SUPPLY CHAIN OPTIMIZATION AND MODULAR PROCESS DESIGN USING MACHINE LEARNING-BASED FRAMEWORKS

By

ATHARV BHOSEKAR

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 Ierapetritou

And approved by

____

____

____

____

New Brunswick, New Jersey

October 2020

ABSTRACT OF THE DISSERTATION

SUPPLY CHAIN OPTIMIZATION AND MODULAR PROCESS DESIGN USING MACHINE LEARNING-BASED FRAMEWORKS

by ATHARV BHOSEKAR

Dissertation Director:

Marianthi Ierapetritou

Globalization and the sudden increase in the exchange of information, trade, and capital all around the world, driven by technological innovation, has given rise to complex global supply chain networks. Since optimally designing such networks can yield significant profits in the long run, supply chain optimization is an area of active interest. The motivation for the problem considered in this work is two-fold. First, in a modern supply chain network, data plays an important role. However, since traditional optimization solvers cannot readily make use of this data, there is a need for frameworks that can utilize data to make optimal decisions in such networks. Second, there is a growing interest in considering multiple levels of decisions while designing the supply chain. However, due to differences in scale, level of details, and computational expense of the resulting integrated model, the problem of integrated decision-making is challenging. This work aims to propose machine learning-based frameworks that address these challenges.

The problem of optimal inventory allocation is first solved in a multienterprise supply chain network where the supply chain model is available in the form of a complex simulation. Further, historical data of the process or data generated from process simulations is used to design the process while simultaneously considering the total cost of the process as well as the flexibility of the design obtained. The framework is applied to modular processes. It is shown that using machine learning-based frameworks, process-level details can be incorporated at the supply chain design stage. This approach allows quantitatively assessing the benefits of modular processes such as design standardization, reduced transportation cost due to decentralized manufacturing, and optimal production facility location. Finally, the study is extended to address the problem of multiperiod supply chain optimization under product demand uncertainty. The results demonstrate the efficacy of the machine learning-based optimization framework proposed in this work and yields a set of solutions that minimize the risk as well as the expected total cost of the supply chain network.

Acknowledgements

I have been encouraged by many people without whose support I would not be writing this thesis. I hope to express my gratitude to all of them though I know that words can never be enough. First, I am sincerely indebted to my advisor, Prof. Marianthi Ierapetritou, for her constant trust, support, and guidance throughout my Ph.D. Marianthi, I have learnt a lot from you that has made me a stronger researcher: about not giving up on any idea, about believing in the fundamentals, about breaking down a difficult problem, and about being patient. In the past four years, you have also inspired me to be a better person. I will always remember your positivity, discipline, curiosity to learn new things, and caring nature as I take the next step in my career. I consider myself fortunate to have had an advisor who took an active interest in my development as a researcher.

I would also like to thank my doctoral committee members, Dr. Rohit Ramachandran, Dr. Ioannis P. Androulakis, and Dr. David Coit. Throughout my Ph.D., all of you have been valuable influences. This work has benefited significantly by your guidance along the way. I am grateful to my funding sources. This work was funded by National Science Foundation.

I have been fortunate to get the chance to work with a group of wonderful people at Rutgers: Parham, Zilong, Sebastian, Lisia, Nirupa, Abhay, Ou, Pooja, Yingjie, Yue, Siddharth, Praneeth, Shu. I would especially like to thank Nihar for his support in the initial stages of my research and his help with understanding supply chain simulations. I would also like to thank Lisia for intense research discussions, which have helped my research significantly. I enjoyed working with you and your curiosity and hard work always inspired me. To Ou, thank you for being a great friend and for always cheering me up. Thanks to the many other friends I have had at Rutgers. Chaitanya, Rajan, Rohan, Abhishek, Muthu, Kavya, thank you all for the great time we had. I would also like to thank new group members that I met in the final year of my Ph.D. at University of Delaware.

Dare, Robert, Michael, Jayanth, Huayu, Chaoying; you have all made my time at University of Delaware enjoyable. I would like to thank Prof. Roth and Prof. Celik for their fantastic support as graduate advisors.

My interest in process systems engineering started with my master's at Carnegie Mellon University. I am greatly thankful to my advisor, Prof. Nick Sahinidis, who inspired me to pursue Ph.D. and introduced me to derivative-free optimization. During my work with Nick, I learnt a lot about computational aspects of optimization and the implementation of optimization algorithms. This knowledge has been proven extremely useful throughout my Ph.D. While I was at Carnegie Mellon, I was fortunate to make friends with wonderful people: Nikhil, Vatsal, Kashish.

I want to express special thanks to my lifetime friends, Aadish, Anjoli, Siddhesh. It was great to have you by my side. Finally, I owe my deepest debts of gratitude to my family overseas. I would like to thank my grandparents for their blessings and unconditional love. Above all, to my parents, none of this would have been possible without your endless love and support. Thank you for having faith in me and encouraging me to be a better person. I hope I will make you proud wherever the future leads me. I am finally very thankful to Bhavana for her support and for being one of the most genuinely kind people I’ve ever met.

Table of Contents

List of Figures

List of Tables

1 Introduction

1.1 Supply chain optimization

Globalization and the sudden increase in the exchange of information, trade, and capital all around the world, driven by technological innovation, has given rise to complex global supply chain networks. Since optimally designing such networks can yield significant profits in the long run, supply chain optimization is an area of active interest. As opposed to traditional supply chain networks, modern supply chain networks consist of entities that usually belong to different enterprises and are even spread geographically. Since entities in such supply chain networks operate based on their individual goals as opposed to a common goal in centralized networks, there is a growing interest towards developing new optimization frameworks that take into account different goals and operating policies of different entities in a supply chain [1]. Under such networks, the problem of optimal inventory allocation is known to have a significant impact on the service level and the total cost of the supply chain and thereby impacting the profit that an individual enterprise would make [2]. However, there are challenges associated with modeling and optimization of complex supply chain networks.

The first challenge to address this problem is related to modeling a complex network. The modeling approaches in the existing literature can be broadly classified as analytical approaches and simulation-based approaches. Analytical methods formulate the problem as a mixed integer linear or nonlinear programming problem and solve using state-of-the-art optimization solvers. Even though these approaches benefit from efficient optimization techniques, modeling complex interactions in a network with the help of equation-based models is not always possible. To overcome these shortcomings, simulation-based approaches utilize a detailed simulation model that captures details of the supply chain network. Techniques such as agent-based modeling allow a bottom-up approach towards building a simulation [3][4] and have been found useful in supply chain networks in various areas.

The second challenge is related to optimization using simulation models for which there is a growing need for the development of novel optimization frameworks. Simulations are often built using commercial software, where the information about the underlying network is not available in closed form. As the simulation models aim for more accuracy, the computational expense of running the simulation increases, and numerical evaluation of derivatives to guide the search towards optimum becomes difficult. As a result, traditional optimization techniques cannot be used to solve this problem. To overcome this challenge, derivative-free optimization (DFO) methods are used. DFO algorithms rely only on the data from simulation models or physical experiments and do not require a closed-form expression of the problem. A common strategy to achieve this is to build a machine learning-based model or a surrogate model that approximates the simulation using a limited amount of data. The models are then iteratively updated as more data is collected. It has been shown that for an effective optimization algorithm, the choice of the surrogate model plays an important role. Another challenge in DFO is that many available algorithms and software packages rely on the assumption of continuity of the response. Since the response may not be continuous in some applications, especially for the case of supply chain networks, where the objective function is dependent on several discrete decisions. For this reason, optimization of problems with discontinuous response needs special attention.

In this dissertation, the aim is to review several machine learning-based models along with their use in DFO algorithms. Using suitable models, for the problem of multienterprise supply chain optimization, this dissertation aims to propose a DFO framework that addresses discontinuity of the response. Throughout the dissertation, the terms machine learning-based models, data-driven models, and surrogate models are interchangeably used for models that address the problems related to regression (predicting a real-valued response). For the classification problem, the terms classifier or machine learning-based classifier are used.

