \\
 & \mathbf{x}_l \leq \mathbf{x} \leq \mathbf{x}_u \\
 & \mathbf{x} \in \mathbb{R}^d
 \end{aligned} \tag{148}$$

In Problem (148),  $f(\mathbf{x})$  and  $g(\mathbf{x})$  cannot be directly observed, but only estimated with a stochastic simulation. Thus, the user only has access to the noisy observation of the objective function,  $\tilde{f}_i(\mathbf{x})$ , and noisy observation of the constraint,  $\tilde{g}_i(\mathbf{x})$ :

$$\begin{aligned}
 \tilde{f}_i(\mathbf{x}) &= f(\mathbf{x}) + \omega_i^f(\mathbf{x}) \\
 \tilde{g}_i(\mathbf{x}) &= g(\mathbf{x}) + \omega_i^g(\mathbf{x}),
 \end{aligned} \tag{149}$$

where  $\tilde{f}_i(\mathbf{x})$  is the observed objective value at the  $i^{th}$  replication of simulation evaluated at a specific point  $\mathbf{x}$ , and  $\omega_i^f(\mathbf{x})$  is the observed noise term of the objective function;  $\tilde{g}_i(\mathbf{x})$  is the observed constraint value at the  $i^{th}$  replication of simulation evaluated at  $\mathbf{x}$ , and  $\omega_i^g(\mathbf{x})$  is the observed constraint noise.

Following the settings of SO problems in Ref. [274], we make assumptions that the noise terms,  $\omega_i^f(\mathbf{x})$  and  $\omega_i^g(\mathbf{x})$ , are normally distributed, centered, and independent from one simulation run to another:

$$\begin{aligned}
 \omega_i^f(\mathbf{x}) &\sim \mathcal{N}\left(0, \xi_f^2(\mathbf{x})\right) \\
 \omega_i^g(\mathbf{x}) &\sim \mathcal{N}\left(0, \xi_g^2(\mathbf{x})\right)
 \end{aligned} \tag{150}$$

where the noise variances,  $\xi_f^2(\mathbf{x})$  and  $\xi_g^2(\mathbf{x})$ , are dependent on the inputs  $\mathbf{x}$ . Thus, the inherent noise terms to the simulation have heteroscedastic variances. Further, we also assume that the objective and the constraint are independent.

SO has extensive applications in manufacturing, computer and communication network, and business processes. For a comprehensive review on the current progress on SO, the readers are referred to Refs. [321-323,291]. Below, we overviewed the commonly used approaches for SO, with a focus on constrained problems.

Gradient-based approaches can be applied to SO problems when the objective and constraint are differentiable. The differentiability and continuity of the objective and constraint can be examined by coupling theory [324,323]. Stochastic approximation (SA) methods [325,326] use estimated gradient information to search for an optimal solution, which is closely related to the steepest descent methods in derivative-based optimization of deterministic models. Fu [321] summarized a variety of gradient approximation methods for SA algorithms. Among these methods, the Simultaneous Perturbation Stochastic Approximation (SPSA) approach [297] only requires two simulation runs to estimate the gradient, which makes it very promising for high-dimensional problems. Although SA methods generally return a local optimum, they can be extended by injecting a Monte Carlo randomness term in the SA recursion and combine global search techniques to the algorithm [327]. For stochastically constrained problems, Bhatnagar et al. [328] developed four algorithms for problems with multiple inequality constraints. These algorithms used the Lagrange multiplier method with the gradient (and the Hessian) estimated using the smoothed functional (SF) technique and the SPSA method. It should be noted that, for most SA methods, it remains challenging to wisely choose the algorithm parameters, which is critical for a successful implementation.

For practical problems where the gradient does not exist or is difficult to estimate, simulation optimization problems need to be solved by derivative-free optimization approaches. Direct search methods determine the search directions by simply comparing function values without the need to approximate derivatives. Trosset [329] examined the use of direct search methods in stochastic optimization, and modified the pattern search method to ensure global convergence for the optimization of noisy systems. Kim and Zhang [330] presented a Generating Set Search (GSS) method for simulation optimization problems with stochastic noise, and the optimization problem was approximated using a sample average approximation (SAA) scheme. Chang and Lu [331] proposed a stochastic Nelder-Mead (SNM) simplex method to solve simulation optimization problems with a quantile-based objective and introduced penalty functions to deal with inequality constraint. Alternatively, simulation optimization problems can be solved by random search algorithms. An overview of such algorithms in simulation optimization is given by Andradóttir [332]. For problems with stochastic constraints, Lacksonen [333] compared the performance of a genetic algorithm, simulated annealing, and two direct search methods, and found the genetic algorithm to be more robust for 25 tested problems. Klassen and Yoogalingam [334] developed a scatter search/tabu search optimization procedure to determine optimal rules for a stochastic appointment scheduling problem.

Recently, surrogate-based optimization methods have been extensively developed and mostly used for continuous problems. This type of algorithm approximates the simulation by a surrogate model (also known as metamodel or response surface), which can be used to facilitate the search of the optimal solution. Recent advances in surrogate modeling and its applications in optimization are reviewed in Refs. [137,335,336]. For constrained simulation optimization problems, Angün et al. [337] developed a Generalized Response Surface Methodology (GRSM) strategy that used low-order polynomials to locally approximate the unknown functions. Augustin and Marzouk [338]

introduced a trust-region optimization algorithm (S)NOWPAC (Stochastic Nonlinear Optimization With Path-Augmented Constraints), which utilized local fully linear models combined with Gaussian process models to approximate the objective and constraint functions with inherent noise. Nezhad and Mahlooji [339] presented an Artificial Neural Network (ANN) algorithm for expensive simulation optimization problems with constraints. Li et al. [340] compared five surrogate modeling techniques (ANN, Radial Basis Function, Support Vector Regression, Kriging, and Multivariate Adaptive Regression Splines) and proposed an optimization framework which integrated these models into a genetic algorithm.

Among various surrogate-based methods, the Kriging-based optimization approach (also known as Bayesian optimization [341]) has gained an increasing popularity in different research communities. This type of method uses a stochastic process model to approximate an expensive simulation, and calculates a certain infill criterion to balance exploration and exploitation when searching for a new sample point. Early algorithms involving the use of stochastic process models include the Bayesian algorithm by Mockus [280] and P-algorithm by Žilinskas [342]. It was later popularized by the Efficient Global Optimization (EGO) method of Jones et al. [67], which used an Expected Improvement (EI) function as the infill sampling criterion. Reviews on the development of Kriging-based methods are in Refs. [343,279]. Note that the Kriging-based methods were first presented for deterministic unconstrained problems. For the purpose of handling constraints, several strategies have been developed. Sasena et al. [2012] proposed a penalty method to restrict it from searching in the infeasible region. Schonlau [344] suggested multiply the EI function by the probability that a point is feasible. This Schonlau's method was furthered investigated in Refs. [345-347,230], and applied to solve robust optimization problems [348] and a pultrusion process optimization problem [349]. Parr et al. [346,347] proposed to transform constrained problems into multi-objective unconstrained problems. Bagheri et al. [350] surveyed the existing constraint handling methods for EGO and modified Schonlau's method by introducing

a newly defined probability of feasibility. Wang et al. [351] proposed a two-phase approach, which first performed a feasibility analysis to search for the feasible region and then applied a modified Schonlau's approach with a penalty to find a feasible optimal solution.

The Kriging-based methods have been adapted to solve optimization problems of stochastic systems. For unconstrained problems with homoscedastic noise variances, the Kriging-based algorithms include the Sequential Kriging Optimization (SKO) [161], the Re-Interpolation (RI) method [301], and the Informational Approach to Global Optimization (IAGO) strategy [352]. Picheny et al. [283] presented a comprehensive study on the performance of different Kriging-based optimization techniques for stochastic systems with homoscedastic noise variances. Additionally, for unconstrained problems with heteroscedastic noise variances, different modeling techniques have been developed, such as Stochastic Kriging (SK) [162], Stochastic Intrinsic Kriging (SIK) [353], and Variational Heteroscedastic Gaussian Process Regression (VHGPR) [354]. Such modeling approaches laid a solid foundation for the later developed Kriging-based optimization algorithms, including Expected Quantile Improvement (EQI) [303], Expected Improvement via Stochastic Intrinsic Kriging (EI-SIK) [355], Expected Risk Improvement (ERI) [315], and a One-Stage (OS) approach [356]. Jalali et al. [278] surveyed and compared the performance of various SK-based optimization algorithms. However, for constrained optimization of stochastic simulations, there is a relatively scarce research on applying Kriging-based approaches. Boukouvala and Ierapetritou [64] developed an optimization framework that first performed a global search to improve the surrogate of objective and characterized the feasible region boundary, and then utilized a local search within the feasible region to find an optimal solution. Kleijnen et al. [357] proposed a heuristic which combined the Design of Experiments, Kriging, and mathematical programming for optimization problems with integer input variables.

In this work, we apply the Stochastic Kriging modeling technique and adapt a “constrained Expected Improvement” method to solve constrained SO problems. Further, considering that the inherent noises can introduce uncertainties in determining the feasibility of sample points, we propose a “feasibility-enhanced Expected Improvement” to explicitly improve the feasibility knowledge while searching for a new sample point. The outline of this paper is as follows. The Kriging-based optimization framework is described in Section 8.2, including the stochastic Kriging modeling technique and two infill criteria for adaptive sampling. Section 8.3 mentions the benchmark problems and performance measures that are used to compare different algorithms. Computations results are discussed in Section 8.4. A pharmaceutical case study is included in Section 8.5. Conclusions and future work are given in Section 8.6.

## 8.2 Kriging-based optimization approaches

In this section, we first introduce the basics of Stochastic Kriging, which is used to model both the stochastic objective and stochastic constraint functions. Then, in Section 8.2.2, we explain the two infill criteria (i.e., constrained Expected Improvement, and feasibility-enhanced Expected Improvement) which are used to guide the search for new sample points. Both approaches follow the same optimization framework, which is discussed in Section 8.2.3.

### 8.2.1 Stochastic Kriging Model

Ankenman et al. [162] developed the Stochastic Kriging (SK) as an extension to the ordinary Kriging [275] in order to model a stochastic simulation with heteroscedastic noise variances. It expresses the output of a simulation on its  $j^{th}$  replication run as:

$$y_j(\mathbf{x}) = \beta_0 + M(\mathbf{x}) + \varepsilon_j(\mathbf{x}), \quad (151)$$

In Equation (111),  $\beta_0$  is a parameter representing a constant surface trend.  $M$  is named as the “extrinsic uncertainty” term in Ref. [162], which is a realized random field with zero mean; namely,

$M$  is considered to be randomly sampled from a space of function mapping  $\mathbb{R}^d \rightarrow \mathbb{R}$ , which have characteristic spatial correlations: when  $\mathbf{x}$  is close to  $\mathbf{x}'$ , we have  $M(\mathbf{x})$  and  $M(\mathbf{x}')$  tend to be similar. A variety of kernel functions can be used to model the correlation between  $M(\mathbf{x})$  and  $M(\mathbf{x}')$ . In this work, we apply the cubic correlation function (Equation (152)(153)), which was shown to give a robust modeling performance in our previous studies [358].

$$\text{Corr}(\mathbf{h}|\boldsymbol{\zeta}) = \prod_{i=1}^d \text{Corr}(h_i|\zeta_i), \quad (152)$$

where

$$\text{Corr}(h_i|\vartheta_i) = \begin{cases} 1 - 6(|h_i|/\vartheta_i)^2 + 6(|h_i|/\vartheta_i)^3, & \text{if } |h_i| \leq \vartheta_i/2 \\ 2(1 - |h_i|/\vartheta_i)^3, & \text{if } \vartheta_i/2 \leq |h_i| \leq \vartheta_i \\ 0, & \text{if } |h_i| \geq \vartheta_i \end{cases} \quad (153)$$

$$\mathbf{h} = \mathbf{x} - \mathbf{x}', \boldsymbol{\zeta} = (\vartheta_1, \dots, \vartheta_d), \vartheta_i > 0, \text{ and } \text{Corr}(h_i|0) \equiv 0.$$

On the basis of the correlation function, the covariance between  $M(\mathbf{x})$  and  $M(\mathbf{x}')$  are expressed as  $\text{Cov}(\mathbf{h}|\boldsymbol{\zeta}) = \tau^2 \text{Corr}(\mathbf{h}|\boldsymbol{\zeta})$ , where  $\tau^2$  is a model parameter representing the variance of  $M(\mathbf{x})$  for all  $\mathbf{x}$ .  $\varepsilon_j(\mathbf{x})$  in Equation (111) is known as the “intrinsic uncertainty” term in Ref. [162]. It represents the realized noise on the  $j^{\text{th}}$  replication of a simulation run.

It should be noted that the SK model in Ref. [162] is based on the following assumption:

*Assumption 1. The random field  $M$  is a stationary Gaussian random field, and  $\varepsilon_1(\mathbf{x}_1), \varepsilon_2(\mathbf{x}_1), \dots$  are independent and identically distributed with a normal distribution:  $\mathcal{N}(0, V(\mathbf{x}_i))$ , independent of  $\varepsilon_j(\mathbf{x}_h)$  for all  $j$  and  $h \neq i$ , and independent of  $M$ .*

Ankenman et al. [162] proved that *Assumption 1* implies that, for any set of different sample points  $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_k$ , the random vector  $[M(\mathbf{x}_1), M(\mathbf{x}_2), \dots, M(\mathbf{x}_k)]^T$  has a multivariate normal

distribution with marginal mean 0, a positive variance  $\tau^2$ , and a positive definite correlation matrix  $\mathbf{R}_M$ .

To apply the SK to model a stochastic simulation, we need to estimate model parameters. The intrinsic variance at a sample point  $\mathbf{x}_i$  can be estimated with  $n_i$  replications:

$$\hat{V}(\mathbf{x}_i) = \frac{\sum_{j=1}^{n_i} (y_j(\mathbf{x}_i) - \bar{y}(\mathbf{x}_i))^2}{(n_i - 1)}, \quad (154)$$

where  $\bar{y}(\mathbf{x}_i)$  is the estimated mean at  $\mathbf{x}_i$ :  $\bar{y}(\mathbf{x}_i) = \sum_{j=1}^{n_i} y_j(\mathbf{x}_i) / n_i$ . It has been shown by Ankenman et al. [162] that the estimated  $\hat{V}$  will not introduce much inaccuracy to the surrogate model as long as  $n_i$  is not too small. A rule of thumb is to choose  $n_i \geq 10$  [162]. Additionally, the model parameters (i.e.,  $\beta_0, \vartheta_1, \dots, \vartheta_d, \tau^2$ ) are calculated by maximizing the log-likelihood, which was derived in Ref. [162].

On the basis of previously mentioned *Assumption 1* for SK, the unbiased predictor is derived as follows:

$$\hat{y}(\mathbf{x}) = \beta_0 + \boldsymbol{\Sigma}_M(\mathbf{x}, \cdot)^T [\boldsymbol{\Sigma}_M + \widehat{\boldsymbol{\Sigma}}_\varepsilon]^{-1} (\bar{\mathbf{y}} - \beta_0 \mathbf{1}_k), \quad (155)$$

where  $\boldsymbol{\Sigma}_M$  is the k-by-k covariance matrix for all the  $k$  sample points  $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_k$ ;  $\boldsymbol{\Sigma}_M(\mathbf{x}, \cdot)$  is k-by-1 covariance vector with the  $i^{th}$  element being  $\text{Cov}(\mathbf{x}, \mathbf{x}_i)$ ;  $\widehat{\boldsymbol{\Sigma}}_\varepsilon$  is the estimated covariance diagonal matrix:  $\widehat{\boldsymbol{\Sigma}}_\varepsilon = \text{Diag} \left\{ \frac{\hat{V}(\mathbf{x}_1)}{n_1}, \frac{\hat{V}(\mathbf{x}_2)}{n_2}, \dots, \frac{\hat{V}(\mathbf{x}_k)}{n_k} \right\}$ ;  $\bar{\mathbf{y}}$  is the k-by-1 vector with the  $i^{th}$  element being  $\bar{y}(\mathbf{x}_i)$ .

The derived prediction variance for a SK model is expressed as follows:

$$\hat{s}^2(\mathbf{x}) = \tau^2 - \boldsymbol{\Sigma}_M(\mathbf{x}, \cdot)^T [\boldsymbol{\Sigma}_M + \widehat{\boldsymbol{\Sigma}}_\varepsilon]^{-1} \boldsymbol{\Sigma}_M(\mathbf{x}, \cdot) + \boldsymbol{\delta}^T \boldsymbol{\delta} \left( \mathbf{1}_k^T [\boldsymbol{\Sigma}_M + \widehat{\boldsymbol{\Sigma}}_\varepsilon]^{-1} \mathbf{1}_k \right)^{-1}, \quad (156)$$

where  $\delta = 1 - \mathbf{1}_k^T [\boldsymbol{\Sigma}_M + \widehat{\boldsymbol{\Sigma}}_\varepsilon]^{-1} \boldsymbol{\Sigma}_M(\mathbf{x}, \cdot)$ . Note that the matrix  $[\boldsymbol{\Sigma}_M + \widehat{\boldsymbol{\Sigma}}_\varepsilon]$  is always positive definite on the basis of *Assumption 1* for the SK model.

### 8.2.2 Infill criteria

For a surrogate-based optimization method, it generally consists of two stages. In the first stage, a low-fidelity surrogate model is built with some initial points sampled using a specific sampling plan. In the second stage, the surrogate model gets improved with some update points that aim to increase modeling accuracy towards promising areas. The update points are selected using a certain infill criterion. This second stage is also known as the adaptive sampling stage and it is performed iteratively until a stop criterion is met. For the purpose of constrained optimization, an infill criterion should seek to points that lead to a better objective value and meanwhile prevent any constraint violations. Below, we describe two SK-based infill criteria: the constrained Expected Improvement and the Feasibility-enhanced Expected Improvement.

#### 8.2.2.1 Constrained Expected Improvement

The concept of expected improvement was first proposed by Mockus [280] and later used in the Efficient Global Optimization (EGO) algorithm by Jones et al. [67]. Here, we introduce this concept on the basis of SK models. Let  $f^{**}$  denote the current best objective function value. (Details of choosing  $f^{**}$  will be discussed later.) The SK model treats an unknown objective function value  $y_f(\mathbf{x})$  as an observation of a normally distributed random variable  $Y_f(\mathbf{x})$  with mean  $\hat{y}_f$  and variance  $\hat{s}_f^2$ . The expressions have been given in Equation (115) and (116). Therefore, at any unvisited point  $\mathbf{x}$ , there is some probability that the objective function value will be better than (or “improve upon”)  $f^{**}$ . A formal definition of improvement for the objective function is given as follows.

$$I_f(\mathbf{x}) = \max(f^{**} - Y_f(\mathbf{x}), 0). \quad (157)$$

Following the procedures in Ref. [67], the expected value of  $I_f$  (denoted as  $EI$ ) can be derived as follows:

$$EI(\mathbf{x}) = E[I_f(\mathbf{x})] = (f^{**} - \hat{y}_f) \Phi \left( \frac{f^{**} - \hat{y}_f}{\hat{s}_f} \right) + \hat{s}_f \phi \left( \frac{f^{**} - \hat{y}_f}{\hat{s}_f} \right), \quad (158)$$

where  $\Phi$  is the standard normal cumulative distribution function (CDF);  $\phi$  is the standard normal probability distribution function (PDF).

For stochastic systems, to further account for the influence of simulation noise on future observations, Huang et al. [161] proposed an Augmented Expected Improvement ( $AEI$ ) algorithm that incorporates a multiplicative term. Applying this approach to the SK model, we have the following modified  $AEI$  function:

$$AEI(\mathbf{x}) = E[I_f(\mathbf{x})] \cdot \left( 1 - \tau_f / \sqrt{\tau_f^2 + \hat{s}_f^2} \right), \quad (159)$$

where  $\tau_f^2 = \xi_f^2/n$ ;  $\xi_f^2$  is the noise variance in the objective function at  $\mathbf{x}$ ;  $n$  is the number of replicated simulation runs to be made at  $\mathbf{x}$ . In this paper, since the expression of  $\xi_f^2(\mathbf{x})$  is unknown, it is estimated with  $\hat{V}_f(\mathbf{x})$ . A Kriging model is built to predict  $\hat{V}_f(\mathbf{x})$  at any unvisited point. To ensure the predicted  $\hat{V}_f(\mathbf{x})$  is always non-negative, the Kriging model is built for  $\log \hat{V}_f(\mathbf{x})$ . This method of estimating  $\hat{V}_f(\mathbf{x})$  has also been used in Refs. [262,358]. In Equation (159), the multiplicative term  $\left( 1 - \tau_f / \sqrt{\tau_f^2 + \hat{s}_f^2} \right)$  is introduced to penalize areas where the prediction variance  $\hat{s}_f^2$  is relatively small compared to the estimated noise variance. For stochastic systems with heteroscedastic noise variances, it has been found that  $\hat{s}_f^2$  is more likely to be “inflated” where  $\xi_f^2$  is larger [162]. In that case, the multiplicative term can prevent over-exploiting a local area, and thus result in a better balanced global search. The benefits of using the multiplicative term have been extensively investigated in Refs. [283,278,358].

With the discussions above, we now turn our attention to strategies of handling the constraint. Recall that a SK model considers an unknown constraint value  $y_g(\mathbf{x})$  as an observation of random variable  $Y_g(\mathbf{x})$  subject to a normal distribution:  $Y_g(\mathbf{x}) \sim \mathcal{N}(\hat{y}_g, \hat{s}_g^2)$ , whose expressions are given in Equation (115) and (116). Thus, for any unvisited point  $\mathbf{x}$ , there is some probability that it will be feasible. The probability of this point to be feasible can be expressed as:

$$P(Y_g(\mathbf{x}) \leq 0) = \Phi\left(\frac{0 - \hat{y}_g}{\hat{s}_g}\right). \quad (160)$$

For the purpose of handling a constraint, Schonlau [344] proposed an infill criterion that uses the product of Expected Improvement (Equation (158)) and the probability of being feasible (Equation (160)). We adapt this constraint-handling approach to stochastic systems where  $AEI$  is used (Equation (159)) and propose the following constrained Augmented Expected Improvement (denoted as  $cAEI_f$ ) in Equation (161).

$$\begin{aligned} cAEI_f(\mathbf{x}) &= E[I_f(\mathbf{x})] \cdot \left(1 - \tau_f / \sqrt{\tau_f^2 + \hat{s}_f^2}\right) \cdot P(Y_g(\mathbf{x}) \leq 0) \\ &= \left( (f^{**} - \hat{y}_f) \Phi\left(\frac{f^{**} - \hat{y}_f}{\hat{s}_f}\right) + \hat{s}_f \phi\left(\frac{f^{**} - \hat{y}_f}{\hat{s}_f}\right) \right) \\ &\quad \cdot \left(1 - \frac{\tau_f}{\sqrt{\tau_f^2 + \hat{s}_f^2}}\right) \cdot \Phi\left(\frac{0 - \hat{y}_g}{\hat{s}_g}\right). \end{aligned} \quad (161)$$

This infill criterion determines the next sample point by maximizing  $cAEI_f$ . In the rest of this paper, this method will be referred to as the “constrained EI” approach.