1.2 Modular process supply chain optimization

Globalization and increasing market competition have been a significant impetus for changing the pace and nature of businesses and innovation around the world. More and more customer-orientated products are driving change in many industries, and product cycles are becoming shorter [5]. Modularization, process intensification, and design standardization are increasingly being recognized as critical factors to reduce the time to market for a product [6]. With a wide range of applications in the areas of gas conversion, solid conversion ammonia synthesis, CO_2 conversion, water purification, renewable energy, power generation, and chemical processing along with growing industrial interest, modular manufacturing provides a promising way forward for the process engineering [7][8].

Modular design involves the use of small and standardized modules of fixed size in a production process. Multiple identical devices may be assembled to achieve the desired production. Modular and distributed processes may not only contribute to decreases in distribution costs but also provide an alternative to overcome several manufacturing challenges. Small devices offer inherent safety and can be used for on-demand and on-site production of hazardous materials [9]. They provide a fast path to commercialization since challenges related to the scaling up of chemical processes are not substantial. Moreover, the time for construction of manufacturing facilities may be reduced, since modules can be preassembled in a shop and are not subject to delays related to weather and on-site inspections. Because of standardized units, the process of numbering up as a part of plant expansion becomes faster. Economically, as the standardized units or small modular plants are numbered up, vendors, as well as process engineers, gain experience. As a result of the learning curve, the vendors may be able to sell the equipment for a lower price, and process engineers can reduce the time-to-market. All these factors contribute to a relatively lower risk of investment related to small modular designs.

However, ensuring the feasibility of the process designed using a limited set of standardized modules is an important problem. In the presence of analytical equations for the underlying process, this problem is straightforward and can be solved using existing nonlinear or linear programming approaches. However, in many cases, the only available information is in the form of historical data or simulation models. In such cases, black-box or machine learning-based feasibility analysis methods are used. Black-box feasibility analysis methods rely on building a data-driven approximation of the feasible region[10]. There are two significant challenges in solving the black-box feasibility analysis problem. First, for process feasibility analysis, the existing literature relies on treating the whole process as a black box. This approach does not utilize feasibility information for individual units. Moreover, when designing the process based on several options for each module, the problem of process feasibility analysis may require a large number of surrogate models. The second challenge is related to flexibility analysis to hedge against exceptional realizations of process parameters[11]. In the presence of an equation-oriented model, the problem of flexibility analysis has been explored for over three decades and is still an active area of research[12][13]. In the absence of closed-form expressions, however, the existing literature is limited to feasibility analysis. This dissertation intends to address the challenges related to process feasibility analysis and flexibility analysis by using proposing machine learning-based feasibility analysis framework.

Even though the benefits of modular designs are well-known, quantifying their economic viability and their overall effect on the supply chain is a relatively underexplored problem. Since process feasibility analysis can be carried out for modular processes, several exciting problems can be addressed for supply chains with modular processing. This dissertation aims to propose an integrated design and supply chain optimization framework using machine learning-based feasibility analysis to ensure the feasibility of the process. This way, the cost savings due to design standardization can be quantified. Moreover, the tradeoff between centralized and distributed manufacturing can be assessed. Finally, this dissertation aims to extend the proposed framework for modular supply chain optimization to the problem of optimization under product demand uncertainty.

It should be noted that in the context of this work, modular designs refer to the design and construction of smaller chemical process units or even entire processes of fixed production capacities [14]. It is important to note that this definition includes the possibility of process intensification [15], transportable processing units [16], standardization of equipment modules [17], and even integrated or customized unit operations [8].

1.3 Outline of the dissertation

This dissertation is organized as follows. Machine learning-based methods for regression, classification, and background on the specific problems of optimization and feasibility analysis is provided in Chapter 2. Chapter 3 proposes a framework to solve the optimal inventory allocation problem for a simulation-based optimization problem for a multienterprise supply chain. Chapters 4-6 are aimed at solving the supply chain optimization problem for modular processing with a simultaneous consideration for process design. Chapter 4 first solves the design optimization of a modular process. In doing so, historical data of the process or data generated from process simulations is used to design the process while simultaneously considering the total cost of the process as well as the flexibility of the design obtained. Chapter 5 integrates supply chain optimization with process design optimization. It is shown that using machine learning-based frameworks, process-level details can be incorporated at the supply chain design stage. This approach allows quantitatively assessing the benefits of modular processes such as design standardization, reduced transportation cost due to decentralized manufacturing, and optimal production facility location. The problem of supply chain optimization is further extended to address the problem of multiperiod supply chain optimization under product demand uncertainty in Chapter 6. Finally, Chapter 7 provides a summary of the work and potential future directions for the research.

2 Machine learning-based methods

In this chapter, a background on machine learning-based methods is provided. Specific details related to the methods used in the dissertation are provided. In section 2.1, a review on machine learning-based regression models, or surrogate models is provided. Section 2.2 described support vector machines. Section 2.3 provides a comprehensive review of derivative-free optimization. Section 2.4 and section 2.5 offer a detailed overview of feasibility analysis and flexibility analysis, respectively. Section 2.6 reviews adaptive sampling approaches in the existing literature. Section 2.7 describes the validation metrics used in this work to validate machine learning-based models.

2.1 Surrogate models

In this section, frequently used approaches for obtaining the surrogate $\hat{f}(x)$ are discussed with a focus on the recent advances. The models that are designed to yield unbiased predictions at the sampled data are referred to as interpolation models, whereas models that are built by minimizing the error between given data and model prediction under a certain criterion are referred to as regression models. In this section, regression models such as linear regression, support vector regression are discussed followed by interpolation models such as RBF and Kriging. Finally, approaches utilizing more than one of these surrogates are discussed. With their ability to provide a quantitative measure of uncertainty in prediction, RBF and Kriging surrogates are the most popular choices for optimization and feasibility analysis. Therefore, special emphasis is given on these surrogates.

2.1.1 Linear regression

This is a commonly used approach where a surrogate is represented as a linear combination of the input variables as given by Eq. (1).

$$\hat{\mathrm{f}}(\mathrm{x}) = \mathrm{w}_0 + \sum_{j=1}^{d} \mathrm{x}_\mathrm{j}\mathrm{w}_\mathrm{j} \tag{1}$$

where x is a vector of size d; d is the number of variables; w is the vector of length $d + 1$. To obtain the weight vector, sum of squared errors between the actual data and the surrogate predicted value is minimized. The unconstrained minimization problem can be formulated as given by Eq. (2).

$$\min_{w} \left|\left|Xw - y\right|\right|_2^2 \tag{2}$$

where X is a matrix of size n by d+1 where n is the number of sample points and all elements in the first column of X are 1 and columns 2 through $d + 1$ correspond to the input vector; y is a vector of size n that represents function values at sample points. For the case of ordinary least squares, solution in analytical form is $w = (X^T X)^{-1} X^T y$. When one or more of independent variables are perfectly correlated, the matrix $X^T X$ becomes near singular. As a result, the coefficients w are not uniquely defined. This kind of rank deficiency can occur in high dimensional problems where the number of data points is less than the number of variables. This is usually addressed by reducing number of variables by screening or by utilizing regularization techniques. As the number of variables (d) in this problem increases, either inherently from the problem or from a combination of existing variables, this system is suscpetible to produce high variance. Even though addition of extra variables leads to low bias on the data points used for building the model, high variance makes it difficult to have better predictions on new data points. This phenomenon is known as overfitting. To avoid this issue, the effect of unnecessary variables is either removed using subset selection or suppressed using regularization. Subset selection and regularization strategies are explained below.

Subset selection

Subset selection refers to addressing the trade-off between prediction error and the regression model complexity by selecting a subset of variables. This step is followed by least squares regression for determining coefficients of the regression model. Several approaches for subset selection exist in the literature that are classified here as exhaustive search methods, heuristic methods, methods based on integer programming, methods relying on model fitness measures, Bayesian variable selection methods, and methods based on analyzing correlations between input variables and the output.

Exhaustive search methods try to exhaustively explore all possible subsets of features and select the subset with minimum prediction error. Advantage of exhaustive search is that a number of regression models are obtained with comparable prediction accuracy. Even though these methods guarantee the selection of best possible model, computational complexity of exhaustive search increases rapidly as the number of subsets increases. An implementation of this approach is the leaps and bounds algorithm [18].

Heuristic methods try to overcome this drawback by using greedy approaches such as forward-stepwise regression, backward-stepwise regression, and forward-stagewise regression. In forward-stepwise regression, variable selection starts from an empty set of variables and proceeds by sequentially adding a variable that improves the fit by largest magnitude. Improvement in the fit is usually measured by using the F-statistic. Using sum squared error, F-statistic quantifies the improvement achieved by addition of a new variable. Backward-stepwise regression is an opposite approach that starts from including all variables and sequentially removes variables that have least impact on the fit. Forward-stagewise regression is similar to forward-stepwise regression. However, in this case, only the coefficient of the newly added variable is adjusted keeping other coefficients constant.

Approaches that use integer programming for subset selection formulate the subset selection problem as an optimization problem. In these formulations, an error measure (EM) is minimized subject constraints that ensure subset selection. One way to impose such a constraint is by having an upper bound on the number of nonzero entries [19]. In addition to limiting the number of nonzero entities, these formulations can be adapted to ensure statistical properties such as robustness, selective and general sparsity of the model [20]. These approaches, however, need a prespecified value for number of variables that might not be known a priori. Review of these approaches can be found in Liu and Motoda[21]. For a known number of variables to be selected, an example of problem formulation for these problems is given by Eq. (3-5)[22].

$$\min \mathrm{EM} \tag{3}$$

$$\text{s.t.} \sum_{i=1}^{d} z_l = k \tag{4}$$

$$w_{lL} z_l \le w_l \le w_{lU} z_l, l = 1, \dots, d \tag{5}$$

$$z_l \epsilon \{0,1\}, l = 1, \dots, d \tag{6}$$

where, z_l is a binary variable for selection of variable l; k is the number of subsets to be selected; for the coefficient w_l in the regression model, w_{lL} and w_{lU} represent the lower and upper bounds respectively. Eq. (4) limits the number of nonzero coefficients used in the model. Eq. (5) imposes bounds on w_l and forces w_l to be 0 when z_l is 0.

Methods utilizing model fitness measures tackle the issue of prespecifying number of selected variables by including a penalty for number of nonzero variables. This way these methods address the tradeoff between model complexity and prediction accuracy. Several fitness measures exist in literature. One such measure is mean absolute error (MAE). An algorithm to minimize MAE is proposed and used for finding subset of variables [23]. Other such measures include Mallow's Cp [24], Akaike information criterion (AIC)[25], Bayesian information criterion (BIC) [26], the Hannan-

Quinn information criterion (HQIC) [27], the risk inflation criterion (RIC) [28], and mean squared error (MSE). These measures are shown in Table 2-1. These fitness measures can be used as EM in Eq. (3) to form a mixed integer quadratic program [29], [30]. AIC is based on the idea of minimizing discrepancy between the original distribution of the data and the distribution given by linear regression model. A well-known discrepancy measure called Kullback-Leibler divergence is used. Other discrepancy measures include Kolmogorov-smirnov and Hellinger discrepancy [31]. AIC_c represents correction term added to AIC for finite sample sizes [32]. C_p simply tries to minimize prediction error where mean squared error is the error measure. BIC_C seeks to maximize approximate posterior probability. These metrics are tabulated in (Table 2-1) where, $p < k$ is the number of coefficients, N is the number of sampled points, $\hat{\sigma}^2$ is an estimate of the error variance.

Table 2-1: Model fitness measures

Model fitness measure	**definition**
AIC_c	$N\log\left(\frac{1}{N}\sum_{i=1}^{N}(y_i - X_i w)^2\right) + 2p + \frac{2p(p+1)}{N-p-1}$
HQIC	$N\log\left(\frac{1}{N}\sum_{i=1}^{N}(y_i - X_i w)^2\right) + 2p\log(\log(N))$
BIC_c	$\frac{\sum_{i=1}^{N}(y_i - X_i w)^2}{\hat{\sigma}^2} + p\log(N)$
RIC	$\frac{\sum_{i=1}^{N}(y_i - X_i w)^2}{\hat{\sigma}^2} + 2p\log(k)$
C_p	$\frac{\sum_{i=1}^{N}(y_i - X_i w)^2}{\hat{\sigma}^2} + 2p - N$

Bayesian approach models the uncertainty over unknowns in a surrogate model using probability theory assuming those as random variables. The probability distribution that represents this uncertainty before obtaining samples is referred to as prior distribution and that after obtaining samples is referred to as posterior distribution. Suppose that there are M models under consideration where ith model is represented by $\hat{f}_i$ and unknowns corresponding to each model are represented as θ_m. The aim is to select a model with highest posterior probability, which is given by Eq. (7).

$$\Pr\left(\hat{f}_m \middle| X\right) = \frac{\Pr\left(X \middle| \hat{f}_m\right)\Pr\left(\hat{f}_m\right)}{\sum_{k\epsilon M} \Pr\left(X \middle| \hat{f}_k\right)\Pr\left(\hat{f}_k\right)} \tag{7}$$

where,

$$\Pr\left(X \middle| \hat{f}_k\right) = \int \Pr\left(X \middle| \theta_k, \hat{f}_k\right)\Pr\left(\theta_k \middle| \hat{f}_k\right) d\theta_k$$

and X is sampled data set, θ_k represents unknowns in surrogate model $\hat{f}_k$. Bayesian variable selection problem is usually considered as a special case of the model selection problem where each model consists of a subset of variables. It should be noted that for comparison between two candidate models, the denominator on the right-hand side of Eq. (7) is the same and therefore, only numerator is usually considered for comparison. Finding the model with highest posterior probability is the fundamental motivation behind BIC.

Another class of subset selection approaches is the one that relies on learning the correlation between input variables and the output. One such method is sure independence screening (SIS). Sure independence screening (SIS) relies on learning ranks of input variables according to their marginal correlation with output y. After standardizing columns of matrix X, where each column corresponds to each input variable, a vector $X^T y$ is obtained which directly signifies marginal correlations of input variables with the output. With this method, input variables having least impact on the output can be filtered out. Another famous approach involving a similar strategy of assessing impact of an input variable by monitoring correlation with output is least angles regression [33]. In this approach, coefficients are added in a similar fashion to forward-stepwise regression. However, instead of obtaining a least squares solution, correlation with the output is monitored and new variables are added sequentially.

Finally, subset selection is extremely important especially when the number of input variables is much higher than the size of available data set. Few of the approaches to address this class of problems include Dantzig selector [34], adaptive lasso and sure independence screening [35]. For high dimensional problems, Cadima et al. [36] review heuristic algorithms for subset selection. An extension of BIC for high dimensional problems known as extended BIC is proposed [37].

Regularization

Subset selection methods lead to a discrete decision of either accepting or discarding a certain variable. This leads to high variance in prediction and does not reduce the prediction error of the regression model. Regularization leads to a continuous reduction of the regression model coefficients.