Before applying the “constrained EI” method, we need to determine how to choose the current best point  $f^{**}$ . For deterministic systems,  $f^{**}$  should be the feasible point that has the best objective value among the visited sample points [351]. For stochastic systems, we follow a similar logic and

modify the approach developed by Huang et al. [161] for unconstrained problems. The formulas of calculating  $f^{**}$  are given as follows.

$$f^{**} = \hat{y}_f(\mathbf{x}^{**})$$

where

$$\mathbf{x}^{**} = \arg \min_{\mathbf{x} \in \chi} \left( \hat{y}_f(\mathbf{x}) + c_f \cdot \hat{s}_f(\mathbf{x}) \right) \quad (162)$$

s. t.

$$\hat{y}_g(\mathbf{x}) + c_g \cdot \hat{s}_g(\mathbf{x}) \leq 0$$

where  $\chi$  is the set of  $k$  visited sample points:  $\chi = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_k\}$ ;  $c_f$  and  $c_g$  are non-negative constant parameters that are used to calculate quantile values for the objective and constraint. The term  $\left( \hat{y}_f(\mathbf{x}) + c_f \cdot \hat{s}_f(\mathbf{x}) \right)$  was used by Huang et al. [161] to account for the uncertainty associated with the predicted objective value. In addition, we add a restrictive constraint  $\hat{y}_g(\mathbf{x}) + c_g \cdot \hat{s}_g(\mathbf{x}) \leq 0$  to further account for the uncertainty in the predicted constraint value. In this work,  $c_f = 1$  and  $c_g = 1$  are selected as the default. In practice, however, the formulas in Equation (162) do not guarantee that a feasible point  $\mathbf{x}^{**}$  satisfying the constraint in (162) can always be found among the visited sample points. This can happen when a test problem has a really small feasible region (or several small disjoint feasible regions), which has not yet been covered by the sample points. If that is the case, the infill criterion will be switched to search for a feasible point. Details of this are discussed in Section 8.2.3.

To demonstrate the difference between the three infill criteria ( $EI$ ,  $AEI$ , and  $cAEI_f$ ), we use a 2-dimensional problem, “constrained Branin”, as an example. (Expressions for this test problem are mentioned in Appendix.) This test problem is depicted in Figure 29. For a better visualization of the problem,  $f(\mathbf{x})$  and  $g(\mathbf{x})$  (i.e., the noise-free terms) are shown in Figure 29 (a);  $\xi_f(\mathbf{x})$  and  $\xi_g(\mathbf{x})$  (i.e., the inherent noise terms) are separately shown in Figure 29 (b) and (c).

In Figure 29 (a), the contour shows the surface of  $f(\mathbf{x})$ . It can be observed that the objective function shows a multi-modal characteristic and has three global minima, which are denoted with the three diamond points. The feasible region is confined by three feasible region boundaries (i.e.,  $g(\mathbf{x}) = 0$ ) that are denoted with thick solid lines. The three disjoint infeasible regions are shaded in white. From Figure 29 (a), it should be noted that only two of the global optima are feasible. In Figure 29 (b), the contour shows  $\xi_f(\mathbf{x})$ , which increases linearly in the  $x_2$  direction. Three diamond points are the three global minima of  $f(\mathbf{x})$ . In Figure 29 (c), the contour shows  $\xi_g(\mathbf{x})$ , which increases linearly in the  $x_1$  direction. Thick solid lines are feasible region boundaries where  $g(\mathbf{x}) = 0$ .

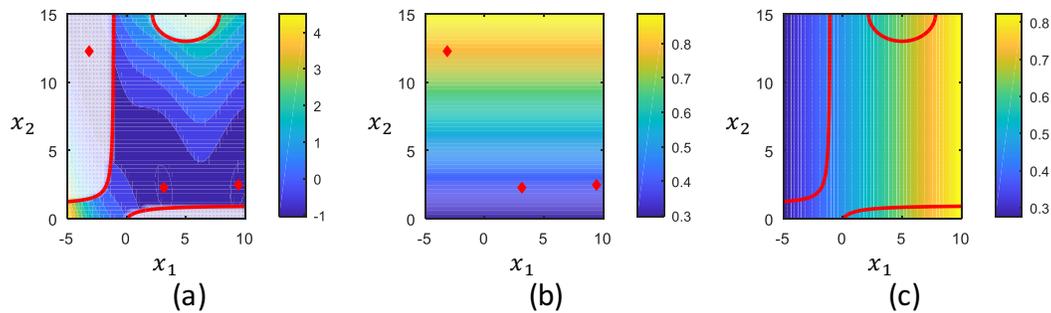


Figure 61. 2D Test problem “constrained Branin” . (a) Objective and constraint. Contours represent the objective function; diamond points are three global minima of the objective function; white-shaded areas are infeasible regions; thick solid lines are feasibility boundaries. (b) Noises in the objective function. Contours represent the standard deviation of the noise in the objective; diamond points are three global minima of the objective function. (c) Noises in the constraint function. Contours represent the standard deviation of the noise in the constraint; thick solid lines are feasibility boundaries.

The three above mentioned criteria ( $EI$ ,  $AEI$ , and  $cAEI_f$ ) are plotted in Figure 62. Initial SK models are built for the objective and constraint using 50 initial points (each with 50 replicated simulation

runs) that are sampled with a space-filling Latin Hypercube Sampling (LHS) design [56]. The initial points are denoted with dot points in Figure 62. The SK predictions can thus be used to calculate the infill criteria. The surface of  $-EI$  (Equation (158)) is shown with the contours in Figure 62 (a). It can be observed that there exist three promising regions that are located close to the three global minima. The top left region is significantly more promising than the other two near the bottom. Figure 62 (b) shows the surface of  $-AEI$  (Equation (159)). It can be found that the relative difference of the  $AEI$  values between three promising areas is smaller in comparison to that of  $EI$  in Figure 62 (a). In other words, the multiplicative term in  $AEI$  causes more penalty on the top left promising region (where the objective noise level is higher) than the bottom two promising regions. As such, the  $AEI$  criterion attempts to reduce the influence arising from the objective noise, and balance the global search by making the three promising regions more closely to be equivalent. Figure 62 (c) shows the surface of  $-cAEI_f$  (Equation (161)). The predicted feasible regions are confined by the predicted feasible region boundaries denoted with thick dashed lines. The predicted infeasible regions are shaded in white. It is obvious that with  $cAEI_f$ , there exist only two promising regions that are located in the predicted feasible regions. This shows the capability of the  $cAEI_f$  criterion in avoiding constraint violations while searching for the next sample point.

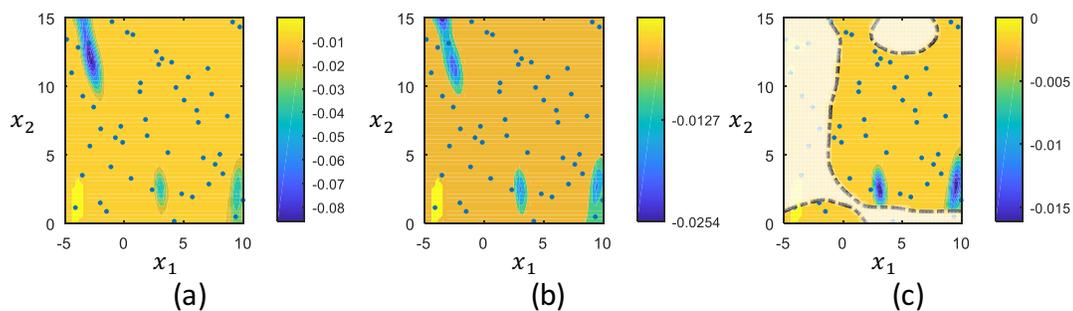


Figure 62. Infill criteria. (a)  $EI$ . (b)  $AEI$ . (c)  $cAEI_f$ . Contour represent the infill criteria; dot points are initial points; thick dashed lines are predicted feasibility boundaries; white-shaded areas are predicted infeasible regions.

### 8.2.2.2 Feasibility-enhanced Expected Improvement

The “constrained EI” approach only focuses on making improvements on the objective function, which maintains a balance between exploration (i.e., searching in highly uncertain areas for a point with a better objective value) and exploitation (i.e., adding points in a local promising area to reduce uncertainties of the current best objective value). However, it does not explicitly consider to improve the knowledge of the feasibility (i.e., constraint value) of the current best point. This requires attention for stochastic systems because the inherent noise can always bring uncertainties in the constraint, even for visited sample points. If there is a high uncertainty in the feasibility of the current best point, the decision-maker may face a high risk of choosing an infeasible point.

A direct way to improve the knowledge of feasibility is by feasibility analysis. By concept, feasibility analysis aims to quantitatively characterize the feasible region of a process. A recent review on this topic is given by Grossmann, et al. [44]. For feasibility analysis of deterministic black-box systems, Boukouvala and Ierapetritou [64] proposed a Kriging-based infill criterion to sequentially search for feasible region boundaries, which is known as “Expected Improvement for Feasibility” (denoted as *EIF*). Below, we give its expression on the basis of a SK model.

$$EIF(\mathbf{x}) = \hat{s}_g \cdot \phi\left(\frac{g^t - \hat{y}_g}{\hat{s}_g}\right). \quad (163)$$

In Equation (163),  $g^t$  is the target contour of feasibility value that needs to be found. By setting  $g^t = 0$ , the algorithm will search for all feasible region boundaries where  $g(\mathbf{x}) = 0$ . The *EIF* works in a similar way as the *EI* function. By maximizing *EIF*, it favors to search where the prediction uncertain  $\hat{s}_g$  is high and where the PDF value  $\phi\left(\frac{g^t - \hat{y}_g}{\hat{s}_g}\right)$  is large. Note that the  $\phi\left(\frac{g^t - \hat{y}_g}{\hat{s}_g}\right)$  is largest when  $\hat{y}_g = g^t$ ; that is, maximizing *EIF* favors areas where SK predictor is close to (or equal to) the target feasibility value. Therefore, maximizing *EIF* function will keep a balance

between a global search (i.e., where  $\hat{s}_g$  is large) and a local search (i.e., where  $\hat{y}_g = g^t$ ) for the contour of feasibility value  $g^t$ . Further mathematical details of this approach was discussed in Ref. [358,261].

For stochastic systems, Wang and Ierapetritou [358] modified the *EIF* function and proposed an infill criterion, known as “Expected Quantile Improvement for Feasibility” (denoted as *EQIF<sub>g</sub>*). This criterion addresses the noise’s influence on feasibility analysis by considering the  $\alpha$ -quantile of the SK prediction:  $q_\alpha = \hat{y}_g + \Phi^{-1}(\alpha) \cdot \hat{s}_g$ , with  $\alpha \in (0,1)$ . It was found that  $q_\alpha$  is subject to a normal distribution [274], which makes the derivation of *EQIF<sub>g</sub>* criterion to be closely related to that of the *EIF* criterion. Below, we directly give the expression of *EQIF<sub>g</sub>* on the basis of a SK model. Further details of the derivation can be seen in Ref. [356].

$$EQIF_g(\mathbf{x}) = \hat{s}_Q \cdot \phi\left(\frac{q^t - \hat{y}_Q}{\hat{s}_Q}\right)$$

where

$$\hat{y}_Q = \hat{y}_g + \Phi^{-1}(\alpha) \cdot \sqrt{\frac{\tau_g^2 \hat{s}_g^2}{\tau_g^2 + \hat{s}_g^2}} \quad (164)$$

$$\hat{s}_Q^2 = \frac{(\hat{s}_g^2)^2}{\tau_g^2 + \hat{s}_g^2}$$

$$q^t = g^t + \Phi^{-1}(\alpha) \cdot \hat{s}_g$$

In Equation (164),  $\hat{y}_Q$  and  $\hat{s}_Q^2$  represent the mean and variance of the normally distributed  $q_\alpha$ . In the expressions of  $\hat{y}_Q$  and  $\hat{s}_Q^2$ ,  $\tau_g^2 = \xi_g^2/n$  where  $\xi_g^2$  is the noise variance in the constraint function at  $\mathbf{x}$ ;  $n$  is the number of replicated simulation runs to be made at  $\mathbf{x}$ . In this paper, because the expression of  $\xi_g^2(\mathbf{x})$  is unknown, it is estimated with  $\hat{V}_g(\mathbf{x})$ . A Kriging model is built to predict  $\hat{V}_g(\mathbf{x})$  for an unvisited point. Specifically, the Kriging model is built for  $\log \hat{V}_g(\mathbf{x})$  to guarantee the

predicted  $\hat{V}_g(\mathbf{x})$  is non-negative. Similar techniques were used in Refs. [262,358]. From Equation (164), it can be noticed that both  $\hat{y}_Q$  and  $\hat{s}_Q^2$  implicitly accounts for noise variance, which lays the foundation of the  $EQIF_g$  criterion to be effective to handle stochasticity in the constraint.  $g^t$  is the target feasibility contour that needs to be found.  $\Phi^{-1}(\alpha)$  is the inverse of a standard normal CDF at the level of  $\alpha$ . The  $EQIF_g$  criterion has an expression very close to the above mentioned  $EIF$ , and it also works similarly with the  $EIF$  criterion. An advantage of using  $EQIF_g$  is that the search direction can be controlled by choosing the value of  $\alpha$ . With  $\alpha < 0.5$ ,  $\Phi^{-1}(\alpha) < 0$ ,  $EQIF_g$  attempts to conservatively search from a more feasible area (i.e.,  $\hat{y}_g < g^t$ ) towards the feasibility contour of  $g^t$ . In contrast, with  $\alpha \geq 0.5$ ,  $\Phi^{-1}(\alpha) \geq 0$ ,  $EQIF_g$  aggressively searches from a less feasible area (i.e.,  $\hat{y}_g \geq g^t$ ) towards the feasibility contour of  $g^t$ .

In this work, we apply the  $EQIF_g$  criterion to enhance the knowledge of feasibility of the current best point. Specifically, in Equation (164) the target feasibility value  $g^t$  is replaced  $g^{**}$ , which is defined as the predicted constraint value at  $\mathbf{x}^{**}$ :

$$g^{**} = \hat{y}_g(\mathbf{x}^{**}) \quad (165)$$

The calculation of  $\mathbf{x}^{**}$  is mentioned in Equation (162) of the “constrained EI” approach.  $\alpha = 0.1$  is set as the default for a more conservative search direction. Note that for the purpose of improving feasibility knowledge of an optimization problem, we don’t need to search for the entire feasibility contour of  $g^{**}$ , but should focus on promising areas where the objective is small. Consider that there is some uncertainty associated with  $f^{**}$ , we define  $s_f^{**}$  and  $f^{***}$  as follows.

$$f^{***} = f^{**} + 3s_f^{**}$$

where

$$s_f^{**} = \hat{s}_f(\mathbf{x}^{**}) \quad (166)$$

Recall that with a SK model, we have  $Y_f(\mathbf{x}) \sim \mathcal{N}(\hat{y}_f, \hat{s}_f^2)$ . Then, for any point, the probability of its objective to be smaller than  $f^{***}$  can be expressed as:

$$P(Y_f(\mathbf{x}) \leq f^{***}) = \Phi\left(\frac{f^{***} - \hat{y}_f}{\hat{s}_f}\right) \quad (167)$$

Now, we compute the “penalized feasibility-enhancement term” as the product of  $EQIF_g$  and the probability of the objective to be smaller than  $f^{***}$ :

$$pEQIF_g(\mathbf{x}) = EQIF_g(\mathbf{x}) \cdot P(Y_f(\mathbf{x}) \leq f^{***}) \quad (168)$$

To simultaneously search for a better objective as well as enhance the feasibility knowledge of the current best point, we add  $pEQIF_g$  to  $cAEI_f$ , and propose the “Feasibility-enhanced Expected Improvement” criterion (denoted as  $FEI$ ) as follows:

$$\begin{aligned} FEI(\mathbf{x}) &= cAEI_f(\mathbf{x}) + pEQIF_g(\mathbf{x}) \\ &= \left( (f^{**} - \hat{y}_f) \Phi\left(\frac{f^{**} - \hat{y}_f}{\hat{s}_f}\right) + \hat{s}_f \phi\left(\frac{f^{**} - \hat{y}_f}{\hat{s}_f}\right) \right) \cdot \left( 1 - \frac{\tau_f}{\sqrt{\tau_f^2 + \hat{s}_f^2}} \right) \\ &\quad \cdot \Phi\left(\frac{0 - \hat{y}_g}{\hat{s}_g}\right) + \hat{s}_g \cdot \phi\left(\frac{(g^{**} + \Phi^{-1}(\alpha) \cdot \hat{s}_g) - \hat{y}_g}{\hat{s}_g}\right) \\ &\quad \cdot \Phi\left(\frac{f^{***} - \hat{y}_f}{\hat{s}_f}\right) \end{aligned} \quad (169)$$

This approach will be referred to as the “feasibility-enhanced EI” method in the rest of the paper.

To demonstrate the difference between the “constrained EI” approach and the “Feasibility-enhanced EI” approach, we use a 2-dimensional test problem, “G24”, as an example. (Expressions for this test problem are mentioned in the Appendix.) This test problem is shown in Figure 29 (a). It has two disjoint feasible regions that are bounded by thick solid lines; infeasible

regions are shaded in white. There is a global minimum (denoted as the diamond point) located at a feasible region boundary. Figure 29 (b) shows the standard deviation of the noise in the objective, which increases from bottom-left to top-right. Figure 29 (c) shows the standard deviation of the noise in the constraint, which is noisier near the feasible region boundaries.

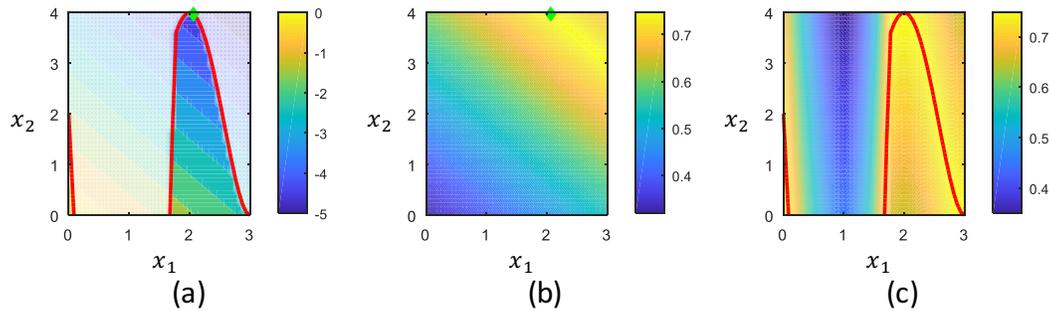


Figure 63. 2D Test problem “G24” . (a) Objective and constraint. Contours represent the objective function; diamond point is the global minimum of the objective function; white-shaded areas are infeasible regions; thick solid lines are feasibility boundaries. (b) Noises in the objective function. Contours represent the standard deviation of the noise in the objective; the diamond point is the global minimum of the objective function. (c) Noises in the constraint function. Contours represent the standard deviation of the noise in the constraint; thick solid lines are feasibility boundaries.

The criteria of  $cAEI_f$ ,  $EQIF_g$ , and  $FEI$  are plotted with contours in Figure 29. Initial SK models are built with 50 LHS points, each with 50 replicated simulation runs. These initial points are denoted with dot points in each plot. Figure 29 (a) shows the surface of  $-cAEI_f$ . The predicted feasible regions are bounded by thick solid lines, with predicted infeasible regions shaded in white. It can be noticed that the promising area lies closely towards the top of the predicted feasible region, which is near to the location of the global optimum. Figure 29 (b) depicts the surface of  $-EQIF_g$ , which is the “feasibility-enhancement” term. The thin solid line is the predicted feasibility

contour that has the value equal to the feasibility of the current best point:  $\hat{y}_g = g^{**}$ . It can be observed that  $EQIF_g$  determines the promising area as the vicinity of the feasibility contour  $\hat{y}_g = g^{**}$ , especially the inner side (i.e., the more feasible side) of the contour. The promising area is bounded from below by the thin dotted line, which is the contour of the objective:  $\hat{y}_f = f^{***}$ . This is caused by the probability term  $P(Y_f(\mathbf{x}) \leq f^{***})$  introduced in  $EQIF_g$ . Therefore, it can be seen that  $EQIF_g$  attempts to improve the feasibility knowledge of the current best point, (or in other words, reduce the uncertainty of  $g^{**}$ ) by searching towards the bounded local areas of the contour  $\hat{y}_g = g^{**}$ . The sum of  $cAEI_f$  and  $EQIF_g$  constitutes the  $FEI$  criterion, which is plotted in Figure 29 (c). Compared to  $cAEI_f$  in Figure 29 (a), the effect of incorporating  $EQIF_g$  in  $FEI$  is to slightly pull back the promising region towards the inner side (i.e., the more feasible side) of the predicted feasible region. As such, it keeps a relatively “restrictive” way (in terms of maintaining feasible) to search for a point with a better objective. More discussion on benefits of using the  $FEI$  criterion is given in Section 8.4.

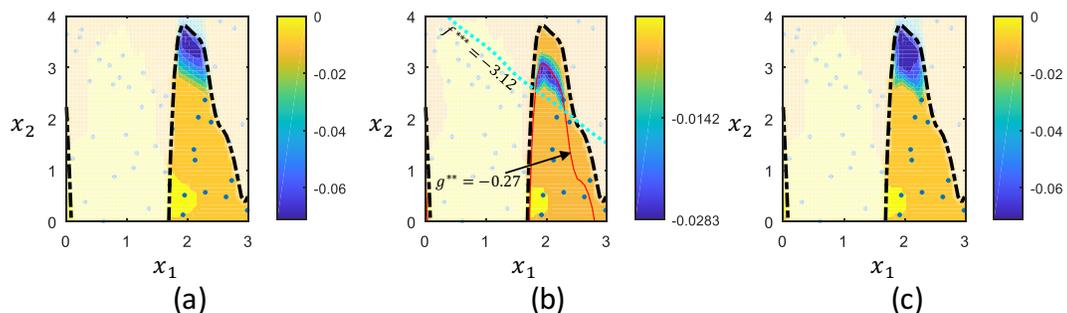


Figure 64. Infill criteria. (a)  $cAEI_f$ ; (b)  $EQIF_g$ ; (c)  $FEI$ . Contours represent the surface of the infill criteria. Dot points are initial point. Thick dashed lines are predicted feasibility boundaries. White-shaded areas are predicted infeasible regions. The thin dotted line in (b) is the contour of the predicted objective equal to  $f^{***}$ . The thin solid line in (b) is the contour of the predicted constraint equal to  $g^{**}$ .

### 8.2.3 Framework of optimization

Both the “constrained EI” method and the “feasibility-enhanced EI” method share the same optimization framework, which is shown in Figure 65.

Step 1 is the initialization step. In Step 1.1, A number of  $10d$  initial points are sampled with a LHS design, each with  $m$  replicated simulation runs. In this work,  $m = 50$  is set as the default. Surrogate models are built separately for the objective and the constraint. For the “constrained EI” method, it requires to build a SK model for the objective ( $SK_f$ ), a SK model for the constraint ( $SK_g$ ), and a Kriging model for the noise in the objective ( $KG_f$ ). For the “feasibility-enhanced EI” method, it requires an additional Kriging model for the noise in the constraint ( $KG_g$ ). In Step 1.2, the current best point  $\mathbf{x}^{**}$  is found by the method in Equation (162). For the “constrained EI” method, it requires to calculate  $f^{**}$ . For the “feasibility-enhanced EI” method, it requires the additional calculation of  $g^{**}$  (Equation (165)),  $f^{***}$  and  $s^{**}$  (Equation (166)).