Regularization penalizes magnitude of regression coefficients w to modify the problem given in Eq. (2) to the form given in Eq. (8) [38].

$$\min_{w} \left|\left|Xw - y\right|\right|_2^2 + \mathrm{C}\left|\left|w\right|\right|_{\mathrm{q}} \tag{8}$$

where,

$$\left|\left|w\right|\right|_{\mathrm{q}} = \left(\sum_{i=1}^{d} w_i^q\right)^{1/q} \tag{9}$$

and C is the parameter that decides magnitude of regularization. Eq. (9) is the expression of q^{th} norm. Value of q has a significant effect on properties of the regression model. Values 1 and 2 represent two commonly used variants of regularization known as lasso and ridge regression respectively [39]. In contrast to ridge regression, lasso has the ability to set regression coefficients exactly to 0. Values of q between 1 and 2 describe a mix between properties of lasso and ridge regression. Another approach to obtain a similar mix is known as elastic-net regression where a linear combination of lasso and ridge regression penalty terms is used.

$$\min_{w} \left||Xw - y|\right|_2^2 + \mathrm{C}\sum_{j=1}^{d}(\alpha w^2 + (1-\alpha)|w|) \tag{10}$$

where, α is a tuning parameter. Other extensions of Lasso include adaptive lasso [40] where the penalty term is a weighted summation where the weight depends on magnitude of the coefficient itself. Another for reducing the absolute value of regression coefficients uses a non-negative garrotte estimator [41]. This estimator is obtained by scaling coefficients of least squared regression. A penalty is associated with the scaling parameters and the problem is to find these scaling parameters. In this case, a closed form expression for these parameters is available as a function of coefficients obtained using ordinary least squares.

2.1.2 Support vector regression

The Support Vector Regression (SVR) surrogates are represented as the weighted sum of basis functions added to a constant term. A general form of SVR surrogate is given in Eq. (11).

$$\hat{f}(X) = \mu + \sum_{i=1}^{n} w^i \psi(X, X^i) \tag{11}$$

Assuming a simple basis function $\psi(.) = X$, the surrogate can be written as per Eq. (12).

$$\hat{f}(x) = \mu + w^T X \tag{12}$$

This form of the surrogate is similar to that of RBF as well as Kriging. However, the way to calculate unknown parameters for this surrogate differs significantly from that of RBF and Kriging surrogates. The unknown parameters μ and w in the model are obtained by formulating a mathematical optimization problem given by Eq. (13-16).

$$\min\frac{1}{2}|w|^2 + \mathrm{C}\frac{1}{n}\sum_{i=1}^{n}(\xi^{+(i)} + \xi^{-(i)}) \tag{13}$$

$$\mathrm{s.t.}$$

$$w.x^i + \mu - y^i \leq \epsilon + \xi^{-(i)} \tag{14}$$

$$y^i - w.x^i - \mu \leq \epsilon + \xi^{+(i)} \tag{15}$$

$$\xi^{+(i)}, \xi^{-(i)} \geq 0 \tag{16}$$

Eq. (14) and Eq. (15) allow the sample points to lie within $\pm\,\epsilon$ deviation from the value at sampled points without affecting the surrogate model. This band of allowed deviation is referred to as ϵ insensitive tube. Slack variables $\xi^{+(i)}$ and $\xi^{-(i)}$ ensure feasibility of the problem by allowing outliers that do not fall within ϵ insensitive tube. Trade-off between model complexity and fit is achieved by penalizing outliers by a pre-defined constant $\mathrm{C} \geq 0$. Combined contribution of the model complexity and the penalty for outliers (Eq. (13)) is minimized.

The above-mentioned formulation is obtained under the assumption of a linear basis function. Using a different basis function might require determining additional hyper-parameters associated with that specific basis function. Details and mathematical derivations related to SVR can be found in the work by Smola and Schölkopf [42].

Finally, SVR is shown to achieve comparable accuracy with that of other surrogates [43]. SVR models are accurate as well as fast in prediction; however, the time required to build this model is high because finding the unknown parameters requires solving a quadratic programming problem. This added complexity hinders the popularity of SVR [44].

2.1.3 Radial basis functions

Given n distinct sample points, RBF surrogates can be represented as given in Eq. (17).

$$\hat{f}(x) = \sum_{i=1}^{n} \lambda_i \phi\left(\left||x - x_i|\right|_2\right) + p(x) \tag{17}$$

where $\lambda_1, \dots, \lambda_n \epsilon R$ are the weights to be determined; $||.||$ is the Euclidean norm; $\phi(.)$ is the basis function. There are several options for choosing the basis function $\phi(.)$ as shown in Table 2-2.

Table 2-2: Commonly used basis functions

Type	**Function $\phi(.)$**
Linear	$\phi(r) = r$
Cubic	$\phi(r) = r^3$
Thin plate spline	$\phi(r) = r^2 \log(r)$
Multi-quadratic	$\phi(r) = \sqrt{r^2 + \gamma^2}$
Gaussian	$\phi(r) = e^{-\gamma r^2}$

In the case of multi-quadratic and Gaussian basis functions, $r \geq 0$, and γ is a positive constant. There is no solid conclusion in literature that decisively concludes one of these basis functions is better than others. However, use of cubic basis function with linear tail has been found to be successful [45]. It can be represented by Eq. (18).

$$\hat{f}(x) = \sum_{i=1}^{n} \lambda_i \phi\left(\left||x - x_i|\right|_2\right) + a^T x + b \tag{18}$$

The weights λ, a and b in Eq. (18) can be determined uniquely by solving the system of equations given by Eq. (19).

$$\begin{pmatrix} \Phi & P \\ P^T & 0 \end{pmatrix} \begin{pmatrix} \lambda \\ c \end{pmatrix} = \begin{pmatrix} F \\ 0 \end{pmatrix} \tag{19}$$

where Φ is an n by n matrix with $\Phi_{ij} = \phi(\left\| x - x_i \right\|_2)$;

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

An extension of RBF for the purposes of global optimization using a function called the bumpiness function (described in section 2.6.2) is proposed [46]. Several variations of this approach are discussed in section 2.6 [45], [47].

2.1.4 Kriging

Kriging surrogate model, also known as Gaussian Process regression, represents the underlying simulation or unknown function as a realization of a stochastic process [10]. A Kriging surrogate can be formulated as given in Eq. (20).

$$\hat{f}(x) = \sum_{i=1}^{m} \beta_j \, f_j(x) + \epsilon(x) \tag{20}$$

where $f_j(x)$ are m known independent basis functions that define the trend of mean prediction at location x; β_j are unknown parameters; $\epsilon(x)$ is a normally distributed random error at location x. The Kriging predictor has the form shown in Eq. (21).

$$\hat{f}(x) = f(x)^T \beta^* + r(x)^T \gamma^* \tag{21}$$

where, $f(x) = [f_1(x), \dots, f_m(x)]^T$; β^* is the vector of generalized least-square estimates of $\beta = [\beta_1, \dots, \beta_m]^T$; $r(x)$ is the correlation vector of size n x 1 between $\epsilon(x)$ and $\epsilon(x_i)$. β^* and γ^* are given in Eq. (22) and Eq. (23) respectively.

$$\beta^* = (F^T R^{-1} F)^{-1} F^T R^{-1} y \tag{22}$$

$$\gamma^* = R^{-1}(y - F\beta^*) \tag{23}$$

where, R is the covariance matrix of size n x n where (i, j) element is the correlation between $\epsilon(x_i)$ and $\epsilon(x_j)$; $F = \left[f\left(x^{(1)}\right), \dots, f\left(x^{(n)}\right)\right]^T$ is n x m matrix; y are observations at available data.

Having a random error term allows Kriging surrogates to provide an estimate of uncertainty in addition to the predicted value at a specific location. Prediction variance can be computed with the help of Eq. (24).

$$s^2(x) = \hat{\sigma}^2[1 - r^T R^{-1} r] \tag{24}$$

where, $\hat{\sigma}^2 = \frac{1}{n}(y - F^T\beta^*)^T R^{-1}(y - F^T\beta)$.

Several correlation models can be used for obtaining R and r as shown in Table 2-3.The correlation models depend on a set of unknowns also known as hyper-parameters. The hyper-parameters are estimated by maximizing the likelihood estimator (ML). For convenience, the log ML estimate (Eq. (25)) is often used.

$$\log \mathrm{ML}(\theta) = -\frac{1}{2}\left[\mathrm{n}\ln(2\pi\sigma^2) + \ln\det\big(\mathrm{R}(\theta)\big) + (\mathrm{y} - \mathrm{F}\beta^*)^{\mathrm{T}}\mathrm{R}(\theta)^{-1}(\mathrm{y} - \mathrm{F}\beta^*)/\sigma^2\right] \tag{25}$$

Variants of Kriging

Depending on the basis function (usually constant or polynomial of first or second degree) and the correlation model (Table 2-3) used, several structures of the Kriging model could be used.

Table 2-3 Commonly used correlation models in Kriging surrogates

Name	**Mathematical expression**
Exponential	$\exp\,(-\sum_{j=1}^{d}\theta_j\left|m_j\right|^{p_j}), 0 < p_j < 2$

Squared exponential	$\exp\,(-\sum_{j=1}^{d}\theta_j\lvert m_j\rvert^2)$
Linear	$\max\{0,1-\sum_{j=1}^{d}\theta_j\lvert m_j\rvert\}$
Spherical	$1-1.5\xi_j+0.5\xi_j^3, \xi_j=\min\{1,\sum_{j=1}^{d}\theta_j\lvert m_j\rvert\}$
Matern	$\prod_{j=1}^{d}\frac{1}{\Gamma\,(\nu_j)2^{\nu_j-1}}(\theta_j\lvert m_j\rvert)^{\nu_j}K_{\nu_j}(\theta_j\lvert m_j\rvert)$

Kriging surrogate shown in Eq. (20) consists of regression component given by $\sum_{i=1}^{m}\beta_j\, f_j(x)$ and correlation component implied by ϵ. Several choices for both of these components are proposed in literature combinations of which lead to multiple variants of Kriging.

Correlation models

The random variables $\epsilon(x)$ in a Kriging surrogate are assumed to be correlated according to a correlation model. For a deterministic and continuous function, if two samples are close to each other, their predicted values are close. As a result, the correlation between random variables is high. Correlation models consider the effect that the correlation decreases as the distance between two distinct samples increases. Commonly used correlation models are depicted in Table 2-3 where m_j is the distance between two points; θ_j and p_j are hyper-parameters; d is the number of dimensions of the original problem. In case of Matern correlation model, Γ is the Gamma function, K_{ν_j} is the modified Bessel function of order ν_j. The parameter $\nu_j > 0$ provides control over the differentiability of correlation model with respect to input variable x_j and therefore that of the Kriging predictor. Chen et al. [48] compare some of these correlation models and their results show that the squared exponential correlation performs worse than the exponential correlation. However, it is important to note that the generalized exponential correlation model has a higher number of hyper-parameters $(2d)$ as opposed to d in case of squared exponential correlation. They also suggest choosing Matern correlation model (Table 2-3) as an alternative to exponential correlation model. Differentiability of this correlation model can be controlled by choosing an appropriate value of ν. For example, $\nu_j = 1 + \frac{1}{2}$ or $\nu_j = 2 + \frac{1}{2}$ make sure that there are 1 or 2 derivatives of the correlation model respectively.

Regression models

Based on the choice of the mean prediction model $f(x)^T\beta$ (given in Eq. (20)) there are several variants of Kriging such as simple Kriging, ordinary Kriging, and universal Kriging (also known as Kriging with a trend). Simple Kriging assumes the term $f(x)^T\beta$ to be a known constant, ordinary Kriging assumes it to be an unknown constant, and universal Kriging assumes $f(x)$ to be any other prespecified function of x. In universal Kriging, usually, $f(x)$ takes form of a lower order polynomial regression. However, specifying a trend or a value for the mean when the underlying function is unknown may lead to inaccuracy in prediction. To avoid this problem blind Kriging is used [49]. In blind Kriging, the unknown trend is identified using a Bayesian variable selection technique. From a given set of candidate models, Bayesian approach tries to select models that have maximum posterior probability (section 2.1.1). Several other approaches for variable selection exist in the literature for blind Kriging. For example, Huang and Chen [50] propose a metric known as generalized degrees of freedom which is an estimator of mean squared error. Variable selection is done by trying to minimize this estimator.

There are a few strategies developed based on penalized likelihood function for variable selection in Kriging where the idea of adding a penalty term in regularization (discussed in section 2.1.1) is implemented in the context of likelihood functions [51]. Unlike penalized least squares approaches discussed in section 2.1.1, algorithms involving penalized likelihood functions involve operations with covariance matrix which is of a size of the order of size of sampled data set (discussed in section 2.1.4). As this problem occurs frequently in building Kriging surrogates, efficient optimization algorithms are developed specifically for this problem [52] [53][54] [55].

By having different combinations of mean prediction terms $f(x)$ and correlation models used for random error $\epsilon(x)$, multiple Kriging models can be obtained. One such comparison between Kriging models was made by Chen et al. [48]. For regression terms, their results reveal that adding complex regression terms to Kriging might not be of advantage over ordinary Kriging in terms of prediction accuracy. Moreover, adding these complex terms might result in multimodal ML function, thus adding an extra computational expense (section 2.1.4).

Nugget effect

Kriging by its fundamental problem formulation is an exact interpolation technique. This means that Kriging surrogate predicted value matches exactly with the underlying black-box function at the sample points used to build the Kriging model. This nature of Kriging might lead to highly oscillating behavior of the prediction. To suppress this, Kriging regression is an approach that attempts to add a regression component to Kriging.

In this approach, the covariance matrix is augmented by a term known as the nugget. The effect of this added term on Kriging surrogates is known as nugget effect. Mathematically, the correlation matrix obtained after adding the nugget term ϵ is shown in Eq. (26).

$$R^{\text{mod}} = R + \epsilon \mathrm{I} \tag{26}$$

Because of this modification, as distance between two points approaches zero, the correlation no longer equals 1. A singular or ill-conditioned covariance matrix occurs when two of the sample locations are very close to each other or hyper-parameters in the covariance model are near zero [56]. Incorporating nugget effect in such cases helps in maintaining conditioning of covariance matrix. The remainder of the procedure to obtain Kriging predictor remains the same as before.

Computational aspects of Kriging

A few key computational aspects of Kriging need to be understood before choosing Kriging surrogate for the problem at hand. First, obtaining a Kriging surrogate involves inversion of a covariance matrix. The size of this matrix depends on the number of samples and thus its inversion may become computationally demanding as the number of samples grows. Second, to obtain hyper-parameters of the correlation model, Kriging maximum likelihood (ML) estimator is optimized. This ML estimator is highly non-convex and has a strong dependence on the inverse of the covariance matrix. The non-convex nature of this function demands multiple evaluations to search for global optima. Couple of approaches are proposed to tackle this problem with likelihood maximization [57, 58]. From the equations, one can observe that getting stuck at a local optimum affects Kriging surrogate prediction as well as quantified uncertainty at unsampled locations. However, with simple covariance functions, experience shows that getting stuck at local optimum is not a serious problem and often there is no point in finding the minimizer with great accuracy [59], [60].

The non-convex and computationally intensive nature of ML estimator becomes a bigger problem as the dimensionality of the problem increases. It can be observed from Table 2-3, that irrespective of the correlation model chosen, the number of hyper-parameters, depends on the dimensionality of the problem. To reduce the number of hyper-parameters, Bouhlel et al. [61] use partial least squares. This way they address problems up to 100 dimensions more efficiently than other existing approaches. Another way to optimize hyper-parameters is to use cross validation instead of maximum likelihood. Use of cross-validation is found to be more robust with respect to correlation model misspecification compared to using maximum likelihood. However, the variance obtained by Kriging surrogates employing cross validation is larger [62].