Step 2 to Step 6 is the adaptive sampling stage. Step 2 calculates an infill criterion to determine the next sample point  $\mathbf{x}'$ . As is mentioned in Section 8.2.2, using Equation (162) in Step 1.2 does not guarantee to always return a feasible point  $\mathbf{x}^{**}$  (among visited points) that satisfies the constraint in (162). In that case, the infill criterion should be switched to find a feasible point without considering the objective. This is achieved by using the Expected Quantile Improvement (denoted as  $EQI_g$ ) criterion developed by Picheny et al. [274] to search for the minimum of the constraint. The  $EQI_g$  criterion is given in Equation (170).

$$EQI_g(\mathbf{x}) = (q^{**} - \hat{y}_Q) \cdot \Phi\left(\frac{q^{**} - \hat{y}_Q}{\hat{s}_Q}\right) + \hat{s}_Q \cdot \phi\left(\frac{q^{**} - \hat{y}_Q}{\hat{s}_Q}\right)$$

where

$$\hat{y}_Q = \hat{y}_g + c_g \cdot \sqrt{\frac{\tau_g^2 \hat{s}_g^2}{\tau_g^2 + \hat{s}_g^2}} \quad (170)$$

$$\hat{s}_Q^2 = \frac{(\hat{s}_g^2)^2}{\tau_g^2 + \hat{s}_g^2}$$

$$q^{**} = \min_{\mathbf{x} \in \mathcal{X}} \hat{y}_g(\mathbf{x}) + c_g \cdot \hat{s}_g(\mathbf{x})$$

Therefore, Step 2 involves a “checking” step on the result from Step 1.2. If no feasible point  $\mathbf{x}^{**}$  can be returned from Step 1.2, then both the “constrained EI” method and the “feasibility-enhanced EI” method will switch to find the next sample point  $\mathbf{x}'$  by maximizing  $EQI_g$  (Equation (170)) for the purpose of finding a feasible point. Otherwise, if a feasible point  $\mathbf{x}^{**}$  is returned from Step 1.2, then the “constrained EI” method determines  $\mathbf{x}'$  by maximizing the  $cAEI_f$  criterion (Equation (161)); the “feasibility-enhanced EI” method finds  $\mathbf{x}'$  by maximizing the  $FEI$  criterion (Equation (169)). After  $\mathbf{x}'$  is determined, Step 3 checks whether  $\mathbf{x}'$  has been visited. In Step 4, if  $\mathbf{x}'$  is a new sample point that hasn't been visited,  $m$  replicated simulation runs are made at  $\mathbf{x}'$ . The calculated  $\bar{y}_f(\mathbf{x}')$ ,  $\hat{V}_f(\mathbf{x}')$ ,  $\bar{y}_g(\mathbf{x}')$ , and  $\hat{V}_g(\mathbf{x}')$  are added to the dataset. Otherwise, if  $\mathbf{x}'$  has already been visited, with the additional  $m$  replicated simulation runs at  $\mathbf{x}'$ , the values of  $\bar{y}_f(\mathbf{x}')$ ,  $\hat{V}_f(\mathbf{x}')$ ,  $\bar{y}_g(\mathbf{x}')$ , and  $\hat{V}_g(\mathbf{x}')$  are re-estimated and the dataset is updated with these values. Step 5.1 update surrogate models. For the “constrained EI” method, the models  $SK_f$ ,  $SK_g$ , and  $KG_f$  are updated. For the “feasibility-enhanced EI” method, an additional model  $SK_g$  is updated. Step 5.2 update the information on the current best point  $\mathbf{x}^{**}$ . The “constrained EI”

method updates  $f^{**}$ , while the “feasibility-enhanced” methods updates the additional terms  $g^{**}$ ,  $f^{***}$  and  $s^{**}$ . The adaptive sampling stage is performed iteratively until Step 6 detects that the total sampling budget  $N_{max}$  is used up.

Step 7 returns the predicted near-optimal solution  $\hat{\mathbf{x}}^*$  and the predicted objective value  $\hat{f}^*$ , which are calculated as follows:

$$\begin{aligned} \hat{f}^* &= \hat{y}_f(\hat{\mathbf{x}}^*) \\ \text{where} \\ \hat{\mathbf{x}}^* &= \arg \min_{\mathbf{x} \in \chi} \hat{y}_f(\mathbf{x}) \\ \text{s. t.} \\ \hat{y}_g(\mathbf{x}) + c_g \cdot \hat{s}_g(\mathbf{x}) &\leq 0 \\ \chi &= \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_k\} \end{aligned} \tag{171}$$

A complete list of differences between the “constrained EI” method and the “feasibility-enhanced EI” method in the optimization framework is shown in Table 26.

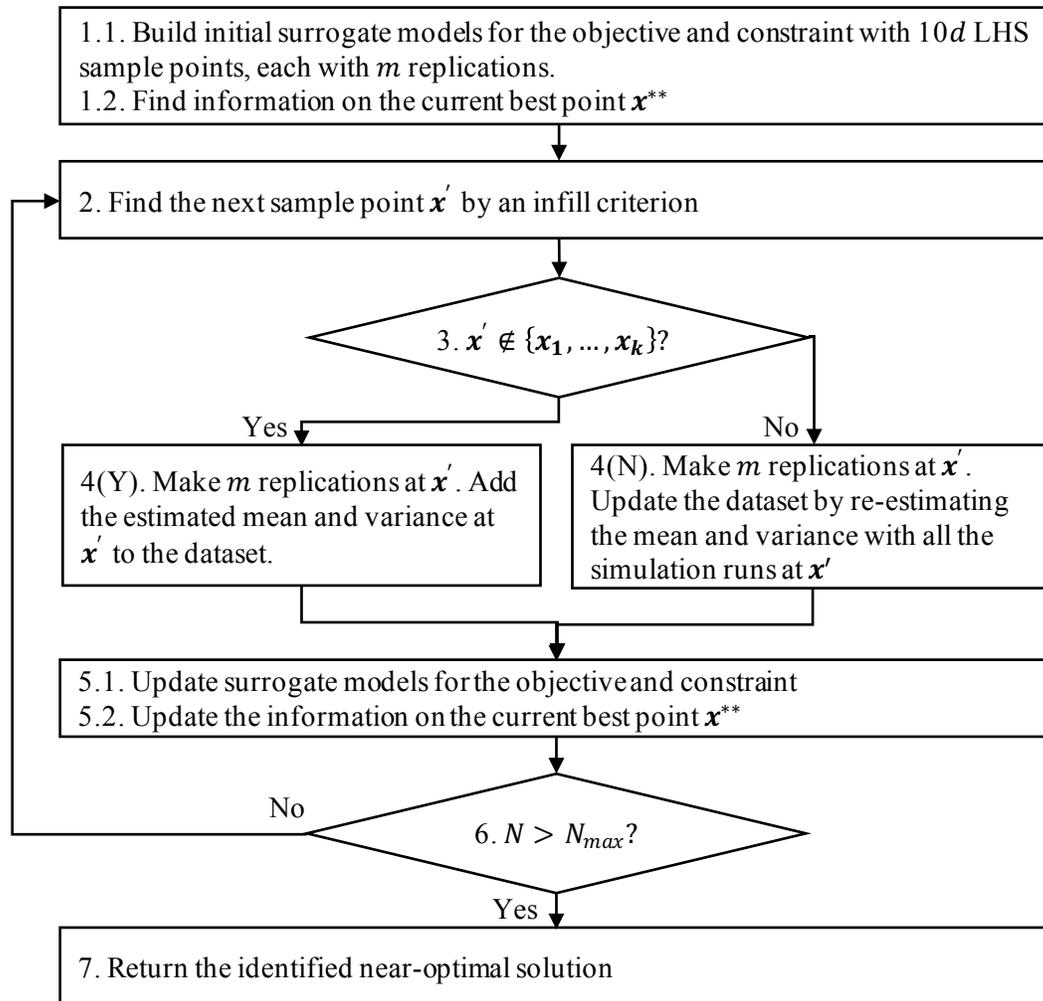


Figure 65. Optimization framework of the Kriging-based approach

Table 26. Differences between the “constrained EI” method and the “feasibility-enhanced EI” method

Steps	“Constrained EI” method	“Feasibility-enhanced EI” method
1.1	Build surrogate models: $SK_f$ , $SK_g$ , $KG_f$	Build surrogate models: $SK_f$ , $SK_g$ , $KG_f$ , $KG_g$
1.2	Calculate $\mathbf{x}^{**}$ and $f^{**}$	Calculate $\mathbf{x}^{**}$ , $f^{**}$ , $g^{**}$ , $f^{***}$ , and $s^{**}$

2	If a feasible $\mathbf{x}^{**}$ is returned from Step 1.2, $\mathbf{x}' = \arg \max cAEI_f(\mathbf{x})$	If a feasible $\mathbf{x}^{**}$ is returned from Step 1.2, $\mathbf{x}' = \arg \max FEI(\mathbf{x})$
5.1	Update surrogate models: $SK_f, SK_g, KG_f$	Update surrogate models: $SK_f, SK_g, KG_f, KG_g$
5.2	Update $\mathbf{x}^{**}$ and $f^{**}$	Update $\mathbf{x}^{**}, f^{**}, g^{**}, f^{***}$ , and $s^{**}$

### 8.3 Computational studies

#### 8.3.1 Test problems and additional competitive solvers

The performance of the two Kriging-based approaches are compared by using eight standard test problems which have been widely adopted in the optimization literature. The dimensionality of the problems ranges from 2 to 5. Note that only low-dimensional problems are chosen because it is known that Kriging-based algorithms are mostly viable for problems with not too large dimensions [350]. Some of the test problems have multimodal characteristic in the objective and disjoint feasible regions, which increases the difficulty of finding the feasible global optimal solution. A summary on these test problems is given in Table 22, where  $R_g$  is defined as:  $R_g = \max |g(x)|$ ,  $R_f$  is defined as:  $R_f = \max f(x) - \min f(x)$ . Functions of the test problems are mentioned in the Appendix.

Table 27. Summary of the test problems

#	Test Problems	$d$	$f$ is multimodal	$g$ has disjoint feasible regions	# of global optima	$R_f$	$R_g$
1	Gomez	2	yes	yes	1	4.2650	3.0000
2	Constrained Branin	2	yes	no	2	5.9236	5.5000

3	New Branin	2	no	yes	1	5.6250	5.8350
4	Sasena	2	no	yes	1	1.2500	4.0000
5	qcp4	3	-	-	1	10.0000	6.0000
6	G4	5	-	-	1	4.9573	5.0430
7	G24	2	no	yes	1	5.0000	5.0000
8	Angun	2	no	no	1	5.0000	5.0000

In addition to the proposed Kriging-based algorithms, two other competitive solvers are added for a comparison purpose: NOMAD (Nonlinear Optimization by Mesh Adaptive Direct Search) [359] and ISRES (Improved Stochastic Ranking Evolution Strategy) [360]. These two algorithms are selected because they are suitable for constrained black-box optimization problems and have been tested in other global optimization literatures [64,361].

### 8.3.2 Noise scenarios

In this paper, we consider stochastic systems where the inherent noises have heteroscedastic variances. Thus, for each test problem, two noise scenarios are constructed. In the “easy” noise scenario,  $\xi_f(\mathbf{x})$  decreases linearly as the objective value  $f(\mathbf{x})$  approaches to the feasible global optimum  $f^*$ ;  $\xi_g(\mathbf{x})$  decreases linearly as the constraint value  $g(\mathbf{x})$  gets close to the feasible region boundary ( $g(\mathbf{x}) = 0$ ). As such,  $\xi_f(\mathbf{x})$  and  $\xi_g(\mathbf{x})$  introduce small uncertainties to the identification of promising areas where  $f(\mathbf{x})$  is close to  $f^*$  as well as the determination of whether a point is feasible. On the other hand, in the “hard” noise scenario,  $\xi_f(\mathbf{x})$  increases linearly as the objective value  $f(\mathbf{x})$  approaches to  $f^*$ ;  $\xi_g(\mathbf{x})$  increases linearly as the constraint value  $g(\mathbf{x})$  gets close to the feasible region boundary. In this case,  $\xi_f(\mathbf{x})$  and  $\xi_g(\mathbf{x})$  cause large uncertainties to the search of promising areas where  $f(\mathbf{x})$  is close to  $f^*$  as well as determining feasibility near feasible region

boundary. Similar settings of constructing noise functions have been considered in Refs. [278,356,358]. The expressions for  $\xi_f(\mathbf{x})$  and  $\xi_g(\mathbf{x})$  are included in Table 28.

Table 28. Expressions for the noise functions

Noise scenarios	Noise functions
“easy” noise	$\xi_f(\mathbf{x}) = (0.15 - 0.05) f(\mathbf{x}) - f^*  + 0.05R_f$
	$\xi_g(\mathbf{x}) = (0.15 - 0.05) g(\mathbf{x}) - 0  + 0.05R_g$
“hard” noise	$\xi_f(\mathbf{x}) = (0.05 - 0.15) f(\mathbf{x}) - f^*  + 0.15R_f$
	$\xi_g(\mathbf{x}) = (0.05 - 0.15) g(\mathbf{x}) - 0  + 0.15R_g$

### 8.3.3 Performance measures

For a stochastically constrained optimization problem, the quality of a returned near-optimal solution depends on the following three aspects: (A) whether the returned solution is feasible at a specified tolerance; (B) how accurate the algorithm is in “locating” the optimal solution; (C) how accurate the algorithm is in “predicting” the objective value.

By selecting a tolerance of  $10^{-6}$  as the default, Quality (A) can be measured by checking whether the following condition is satisfied.

$$g(\hat{\mathbf{x}}^*) < 10^{-6} \quad (172)$$

Quality (B) can be measured by the difference between  $f(\hat{\mathbf{x}}^*)$  and  $f^*$ , which is denoted as *gap1* and expressed as follows.

$$gap1 = |f(\hat{\mathbf{x}}^*) - f^*| \quad (173)$$

Quality (C) can be measured by the difference between  $\hat{y}_f(\hat{\mathbf{x}}^*)$  and  $f(\hat{\mathbf{x}}^*)$ , which is noted as *gap2* with the following expression.

$$gap2 = |\hat{y}_f(\hat{\mathbf{x}}^*) - f(\hat{\mathbf{x}}^*)| \quad (174)$$

Similar performance measures have been adopted in Refs. [278,356].

#### 8.3.4 Implementation details

Computational studies are performed in Matlab 2017a. Implementations of NOMAD and ISRES are from the OPTimization Interface (OPTI) Matlab toolbox available at ([www.inverseproblem.co.nz/OPTI/](http://www.inverseproblem.co.nz/OPTI/)). SK models are built on the basis of the work by Ankenman et al. [162] with Matlab codes available at ([stochastickriging.net/](http://stochastickriging.net/)). Kriging models are built with the DACE toolbox [226] and the Matlab codes are available at ([www2.imm.dtu.dk/projects/dace/](http://www2.imm.dtu.dk/projects/dace/)). For each Kriging-based algorithm, the infill criterion is maximized using “fmincon” in Matlab, which implements a sequential quadratic programming algorithm. A multi-start strategy is used to increase the probability of finding the global optimum. In specific, a number of  $10d$  LHS points are used every time the  $cAEI_f$  or  $FEI$  is maximized. In each test, 80 iterations of the adaptive sampling stage are used. Therefore, the total sampling budget for each test is:  $N_{max} = (10d \text{ initial points} + 80 \text{ iterations}) \times 50 \text{ replications}$ .

To account for the randomness in the initial LHS sampling and stochastic simulations, we make 100 mega-repetitions for every algorithm: an algorithm is run 100 times for the two noise scenarios of every test problem. For a fair comparison, each algorithm shares the same 100 sets of initial sample points (same noisy observations at the same sample points) for each noise scenario.

## 8.4 Results

### 8.4.1 Capabilities of returning a truly feasible solution

We first examine different algorithms’ capabilities of returning a near-optimal solution that is actually feasible, which corresponds to the Quality (A) mentioned in Section 8.3.3. This is reflected

by the number of mega-repetitions (among the total number of 100) of each algorithm that fulfill the requirement  $g(\hat{\mathbf{x}}^*) < 10^{-6}$ . The results are summarized in Table 29.

Among the four algorithms, NOMAD performs generally worse than the other three algorithms. It is noticed that NOMAD returns a significantly smaller number of  $\hat{\mathbf{x}}^*$  that satisfy the feasibility requirement, and it happens in both the “easy” and “hard” noise scenarios. The only exception that NOMAD shows competitiveness is in the test problem “constrained Branin”, which is the only problem where the global optimal solution  $\mathbf{x}^*$  lies inside the feasible region and the constraint is not active at  $\mathbf{x}^*$ . This characteristic makes “constrained Branin” the easiest one in terms of returning a truly feasible solution because the promising area surrounding  $\mathbf{x}^*$  is mostly feasible and  $\mathbf{x}^*$  is relatively far from infeasible regions.

For the ISRES algorithm, it shows a competitive performance to the two Kriging-based algorithms in two test problems (i.e., “Gomez” and “constrained Branin”) for both noise scenarios. Further, ISRES exceeds other algorithms in “G24” for the “easy” noise scenario, and shows comparable to “feasibility-enhanced EI” for the “hard” noise scenario. However, notable disadvantages of using ISRES are found for “new Branin”, “Sasena”, and “G4” in their “easy” noise scenarios. Such disadvantages become even more significant in the “hard” noise scenarios of these test problems. In addition, two more test problems (“qcp4” and “Angun”) become more challenging in their “hard” noise scenarios for ISRES than the two Kriging-based algorithms.

Comparing the “constrained EI” method and the “feasibility-enhanced EI” method, it is found that, for “easy” noise scenarios, it is more favorable to use “feasibility-enhanced EI” for the

“Sasena” test problem. For “hard” noise scenarios, significant advantages of using the “feasibility-enhanced EI” are found for “new Branin” , “Sasena” , and “G24” .

For each algorithm, the total numbers of mega-repetitions that satisfy  $g(\hat{\mathbf{x}}^*) < 10^{-6}$  are listed at the end of Table 29. It can be found that “hard” noise scenarios generally cause more difficulties than “easy” noise scenarios for all the algorithms. By using the “constrained EI” method as a reference, for “easy” noise scenarios, NOMAD is significantly worse than ISRES, which is slightly less competent than “constrained EI” which is slightly worse than “feasibility-enhanced EI” . For “hard” noise scenarios, although this rank remains the same, the differences between ISRES, “constrained EI” , and “feasibility-enhanced EI” get sufficiently enlarged.

In summary, in terms of capabilities of returning a truly feasible solution, the two Kriging-based algorithms are generally more competitive than NOMAD and ISRES, with NOMAD being the worst. For some test problems, the “feasibility-enhanced EI” method shows improved performance than the “constrained EI” method. Such an improvement becomes more significant for “hard” noise scenarios where the stochasticity in the constraint introduces a larger uncertainty near feasible region boundaries. This type of noise structures can make problems especially difficult to solve when the global optimum is located at an active constraint.

To further understand in what cases the “feasibility-enhanced EI” method becomes significantly useful, we check the landscapes of constraints for “new Branin” and “Sasena” . These two problems are selected because they appear to be most difficult for all four algorithms according to the data in Table 29, while “feasibility-enhanced EI” tends to give a significantly better performance than the other three algorithms. Figure 29 shows the contours of constraint functions

for “new Branin” (Figure 29 (a)) and “Sasena” (Figure 29 (b)), where thick solid lines are feasible region boundaries. It can be observed that both problems have feasible region boundaries located in a relatively flat neighborhood. In such cases, a relatively large noise variance in the vicinity of feasible region boundaries can drastically degrade the estimation of constraint values. When the global optimum is located at (or near) the feasible region boundary, it becomes greatly difficult for all algorithms to determine the current best point as a truly feasible one. In such situations, the “feasibility-enhanced EI” method can be especially beneficial because it explicitly accounts for improving the knowledge on feasibilities in the infill criterion as it searches for the next sample point.

Table 29. Total numbers of returned near-optimal solutions being feasible

#	Test problems	"Easy" noise				"Hard" noise			
		NOMAD	ISRES	$cAEI_f$	$FEI$	NOMAD	ISRES	$cAEI_f$	$FEI$
1	Gomez	84	100	100	100	82	98	98	97
2	constrained Branin	98	100	100	100	99	100	100	100
3	new Branin	14	77	84	86	16	54	76	87
4	Sasena	5	63	72	86	1	22	47	71
5	qcp4	88	97	95	99	82	84	95	95
6	G4	82	90	99	95	70	79	91	93
7	G24	25	88	80	81	22	73	63	73
8	Angun	51	87	88	94	50	76	93	92
<b>Total numbers</b>		447	702	718	741	422	586	663	708
<b>Overall difference</b>		-271	-16	$Ref_{easy} +23$		-241	-77	$Ref_{hard} +45$	

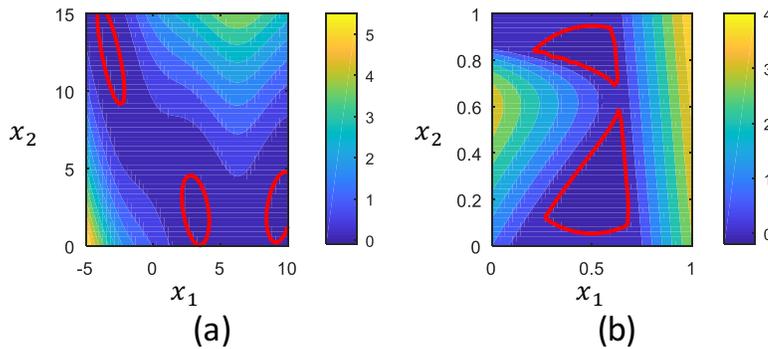


Figure 66. Contours of constraint functions. (a) new Branin; (b) Sasena. Thick solid lines are feasible region boundaries

#### 8.4.2 Accuracies of “locating” the feasible global optimum

As is discussed in Section 8.3.3,  $gap1$  can be computed to measure how accurate an algorithm can “locate” the global optimum solution. The results are shown with boxplots of  $\log_{10}(gap1)$  at the last iteration of each algorithm (Figure 67). Note that the boxplots are created only on the basis of those mega-repetitions that return a truly feasible solution satisfying  $g(\hat{\mathbf{x}}^*) < 10^{-6}$ . In Figure 67, each subplot (corresponding to one test problem) consists of two groups of boxplots: the left group of four boxplots shows algorithms’ performances in the “easy” noise scenario; the right group of four boxplots is for the “hard” noise scenario.

Compared to the two Kriging-based methods, NOMAD has relatively larger values of  $\log_{10}(gap1)$  for six test problems in both “easy” and “hard” noise scenarios. These test problems include “Gomez”, “constrained Branin”, “new Branin”, “Sasena”, “qcp4”, “G4”. Among these test problems, “Sasena” in the “hard” noise scenario only has a boxplot as a short line for the NOMAD algorithm. This is because the NOMAD algorithm only has one

mega-repetition (among total 100 runs) that satisfies  $g(\hat{\mathbf{x}}^*) < 10^{-6}$ , which is reflected in Table 29. Although this run has a smaller  $\log_{10}(\text{gap1})$  than other algorithms in the “Sasena” - “hard” noise scenario, it is only a rare case and does not prove NOMAD to be effective in this test problem. For the test problem “G24” in both noise scenarios, NOMAD shows a competitive performance to Kriging-based methods, but the boxplots for NOMAD in “G24” have wider spans. This indicates that NOMAD can be less capable of returning a consistently good near-optimal solution than the Kriging-based methods. For the test problem “Angun” in the “easy” noise scenario, NOMAD has a performance comparable to Kriging-based methods. However, in the “hard” noise scenario, NOMAD becomes much worse.

The ISRES algorithm performs generally worse than other three algorithms in almost all test problems. Only in the “hard” noise scenario of “Sasena”, ISRES has a boxplot that are close to Kriging-based methods. However, recalling the data in Table 29, ISRES only has 22 mega-repetitions that return a truly feasible solution, which is significantly fewer than that of the “constrained EI” approach (47 feasible) and the “feasibility-enhanced EI” (71 feasible). Therefore, ISRES is still far less competent than Kriging-based algorithms for “Sasena” .

Comparing the two Kriging-based methods, it is found that the “constrained EI” method performs better than “feasibility-enhanced” method for “Gomez” and “qcp4” in both noise scenarios and for “Sasena” in the “easy” noise scenario. For the rest test problems, these two Kriging-based methods show performances that are almost indistinguishable.

In summary, for almost all test problems, the Kriging-based methods are sufficiently better at “locating” the global optimal solution than the NOMAD and ISRES when using the same

sampling budget. Only in rare cases (e.g., “G24” , “Sasena” ), NOMAD or ISRES shows a competitive performance to Kriging-based algorithms. However, in such cases, given the fact that NOMAD and ISRES can only return much fewer truly feasible solutions than Kriging-based methods, it is still less reliable of using NOMAD or ISRES than using Kriging-based methods for the optimization of stochastic systems. The advantages of using Kriging-based methods are expected because direct-search methods (e.g., NOMAD) and evolutionary algorithms (ISRES) generally require more sample points to find a near-optimal solution that is close enough to the global optimal solution [361]. In addition, it is found that in a few cases, the “feasibility-enhanced EI” method is can be slightly less accurate than the “constrained EI” method in “locating” the global optimum.

To further understand the difference between “constrained EI” and “feasibility-enhanced EI” , we plot the profiles of  $\log_{10}(gap1)$  versus iterations for each test problem using these Kriging-based algorithms (Figure 68). Similar with the boxplots, only those mega-repetitions that result in a truly feasible solution are considered when making the profile plots. In Figure 68, the thick solid line represents the median profile of “constrained EI” ; the thick dashed line represents the median profile of “Feasibility-enhanced EI” . The lower and upper thin solid lines represent the 1<sup>st</sup> quartile and 3<sup>rd</sup> quartile profile of “constrained EI” ; the lower and upper thin dashed lines represent the 1<sup>st</sup> quartile and 3<sup>rd</sup> quartile profile of “feasibility-enhanced EI” .

It is observed that profiles of “feasibility-enhanced EI” generally show a slightly slower decrease in  $\log_{10}(gap1)$  than “constrained EI” , especially for “Gomez” and “qcp4” . This can be attributed to the “penalized feasibility-enhancement term” ( $pEQIF_g$  in Eqaition (168), which requires to search near a bounded vicinity of the feasibility contour of the current best point. The

addition of  $pEQIF_g$  causes a small portion of sampling budget to be allocated to locally increase the knowledge of the feasibility, and thus slightly slows down the progress of searching for a point with a better objective. This is a compromise that we need to make as we attempt to maintain feasible while approaching to the optimum of stochastically constrained problems. For black-box systems where the noise variance in the constraint is unknown, it is worthwhile of considering “feasibility-enhanced EI” approach because it makes the algorithm less vulnerable to the stochasticity in the constraint and thus more reliable than the “constrained EI” approach in returning a truly feasible near-optimal solution.

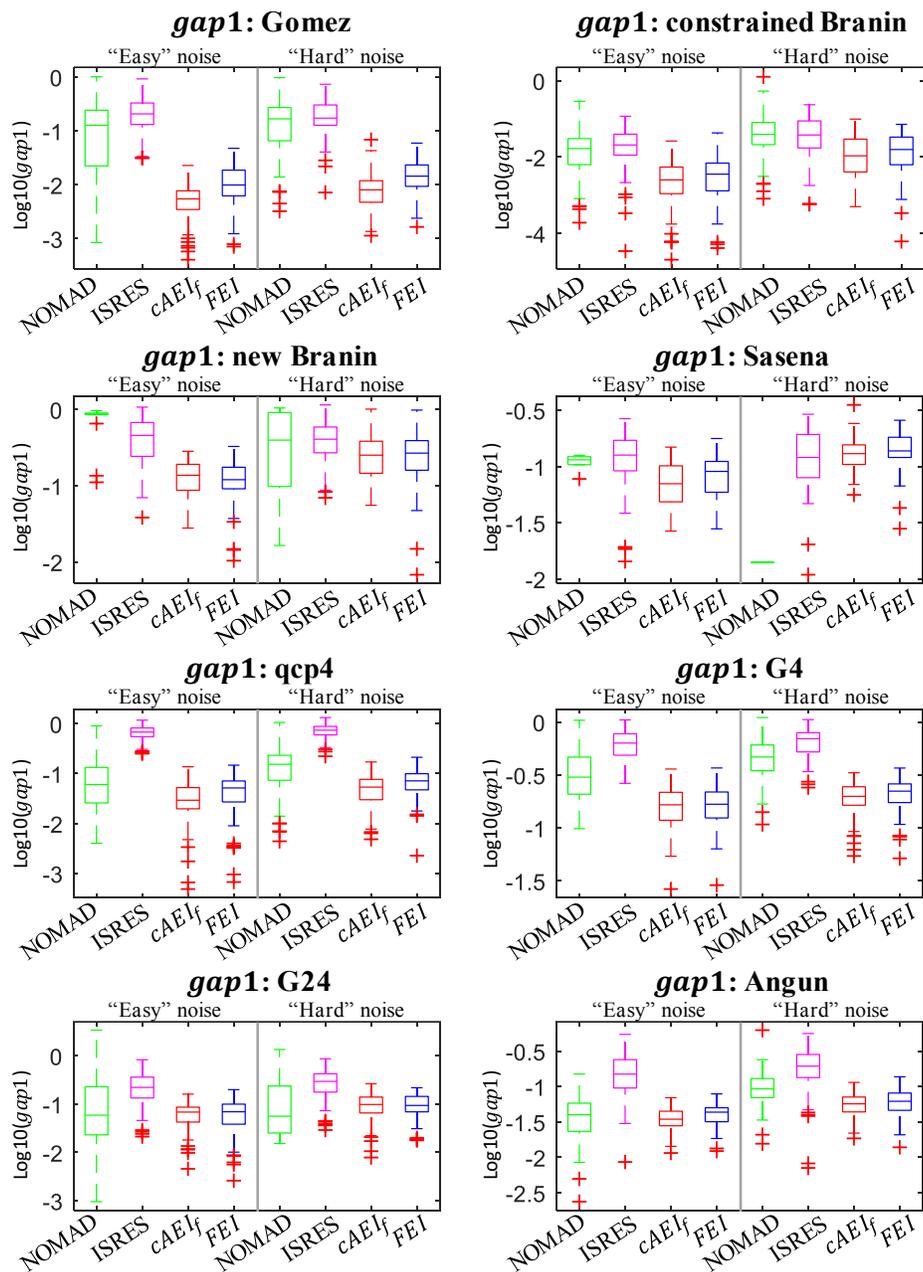


Figure 67. Boxplots of  $\text{log}_{10}(gap1)$ .

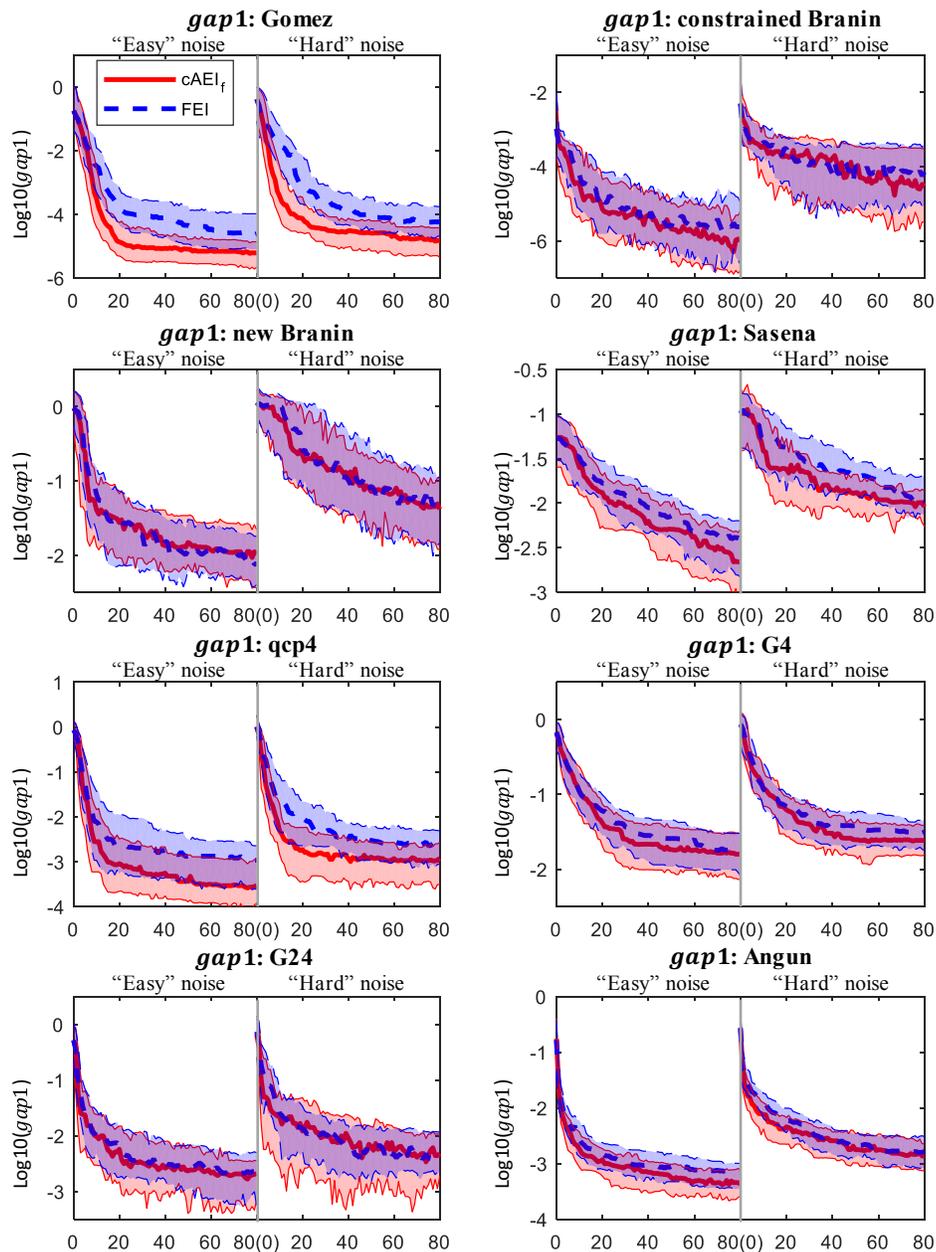


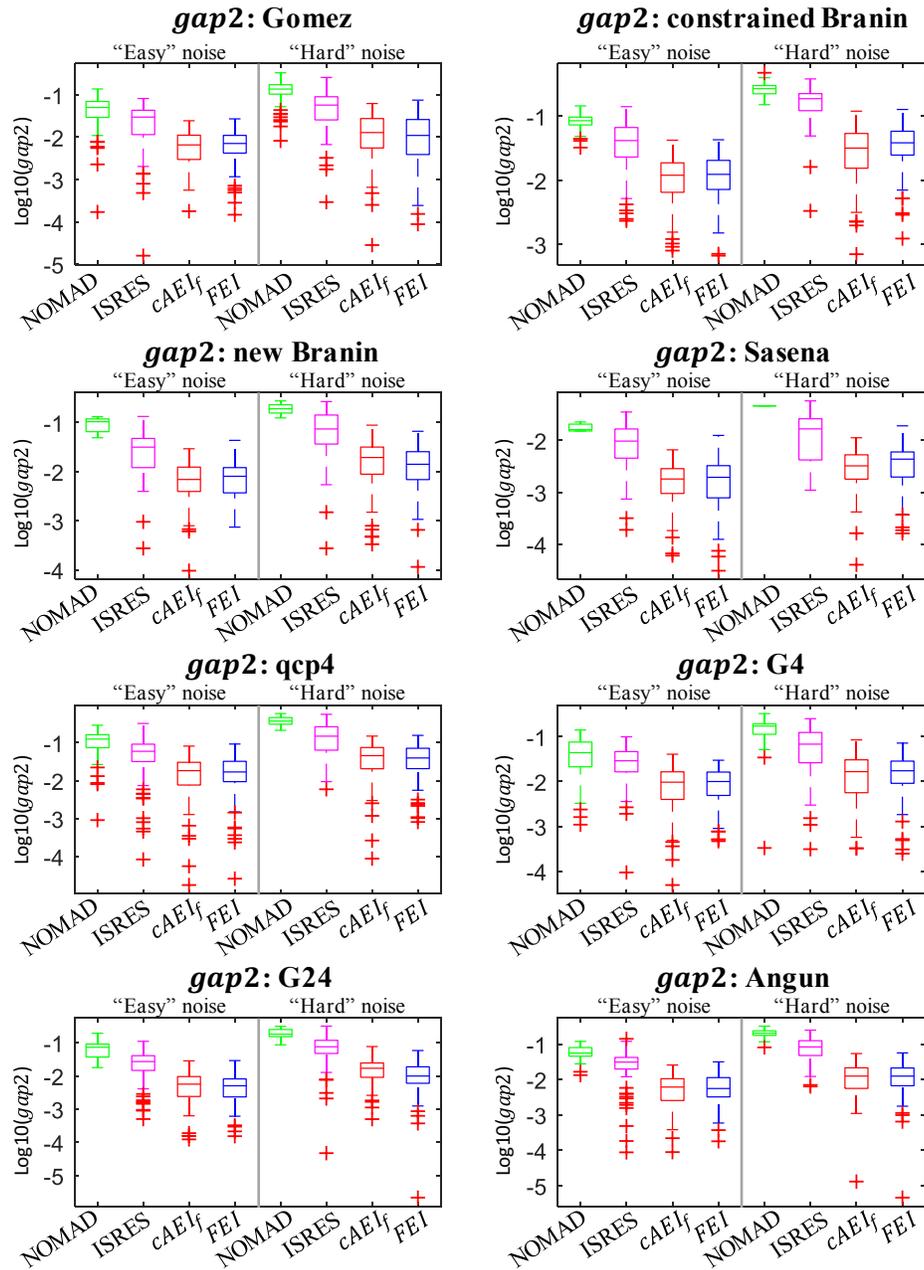
Figure 68. Profiles of  $\log_{10}(\text{gap1})$ . Thick solid lines represent median profiles for “constrained EI” ; thick dashed line represent median profiles for “Feasibility-enhanced EI” ; lower and upper thin solid lines represent the 1<sup>st</sup> quartile and 3<sup>rd</sup> quartile profiles for “constrained EI” ; lower and

upper thin dashed lines represent the 1<sup>st</sup> quartile and 3<sup>rd</sup> quartile profiles for “feasibility-enhanced EI” .

#### 8.4.3 Accuracies of “predicting” the objective value at the near-optimal solution

For each algorithm, *gap2* is calculated to measure the accuracy of “predicting” the objective value. Figure 69 shows the boxplots of  $\log_{10}(\text{gap2})$  at the last iteration of each algorithm. Similar with *gap1*, the calculation of *gap2* only considers those mega-repetitions that return a truly feasible solution satisfying  $g(\hat{\mathbf{x}}^*) < 10^{-6}$ . In Figure 69, each subplot (corresponding to one test problem) consists of two groups of boxplots: the left group shows boxplots for the “easy” noise scenario; the right group is for the “hard” noise scenario.

For all test problems, the performances of NOMAD and ISRES are much worse than the two Kriging-based methods, and there is no obvious difference between “constrained EI” and “feasibility-enhanced EI” . This can be attributed to the different methods that algorithms use to estimate the near-optimal objective value: NOMAD and ISRES use sampling average  $\bar{y}_f(\hat{\mathbf{x}}^*)$ , while Kriging-based algorithms use the SK predictor  $\hat{y}_f(\hat{\mathbf{x}}^*)$ . Mathematically, Chen and Zhou [262] demonstrated that the uncertainty associated with  $\hat{y}_f(\hat{\mathbf{x}}^*)$  is upper bounded by the uncertainty associated with  $\bar{y}_f(\hat{\mathbf{x}}^*)$ . Therefore, it is more accurate to use  $\hat{y}_f(\hat{\mathbf{x}}^*)$  than  $\bar{y}_f(\hat{\mathbf{x}}^*)$  to estimate the near-optimal objective. Such results further indicate the advantages of using SK models to approximate stochastic systems in an optimization framework.

Figure 69. Boxplots of  $\text{log}_{10}(\text{gap}2)$ .

### 8.5 Optimization of a continuous pharmaceutical manufacturing process

After realizing the advantages of the Kriging-based algorithms, we apply them to a pharmaceutical manufacturing case study. The goal is to optimize the operation of a Continuous Direct Compaction process which is used to produce drug products in a solid-oral dosage form. The process flowsheet is shown in Figure 29. Three feeders are used to continuously feed powder components (including Active Pharmaceutical Ingredient (API), Excipient, and Lubricant). A co-mill is used to de-lump any large-sized chunks in API and Excipient. A blender is used to continuously mix the three raw materials. The mixtures are sent to a tablet press unit where tablet products are made. A more detailed description on the mathematical models of this process is given in Ref. [362].

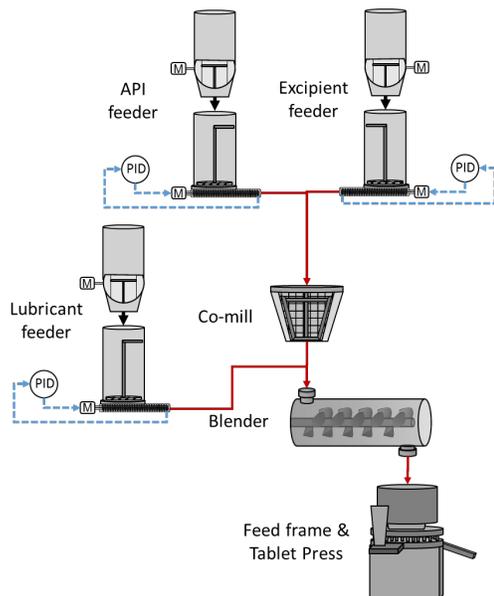


Figure 70. Flowsheet of a Continuous Direct Compaction process

In this work, the optimization problem is formulated with Problem (175). The objective is to minimize the total operation cost per day, which involves the material, utility and waste costs. For a pharmaceutical manufacturing process, it is critical to reduce wasted products. Therefore, a constraint is added to enforce that wasted products can only be generated during the starting stage of the process. Two decision variables are considered: the API flow rate set point ( $FR_{API}$ ) and the

refilling strategy of the API feeder ( $RS_{API}$ ).  $FR_{API}$  controls how fast API is fed to the system.  $RS_{API}$  is defined as the fill level of the API feeder where a refilling operation is conducted. For the operation of a feeder, it requires a certain amount of materials to be refilled to the equipment periodically to prevent the feeder gets empty. Therefore,  $RS_{API}$  determines the frequency of refilling.  $FR_{API}$  and  $RS_{API}$  are selected as the decision variables because they are influential to the calculation of total costs as well as the generation of wasted products caused by temporal process variations [362]. To account for the uncertainty in estimations of the total cost and the time of generating wasted products, two noise terms ( $\varepsilon_f$  and  $\varepsilon_g$ ) are added respectively to the objective and constraint. We further assume that  $\varepsilon_f$  and  $\varepsilon_g$  are subject to normal distributions, with the variances dependent on the decision variables:  $\varepsilon_f \sim \mathcal{N}(0, \sigma_f^2(FR_{API}, RS_{API}))$ ,  $\varepsilon_g \sim \mathcal{N}(0, \sigma_g^2(FR_{API}, RS_{API}))$ . This assumption is plausible because it was experimentally found that  $FR_{API}, RS_{API}$  can introduce a normally distributed process variation [17], which is the main source of the uncertainties in the objective and constraint function. However, further experimental studies are needed to verify this assumption.

$$\begin{aligned}
 & \min_{FR_{API}, RS_{API}} \text{cost}_{total} = \text{cost}_{material} + \text{cost}_{utility} + \text{cost}_{waste} + \varepsilon_f \\
 & \text{s. t.} \\
 & \text{WasteProductTime}(FR_{API}, RS_{API}) + \varepsilon_g \leq \text{StartupTime} \quad (175) \\
 & RS_{API}^{lb} \leq RS_{API} \leq RS_{API}^{ub} \\
 & FR_{API}^{lb} \leq FR_{API} \leq FR_{API}^{ub}
 \end{aligned}$$

The flowsheet model is built in the gPROMS simulation platform. The interface “gO:MATLAB” is used to transfer information between gPROMS and Matlab. Each Kriging-based algorithm uses a total sampling budget of 100 sample points (20 initial points + 80 iterations of adaptive sampling), each point with 50 replicated simulation runs. The returned near-optimal solutions and the predicted objective values are listed in Table 30. The results suggest a small value for  $FR_{API}$ ,

which is beneficial to reduce the total cost. Further, both algorithms suggest a refilling frequency in a mid-high range. According to experimental studies, this is helpful to reduce temporal process variations [254] and thus reduce potential wasted products. The estimated total costs from both algorithms are over 98,000 \$/day, and the values are close.

Table 30. Optimization results for the pharmaceutical case study

		$cAEI_f$	$FEI$
$FR_{API}$	[kg/hr]	2.8691	2.8622
	(lb~ub)	(2.85~3.15)	(2.85~3.15)
$RS_{API}$	[%]	64.54	54.49
	(lb~ub)	(10~70)	(10~70)
Predicted	total		
cost [\$/day]		98,193	98,175

## 8.6 Summary and future work

A Kriging-based optimization framework is used to solve stochastically constrained simulation optimization problems. Stochastic Kriging is used to model black-box objective and constraint functions which have inherent noises with heteroscedastic variances. A “constrained Expected Improvement” infill criterion is adapted for stochastic systems to seek for the next sample point. To better account for the stochasticity in the constraint, we propose a “feasibility-enhanced Expected Improvement” infill criterion that explicitly considers to improve feasibility knowledge while searching for new sample points with a better objective value. The two Kriging-based algorithms are shown to perform better than a direct search method (NOMAD) and an evolutionary algorithm (ISRES) in three aspects: in most cases, Kriging-based methods are (1) more reliable in returning a truly feasible solution; (2) more accurate in “locating” the feasible global optimal

solution; (3) more accurate in “predicting” the objective value. Additionally, the comparison between the two Kriging-based methods reveals that the “feasibility-enhanced Expected Improvement” approach is more robust than “constrained Expected Improvement” to return a truly feasible solution when the stochasticity in the constraint increases, although the former approach may result in a slightly slower progress in approaching to the feasible global optimum. The two Kriging-based algorithms are further applied to optimize the operations of a pharmaceutical manufacturing process.

For future work, there are still several aspects that are worth further attention for the Kriging-based algorithms. First, theoretical proof on the convergence of the algorithms needs to be investigated. This is helpful to understand the average performance of the algorithms. Second, the current Kriging-based methods are based on the assumption that stochastic systems have noise terms subject to a normal distribution. Further studies are needed to expand the algorithms to be suitable for simulations with other types of stochasticity. Finally, the proposed Kriging-based algorithms are developed for simulation optimization problem with one constraint. They still need to be improved to address multi-constraint problems. In Refs. [344,350], the “constrained Expected Improvement” method was used to address problems with  $n$  constraints by multiplying  $n$  probabilities of being feasible for each constraint. However, such methods become less effective when there are many active constraints [350]. Alternatively, we envision that a more effective way is to transform multiple constraints into a single new constraint. This can be achieved by defining the new constraint as the maximum violation of all the constraints. Similar approaches have been used in feasibility analysis problems in Ref. [64,261] for deterministic models. However, for multiple stochastic constraints, this transformation will cause the newly defined constraint to possibly have a completely different (and unknown) distribution. Therefore, the effectiveness of this rough idea should depend on the second future work that we mentioned above, which is to

expand the Kriging-based algorithms to stochastic systems with a diverse range of noise distributions.

## V. Conclusions

## 9 Conclusions and Future Work

In this thesis, we investigate to enhance pharmaceutical process knowledge via simulation-based process analysis and optimization approaches. A framework of global sensitivity, feasibility, and optimization is proposed to have a comprehensive understanding of the process and provide guide to improve the process performance. This framework is applied to a direct compaction process where deterministic simulations are available to approximate process behaviors. In order to further expand the applications of such framework, we have developed process analysis and optimization approaches that are suitable for stochastic systems. These approaches can be useful when stochastic simulations are adopted to model the random behaviors of a real-life pharmaceutical process.