For the problem with large number of data points, there are several successful applications of Kriging in the literature. One way to do this is by representing covariance matrix in terms of small matrices of size r, where r is the number of basis functions used [63]. Similar approach of reducing size of covariance matrix from number of sample points (n) to a smaller number r is used by Nychka et al. [64], and Banerjee et al. [65]. Another approach is using covariance tapering, where a sparse covariance matrix is obtained by setting majority of the insignificant elements to zero. Sparse matrix inversion techniques are then used to achieve attractive computational complexity [54]. Another way is to choose only a subset of data for building Kriging model [66]. There exists a large amount of literature for using Kriging on large datasets by combination of the above-mentioned approaches [67], [68] or by other frameworks [69].

Finally, even though inversion of covariance matrix is a computationally intensive task, positive definiteness of the covariance matrix helps current software implementations reduce the computational complexity by a significant factor. For obtaining maximum likelihood within a limited number of function evaluations, some software implementations make use of DFO algorithms.

2.1.5 Mixture of surrogates

Realizing the fact that no single type of surrogates outperforms all other types for all types of problems, choosing the best type of surrogate for the problem at hand is a challenging task. It is not always possible to try multiple choices of surrogate models and choose the surrogate model that shows the best performance. This motivates approaches utilizing a combination of surrogates. In general, prediction using a mixture of surrogates can be given by Eq. (27).

$$\hat{f}(x) = \sum_{i=1}^{n} w_i(x)\widehat{f_i}(x) \tag{27}$$

where, $w_i(x)$ is the weight associated with the i^{th} surrogate at design point x. Finally, summation of weights is set to one $\sum_{i=1}^{N} w_i = 1$. This implies that if all surrogate predictions $\widehat{f_i}(x)$ are equal, the weighted mixture will predict the same value.

Different approaches to determine the weights w_i are used in the literature. For example, Zerpa et al. [70] use a mixture of surrogates to optimize alkaline-surfactant-polymer flooding processes. They use a weighted combination of Kriging, RBF and polynomial regression where weights are determined based on the variance of individual surrogates. Weights can also be determined using a global cross validation metric called PRESS (discussed in section 2.7)[71]. Another approach for identifying weights is by weighing the surrogates with the help of an error metric proposed by Müller and Piché [72]. They assign probability to surrogates with the help of an error metric. These probability assignments are then used to determine weights. A variant of efficient global optimization utilizes mixture of surrogates [73]. They propose multiple surrogate efficient global optimization approach that is able add multiple candidate points for global optimization in a single iteration. Use of multiple surrogates, in general, provides a flexibility to emphasize more on good surrogates and put less emphasis on bad surrogates as per the need.

2.2 Machine learning-based classification

2.2.1 Support vector machines

Support vectors machines (SVM) are motivated by the idea of finding a hyperplane that creates the biggest margin between two training classes. The theory of support vectors classifier was explored a long time ago for linearly separable data [74], [75]. Given n pairs $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ with $x_i \in \mathbb{R}^p$ $y_i \in \{-1, 1\}$, we define a hyperplane by $f(x) = \gamma^T x + \gamma_0$ where $||\gamma|| = 1$. If the data is separable, we can find the values of γ and γ_0 such that:

$$\gamma^T x + \gamma_0 \geq 1, \forall x_m, y_m = 1 \qquad (28)$$

$$\gamma^T x + \gamma_0 \le -1, \forall x_m, y_m = -1$$

It can be shown that the problem of finding such a separating hyperplane is solved by minimizing the squared norm of γ as shown by Eq. (29)

$$\min_{\beta\,,\,\beta_0} ||\gamma|| \\ \text{s.t.}\, y_i\left(x_i^T\gamma + \gamma_0\right) \ge 1, \forall i \in \{1, \dots, n\} \tag{29}$$

where, x_i is the input data and y_i are the classification labels in the input data; i is a set of n data points; γ and γ_0 are parameters of the linear support vector; n is the number of samples.

Suppose the data has some overlapping points or outliers that cannot be separated using a hyperplane, the problem is addressed by allowing some points to be on the wrong side of the margin with the help of slack variables. We define slack variables ξ_i and modify the constraint given in Eq. (29) and reformulate as shown in Eq. (30)

$$\min_{\beta\,,\,\beta_0} ||\gamma|| \\ \text{s.t.}\, y_i\left(x_i^T\gamma + \gamma_0\right) \ge 1 - \xi_i, \forall i \in \{1, \dots, n\} \\ \xi_i \ge 0, \sum_{i \in \{1,\dots,n\}} \xi_i \le \text{constant} \tag{30}$$

where ξ_i are slack variables for allowing misclassification and the second constraint in Eq. (30) puts a bound on the total number of misclassifications. From a computational point of view, we find it convenient to re-express the problem given by Eq. (31)

$$\min_{\beta\,,\,\beta_0} \frac{1}{2}||\gamma||^2 + C\sum_{i=1}^{n} \xi_i \\ \text{s.t.}\, y_i\left(x_i^T\gamma + \gamma_0\right) \ge 1 - \xi_i, \forall i \in \{1, \dots, n\} \\ \xi_i \ge 0 \tag{31}$$

where, C is the penalty for misclassification.

By deriving the Lagrangean of this problem, we obtain the dual function.

$$\max_{\alpha} \sum_{i=1}^{n}\left[\alpha_i - \frac{1}{2}\sum_{i'=1}^{n}\alpha_i\alpha_{i'}y_iy_{i'}x_i^Tx_{i'}\right]$$
$$\text{s.t.}\sum_{i=1}^{n}\alpha_iy_i = 0 \qquad (32)$$
$$0 \le \alpha_i \le C$$

The solution of this problem leads to finding γ to obtain the separating hyperplane. The expressions for γ and γ_0 are shown by Eq. (33) and Eq. (34)

$$\gamma = \sum_{i=1}^{n}\alpha_iy_ix_i \qquad (33)$$

$$\gamma_0 = \frac{1}{N_{SV}}\sum_{i=1}^{N_{SV}}(y_i - \gamma x_i) \qquad (34)$$

where, N_{SV} represents the number of support vectors, which is the same as the number of data points that lie on the separating hyperplane.

However, most realistic process engineering problems will lead to data that is not linearly separable. In such a case, SVM can still be used by transforming the data into a higher dimensional space where it becomes linearly separable. The transformation functions, in this case, are referred to as nonlinear kernel functions. Commonly used kernel functions are displayed in Table 2-4.

Table 2-4 Kernels used for separating data using SVM

Kernel = $\boldsymbol{K(x,x')}$	**expression**
dth degree polynomial	$(1+< x,x' >)^d$
Radial basis	$\exp\left(-\rho\left\|\|x - x'\|\right\|^2\right)$
sigmoid	$\tanh\,(k_1 < x,x' > +k_2)$

Where, ρ, k_1, and k_2 are hyperparameters of the kernel function.

By using the values of γ and γ_0 from Eq. (33) and Eq. (34), and using the kernel functions, the final predictor $\hat{f}(x)$ can be expressed as shown in Eq. (35)

$$\hat{f}(x) = \text{sign}(\sum_{i=1}^{n} y_i \alpha_i K(x, x_i) + \gamma_0) \tag{35}$$

where, $K(.)$ is the kernel function.

The methodology can be extended to multiclass classification [76]. The implementation used in this work is from scikit-learn python toolbox [77], where the quadratic optimization problem given by Eq. (32) is solved. One algorithm to solve the optimization problem is given by Chang et al. [49]. The kernel function used in this work is the most popularly used radial basis kernel with the value of γ is chosen to be the inverse of the number of variables. In practice, the choice of a kernel function in SVM is usually made by the user based on her experience or by trying out multiple kernel functions. However, more rigorous approaches for this choice exist and the reader is referred to Jebara [78]. All inputs are standardized to have zero mean and unit variance before training the SVM model.