Below, we outline the major contributions made in this thesis, and provide some future directions that are worth research attention.

### 9.1 Major contributions

- Literature reviews are given respectively on process analysis (Chapter 2) and optimization (Chapter 3) for pharmaceutical manufacturing processes. These two reviews can be used as a guide on choosing the proper method in a real-life case to conduct simulation-based process analysis and optimization. This choice depends on the problem dimensionality, computational cost (and sampling budget), transparency of the model (i.e., whether closed-forms model functions are available), and types of constraints of the problem.
- For deterministic systems, we focus on developing and applying efficient approaches that are suitable for computationally expensive simulations. Specifically, a Radial Basis Function (RBF-based) adaptive sampling approach is developed for feasibility analysis

problems (Chapter 4). This method proves to be more efficient and accurate than a competitive Kriging-based algorithm. Additionally, a process analysis and optimization framework is proposed and demonstrated with a direct compaction process (Chapter 5). This framework provides a systematic way to extract process knowledge via simulations.

- For stochastic systems, we first address the feasibility analysis problems by comparing three Kriging-based approaches (Chapter 6). Among these approaches, two of them prove to be more reliable for noisy systems, as well as more accurate in identifying non-convex feasible regions. Then, we propose a “one-stage” algorithm for stochastically unconstrained optimization problems (Chapter 7). This algorithm is more capable than a “two-stage” algorithm in “locating” the global optimal solution when the noise level is high in promising regions. Finally, we solve stochastically constrained optimization problems by developing a Kriging-based algorithm with feasibility enhancements (Chapter 8). This algorithm shows more robustness than competitive solvers in returning a feasible and near-optimal solution in a variety of stochastic systems.

## 9.2 Future work

With respect to deterministic systems, there are a few directions that require further research, that are listed as follows.

- Currently, the global sensitivity analysis and feasibility analysis are conducted separately in two stages. In this case, these two types of analysis each require an independent set of data samples, which can be less efficient for the analysis of computationally expensive simulations. There still lacks a “bridge” that can connect sensitivity analysis and feasibility analysis, and make it possible for them to share data samples. A possible way to build this “bridge” is by using a common surrogate model for both the sensitivity

analysis and the feasibility analysis. Further research is needed to choose an appropriate model, and choose the sampling plan.

- The proposed RBF-based method for feasibility analysis can be computationally expensive as the problem dimension increases. It would be interesting to combine this approach with model-reduction (or dimension-reduction) techniques, and apply it to higher-dimensional problems. An attempt has been made recently to combine the Partial Least Squares (PLS) approach to the RBF-based algorithm.
- The visualization of design space becomes an issue as the problem dimension increases. It requires additional research on displaying the high-dimensional space that can show interactions between input factors when characterizing the design space.

With respect to stochastic systems, the suggested future research directions are mentioned as follows.

- The stochastic Kriging used in the feasibility analysis and optimization algorithms requires a pre-defined number of replications at each sample point. Currently, there still lacks a proper way of choosing this number depending on different problems. This problem has been addressed for systems with discrete variables. It would be interesting if we can adapt those approaches to problems with continuous variables.
- The currently developed algorithms are still suitable for relatively small-scale problems. More research is needed to extend the algorithms to larger-scale problems.
- We have only considered stochastic systems where the inherent noise is subject to a normal distribution. This can be a limitation for a broader range of applications when other types of stochasticity are existent.
- The proposed simulation optimization algorithms can be used in other applications than pharmaceutical processes, including supply chain management, inventory management

problems, control under uncertainties. It would be interesting if we can gather more types of problems to test our proposed algorithm.

### **Acknowledgment of previous publications**

Several chapters of the thesis have been published or being prepared for publications. The following publications are acknowledged.

- Chapter 1.1 and Chapter 5 have been published in full under the citation:

**Wang Z**, Escotet-Espinoza M S, Ierapetritou M. Process analysis and optimization of continuous pharmaceutical manufacturing using flowsheet models[J]. Computers & Chemical Engineering, 2017, 107: 77-91.

- Chapter 2 has been published in full under the citation:

**Wang Z**, Ierapetritou M. Global sensitivity, feasibility, and flexibility analysis of continuous pharmaceutical manufacturing processes[J]. Computer Aided Chemical Engineering, 2018, 41: 189-213.

- Chapter 3 is being prepared for publication under the citation:

**Wang Z**, Ierapetritou M. Applications of Optimization in the Pharmaceutical Process Development.

- Chapter 4 has been published in full under the citation:

**Wang Z**, Ierapetritou M. A novel feasibility analysis method for black - box processes using a radial basis function adaptive sampling approach[J]. AIChE Journal, 2017, 63(2): 532-550.

- Chapter 6 has been published in full under the citation:

**Wang Z**, Ierapetritou M. Surrogate-based feasibility analysis for black-box stochastic simulations with heteroscedastic noise[J]. Journal of Global Optimization, <https://doi.org/10.1007/s10898-018-0615-4>

- Chapter 7 has been published in full under the citation:

**Wang Z**, Ierapetritou M. A Novel Surrogate-Based Optimization Method for Black-Box Simulation with Heteroscedastic Noise[J]. Industrial & Engineering Chemistry Research, 2017, 56(38): 10720-10732.

- Chapter 8 is being prepared for publication under the citation:

**Wang Z**, Ierapetritou M. A Kriging-based method with feasibility enhancements for constrained optimization of black-box stochastic systems

## Appendix

### A. Derivation of $E I_{feas}$

Rewrite the improvement of feasibility as follows:

$$I_{feas} = \begin{cases} \hat{s}(0 - z), & \text{if } z \leq f^{t'} \leq 0 \\ 0, & \text{else} \\ \hat{s}(z - 0), & \text{if } z \geq f^{t'} > 0, \end{cases}$$

where  $z = \frac{y - \hat{\mu}}{\hat{s}}$ ,  $f^{t'} = \frac{f^t - \hat{\mu}}{\hat{s}}$ .

Due to the assumed normal distribution for  $y$ , in the case when  $f^{t'} \leq 0$ , the expected value of  $I_{feas}$  can be derived as follows:

$$\begin{aligned} E[I_{feas}] &= \hat{s} \int_{-\infty}^{f^{t'}} (0 - z) \phi(z) dz \\ &= \hat{s} \left[ \frac{e^{-\frac{z^2}{2}}}{\sqrt{2\pi}} \right]_{-\infty}^{f^{t'}} \\ &= \hat{s} \phi(f^{t'}) \\ &= \hat{s} \phi\left(\frac{f^t - \hat{\mu}}{\hat{s}}\right). \end{aligned}$$

In the case when  $f^{t'} > 0$ , the same expression for  $E[I_{feas}]$  can be obtained by taking similar derivation steps as mentioned above.

### B. Figures for the 2D test problems

Below, we show the figures depicting the feasible region boundaries (denoted with thick-dashed line) and noise standard deviation  $\xi$  (denoted with filled contour) for the four 2D test functions. “Branin” in Figure 71; “Camelback” in Figure 72; “Example3” in Figure 73; “Sasena” in Figure 74.

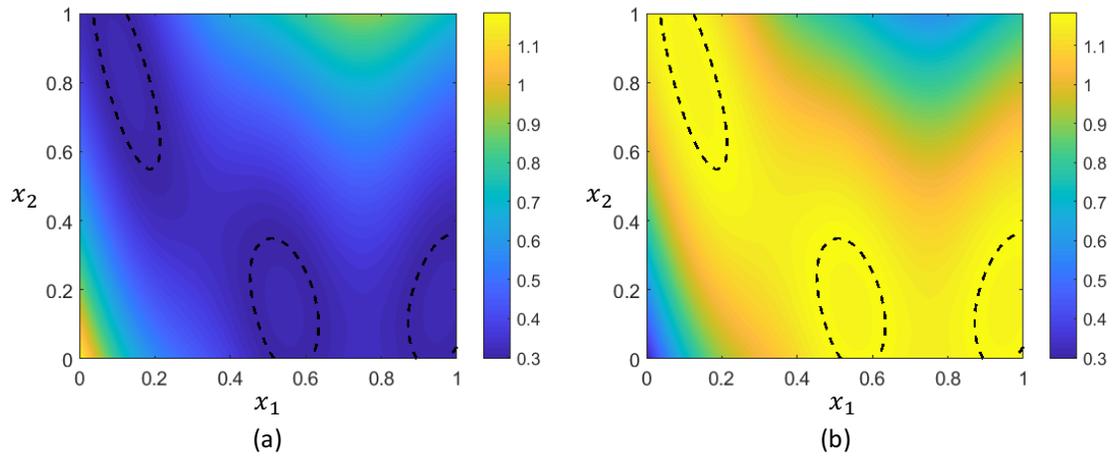


Figure 71. “Branin” test function. (a) “easy” noise; (b) “hard” noise. Thick-dashed line: feasible region boundaries; filled contour: standard deviation of the noise term

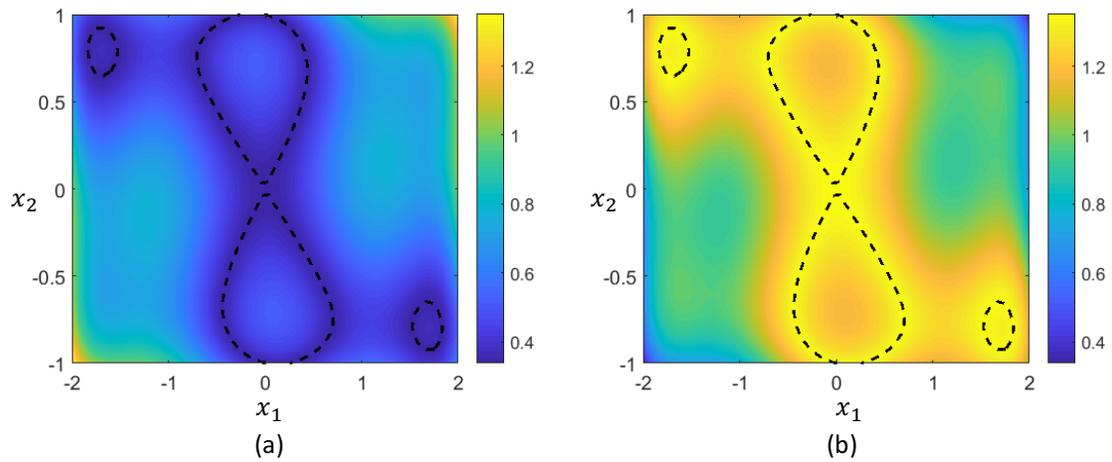


Figure 72. “Camelback” test function. (a) “easy” noise; (b) “hard” noise. Thick-dashed line: feasible region boundaries; filled contour: standard deviation of the noise term

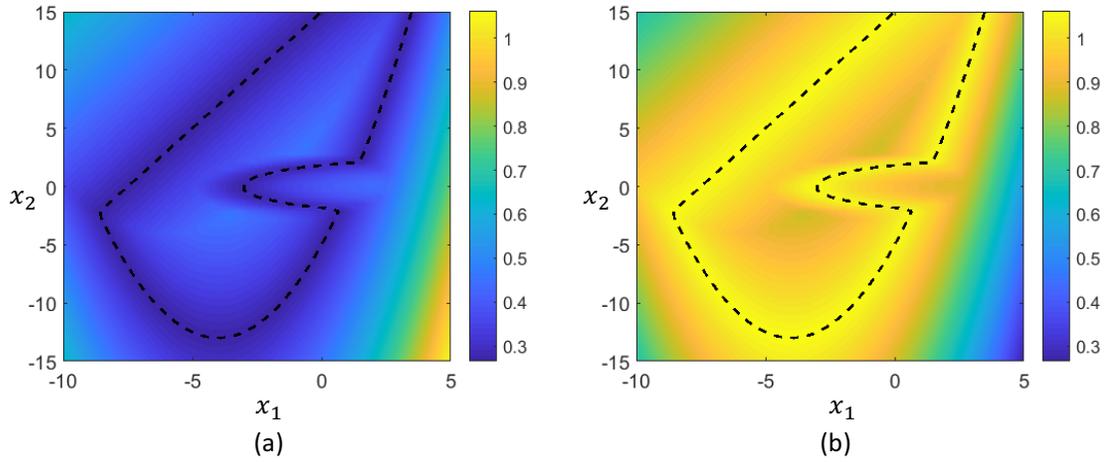


Figure 73. “Example3” test function. (a) “easy” noise; (b) “hard” noise. Thick-dashed line: feasible region boundaries; filled contour: standard deviation of the noise term

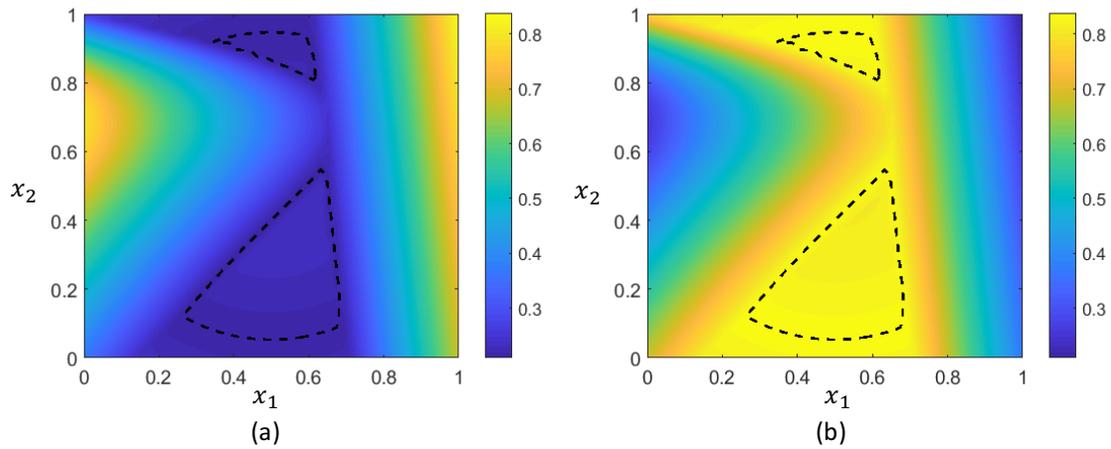


Figure 74. “Sasena” test function. (a) “easy” noise; (b) “hard” noise. Thick-dashed line: feasible region boundaries; filled contour: standard deviation of the noise term

### C. Test problem functions

#### Gomez [363,364]

$$\min f = \left(4 - 2.1x_1^2 + \frac{1}{3}x_1^4\right)x_1^2 + x_1x_2 + (-4 + 4x_2^2)x_2^2$$

s. t.

$$g = -\sin(4\pi x_1) + 2\sin^2(2\pi x_2) \leq 0$$

$$-1 \leq x_i \leq 1 \text{ for } i = 1, 2$$

The best known solution is at  $x^* = (0.1093, -0.6234)$  where  $f^* = -0.9711$

#### Constrained Branin

This is a modified version of the “constrained Branin Function” in Ref. [64].

$$\min f = \frac{1}{51.95}((ax_2 - bx_1^2 + cx_1 - d)^2 + h(1 - ff) \cos(x_1) - 44.81)$$

s. t.

$$g = \max_{j \in \{1, 2\}} \{g_j\} \leq 0$$

$$g_1 = \frac{1}{10}(x_1(1 - x_2) - x_2)$$

$$g_2 = \frac{1}{10} \left( 1 - \left( \frac{(x_1 - 5)^2}{8} + \frac{(x_2 - 15)^2}{4} \right) \right)$$

$$a = 1, b = \frac{5.1}{4\pi^2}, c = \frac{5}{\pi}, d = 6, h = 10, ff = \frac{1}{8\pi}$$

$$-5 \leq x_1 \leq 10, 0 \leq x_2 \leq 15$$

The best known solutions are at  $x^* = (9.42478, 2.475)$  and  $(\pi, 2.275)$  where  $f^* = -1.0474$

**New Branin**

This is a scaled version of the “newBranin” function in Ref. [364].

$$\min f = \frac{1}{80}(-(x_1 - 10)^2 - (x_2 - 15)^2)$$

s. t.

$$g = \frac{1}{51.95}((ax_2 - bx_1^2 + cx_1 - d)^2 + h(1 - ff) \cos(x_1) - 5 + h) \leq 0$$

$$a = 1, b = \frac{5.1}{4\pi^2}, c = \frac{5}{\pi}, d = 6, h = 10, ff = \frac{1}{8\pi}$$

$$-5 \leq x_1 \leq 10, 0 \leq x_2 \leq 15$$

The best known solutions are at  $x^* = (3.273, 0.0489)$  where  $f^* = -3.3599$

**Sasena [364,365]**

$$\min f = -(x_1 - 1)^2 - (x_2 - 0.5)^2$$

s. t.

$$g = \max_{j \in \{1,2,3\}} \{g_j\} \leq 0$$

$$g_1 = ((x_1 - 3)^2 + ((x_2 + 2)^2)e^{-x_2^7} - 12$$

$$g_2 = 10x_1 + x_2 - 7$$

$$g_3 = (x_1 - 0.5)^2 + (x_2 - 0.5)^2 - 0.2$$

$$0 \leq x_i \leq 1 \text{ for } i = 1,2$$

The best known solutions are at  $x^* = (0.2017, 0.8332)$  where  $f^* = -0.7483$

**Qcp4 [233]**

$$\min f = -2x_1 + x_2 - x_3$$

s. t.

$$g = \max_{j \in \{1,2,3\}} \{g_j\} \leq 0$$

$$g_1 = x_1 + x_2 + x_3 - 4$$

$$g_2 = 3x_2 + x_3 - 6$$

$$g_3 = -\mathbf{x}' \mathbf{A}' \mathbf{A} \mathbf{x} + 2\mathbf{y}' \mathbf{A} \mathbf{x} - \|\mathbf{y}\|^2 + 0.25\|\mathbf{b} - \mathbf{z}\|^2$$

$$\mathbf{A} = [0,0,1; 0, -1,0; -2,1, -1],$$

$$\mathbf{b} = [3; 0; -4],$$

$$\mathbf{y} = [1.5; -0.5; -5],$$

$$\mathbf{z} = [0; -1; -6]$$

$$\mathbf{x} = [x_1; x_2; x_3]$$

$$0 \leq x_1 \leq 2, 0 \leq x_i \leq 3 \text{ for } i = 2,3$$

The best known solutions are at  $\mathbf{x}^* = (0.5, 0, 3)$  where  $f^* = -4$

#### G4

This is a scaled version of the “G4” function in Ref. [366].

$$\min f = \frac{5.3578547x_3^2 + 0.8356891x_1x_5 + 37.293239x_1 - 40792.141}{2000}$$

s. t.

$$g = \max_{j \in \{1,2,3,4,5,6\}} \{g_j\} \leq 0$$

$$g_1 = 0 - u$$

$$g_2 = u - 92$$

$$g_3 = 90 - v$$

$$g_4 = v - 110$$

$$g_5 = 20 - w$$

$$g_6 = w - 25$$

$$u = 85.334407 + 0.0056858x_2x_5 + 0.0006262x_1x_4 - 0.0022053x_3x_5$$

$$v = 80.51249 + 0.0071317x_2x_5 + 0.0029955x_1x_2 + 0.0021813x_3^2$$

$$w = 9.300961 + 0.0047026x_3x_5 + 0.0012547x_1x_3 + 0.0019085x_3x_4$$

$$78 \leq x_1 \leq 102, 33 \leq x_2 \leq 45, 27 \leq x_i \leq 45 \text{ for } i = 3, 4, 5$$

The best known solutions are at  $x^* = (78, 33, 29.995, 45, 36.7758)$  where  $f^* = -15.3328$

## G24

This is a scaled version of the “G24” function in Ref. [366].

$$\min f = \frac{5}{7}(-x_1 - x_2)$$

s. t.

$$g = \max_{j \in \{1,2\}} \{g_j\} \leq 0$$

$$g_1 = \frac{1}{4}(-2x_1^4 + 8x_1^3 - 8x_1^2 + x_2 - 2)$$

$$g_2 = \frac{1}{4}(-4x_1^4 + 32x_1^3 - 88x_1^2 + 96x_1 + x_2 - 36)$$

$$0 \leq x_1 \leq 3, 0 \leq x_2 \leq 4$$

The best known solutions are at  $x^* = (2.0627, 3.9686)$  where  $f^* = -4.3081$

## Angun

This is a scaled version of the “Angun” function in Ref. [337]

$$\min f = \frac{5}{34.8}(5(x_1 - 1)^2 + (x_2 - 5)^2 + 4x_1x_2)$$

s. t.

$$g = \max_{j \in \{1,2\}} \{g_j\} \leq 0$$

$$g_1 = \frac{5}{17.1544} ((x_1 - 3)^2 + x_2^2 + x_1 x_2 - 4)$$

$$g_2 = \frac{5}{17.1544} (x_1^2 + 3(x_2 + 1.061)^2 - 9)$$

$$0 \leq x_1 \leq x_3, -2 \leq x_2 \leq 1$$

The best known solutions are at  $x^* = (1.2411, 0.5159)$  where  $f^* = 3.2987$

## Bibliography

1. Fisher, A.C., Lee, S.L., Harris, D.P., Buhse, L., Kozlowski, S., Yu, L., Kopcha, M., Woodcock, J.: Advancing pharmaceutical quality: An overview of science and research in the US FDA's Office of Pharmaceutical Quality. *International Journal of Pharmaceutics* **515**(1), 390-402 (2016).
2. Lee, S.L., O' Connor, T.F., Yang, X., Cruz, C.N., Chatterjee, S., Madurawe, R.D., Moore, C.M., Lawrence, X.Y., Woodcock, J.: Modernizing pharmaceutical manufacturing: from batch to continuous production. *Journal of Pharmaceutical Innovation* **10**(3), 191-199 (2015).
3. Badman, C., Trout, B.L.: Achieving continuous manufacturing. May 20 - 21, 2014 continuous manufacturing symposium. *Journal of pharmaceutical sciences* **104**(3), 779-780 (2015).
4. Lee, H.: *Pharmaceutical industry practices on genotoxic impurities*. CRC Press, (2014)
5. Hlinak, A.J., Kuriyan, K., Morris, K.R., Reklaitis, G.V., Basu, P.K.: Understanding critical material properties for solid dosage form design. *Journal of Pharmaceutical Innovation* **1**(1), 12-17 (2006).
6. Gao, Y., Vanarase, A., Muzzio, F., Ierapetritou, M.: Characterizing continuous powder mixing using residence time distribution. *Chemical Engineering Science* **66**(3), 417-425 (2011).
7. Engisch, W., Muzzio, F.: Using residence time distributions (RTDs) to address the traceability of raw materials in continuous pharmaceutical manufacturing. *Journal of pharmaceutical innovation* **11**(1), 64-81 (2016).
8. Lawrence, X.Y., Amidon, G., Khan, M.A., Hoag, S.W., Polli, J., Raju, G., Woodcock, J.: Understanding pharmaceutical quality by design. *The AAPS journal* **16**(4), 771-783 (2014).
9. Ketterhagen, W.R., am Ende, M.T., Hancock, B.C.: Process modeling in the pharmaceutical industry using the discrete element method. *Journal of pharmaceutical sciences* **98**(2), 442-470 (2009).
10. Immanuel, C.D., Doyle, F.J.: Solution technique for a multi-dimensional population balance model describing granulation processes. *Powder Technology* **156**(2), 213-225 (2005).
11. Barrasso, D., Oka, S., Muliadi, A., Litster, J.D., Wassgren, C., Ramachandran, R.: Population Balance Model Validation and Prediction of CQAs for Continuous Milling Processes: toward QbD in Pharmaceutical Drug Product Manufacturing. *Journal of Pharmaceutical Innovation* **8**(3), 147-162 (2013).
12. Boukouvala, F., Dubey, A., Vanarase, A., Ramachandran, R., Muzzio, F.J., Ierapetritou, M.: Computational Approaches for Studying the Granular Dynamics of Continuous Blending Processes, 2 - Population Balance and Data - Based Methods. *Macromolecular Materials and Engineering* **297**(1), 9-19 (2012).
13. Boukouvala, F., Muzzio, F., Ierapetritou, M.G.: Dynamic data-driven modeling of pharmaceutical processes. *Industrial & Engineering Chemistry Research* **50**(11), 6743-6754 (2011).
14. Rogers, A.J., Inamdar, C., Ierapetritou, M.G.: An integrated approach to simulation of pharmaceutical processes for solid drug manufacture. *Industrial & Engineering Chemistry Research* **53**(13), 5128-5147 (2013).
15. Boukouvala, F., Niotis, V., Ramachandran, R., Muzzio, F.J., Ierapetritou, M.G.: An integrated approach for dynamic flowsheet modeling and sensitivity analysis of a continuous tablet manufacturing process. *Computers & Chemical Engineering* **42**, 30-47 (2012).
16. Sen, M., Dubey, A., Singh, R., Ramachandran, R.: Mathematical development and comparison of a hybrid PBM-DEM description of a continuous powder mixing process. *Journal of Powder Technology* **2013** (2012).