2.3 Derivative-free optimization and surrogates

The optimization problems for which function derivative information is not symbolically or numerically available are classified as DFO problems. There are two sub-categories in algorithms addressing DFO problems, one is local search (referred to as local DFO) algorithms and the other is global search algorithms (referred to as global DFO). Local search algorithms are effective in refining the solution or reaching a local optimum from an initial guess. Global search algorithms, on the other hand, have a component that allows escaping from a local minimum. For the purposes of this chapter, it is convenient to classify DFO algorithms as algorithms that do not use surrogate models and model-based algorithms. A major class of local DFO algorithms that do not rely on surrogate models is direct search algorithms. Direct search algorithms sequentially examine candidate points generated by a certain strategy sometimes recognizing geometric patterns. Well known examples of the direct search are Hooke and Jeeve's algorithm [79] and simplex method [80]. Model-based approaches, as the name suggests, rely on surrogate models to guide the search. For the case of global DFO algorithms, the majority of the algorithms that do not use surrogate models use an approach such as partitioning of the feasible space or a stochastic approach. An example of partitioning algorithm is DIviding RECTangles (DIRECT) algorithm [81]. Examples of stochastic algorithms include several approaches such as simulated annealing or genetic algorithms. For details on advances in DFO and an extensive comparative study on box-bounded problems, the readers are referred to the review work by Rios and Sahinidis [82].

As model-based search algorithms have been shown to display superior performance compared to these algorithms, it is important to discuss the role of surrogate models in the context of DFO. A major class of model-based local DFO methods known as trust-region methods is discussed followed by model-based global DFO methods.

2.3.1 Model-based local DFO

Trust-region methods are local search methods that rely on a surrogate model in a neighborhood of a given sample location. This neighborhood is called as trust region and the model is presumed to be accurate within trust region. The size of the trust region is defined with the help of radius which is adjusted based on a measure of the accuracy of the surrogate. The sufficiently small value of trust-region radius usually indicates termination. Because of the general nature of trust-region framework, several surrogates have been used in the literature to achieve local approximation. For example, Powell [83] use linear interpolation models to approximate objective and constraint functions in the algorithm COBYLA (Constrained Optimization BY Linear Approximation). Linear

Interpolating surrogates are easy to construct but these surrogates face difficulty in capturing curvature of the original problem. Use of quadratic models of the form given in Eq. (36) is proposed [84].

$$\hat{f}(x_k + s) = f(x_k) + s^T g_k + \frac{1}{2} s^T H_K s \tag{36}$$

where, k corresponds to iteration k, x_k is the current iterate, $g_k \epsilon \mathrm{R}^{\mathrm{d}}$, H_k is a matrix of of size d by d. Uniquely determining g_k and H_k requires $\frac{(d+1)(d+2)}{2}$ sample points. This number becomes significantly high as the number of dimensions increase. For example, for a 30 dimensional problem, this number becomes nearly 500. To avoid this high sampling requirement, Powell [85] proposed underdetermined quadratic interpolation models. These models are proven to attain stationary local optimum and thus are called locally convergent. Futher, Oeuvray and Bierlaire [86] use RBF interpolation models with cubic basis functions and a linear tail. With modifications to the set of points used for building RBF models, Wild et al. [87] proposed Optimization by RBF Interpolation in Trust-regions (ORBIT) algorithm. This algorithm was later extended to handle constrained optimization problems [88]. RBF based trust region algorithms are proven to be globally convergent [89]. A similar strategy was used recently where Kriging based efficient global optimization (EGO) where Kriging surrogate was used inside trust-region framework [90].

2.3.2 Model-based global DFO

One of the reasons surrogates are promising in the context of global DFO is the progress made in the area of global optimization algorithms in the past decade [91], [92]. With the help of these algorithms, non-convex surrogates can be optimized and used to guide the search. For global DFO, a surrogate model is generated over the entire feasible space or multiple parts of the feasible space. For example, in the algorithm EGO [93], Kriging surrogate is built over the entire feasible space. With the help of expected improvement (EI) function (discussed in section 2.6.1), the surrogate is updated. In this case, a balance between local search and global search is achieved by maximizing the EI function. A similar balance is achieved for RBF surrogates using a measure called bumpiness function (Eq. (42)) [46]. Minimizing the bumpiness function for a given target value can be used to focus on global and local search. This property of the bumpiness function is exploited for global DFO [94]. EI function, as well as bumpiness function, are discussed in more detail in section 2.6. Another approach using RBF surrogate relies on optimization and sequential updating of RBF surrogate over the feasible space [95]. Finally, global search is also achieved by conducting local search starting from multiple starting points obtained using a certain strategy. A complete restart strategy is proposed that suggests starting from a new sample design if algorithm gets stuck in a local minimum [47]. A more recent example of successful use of surrogates to address constrained global DFO problems is shown by Boukouvala and Floudas [96]. They developed a framework for constrained grey box optimization named Algorithms for Global Optimization of coNstrAined grey-box compUTational problems (ARGONAUT) that was shown to address a difficult class of problems successfully. In this framework, the surrogate is chosen from a set containing linear, general quadratic, sigmoidal, RBF and Kriging models based on the accuracy of prediction.

2.4 Feasibility analysis

A process is said to be feasible if all the relevant constraints are satisfied. Feasibility analysis relates to identifying conditions under which the process is feasible. Since identifying the optimal design requires the user to ensure that the design satisfies the constraints and meets the desired demand for products, feasibility analysis plays an important role in this work. A precise estimation of feasibility is crucial for conducting a systematic study of multiple design alternatives and achieve objectives such as maximizing profit.

Feasibility is quantified with the help of a measure known as the feasibility function $\psi(d,\theta)$ as given in Eq. (37). A positive value of the feasibility function implies that the design is infeasible[12].

$$\psi(d,\theta) = \min_{z} \max_{j \in J} \{f_j(d,z,\theta)\} \tag{37}$$

where d and z represent design variables and control variables respectively, bounds on z are written as $z \in \ Z = \{z\colon z^L \leq z \leq z^U\}$; θ represents uncertain parameters $\theta \in T = \{\theta\colon \theta^L \leq \theta \leq \theta^U\}$; $f_j(d,z,\theta)$ represents constraints. The problem is to check if all constraints f_j can be satisfied for a given design d by adjusting the control variables z. Thus, $\psi(d,\theta) > 0$ implies one or more constraints are violated and $\psi(d,\theta) = 0$ implies the boundary of the feasible region. In the presence of analytical equations for the constraints f_j, the problem of feasibility analysis can be handled using an equation-oriented optimization solver. However, the feasibility analysis of simulation models often requires substantial computational efforts because of the unavailability of the closed form. In such a case, the literature relies on the black-box feasibility analysis. Several approaches for black-box feasibility analysis exist in the literature based on the type of data-driven approximation or a surrogate model used. Previous techniques have used Kriging [97][98], RBF [99], HDMR [100], and CRS for approximating the feasibility function $\psi(d,\theta)$ over the entire domain.

As described in section 2.6, quality of surrogates has a strong dependence on the required quantity and quality of sampling set. Increasing sample size may lead to a better prediction but it will result in increased sampling cost. For feasibility analysis problems, sampling requirement is higher than that of single objective prediction due to the presence of constraints. To control the sampling cost, approaches employing adaptive sampling are used. Kriging surrogates and a modified version of EI function given in Eq. (38) for adaptive sampling is proposed [101].

$$\max_{x} \mathrm{EI}_{\mathrm{feas}}(x) = s\phi\left(-\frac{y}{x}\right) = s\left(\frac{1}{\sqrt{2\pi}} e^{-0.5\left(\frac{y^2}{s^2}\right)}\right) \tag{38}$$

where $\mathrm{EI}_{\mathrm{feas}}(x)$ is the modified EI function value at x; y is the surrogate model predictor; s is the standard error of the predictor; $\phi(.)$ is the normal probability distribution function.