17. Engisch, W.E., Muzzio, F.J.: Method for characterization of loss-in-weight feeder equipment. *Powder technology* **228**, 395-403 (2012).
18. Boukouvala, F., Muzzio, F.J., Ierapetritou, M.G.: Design space of pharmaceutical processes using data-driven-based methods. *Journal of Pharmaceutical Innovation* **5**(3), 119-137 (2010).
19. Escotet-Espinoza, M.S., Jayjock, E., Singh, R., Vanarase, A., Muzzio, F.J., Ierapetritou, M., Ketterhagen, W.R., Blackwood, D.O., Moriarty, J., Liu, Y.: 2015 Annual Meeting November 8-13, 2015.
20. Abdullah, E., Geldart, D.: The use of bulk density measurements as flowability indicators. *Powder Technology* **102**(2), 151-165 (1999).
21. Singh, R., Sahay, A., Karry, K.M., Muzzio, F., Ierapetritou, M., Ramachandran, R.: Implementation of an advanced hybrid MPC - PID control system using PAT tools into a direct compaction continuous pharmaceutical tablet manufacturing pilot plant. *International journal of pharmaceutics* **473**(1), 38-54 (2014).
22. Vanarase, A.U., Osorio, J.G., Muzzio, F.J.: Effects of powder flow properties and shear environment on the performance of continuous mixing of pharmaceutical powders. *Powder technology* **246**, 63-72 (2013).
23. Deng, X., Scicolone, J., Han, X., Davé, R.N.: Discrete element method simulation of a conical screen mill: A continuous dry coating device. *Chemical Engineering Science* **125**, 58-74 (2015).
24. Pernenkil, L., Cooney, C.L.: A review on the continuous blending of powders. *Chemical Engineering Science* **61**(2), 720-742 (2006).
25. Zhu, H., Zhou, Z., Yang, R., Yu, A.: Discrete particle simulation of particulate systems: a review of major applications and findings. *Chemical Engineering Science* **63**(23), 5728-5770 (2008).
26. Bertrand, F., Leclaire, L.-A., Levecque, G.: DEM-based models for the mixing of granular materials. *Chemical Engineering Science* **60**(8), 2517-2531 (2005).
27. Gao, Y., Ierapetritou, M., Muzzio, F.: Periodic section modeling of convective continuous powder mixing processes. *AIChE Journal* **58**(1), 69-78 (2012).
28. Sen, M., Chaudhury, A., Singh, R., John, J., Ramachandran, R.: Multi-scale flowsheet simulation of an integrated continuous purification - downstream pharmaceutical manufacturing process. *International journal of pharmaceutics* **445**(1), 29-38 (2013).
29. Rogers, A., Ierapetritou, M.G.: Discrete element reduced - order modeling of dynamic particulate systems. *AIChE Journal* **60**(9), 3184-3194 (2014).
30. Gao, Y., Ierapetritou, M., Muzzio, F.: Investigation on the effect of blade patterns on continuous solid mixing performance. *The Canadian Journal of Chemical Engineering* **89**(5), 969-984 (2011).
31. Vanarase, A.U., Alcalà, M., Rozo, J.I.J., Muzzio, F.J., Románach, R.J.: Real-time monitoring of drug concentration in a continuous powder mixing process using NIR spectroscopy. *Chemical Engineering Science* **65**(21), 5728-5733 (2010).
32. Ierapetritou, M., Reklaitis, G., Muzzio, F.: Perspectives on continuous manufacturing of powder - based pharmaceutical processes. *AIChE Journal* **62**(6), 1846-1862 (2016).
33. Sinka, I., Motazedian, F., Cocks, A., Pitt, K.: The effect of processing parameters on pharmaceutical tablet properties. *Powder Technology* **189**(2), 276-284 (2009).
34. Pitt, K., Sinka, C.: Tableting. *Handbook of powder technology* **11**, 735-778 (2007).
35. Mateo-Ortiz, D., Muzzio, F.J., Méndez, R.: Particle size segregation promoted by powder flow in confined space: the die filling process case. *Powder Technology* **262**, 215-222 (2014).

36. Koynov, A., Akseli, I., Cuitino, A.: Modeling and simulation of compact strength due to particle bonding using a hybrid discrete-continuum approach. *International journal of pharmaceutics* **418**(2), 273-285 (2011).
37. Siiriä, S.M., Antikainen, O., Heinämäki, J., Yliruusi, J.: 3d simulation of internal tablet strength during tableting. *Aaps Pharmscitech* **12**(2), 593-603 (2011).
38. Heckel, R.: Density-pressure relationships in powder compaction. *Trans Metall Soc AIME* **221**(4), 671-675 (1961).
39. Kawakita, K., Lüdde, K.-H.: Some considerations on powder compression equations. *Powder technology* **4**(2), 61-68 (1971).
40. Kuentz, M., Leuenberger, H.: A new model for the hardness of a compacted particle system, applied to tablets of pharmaceutical polymers. *Powder technology* **111**(1), 145-153 (2000).
41. Singh, R., Gernaey, K.V., Gani, R.: ICAS-PAT: A software for design, analysis and validation of PAT systems. *Computers & chemical engineering* **34**(7), 1108-1136 (2010).
42. Singh, R., Ierapetritou, M., Ramachandran, R.: An engineering study on the enhanced control and operation of continuous manufacturing of pharmaceutical tablets via roller compaction. *International journal of pharmaceutics* **438**(1), 307-326 (2012).
43. Saltelli, A., Ratto, M., Andres, T., Campolongo, F., Cariboni, J., Gatelli, D., Saisana, M., Tarantola, S.: *Global sensitivity analysis: the primer*. John Wiley & Sons, (2008)
44. Grossmann, I.E., Calfa, B.A., Garcia-Herreros, P.: Evolution of concepts and models for quantifying resiliency and flexibility of chemical processes. *Computers & Chemical Engineering* **70**, 22-34 (2014).
45. Grossmann, I.E., Morari, M.: *Operability, resiliency, and flexibility: Process design objectives for a changing world*. (1983).
46. Brueggemeier, S.B., Reiff, E.A., Lyngberg, O.K., Hobson, L.A., Tabora, J.E.: Modeling-Based Approach Towards Quality by Design for the Ibipinabant API Step. *Organic Process Research & Development* **16**(4), 567-576 (2012).
47. Prpich, A., am Ende, M.T., Katzschner, T., Lubczyk, V., Weyhers, H., Bernhard, G.: Drug product modeling predictions for scale-up of tablet film coating—a quality by design approach. *Computers & chemical engineering* **34**(7), 1092-1097 (2010).
48. Rogers, A., Ierapetritou, M.: Feasibility and flexibility analysis of black-box processes Part 2: surrogate-based flexibility analysis. *Chemical Engineering Science* **137**, 1005-1013 (2015).
49. Boukouvala, F., Ierapetritou, M.G.: Surrogate-based optimization of expensive flowsheet modeling for continuous pharmaceutical manufacturing. *Journal of Pharmaceutical Innovation* **8**(2), 131-145 (2013).
50. Ekstrom, P.-A.: *Eikos: a simulation toolbox for sensitivity analysis in Matlab*. FACILIA AB (2005).
51. Iooss, B., Lemaître, P.: A review on global sensitivity analysis methods. In: *Uncertainty Management in Simulation-Optimization of Complex Systems*. pp. 101-122. Springer, (2015)
52. Campolongo, F., Saltelli, A., Cariboni, J.: From screening to quantitative sensitivity analysis. A unified approach. *Computer Physics Communications* **182**(4), 978-988 (2011).
53. Sobol, I.M.: Uniformly distributed sequences with an additional uniform property. *USSR Computational Mathematics and Mathematical Physics* **16**(5), 236-242 (1976).
54. Sohier, H., Farges, J.-L., Piet-Lahanier, H.: Improvement of the Representativity of the Morris Method for Air-Launch-to-Orbit Separation. *IFAC Proceedings Volumes* **47**(3), 7954-7959 (2014).
55. Marino, S., Hogue, I.B., Ray, C.J., Kirschner, D.E.: A methodology for performing global uncertainty and sensitivity analysis in systems biology. *Journal of theoretical biology* **254**(1), 178-196 (2008).

56. McKay, M.D., Beckman, R.J., Conover, W.J.: Comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics* **21**(2), 239-245 (1979).
57. Helton, J.C., Davis, F.J.: Latin hypercube sampling and the propagation of uncertainty in analyses of complex systems. *Reliability Engineering & System Safety* **81**(1), 23-69 (2003).
58. Saltelli, A., Annoni, P., Azzini, I., Campolongo, F., Ratto, M., Tarantola, S.: Variance based sensitivity analysis of model output. Design and estimator for the total sensitivity index. *Computer Physics Communications* **181**(2), 259-270 (2010).
59. Saltelli, A., Chan, K., Scott, E.M.: *Sensitivity analysis*, vol. 1. Wiley New York, (2000)
60. Cukier, R., Fortuin, C., Shuler, K.E., Petschek, A., Schaibly, J.: Study of the sensitivity of coupled reaction systems to uncertainties in rate coefficients. I Theory. *The Journal of chemical physics* **59**(8), 3873-3878 (1973).
61. Ziehn, T., Tomlin, A.S.: GUI - HDMR - A software tool for global sensitivity analysis of complex models. *Environmental Modelling & Software* **24**(7), 775-785 (2009).
62. Li, G., Rosenthal, C., Rabitz, H.: High dimensional model representations. *The Journal of Physical Chemistry A* **105**(33), 7765-7777 (2001).
63. Huang, J., Kaul, G., Cai, C., Chatlapalli, R., Hernandez-Abad, P., Ghosh, K., Nagi, A.: Quality by design case study: an integrated multivariate approach to drug product and process development. *International journal of pharmaceuticals* **382**(1), 23-32 (2009).
64. Boukouvala, F., Ierapetritou, M.G.: Derivative - free optimization for expensive constrained problems using a novel expected improvement objective function. *AIChE Journal* **60**(7), 2462-2474 (2014).
65. Rogers, A., Ierapetritou, M.: Feasibility and flexibility analysis of black-box processes Part 1: Surrogate-based feasibility analysis. *Chemical Engineering Science* **137**, 986-1004 (2015).
66. Oliver, M.A., Webster, R.: *Basic steps in geostatistics: the variogram and kriging*. Springer, (2015)
67. Jones, D.R., Schonlau, M., Welch, W.J.: Efficient global optimization of expensive black-box functions. *Journal of Global optimization* **13**(4), 455-492 (1998).
68. Lai, S.M., Hui, C.-W.: Process flexibility for multivariable systems. *Industrial & engineering chemistry research* **47**(12), 4170-4183 (2008).
69. Adi, V.S.K., Chang, C.-T.: A mathematical programming formulation for temporal flexibility analysis. *Computers & Chemical Engineering* **57**, 151-158 (2013).
70. Process Systems Enterprise. <https://www.psenderprise.com/products/gproms/modelbuilder>. Accessed 13 Feb 2017
71. Process Systems Enterprise. <https://www.psenderprise.com/products/gsolids>. Accessed 12 Feb 2017
72. SimLab. <https://ec.europa.eu/jrc/en/samo/simlab>. Accessed 13 Feb 2017
73. GUI-HDMR. <http://www.gui-hdmr.de/index.html>.
74. DACE. <http://www2.imm.dtu.dk/projects/dace/>. Accessed 13 Feb 2017
75. Jacques, J., Lavergne, C., Devictor, N.: Sensitivity analysis in presence of model uncertainty and correlated inputs. *Reliability Engineering & System Safety* **91**(10), 1126-1134 (2006).
76. Shan, S., Wang, G.G.: Survey of modeling and optimization strategies to solve high-dimensional design problems with computationally-expensive black-box functions. *Structural and Multidisciplinary Optimization* **41**(2), 219-241 (2010).
77. Food, U., Administration, D.: *Pharmaceutical CGMPs for the 21st Century—a risk-based approach*. Final report. Rockville, MD (2004).

78. Food, Administration, D.: Guidance for industry: PAT—A framework for innovative pharmaceutical development, manufacturing, and quality assurance. DHHS, Rockville, MD (2004).
79. Lawrence, X.Y.: Pharmaceutical quality by design: product and process development, understanding, and control. *Pharmaceutical research* **25**(4), 781-791 (2008).
80. Khaled, S.A., Burley, J.C., Alexander, M.R., Roberts, C.J.: Desktop 3D printing of controlled release pharmaceutical bilayer tablets. *International journal of pharmaceutics* **461**(1), 105-111 (2014).
81. O' Connor, T.F., Lawrence, X.Y., Lee, S.L.: Emerging technology: A key enabler for modernizing pharmaceutical manufacturing and advancing product quality. *International journal of pharmaceutics* **509**(1), 492-498 (2016).
82. McKenzie, P., Kiang, S., Tom, J., Rubin, A.E., Futran, M.: Can pharmaceutical process development become high tech? *AIChE Journal* **52**(12), 3990-3994 (2006).
83. Rogers, A., Ierapetritou, M.: Challenges and opportunities in modeling pharmaceutical manufacturing processes. *Computers & Chemical Engineering* **81**, 32-39 (2015).
84. Kleinebudde, P., Khinast, J., Rantanen, J.: *Continuous Manufacturing of Pharmaceuticals*, vol. 7703. John Wiley & Sons, (2017)
85. Wu, C.-Y., Ruddy, O., Bentham, A., Hancock, B., Best, S., Elliott, J.: Modelling the mechanical behaviour of pharmaceutical powders during compaction. *Powder Technology* **152**(1), 107-117 (2005).
86. Chaudhury, A., Barrasso, D., Pandey, P., Wu, H., Ramachandran, R.: Population balance model development, validation, and prediction of CQAs of a high-shear wet granulation process: towards QbD in drug product pharmaceutical manufacturing. *Journal of Pharmaceutical Innovation* **9**(1), 53-64 (2014).
87. Pal, T.K., Dan, S., Dan, N.: Application of Response Surface Methodology (RSM) in Statistical Optimization and Pharmaceutical Characterization of a Matrix Tablet Formulation using Metformin HCl as a Model Drug. *Innoriginal: International Journal Of Sciences* (2014).
88. Shirazian, S., Kuhs, M., Darwish, S., Croker, D., Walker, G.M.: Artificial neural network modelling of continuous wet granulation using a twin-screw extruder. *International journal of pharmaceutics* **521**(1), 102-109 (2017).
89. Tabora, J.E., Domagalski, N.: *Multivariate Analysis and Statistics in Pharmaceutical Process Research and Development*. Annual Review of Chemical and Biomolecular Engineering(0) (2017).
90. Boukouvala, F., Chaudhury, A., Sen, M., Zhou, R., Mioduszewski, L., Ierapetritou, M.G., Ramachandran, R.: Computer-aided flowsheet simulation of a pharmaceutical tablet manufacturing process incorporating wet granulation. *Journal of Pharmaceutical Innovation* **8**(1), 11-27 (2013).
91. Boukouvala, F., Muzzio, F.J., Ierapetritou, M.G.: Methods and Tools for Design Space Identification in Pharmaceutical Development. *Comprehensive Quality by Design for Pharmaceutical Product Development and Manufacture*, 95-123 (2017).
92. Singh, R., Muzzio, F.J., Ierapetritou, M., Ramachandran, R.: A combined feed-forward/feed-back control system for a QbD-based continuous tablet manufacturing process. *Processes* **3**(2), 339-356 (2015).
93. Biegler, L.T., Grossmann, I.E.: Retrospective on optimization. *Computers & Chemical Engineering* **28**(8), 1169-1192 (2004).
94. Velásco-Mejía, A., Vallejo-Becerra, V., Chávez-Ramírez, A., Torres-González, J., Reyes-Vidal, Y., Castañeda-Zaldivar, F.: Modeling and optimization of a pharmaceutical crystallization process by using neural networks and genetic algorithms. *Powder Technology* **292**, 122-128 (2016).

95. Monteagudo, E., Langenheim, M., Salerno, C., Buontempo, F., Bregni, C., Carlucci, A.: Pharmaceutical optimization of lipid-based dosage forms for the improvement of taste-masking, chemical stability and solubilizing capacity of phenobarbital. *Drug development and industrial pharmacy* **40**(6), 783-792 (2014).
96. Chavez, P.-F., Lebrun, P., Sacre, P.-Y., De Bleye, C., Netchacovitch, L., Cuypers, S., Mantanus, J., Motte, H., Schubert, M., Evrard, B.: Optimization of a pharmaceutical tablet formulation based on a design space approach and using vibrational spectroscopy as PAT tool. *International journal of pharmaceutics* **486**(1), 13-20 (2015).
97. Sheikholeslamzadeh, E., Chen, C.-C., Rohani, S.: Optimal solvent screening for the crystallization of pharmaceutical compounds from multisolvent systems. *Industrial & Engineering Chemistry Research* **51**(42), 13792-13802 (2012).
98. Zhang, C., Huang, J.: Optimization of process parameters for pharmaceutical wastewater treatment. *Polish Journal of Environmental Studies* **24**(1), 391-395 (2015).
99. Grom, M., Stavber, G., Drnovšek, P., Likozar, B.: Modelling chemical kinetics of a complex reaction network of active pharmaceutical ingredient (API) synthesis with process optimization for benzazepine heterocyclic compound. *Chemical Engineering Journal* **283**, 703-716 (2016).
100. Jolliffe, H.G., Gerogiorgis, D.I.: Technoeconomic Optimization of a Conceptual Flowsheet for Continuous Separation of an Analgesic Active Pharmaceutical Ingredient (API). *Industrial & Engineering Chemistry Research* **56**(15), 4357-4376 (2017).
101. Abejón, R., Garea, A., Irabien, A.: Analysis and optimization of continuous organic solvent nanofiltration by membrane cascade for pharmaceutical separation. *AIChE Journal* **60**(3), 931-948 (2014).
102. Grossmann, I.E., Morari, M.: Operability, resiliency, and flexibility: Process design objectives for a changing world. In: Westerberg, A.W., Chien, H.H. (eds.) *Proc. 2nd international conference on foundations computer aided process design 1983. CACHE*
103. Adi, V.S.K., Laxmidewi, R.: Design and Operability Analysis of Membrane Module based on Volumetric Flexibility. In: *Computer Aided Chemical Engineering*, vol. 40. pp. 1231-1236. Elsevier, (2017)
104. Finnveden, G., Hauschild, M.Z., Ekvall, T., Guinée, J., Heijungs, R., Hellweg, S., Koehler, A., Pennington, D., Suh, S.: Recent developments in life cycle assessment. *Journal of environmental management* **91**(1), 1-21 (2009).
105. Ott, D., Kralisch, D., Denčić, I., Hessel, V., Laribi, Y., Perrichon, P.D., Berguerand, C., Kiwi - Minsker, L., Loeb, P.: Life cycle analysis within pharmaceutical process optimization and intensification: case study of active pharmaceutical ingredient production. *ChemSusChem* **7**(12), 3521-3533 (2014).
106. Ott, D., Borukhova, S., Hessel, V.: Life cycle assessment of multi-step rufinamide synthesis – from isolated reactions in batch to continuous microreactor networks. *Green Chemistry* **18**(4), 1096-1116 (2016).
107. Nha, V.T., Shin, S., Jeong, S.H.: Lexicographical dynamic goal programming approach to a robust design optimization within the pharmaceutical environment. *European Journal of Operational Research* **229**(2), 505-517 (2013).
108. Jeong, S.H., Kongsuwan, P., Truong, N.K.V., Shin, S.: Optimal tolerance design and optimization for a pharmaceutical quality characteristic. *Mathematical Problems in Engineering* **2013** (2013).
109. Yoshizaki, R., Kano, M., Tanabe, S., Miyano, T.: Process Parameter Optimization based on LW-PLS in Pharmaceutical Granulation Process\*\* This work was partially supported by Japan Society for the Promotion of Science (JSPS), Grant-in-Aid for Scientific Research (C) 24560940. *IFAC-PapersOnLine* **48**(8), 303-308 (2015).

110. Ardakani, M.K., Wulff, S.S.: An overview of optimization formulations for multiresponse surface problems. *Quality and Reliability Engineering International* **29**(1), 3-16 (2013).
111. Deb, K.: Multi-objective optimization. In: *Search methodologies*. pp. 403-449. Springer, (2014)
112. Brunet, R., Guillén-Gosálbez, G., Jiménez, L.: Combined simulation – optimization methodology to reduce the environmental impact of pharmaceutical processes: application to the production of Penicillin V. *Journal of cleaner production* **76**, 55-63 (2014).
113. Derringer, G.C.: A balancing act-optimizing a products properties. *Quality progress* **27**(6), 51-58 (1994).
114. Uttekar, P., Chaudhari, P.: Formulation and Evaluation of Engineered Pharmaceutical Fine Particles of Budesonide for Dry Powder Inhalation (Dpi) Produced by Amphiphilic Crystallization Technique: Optimization of Process Parameters. *International Journal of Pharmaceutical Sciences and Research* **4**(12), 4656 (2013).
115. Sato, H., Watanabe, S., Takeda, D., Yano, S., Doki, N., Yokota, M., Shimizu, K.: Optimization of a Crystallization Process for Orantinib Active Pharmaceutical Ingredient by Design of Experiment To Control Residual Solvent Amount and Particle Size Distribution. *Organic Process Research & Development* **19**(11), 1655-1661 (2015).
116. Chakraborty, P., Dey, S., Parcha, V., Bhattacharya, S.S., Ghosh, A.: Design expert supported mathematical optimization and predictability study of buccoadhesive pharmaceutical wafers of Loratadine. *BioMed research international* **2013** (2013).
117. Kermet-Said, H., Moulai-Mostefa, N.: Optimization of Turbidity and COD Removal from Pharmaceutical Wastewater by Electrocoagulation. *Isotherm Modeling and Cost Analysis. Polish Journal of Environmental Studies* **24**(3) (2015).
118. Li, Z., Cho, B.R., Melloy, B.J.: Quality by design studies on multi-response pharmaceutical formulation modeling and optimization. *Journal of Pharmaceutical Innovation* **8**(1), 28-44 (2013).
119. Solomatine, D., See, L.M., Abrahart, R.: Data-driven modelling: concepts, approaches and experiences. In: *Practical hydroinformatics*. pp. 17-30. Springer, (2009)
120. Razavi, S., Tolson, B.A., Burn, D.H.: Review of surrogate modeling in water resources. *Water Resources Research* **48**(7) (2012).
121. Wang, G.G., Shan, S.: Review of metamodeling techniques in support of engineering design optimization. *Journal of Mechanical design* **129**(4), 370-380 (2007).
122. Rogers, A.J., Hashemi, A., Ierapetritou, M.G.: Modeling of particulate processes for the continuous manufacture of solid-based pharmaceutical dosage forms. *Processes* **1**(2), 67-127 (2013).
123. Jones, D.R.: A taxonomy of global optimization methods based on response surfaces. *Journal of global optimization* **21**(4), 345-383 (2001).
124. Boukouvala, F., Muzzio, F.J., Ierapetritou, M.G.: Predictive modeling of pharmaceutical processes with missing and noisy data. *AIChE journal* **56**(11), 2860-2872 (2010).
125. Singh, B., Kumar, R., Ahuja, N.: Optimizing drug delivery systems using systematic" design of experiments." Part I: fundamental aspects. *Critical Reviews™ in Therapeutic Drug Carrier Systems* **22**(1) (2005).
126. Elkhoudary, M.M., Abdel Salam, R.A., Hadad, G.M.: Development and Optimization of HPLC Analysis of Metronidazole, Diloxanide, Spiramycin and Cliquinol in Pharmaceutical Dosage Forms Using Experimental Design. *Journal of chromatographic science* **54**(10), 1701-1712 (2016).
127. Sahu, P.K., Swain, S., Prasad, G.S., Panda, J., Murthy, Y.: RP-HPLC Method for Determination of Metaxalone Using Box-Behnken Experimental Design. *Journal of Applied Biopharmaceutics and Pharmacokinetics* **2**(2), 40-49 (2015).

128. Sharma, D., Maheshwari, D., Philip, G., Rana, R., Bhatia, S., Singh, M., Gabrani, R., Sharma, S.K., Ali, J., Sharma, R.K.: Formulation and optimization of polymeric nanoparticles for intranasal delivery of lorazepam using Box-Behnken design: in vitro and in vivo evaluation. *BioMed research international* **2014** (2014).
129. Kalyani, A., Naga Sireesha, G., Aditya, A., Girija Sankar, G., Prabhakar, T.: Production Optimization of Rhamnolipid Biosurfactant by *Streptomyces coelicoflavus* (NBRC 15399T) using Plackett-Burman design. *Eur. J. Biotechnol. Biosci* **1**(5), 07-13 (2014).
130. Agarabi, C.D., Schiel, J.E., Lute, S.C., Chavez, B.K., Boyne, M.T., Brorson, K.A., Khan, M.A., Read, E.K.: Bioreactor process parameter screening utilizing a plackett – burman design for a model monoclonal antibody. *Journal of pharmaceutical sciences* **104**(6), 1919-1928 (2015).
131. Sacks, J., Welch, W.J., Mitchell, T.J., Wynn, H.P.: Design and analysis of computer experiments. *Statistical science*, 409-423 (1989).
132. Huntington, D., Lyrantzis, C.: Improvements to and limitations of Latin hypercube sampling. *Probabilistic Engineering Mechanics* **13**(4), 245-253 (1998).
133. Kalagnanam, J.R., Diwekar, U.M.: An efficient sampling technique for off-line quality control. *Technometrics* **39**(3), 308-319 (1997).
134. Owen, A.B.: Orthogonal arrays for computer experiments, integration and visualization. *Statistica Sinica*, 439-452 (1992).
135. Fang, K.-T., Lin, D.K., Winker, P., Zhang, Y.: Uniform design: theory and application. *Technometrics* **42**(3), 237-248 (2000).
136. Santner, T.J., Williams, B.J., Notz, W.I.: The design and analysis of computer experiments. Springer Science & Business Media, (2013)
137. Forrester, A.I., Keane, A.J.: Recent advances in surrogate-based optimization. *Progress in Aerospace Sciences* **45**(1), 50-79 (2009).
138. Box, G.E., Wilson, K.B.: On the experimental attainment of optimum conditions. *Journal of the Royal Statistical Society. Series B (Methodological)* **13**, 1-45 (1951).
139. Khuri, A.I., Mukhopadhyay, S.: Response surface methodology. *Wiley Interdisciplinary Reviews: Computational Statistics* **2**(2), 128-149 (2010).
140. Khuri, A.I., Cornell, J.A.: Response surfaces: designs and analyses, vol. 152. CRC press, (1996)
141. Geladi, P., Kowalski, B.R.: Partial least-squares regression: a tutorial. *Analytica chimica acta* **185**, 1-17 (1986).
142. Wold, S., Sjöström, M., Eriksson, L.: PLS-regression: a basic tool of chemometrics. *Chemometrics and intelligent laboratory systems* **58**(2), 109-130 (2001).
143. Nelson, P.R., Taylor, P.A., MacGregor, J.F.: Missing data methods in PCA and PLS: Score calculations with incomplete observations. *Chemometrics and intelligent laboratory systems* **35**(1), 45-65 (1996).
144. Shao, J.: Linear model selection by cross-validation. *Journal of the American statistical Association* **88**(422), 486-494 (1993).
145. Eriksson, L., Johansson, E., Kettaneh-Wold, N., Wold, S.: Multi-and megavariate data analysis. Part I: Basic Principles and Applications **2**, 425 (2001).
146. Rosipal, R.: Nonlinear partial least squares: An overview. *Chemoinformatics and advanced machine learning perspectives: complex computational methods and collaborative techniques*, 169-189 (2010).
147. Tomba, E., Facco, P., Bezzo, F., Barolo, M.: Latent variable modeling to assist the implementation of Quality-by-Design paradigms in pharmaceutical development and manufacturing: a review. *International journal of pharmaceutics* **457**(1), 283-297 (2013).

148. Agatonovic-Kustrin, S., Beresford, R.: Basic concepts of artificial neural network (ANN) modeling and its application in pharmaceutical research. *Journal of pharmaceutical and biomedical analysis* **22**(5), 717-727 (2000).
149. Simpson, P.K.: *Artificial neural systems: foundations, paradigms, applications, and implementations*. Pergamon, (1990)
150. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning internal representations by error propagation. In: *California Univ San Diego La Jolla Inst for Cognitive Science*, (1985)
151. Ripley, B.D.: *Pattern recognition and neural networks*. Cambridge university press, (2007)
152. Yu, H., Fu, J., Dang, L., Cheong, Y., Tan, H., Wei, H.: Prediction of the particle size distribution parameters in a high shear granulation process using a key parameter definition combined artificial neural network model. *Industrial & Engineering Chemistry Research* **54**(43), 10825-10834 (2015).
153. Masters, T.: *Practical neural network recipes in C++*. Morgan Kaufmann, (1993)
154. Basheer, I., Hajmeer, M.: Artificial neural networks: fundamentals, computing, design, and application. *Journal of microbiological methods* **43**(1), 3-31 (2000).
155. Orr, M.J.: Introduction to radial basis function networks. In: *Technical Report, Center for Cognitive Science, University of Edinburgh*, (1996)
156. Maher, H.M.: Development and validation of a stability-indicating HPLC-dad method with ANN optimization for the determination of diflunisal and naproxen in pharmaceutical tablets. *Journal of Liquid Chromatography & Related Technologies* **37**(5), 634-652 (2014).
157. Patel, T.B., Patel, L., Patel, T.R., Suhagia, B.: Artificial neural network as tool for quality by design in formulation development of solid dispersion of fenofibrate. *Bulletin of Pharmaceutical Research* **5**(1), 20-27 (2015).
158. Li, Y., Abbaspour, M.R., Grootendorst, P.V., Rauth, A.M., Wu, X.Y.: Optimization of controlled release nanoparticle formulation of verapamil hydrochloride using artificial neural networks with genetic algorithm and response surface methodology. *European Journal of Pharmaceutics and Biopharmaceutics* **94**, 170-179 (2015).
159. Cressie, N.: The origins of kriging. *Mathematical geology* **22**(3), 239-252 (1990).
160. Rasmussen, C.E., Williams, C.K.: *Gaussian processes for machine learning*, vol. 1. MIT press Cambridge, (2006)
161. Huang, D., Allen, T.T., Notz, W.I., Zeng, N.: Global optimization of stochastic black-box systems via sequential kriging meta-models. *Journal of global optimization* **34**(3), 441-466 (2006).
162. Ankenman, B., Nelson, B.L., Staum, J.: Stochastic kriging for simulation metamodeling. *Operations research* **58**(2), 371-382 (2010).
163. Kleijnen, J.P., Sargent, R.G.: A methodology for fitting and validating metamodels in simulation. *European Journal of Operational Research* **120**(1), 14-29 (2000).
164. Meckesheimer, M., Booker, A.J., Barton, R.R., Simpson, T.W.: Computationally inexpensive metamodel assessment strategies. *AIAA journal* **40**(10), 2053-2060 (2002).
165. Mitchell, T.J., Morris, M.D.: Bayesian design and analysis of computer experiments: two examples. *Statistica Sinica*, 359-379 (1992).
166. Queipo, N.V., Haftka, R.T., Shyy, W., Goel, T., Vaidyanathan, R., Tucker, P.K.: Surrogate-based analysis and optimization. *Progress in aerospace sciences* **41**(1), 1-28 (2005).
167. Lin, Y.: *An efficient robust concept exploration method and sequential exploratory experimental design*. Georgia Institute of Technology (2004)
168. Boggs, P.T., Tolle, J.W.: Sequential quadratic programming. *Acta numerica* **4**, 1-51 (1995).
169. Schittkowski, K.: More test examples for nonlinear programming codes. *Lecture Notes in Econom. and Math. Systems* **282** (1987).
170. Binder, T., Blank, L., Bock, H.G., Bulirsch, R., Dahmen, W., Diehl, M., Kronseder, T., Marquardt, W., Schlöder, J.P., von Stryk, O.: Introduction to model based optimization of

- chemical processes on moving horizons. In: Online optimization of large scale systems. pp. 295-339. Springer, (2001)
171. Bonnans, J.F., Panier, E.R., Tits, A.L., Zhou, J.L.: Avoiding the Maratos effect by means of a nonmonotone line search. II. Inequality constrained problems—feasible iterates. *SIAM Journal on Numerical Analysis* **29**(4), 1187-1202 (1992).
  172. Bussieck, M.R., Meeraus, A.: General algebraic modeling system (GAMS). *Applied Optimization* **88**, 137-158 (2004).
  173. Fourer, R., Gay, D.M., Kernighan, B.W.: *AMPL: A mathematical programming language*. Citeseer, (1987)
  174. Sen, M., Chaudhury, A., Singh, R., Ramachandran, R.: Two-dimensional population balance model development and validation of pharmaceutical crystallization processes. *American Journal of Modern Chemical Engineering* **1**, 13-29 (2014).
  175. Acevedo, D., Tandy, Y., Nagy, Z.K.: Multiobjective optimization of an unseeded batch cooling crystallizer for shape and size manipulation. *Industrial & Engineering Chemistry Research* **54**(7), 2156-2166 (2015).
  176. Yang, Y., Nagy, Z.K.: Combined Cooling and Antisolvent Crystallization in Continuous Mixed Suspension, Mixed Product Removal Cascade Crystallizers: Steady-State and Startup Optimization. *Industrial & Engineering Chemistry Research* **54**(21), 5673-5682 (2015).
  177. Gagnon, F., Desbiens, A., Poulin, É., Lapointe-Garant, P.-P., Simard, J.-S.: Nonlinear model predictive control of a batch fluidized bed dryer for pharmaceutical particles. *Control Engineering Practice* **64**, 88-101 (2017).
  178. Wang, J., Lakerveld, R.: Continuous Membrane-Assisted Crystallization To Increase the Attainable Product Quality of Pharmaceuticals and Design Space for Operation. *Industrial & Engineering Chemistry Research* **56**(19), 5705-5714 (2017).
  179. Emenike, V.N., Schenkendorf, R., Krewer, U.: A systematic reactor design approach for the synthesis of active pharmaceutical ingredients. *European Journal of Pharmaceutics and Biopharmaceutics* (2017).
  180. Rios, L.M., Sahinidis, N.V.: Derivative-free optimization: a review of algorithms and comparison of software implementations. *Journal of Global Optimization* **56**(3), 1247-1293 (2013).
  181. Nelder, J.A., Mead, R.: A simplex method for function minimization. *The computer journal* **7**(4), 308-313 (1965).
  182. McKinnon, K.I.: Convergence of the Nelder--Mead Simplex Method to a Nonstationary Point. *SIAM Journal on Optimization* **9**(1), 148-158 (1998).
  183. Conn, A.R., Scheinberg, K., Vicente, L.N.: *Introduction to derivative-free optimization*. SIAM, (2009)
  184. Torczon, V.: On the convergence of pattern search algorithms. *SIAM Journal on optimization* **7**(1), 1-25 (1997).
  185. Hooke, R., Jeeves, T.A.: "Direct Search" Solution of Numerical and Statistical Problems. *Journal of the ACM (JACM)* **8**(2), 212-229 (1961).
  186. Audet, C.: A survey on direct search methods for blackbox optimization and their applications. In: *Mathematics Without Boundaries*. pp. 31-56. Springer, (2014)
  187. Lewis, R.M., Torczon, V.: Pattern search algorithms for bound constrained minimization. *SIAM Journal on Optimization* **9**(4), 1082-1099 (1999).
  188. Lewis, R.M., Torczon, V.: Pattern search methods for linearly constrained minimization. *SIAM Journal on Optimization* **10**(3), 917-941 (2000).
  189. Kolda, T.G., Lewis, R.M., Torczon, V.: Optimization by direct search: New perspectives on some classical and modern methods. *SIAM review* **45**(3), 385-482 (2003).

190. Grimard, J., Dewasme, L., Thiry, J., Krier, F., Evrard, B., Wouwer, A.V.: Modeling, sensitivity analysis and parameter identification of a twin screw extruder. *IFAC-PapersOnLine* **49**(7), 1127-1132 (2016).
191. Besenhard, M.O., Chaudhury, A., Vetter, T., Ramachandran, R., Khinast, J.G.: Evaluation of parameter estimation methods for crystallization processes modeled via population balance equations. *Chemical Engineering Research and Design* **94**, 275-289 (2015).
192. Paul, J., Jensen, S., Dukart, A., Cornelissen, G.: Optimization of a preparative multimodal ion exchange step for purification of a potential malaria vaccine. *Journal of Chromatography A* **1366**, 38-44 (2014).
193. Zuo, J., Gao, Y., Bou-Chacra, N., Löbenberg, R.: Evaluation of the DDSolver software applications. *BioMed research international* **2014** (2014).
194. Xi, J., Yuan, J.E., Si, X.A., Hasbany, J.: Numerical optimization of targeted delivery of charged nanoparticles to the ostiomeatal complex for treatment of rhinosinusitis. *International journal of nanomedicine* **10**, 4847 (2015).
195. Moudjari, Y., Louaer, W., Meniai, A.: Modeling of the solubility of Naproxen and Trimethoprim in different solvents at different temperature. In: *MATEC Web of Conferences 2013*, p. 01057. EDP Sciences
196. Holland, J.H.: *Adaptation in natural and artificial systems*. The University of Michigan Press, Ann Arbor (1975).
197. Whitley, D.: A genetic algorithm tutorial. *Statistics and computing* **4**(2), 65-85 (1994).
198. Kumar, M., Husian, M., Upreti, N., Gupta, D.: Genetic algorithm: Review and application. *International Journal of Information Technology and Knowledge Management* **2**(2), 451-454 (2010).
199. Ridder, B.J., Majumder, A., Nagy, Z.K.: Population balance model-based multiobjective optimization of a multisegment multiaddition (MSMA) continuous plug-flow antisolvent crystallizer. *Industrial & Engineering Chemistry Research* **53**(11), 4387-4397 (2014).
200. Zaki, M.R., Varshosaz, J., Fathi, M.: Preparation of agar nanospheres: comparison of response surface and artificial neural network modeling by a genetic algorithm approach. *Carbohydrate polymers* **122**, 314-320 (2015).
201. Allmendinger, R., Simaria, A.S., Turner, R., Farid, S.S.: Closed - loop optimization of chromatography column sizing strategies in biopharmaceutical manufacture. *Journal of Chemical Technology and Biotechnology* **89**(10), 1481-1490 (2014).
202. Rostamizadeh, K., Rezaei, S., Abdouss, M., Sadighian, S., Arish, S.: A hybrid modeling approach for optimization of PMAA - chitosan - PEG nanoparticles for oral insulin delivery. *RSC Advances* **5**(85), 69152-69160 (2015).
203. Kalkhoran, A.H.Z., Vahidi, O., Naghib, S.M.: A new mathematical approach to predict the actual drug release from hydrogels. *European Journal of Pharmaceutical Sciences* **111**, 303-310 (2018).
204. Wang, Z., He, Z., Zhang, L., Zhang, H., Zhang, M., Wen, X., Quan, G., Huang, X., Pan, X., Wu, C.: Optimization of a doxycycline hydroxypropyl-  $\beta$  -cyclodextrin inclusion complex based on computational modeling. *Acta Pharmaceutica Sinica B* **3**(2), 130-139 (2013).
205. Huan, X., Marzouk, Y.M.: Sequential Bayesian optimal experimental design via approximate dynamic programming. *arXiv preprint arXiv:1604.08320* (2016).
206. Lam, R., Willcox, K., Wolpert, D.H.: Bayesian Optimization with a Finite Budget: An Approximate Dynamic Programming Approach. In: *Advances in Neural Information Processing Systems 2016*, pp. 883-891
207. Gutmann, H.-M.: A radial basis function method for global optimization. *Journal of global optimization* **19**(3), 201-227 (2001).

208. Regis, R.G., Shoemaker, C.A.: Constrained global optimization of expensive black box functions using radial basis functions. *Journal of Global optimization* **31**(1), 153-171 (2005).
209. Regis, R.G.: Constrained optimization by radial basis function interpolation for high-dimensional expensive black-box problems with infeasible initial points. *Engineering Optimization* **46**(2), 218-243 (2014).
210. Luna, M., Martínez, E.: A Bayesian approach to run-to-run optimization of animal cell bioreactors using probabilistic tendency models. *Industrial & Engineering Chemistry Research* **53**(44), 17252-17266 (2014).
211. Mehrian, M., Guyot, Y., Papantoniou, I., Olofsson, S., Sonnaert, M., Misener, R., Geris, L.: Maximizing neotissue growth kinetics in a perfusion bioreactor: an in silico strategy using model reduction and Bayesian optimization. *Biotechnology and bioengineering* (2017).
212. Metta, N., Ierapetritou, M., Ramachandran, R.: A multiscale DEM-PBM approach for a continuous comilling process using a mechanistically developed breakage kernel. *Chemical Engineering Science* (2017). doi:<https://doi.org/10.1016/j.ces.2017.12.016>
213. Halemane, K.P., Grossmann, I.E.: Optimal process design under uncertainty. *AIChE Journal* **29**(3), 425-433 (1983).
214. Grossmann, I.E., Floudas, C.A.: Active constraint strategy for flexibility analysis in chemical processes. *Computers & Chemical Engineering* **11**(6), 675-693 (1987).
215. Banerjee, I., Ierapetritou, M.G.: Feasibility evaluation of nonconvex systems using shape reconstruction techniques. *Industrial & engineering chemistry research* **44**(10), 3638-3647 (2005).
216. Adi, V.S.K., Laxmidewi, R., Chang, C.-T.: An effective computation strategy for assessing operational flexibility of high-dimensional systems with complicated feasible regions. *Chemical Engineering Science* **147**, 137-149 (2016).
217. Michalewicz, Z., Schoenauer, M.: Evolutionary algorithms for constrained parameter optimization problems. *Evolutionary computation* **4**(1), 1-32 (1996).
218. Goyal, V., Ierapetritou, M.G.: Determination of operability limits using simplicial approximation. *AIChE journal* **48**(12), 2902-2909 (2002).
219. Banerjee, I., Pal, S., Maiti, S.: Computationally efficient black-box modeling for feasibility analysis. *Computers & chemical engineering* **34**(9), 1515-1521 (2010).
220. Boukouvala, F., Ierapetritou, M.G.: Feasibility analysis of black-box processes using an adaptive sampling Kriging-based method. *Computers & Chemical Engineering* **36**, 358-368 (2012).
221. Park, J., Sandberg, I.W.: Universal approximation using radial-basis-function networks. *Neural computation* **3**(2), 246-257 (1991).
222. Björkman, M., Holmström, K.: Global optimization of costly nonconvex functions using radial basis functions. *Optimization and Engineering* **1**(4), 373-397 (2000).
223. Koch, P., Bagheri, S., Konen, W., Foussette, C., Krause, P., Bäck, T.: A new repair method for constrained optimization. In: *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation 2015*, pp. 273-280. ACM
224. Simpson, T.W., Lin, D.K., Chen, W.: Sampling strategies for computer experiments: design and analysis. *International Journal of Reliability and Applications* **2**(3), 209-240 (2001).
225. Rasmussen, C.E.: Gaussian processes in machine learning. In: *Advanced lectures on machine learning*. pp. 63-71. Springer, (2004)
226. Lophaven, S.N., Nielsen, H.B., Søndergaard, J.: *DACE: a Matlab kriging toolbox*, vol. 2. Citeseer, (2002)
227. Regis, R.G., Shoemaker, C.A.: Improved strategies for radial basis function methods for global optimization. *Journal of Global Optimization* **37**(1), 113-135 (2007).

228. Wild, S.M., Shoemaker, C.: Global convergence of radial basis function trust-region algorithms for derivative-free optimization. *SIAM Review* **55**(2), 349-371 (2013).
229. Holmström, K., Göran, A.O., Edvall, M.M.: User's Guide for Tomlab/Conopt. In. (2007)
230. Sasena, M.J., Papalambros, P., Goovaerts, P.: Exploration of metamodeling sampling criteria for constrained global optimization. *Engineering optimization* **34**(3), 263-278 (2002).
231. Goel, T., Haftka, R.T., Shyy, W., Queipo, N.V.: Ensemble of surrogates. *Structural and Multidisciplinary Optimization* **33**(3), 199-216 (2007).
232. Koziel, S., Michalewicz, Z.: Evolutionary algorithms, homomorphous mappings, and constrained parameter optimization. *Evolutionary computation* **7**(1), 19-44 (1999).
233. Floudas, C.A., Pardalos, P.M.: A collection of test problems for constrained global optimization algorithms, vol. 455. Springer Science & Business Media, (1990)
234. Hsu, S.-H., Reklaitis, G.V., Venkatasubramanian, V.: Modeling and control of roller compaction for pharmaceutical manufacturing. Part I: Process dynamics and control framework. *Journal of Pharmaceutical Innovation* **5**(1-2), 14-23 (2010).
235. Lionberger, R.A., Lee, S.L., Lee, L., Raw, A., Lawrence, X.Y.: Quality by design: concepts for ANDAs. *The AAPS journal* **10**(2), 268-276 (2008).
236. Allison, G., Cain, Y.T., Cooney, C., Garcia, T., Bizjak, T.G., Holte, O., Jagota, N., Komar, B., Korakianiti, E., Kourti, D.: Regulatory and quality considerations for continuous manufacturing. May 20 - 21, 2014 Continuous Manufacturing Symposium. *Journal of pharmaceutical sciences* **104**(3), 803-812 (2015).
237. Leuenberger, H.: New trends in the production of pharmaceutical granules: batch versus continuous processing. *European journal of pharmaceutics and biopharmaceutics* **52**(3), 289-296 (2001).
238. Schaber, S.D., Gerogiorgis, D.I., Ramachandran, R., Evans, J.M., Barton, P.I., Trout, B.L.: Economic analysis of integrated continuous and batch pharmaceutical manufacturing: a case study. *Industrial & Engineering Chemistry Research* **50**(17), 10083-10092 (2011).
239. Byrn, S., Futran, M., Thomas, H., Jayjock, E., Maron, N., Meyer, R.F., Myerson, A.S., Thien, M.P., Trout, B.L.: Achieving continuous manufacturing for final dosage formation: Challenges and how to meet them. May 20 - 21, 2014 Continuous Manufacturing Symposium. *Journal of pharmaceutical sciences* **104**(3), 792-802 (2015).
240. Dubey, A., Sarkar, A., Ierapetritou, M., Wassgren, C.R., Muzzio, F.J.: Computational approaches for studying the granular dynamics of continuous blending processes, 1 - DEM based methods. *Macromolecular Materials and Engineering* **296**(3 - 4), 290-307 (2011).
241. Sen, M., Singh, R., Vanarase, A., John, J., Ramachandran, R.: Multi-dimensional population balance modeling and experimental validation of continuous powder mixing processes. *Chemical engineering science* **80**, 349-360 (2012).
242. Barrasso, D., Ramachandran, R.: A comparison of model order reduction techniques for a four-dimensional population balance model describing multi-component wet granulation processes. *Chemical engineering science* **80**, 380-392 (2012).
243. Gao, Y., Muzzio, F.J., Ierapetritou, M.G.: A review of the residence time distribution (RTD) applications in solid unit operations. *Powder technology* **228**, 416-423 (2012).
244. Sin, G., Germaey, K.V., Lantz, A.E.: Good modeling practice for PAT applications: Propagation of input uncertainty and sensitivity analysis. *Biotechnology progress* **25**(4), 1043-1053 (2009).
245. Gétaz, D., Butté, A., Morbidelli, M.: Model-based design space determination of peptide chromatographic purification processes. *Journal of Chromatography A* **1284**, 80-87 (2013).
246. Wang, Z., Ierapetritou, M.: A Novel Feasibility Analysis Method for Black - Box Processes using a Radial Basis Function Adaptive Sampling Approach. *AIChE Journal* (2016).

247. Sobol', I.y.M.: On the distribution of points in a cube and the approximate evaluation of integrals. *Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki* **7**(4), 784-802 (1967).
248. Hamby, D.: A comparison of sensitivity analysis techniques. *Health physics* **68**(2), 195-204 (1995).
249. McKay, M.D., Beckman, R.J., Conover, W.J.: A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics* **42**(1), 55-61 (2000).
250. Pronzato, L., Müller, W.G.: Design of computer experiments: space filling and beyond. *Statistics and Computing* **22**(3), 681-701 (2012).
251. Forrester, A., Sobester, A., Keane, A.: *Engineering design via surrogate modelling: a practical guide*. John Wiley & Sons, (2008)
252. Holmström, K., Edvall, M.M., Göran, A.O.: TOMLAB for large-scale robust optimization. In: *Proceedings of the Nordic MATLAB Conference 2003*, pp. 218-222
253. Process Systems Enterprise.  
<https://www.psenterprise.com/products/gproms/objects/gomatlab>. Accessed 30 Jan 2017
254. Engisch, W.E., Muzzio, F.J.: Feedrate deviations caused by hopper refill of loss-in-weight feeders. *Powder Technology* **283**, 389-400 (2015).
255. Swaney, R.E., Grossmann, I.E.: An index for operational flexibility in chemical process design. Part I: Formulation and theory. *AIChE Journal* **31**(4), 621-630 (1985).
256. Swaney, R.E., Grossmann, I.E.: An index for operational flexibility in chemical process design. Part II: Computational algorithms. *AIChE Journal* **31**(4), 631-641 (1985).
257. Floudas, C.A., Gümüş, Z.H., Ierapetritou, M.G.: Global optimization in design under uncertainty: feasibility test and flexibility index problems. *Industrial & Engineering Chemistry Research* **40**(20), 4267-4282 (2001).
258. Pistikopoulos, E., Mazzuchi, T.: A novel flexibility analysis approach for processes with stochastic parameters. *Computers & Chemical Engineering* **14**(9), 991-1000 (1990).
259. Dimitriadis, V.D., Pistikopoulos, E.N.: Flexibility analysis of dynamic systems. *Industrial & Engineering Chemistry Research* **34**(12), 4451-4462 (1995).
260. Ierapetritou, M.G.: New approach for quantifying process feasibility: Convex and 1 - D quasi - convex regions. *AICHE journal* **47**(6), 1407-1417 (2001).
261. Wang, Z., Ierapetritou, M.: A novel feasibility analysis method for black - box processes using a radial basis function adaptive sampling approach. *AIChE Journal* **63**(2), 532-550 (2017).
262. Chen, X., Zhou, Q.: Sequential design strategies for mean response surface metamodeling via stochastic kriging with adaptive exploration and exploitation. *European Journal of Operational Research* **262**(2), 575-585 (2017).
263. Shao, X., Shi, L.: Enhancing feasibility determination with kriging metamodels. In: *Automation Science and Engineering (CASE), 2016 IEEE International Conference on 2016*, pp. 476-481. IEEE
264. Wang, Z., Wang, P.: An integrated performance measure approach for system reliability analysis. *Journal of Mechanical Design* **137**(2), 021406 (2015).
265. Sadoughi, M.K., Hu, C., MacKenzie, C.A., Eshghi, A.T., Lee, S.: Sequential exploration-exploitation with dynamic trade-off for efficient reliability analysis of complex engineered systems. *Structural and Multidisciplinary Optimization* **57**(1), 235-250 (2018).
266. Birjandi, M.R.S., Shahraki, F., Razzaghi, K.: Hydrogen network retrofit via flexibility analysis: The steady-state flexibility index. *Chemical Engineering Research and Design* **117**, 83-94 (2017).

267. Garcia-Munoz, S., Luciani, C.V., Vaidyaraman, S., Seibert, K.D.: Definition of Design Spaces Using Mechanistic Models and Geometric Projections of Probability Maps. *Organic Process Research & Development* **19**(8), 1012-1023 (2015).
268. Wang, Z., Escotet-Espinoza, M.S., Ierapetritou, M.: Process analysis and optimization of continuous pharmaceutical manufacturing using flowsheet models. *Computers & Chemical Engineering* (2017).
269. Sharifzadeh, M., Meghdari, M., Rashtchian, D.: Multi-objective design and operation of solid oxide fuel cell (SOFC) triple combined-cycle power generation systems: Integrating energy efficiency and operational safety. *Applied Energy* **185**, 345-361 (2017).
270. Sahay, N., Ierapetritou, M.: Flexibility assessment and risk management in supply chains. *AIChE Journal* **61**(12), 4166-4178 (2015).
271. Wang, H., Mastragostino, R., Swartz, C.L.: Flexibility analysis of process supply chain networks. *Computers & Chemical Engineering* **84**, 409-421 (2016).
272. Zhang, Q., Grossmann, I.E., Lima, R.M.: On the relation between flexibility analysis and robust optimization for linear systems. *AIChE Journal* **62**(9), 3109-3123 (2016).
273. Amaran, S., Sahinidis, N.V., Sharda, B., Bury, S.J.: Simulation optimization: A review of algorithms and applications. *4OR* **12**(4), 301-333 (2014).
274. Picheny, V., Ginsbourger, D., Richet, Y., Caplin, G.: Quantile-based optimization of noisy computer experiments with tunable precision. *Technometrics* **55**(1), 2-13 (2013).
275. Cressie, N.: *Statistics for spatial data*. John Wiley & Sons, (2015)
276. Kleijnen, J.P.: Kriging metamodeling in simulation: A review. *European journal of operational research* **192**(3), 707-716 (2009).
277. Martin, J.D., Simpson, T.W.: Use of kriging models to approximate deterministic computer models. *AIAA J* **43**(4), 853-863 (2005).
278. Jalali, H., Van Nieuwenhuysse, I., Picheny, V.: Comparison of Kriging-based algorithms for simulation optimization with heterogeneous noise. *European Journal of Operational Research* **261**(1), 279-301 (2017).
279. Žilinskas, A., Zhigljavsky, A.: Stochastic global optimization: a review on the occasion of 25 years of Informatica. *Informatica* **27**(2), 229-256 (2016).
280. Močkus, J.: On Bayesian methods for seeking the extremum. In: *Optimization Techniques IFIP Technical Conference 1975*, pp. 400-404. Springer
281. Žilinskas, A.: Single-step Bayesian search method for an extremum of functions of a single variable. *Cybernetics and Systems Analysis* **11**(1), 160-166 (1975).
282. Schonlau, M., Welch, W.J., Jones, D.: Global optimization with nonparametric function fitting. *Proceedings of the ASA, section on physical and engineering sciences*, 183-186 (1996).
283. Picheny, V., Wagner, T., Ginsbourger, D.: A benchmark of kriging-based infill criteria for noisy optimization. *Structural and Multidisciplinary Optimization* **48**(3), 607-626 (2013).
284. Ranjan, P., Bingham, D., Michailidis, G.: Sequential experiment design for contour estimation from complex computer codes. *Technometrics* **50**(4), 527-541 (2008).
285. Lophaven, S., Nielsen, H., Sondergaard, J., Toolbox, D.-A.M.K.: Technical University of Denmark. In: *Technical Report IMM-TR2002-12*, (2002)
286. Al-Asady, R.B., Dhenge, R.M., Hounslow, M.J., Salman, A.D.: Roller compactor: Determining the nip angle and powder compaction progress by indentation of the pre-compacted body. *Powder Technology* **300**, 107-119 (2016).
287. Fu, M.C., Glover, F.W., April, J.: Simulation optimization: a review, new developments, and applications. In: *Simulation conference, 2005 proceedings of the winter 2005*, p. 13 pp. IEEE
288. Rulliere, D., Faleh, A., Planchet, F., Youssef, W.: Exploring or reducing noise? *Structural and Multidisciplinary Optimization* **47**(6), 921-936 (2013).

289. Bazaraa, M.S., Sherali, H.D., Shetty, C.M.: Nonlinear programming: theory and algorithms. John Wiley & Sons, (2013)
290. Birge, J.R., Louveaux, F.: Introduction to stochastic programming. Springer Science & Business Media, (2011)
291. Amaran, S., Sahinidis, N.V., Sharda, B., Bury, S.J.: Simulation optimization: a review of algorithms and applications. *Annals of Operations Research* **240**(1), 351-380 (2016).
292. Deng, G., Ferris, M.C.: Extension of the direct optimization algorithm for noisy functions. Paper presented at the 2007 Winter Simulation Conference,
293. Chang, K.-H.: Stochastic Nelder – Mead simplex method – A new globally convergent direct search method for simulation optimization. *European journal of operational research* **220**(3), 684-694 (2012).
294. Xu, J., Nelson, B.L., Hong, J.: Industrial strength COMPASS: A comprehensive algorithm and software for optimization via simulation. *ACM Transactions on Modeling and Computer Simulation (TOMACS)* **20**(1), 3 (2010).
295. Lee, L.H., Chen, C., Chew, E.P., Li, J., Pujowidianto, N.A., Zhang, S.: A review of optimal computing budget allocation algorithms for simulation optimization problem. *International Journal of Operations Research* **7**(2), 19-31 (2010).
296. Chen, C.-h., Lee, L.H.: Stochastic simulation optimization: an optimal computing budget allocation, vol. 1. World scientific, (2011)
297. Spall, J.C.: Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE transactions on automatic control* **37**(3), 332-341 (1992).
298. Spall, J.C.: Simultaneous perturbation stochastic approximation. *Introduction to Stochastic Search and Optimization: Estimation, Simulation, and Control*, 176-207 (2003).
299. Jakobsson, S., Patriksson, M., Rudholm, J., Wojciechowski, A.: A method for simulation based optimization using radial basis functions. *Optimization and Engineering* **11**(4), 501-532 (2010).
300. Vazquez, E., Villemonteix, J., Sidorkiewicz, M.: Global optimization based on noisy evaluations: an empirical study of two statistical approaches. Paper presented at the *Journal of Physics: Conference Series*,
301. J. Forrester, A.I., Keane, A.J., Bressloff, N.W.: Design and analysis of "Noisy" computer experiments. *AIAA journal* **44**(10), 2331-2339 (2006).
302. Yin, J., Ng, S.H., Ng, K.M.: Kriging metamodel with modified nugget-effect: The heteroscedastic variance case. *Computers & Industrial Engineering* **61**(3), 760-777 (2011).
303. Quan, N., Yin, J., Ng, S.H., Lee, L.H.: Simulation optimization via kriging: a sequential search using expected improvement with computing budget constraints. *Iie Transactions* **45**(7), 763-780 (2013).
304. Ji, Y., Kim, S.: Regularized radial basis function models for stochastic simulation. Paper presented at the Proceedings of the 2014 Winter Simulation Conference,
305. Chen, X., Zhou, Q.: Sequential experimental designs for stochastic kriging. Paper presented at the Proceedings of the 2014 Winter Simulation Conference,
306. Kleijnen, J.P., Mehdad, E.: Estimating the variance of the predictor in stochastic Kriging. *Simulation Modelling Practice and Theory* **66**, 166-173 (2016).
307. Rasmussen, C.E., Williams, C.K.: Gaussian processes for machine learning. the MIT Press (2006).
308. Kersting, K., Plagemann, C., Pfaff, P., Burgard, W.: Most likely heteroscedastic Gaussian process regression. Paper presented at the Proceedings of the 24th international conference on Machine learning,
309. Boukouvalas, A., Cornford, D.: Learning heteroscedastic Gaussian processes for complex datasets. In. Technical report, Neural Computing Research Group, Aston University, (2009)

310. Schneider, M., Ertel, W.: Robot learning by demonstration with local gaussian process regression. Paper presented at the Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on,
311. Muñoz-González, L., Lázaro-Gredilla, M., Figueiras-Vidal, A.R.: Heteroscedastic Gaussian process regression using expectation propagation. Paper presented at the 2011 IEEE International Workshop on Machine Learning for Signal Processing,
312. Lázaro-gredilla, M., Titsias, M.K.: Variational heteroscedastic Gaussian process regression. Paper presented at the Proceedings of the 28th International Conference on Machine Learning (ICML-11),
313. Mehdad, E., Kleijnen, J.P.: Efficient Global Optimization for Black-Box Simulation via Sequential Intrinsic Kriging. (2015).
314. Jalali, H., Van Nieuwenhuysse, I., Picheny, V.: Comparison of Kriging-based algorithms for simulation optimization with heterogeneous noise. *European Journal of Operational Research* (2017).
315. Kuindersma, S.R., Grupen, R.A., Barto, A.G.: Variable risk control via stochastic optimization. *The International Journal of Robotics Research* **32**(7), 806-825 (2013).
316. Bull, A.D.: Convergence rates of efficient global optimization algorithms. *Journal of Machine Learning Research* **12**(Oct), 2879-2904 (2011).
317. Eaton, M.L.: *Multivariate statistics: A vector space approach*. JOHN WILEY & SONS, INC., 605 THIRD AVE., NEW YORK, NY 10158, USA, 1983, 512 (1983).
318. Moré, J.J., Wild, S.M.: Benchmarking derivative-free optimization algorithms. *SIAM Journal on Optimization* **20**(1), 172-191 (2009).
319. Quttineh, N.-H., Holmström, K.: Implementation of a One-Stage Efficient Global Optimization (EGO) Algorithm. *Models and Methods for Costly Global Optimization and Military Decision Support Systems*, 91 (2012).
320. Wang, Z., Escotet-Espinoza, M.S., Ierapetritou, M.: Process Analysis and Optimization of Continuous Pharmaceutical Manufacturing using Flowsheet Models. *Computers & Chemical Engineering* (in press. DOI: <http://dx.doi.org/doi:10.1016/j.compchemeng.2017.02.030>). doi:<http://dx.doi.org/doi:10.1016/j.compchemeng.2017.02.030>
321. Fu, M.C.: Optimization for simulation: Theory vs. practice. *INFORMS Journal on Computing* **14**(3), 192-215 (2002).
322. Tekin, E., Sabuncuoglu, I.: Simulation optimization: A comprehensive review on theory and applications. *IIE transactions* **36**(11), 1067-1081 (2004).
323. Jalali, H., Nieuwenhuysse, I.V.: Simulation optimization in inventory replenishment: a classification. *IIE Transactions* **47**(11), 1217-1235 (2015).
324. Kim, S., Henderson, S.G.: The mathematics of continuous-variable simulation optimization. In: *Proceedings of the 40th Conference on Winter Simulation 2008*, pp. 122-132. Winter Simulation Conference
325. Robbins, H., Monro, S.: A stochastic approximation method. *The annals of mathematical statistics*, 400-407 (1951).
326. Kushner, H.J., Clark, D.S.: *Stochastic approximation methods for constrained and unconstrained systems*, vol. 26. Springer Science & Business Media, (2012)
327. Jones, M.H., White, K.P.: Stochastic approximation with simulated annealing as an approach to global discrete-event simulation optimization. In: *Simulation Conference, 2004. Proceedings of the 2004 Winter 2004*. IEEE
328. Bhatnagar, S., Hemachandra, N., Mishra, V.K.: Stochastic approximation algorithms for constrained optimization via simulation. *ACM Transactions on Modeling and Computer Simulation (TOMACS)* **21**(3), 15 (2011).
329. Trosset, M.W.: On the use of direct search methods for stochastic optimization. In: *Rice University, Department of 2000*. Citeseer

330. Kim, S., Zhang, D.: Convergence properties of direct search methods for stochastic optimization. In: Proceedings of the Winter Simulation Conference 2010, pp. 1003-1011. Winter Simulation Conference
331. Chang, K.-H., Lu, H.-K.: Quantile-based simulation optimization with inequality constraints: Methodology and applications. *IEEE Transactions on Automation Science and Engineering* **13**(2), 701-708 (2016).
332. Andradóttir, S.: An overview of simulation optimization via random search. *Handbooks in operations research and management science* **13**, 617-631 (2006).
333. Lacksonen, T.: Empirical comparison of search algorithms for discrete event simulation. *Computers & Industrial Engineering* **40**(1-2), 133-148 (2001).
334. Klassen, K.J., Yoogalingam, R.: Improving performance in outpatient appointment services with a simulation optimization approach. *Production and Operations Management* **18**(4), 447-458 (2009).
335. Bhosekar, A., Ierapetritou, M.: Advances in surrogate based modeling, feasibility analysis and optimization: A review. *Computers & Chemical Engineering* (2017).
336. Barton, R.R., Meckesheimer, M.: Metamodel-based simulation optimization. *Handbooks in operations research and management science* **13**, 535-574 (2006).
337. Angün, E., Kleijnen, J., den Hertog, D., Gürkan, G.: Response surface methodology with stochastic constraints for expensive simulation. *Journal of the operational research society* **60**(6), 735-746 (2009).
338. Augustin, F., Marzouk, Y.: A trust-region method for derivative-free nonlinear constrained stochastic optimization. *arXiv preprint arXiv:1703.04156* (2017).
339. Nezhad, A.M., Mahlooji, H.: An artificial neural network meta-model for constrained simulation optimization. *Journal of the Operational Research Society* **65**(8), 1232-1244 (2014).
340. Li, Y., Ng, S.H., Xie, M., Goh, T.: A systematic comparison of metamodeling techniques for simulation optimization in decision support systems. *Applied Soft Computing* **10**(4), 1257-1273 (2010).
341. Mockus, J.: Bayesian approach to global optimization: theory and applications, vol. 37. Springer Science & Business Media, (2012)
342. Žilinskas, A.: Axiomatic characterization of a global optimization algorithm and investigation of its search strategy. *Operations Research Letters* **4**(1), 35-39 (1985).
343. Kleijnen, J.P.: Simulation-optimization via Kriging and bootstrapping: a survey. *Journal of Simulation* **8**(4), 241-250 (2014).
344. Schonlau, M.: Computer experiments and global optimization. (1997).
345. Parr, J., Holden, C.M., Forrester, A.I., Keane, A.J.: Review of efficient surrogate infill sampling criteria with constraint handling. (2010).
346. Parr, J.M., Forrester, A.I., Keane, A.J., Holden, C.M.: Enhancing infill sampling criteria for surrogate-based constrained optimization. *Journal of Computational Methods in Sciences and Engineering* **12**(1, 2), 25-45 (2012).
347. Parr, J., Keane, A., Forrester, A.I., Holden, C.: Infill sampling criteria for surrogate-based optimization with constraint handling. *Engineering Optimization* **44**(10), 1147-1166 (2012).
348. ur Rehman, S., Langelaar, M.: Expected improvement based infill sampling for global robust optimization of constrained problems. *Optimization and Engineering* **18**(3), 723-753 (2017).
349. Tutum, C.C., Deb, K., Baran, I.: Constrained efficient global optimization for pultrusion process. *Materials and manufacturing processes* **30**(4), 538-551 (2015).
350. Bagheri, S., Konen, W., Allmendinger, R., Branke, J., Deb, K., Fieldsend, J., Quagliarella, D., Sindhya, K.: Constraint handling in efficient global optimization. In: Proceedings of the Genetic and Evolutionary Computation Conference 2017, pp. 673-680. ACM

351. Wang, Z., Escotet-Espinoza, M.S., Singh, R., Ierapetritou, M.: Surrogate-based Optimization for Pharmaceutical Manufacturing Processes. In: Computer Aided Chemical Engineering, vol. 40. pp. 2797-2802. Elsevier, (2017)
352. Vazquez, E., Villemonteix, J., Sidorkiewicz, M., Walter, E.: Global optimization based on noisy evaluations: an empirical study of two statistical approaches. In: Journal of Physics: Conference Series 2008, vol. 1, p. 012100. IOP Publishing
353. Mehdad, E., Kleijnen, J.P.: Stochastic intrinsic Kriging for simulation metamodeling. Applied Stochastic Models in Business and Industry (2014).
354. Lázaro-Gredilla, M., Titsias, M.: Variational heteroscedastic Gaussian process regression. (2011).
355. Mehdad, E., Kleijnen, J.P.: Efficient global optimisation for black-box simulation via sequential intrinsic Kriging. Journal of the Operational Research Society, 1-13 (2018).
356. Wang, Z., Ierapetritou, M.: A Novel Surrogate-Based Optimization Method for Black-Box Simulation with Heteroscedastic Noise. Industrial & Engineering Chemistry Research **56**(38), 10720-10732 (2017).
357. Kleijnen, J.P., Van Beers, W., Van Nieuwenhuysse, I.: Constrained optimization in expensive simulation: Novel approach. European journal of operational research **202**(1), 164-174 (2010).
358. Wang, Z., Ierapetritou, M.: Surrogate-based feasibility analysis for black-box stochastic simulations with heteroscedastic noise. Journal of Global Optimization (2018). doi:<https://doi.org/10.1007/s10898-018-0615-4>
359. Le Digabel, S.: Algorithm 909: NOMAD: Nonlinear optimization with the MADS algorithm. ACM Transactions on Mathematical Software (TOMS) **37**(4), 44 (2011).
360. Runarsson, T.P., Yao, X.: Search biases in constrained evolutionary optimization. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) **35**(2), 233-243 (2005).
361. Boukouvala, F., Floudas, C.A.: ARGONAUT: Algorithms for Global Optimization of coNstrained grey-box compUTational problems. Optimization Letters **11**(5), 895-913 (2017).
362. Wang, Z., Escotet-Espinoza, M.S., Ierapetritou, M.: Process analysis and optimization of continuous pharmaceutical manufacturing using flowsheet models. Computers & Chemical Engineering **107**, 77-91 (2017).
363. Gomez, S., Levy, A.V.: The tunnelling method for solving the constrained global optimization problem with several non-connected feasible regions. In: Numerical analysis. pp. 34-47. Springer, (1982)
364. Sasena, M.J.: Flexibility and efficiency enhancements for constrained global design optimization with kriging approximations. University of Michigan Ann Arbor (2002)
365. Habib, A., Singh, H.K., Ray, T.: A study on the effectiveness of constraint handling schemes within Efficient Global Optimization framework. In: Computational Intelligence (SSCI), 2016 IEEE Symposium Series on 2016, pp. 1-8. IEEE
366. Liang, J., Runarsson, T.P., Mezura-Montes, E., Clerc, M., Suganthan, P.N., Coello, C.C., Deb, K.: Problem definitions and evaluation criteria for the CEC 2006 special session on constrained real-parameter optimization. Journal of Applied Mechanics **41**(8) (2006).