There is a difference between a search for global optimization and that for feasibility analysis. For feasibility analysis, the problem is to find a surface defining the boundary of the feasible space within the box bounded design space as opposed to finding a single optimum in global optimization [97]. This problem is addressed with the help of Kriging variance as a metric to ensure exploration. To guide the search towards better defining the boundary of feasible space, the product of feasibility function values of nearby samples is used. Samples on the same side of feasible boundary result in a positive product. More recently, Wang and Ierapetritou [99] used an adaptive sampling strategy based on RBF surrogates. They used bumpiness measure (explained in section 2.6.2) to obtain prediction error. Substituting this prediction error in $\mathrm{EI}_{\mathrm{feas}}$ function and maximizing $\mathrm{EI}_{\mathrm{feas}}$ with respect to x, they chose new sample points for evaluation. Their results show that accuracy obtained from both Kriging and RBF is comparable.

2.5 Flexibility analysis

Uncertainty in process parameters can have a significant impact on the feasibility of a design. The ability of a process to remain feasible when subject to deviations of uncertain parameters is referred to as process flexibility. Process flexibility is quantified by solving the flexibility test problem which checks if feasibility function $\psi(d, \theta)$ is non-positive over the entire range of uncertain parameters θ. The flexibility test problem is usually represented as a max-min-max problem, as represented by Eq. (39) [12].

$$\chi(d) = \max_{\theta \in T} \min_{z} \max_{j \in J} \{f_j(d, z, \theta)\} \tag{39}$$

$\chi(d)$ indicates if the design is feasible over the entire range of uncertain parameters. A negative value of $\chi(d)$ indicates that the design is feasible over the entire range of uncertain parameters, and a positive value indicates that the design is not feasible over the entire range. However, it is often important to quantify the actual range over which the design is feasible. For this reason, flexibility index [102] is used as a quantitative measure. The problem for finding flexibility is formulated as shown by Eq. (40).

$$\begin{gathered} F = \max \delta \\ \text{s.t.}\, \chi(d) = \max_{\theta \in T} \min_{z} \max_{j \in J} \{f_j(d, z, \theta)\} \leq 0 \\ \mathrm{T}(\delta) = \{\theta : \theta^N - \delta \Delta\theta^- \leq \theta \leq \theta^N + \delta \Delta\theta^+ \\ \delta \geq 0 \end{gathered} \tag{40}$$

where, θ^N, $\Delta\theta^-$, $\Delta\theta^+ \in R^m$; $\delta \in R^+$; m is the number of uncertain parameters; θ^N are nominal values for uncertain parameters; $\Delta\theta^-$ and $\Delta\theta^+$ represent the expected deviations in negative and positive directions respectively; δ quantifies the amount of uncertainty denoted by the parameter set $T(\delta)$; F is the flexibility index. A value of 1 for F indicates that the design has flexibility just enough to satisfy all the process constraints over the range of uncertain parameters.

2.6 Sampling

The process of generating data points to be able to build surrogates is referred to as sampling. The performance of surrogate models depends strongly on the quality as well as the number of samples. However, as generating data demands evaluation of the true function, sampling contributes towards significant computational cost. To maintain the quality of surrogates without incurring excessive sampling cost, studying sampling strategies is of immense importance.

Sampling strategies are broadly classified as adaptive sampling and stationary sampling. Stationary sampling consists of methods that rely on geometry or pattern such as grid sampling, full and half factorial designs, methods that were derived from the design of experiments literature such as orthogonal sampling, full and half factorial designs, Box-Behnken design. Some of the widely used stationary sampling strategies are Latin Hypercube Sampling (LHS) [103], Sobol [104] sampling and Halton sampling. LHS is a stratified sampling strategy where a sample is drawn from each stratum once. To provide better space-filling properties, LHS is done subject to projection filters. Sobol and Halton sampling are quasi-random strategies where samples are drawn from Sobol and Halton low-discrepancy sequences respectively.

In adaptive sampling, starting from a limited number of samples that are generally obtained from stationary sampling, new sample locations are decided sequentially. This strategy aims to minimize sampling requirement by obtaining more samples that benefit the quality of the surrogate. Most of the new adaptive sampling strategies rely on some criteria to tackle the trade-off between exploring the most unexplored region (exploration) and refining the region near existing samples for better understanding (exploitation). This approach is most common in the context of global optimization where exploration is required to escape local optima and exploitation is required to improve available optimum. For Kriging surrogates, a popular approach is making the use of EI function. For RBF surrogates, a similar quantitative measure is obtained using a function known as bumpiness function. Other approaches employing adaptive sampling make use of different strategies to address this trade-off. In general, these methods have been shown to achieve better accuracy with fewer samples [105].

2.6.1 Expected improvement function

One commonly used approach to handling exploration and exploitation is by using EI function (Eq. (41)) as used by [93].

$$\mathrm{EI}(x) = \left(f_{\min} - \hat{f}(x)\right)\boldsymbol{\Phi}\left(\frac{f_{\min} - \hat{f}(x)}{s}\right) + s\phi\left(\frac{f_{\min} - \hat{f}(x)}{s}\right) \tag{41}$$

where, $\boldsymbol{\Phi}(.)$ represents the standard normal density function; $\phi(.)$ represents the probability distribution function; $\hat{f}$ is the surrogate model predictor; $f_{\min}$ is the current minimum function value and s is the standard deviation. $\mathrm{EI}(x)$ represents the expected improvement at sample location x. The function increases with decreasing $\hat{f}(x)$ that corresponds to the predicted value and increasing standard deviation s. Achieving low $\hat{f}(x)$ and high s correspond to exploration and exploitation, respectively. As both contribute positively towards EI function, the trade-off between exploration and exploitation is addressed by maximizing the EI function. The EI function exhibits multiple local optima that might cause numerical problems.

2.6.2 Bumpiness function

A similar approach of having a single function to balance exploration and exploitation was proposed by Gutmann [46] for RBF surrogates. This relies on the fact that the RBF surrogate that is obtained by solving the system of equations given by Eq. (19) is the one that minimizes bumpiness. A quantitative measure of bumpiness is given by bumpiness function (Eq. (42)).

$$\min g_n(y) = (-1)^{m_0+1}\mu_n(y)\left[\hat{f}(y) - f_n^*\right]^2, y \,\epsilon\, D\backslash\{x_1, x_2, .., x_n\} \tag{42}$$

where, y is an unsampled point; f_n^* is the target value; m_0 is a constant whose value depends on the basis function used (1 for cubic and thin plate splines, 0 for linear and multi-quadratic and -1 for Gaussian); $\mu_n(y)$ is the coefficient of the new term $\phi(||x - y||)_2$ in the surrogate $\hat{f}_n(x)$ if an unsampled point y is added. It is calculated as the n^th element of vector v, and v is calulcated by solving the system of equations given by Eq. (43).

$$\begin{pmatrix} \Phi_y & P_y \\ P_y^T & 0 \end{pmatrix} v = \begin{pmatrix} 0_n \\ 1 \\ 0_{d+1} \end{pmatrix} \tag{43}$$

$$\Phi_y = \begin{pmatrix} \Phi & \phi_y \\ \phi_y^T & 0 \end{pmatrix}; P_y = \begin{pmatrix} & P \\ y^T & 1 \end{pmatrix}; (\Phi_y)_i = \phi\left(\left\|y - x_i\right\|_2\right); i = 1, \dots, n.$$

Minimizing the bumpiness function emphasizes exploration as well as exploitation depending on f_n^*. A large negative value of f_n^* makes the search global and focuses on exploration whereas, a value close to current optimal solution makes the search local and focuses on exploitation. Evaluation of bumpiness function is computationally expensive because obtaining μ_n by solving the system given by Eq. (43) is an $O(n^3)$ operation. However, the cost of this step can be improved by exploiting the structure of Φ after which the operation becomes $O(n^2)$ [45].

2.6.3 Other approaches

The problem of adaptive sampling can be formulated as a DFO problem with the objective function being the difference between the true function and the surrogate [30]. The objective function is given in Eq. (44)

$$\max\left(\frac{f(x) - \hat{f}(x)}{f(x)}\right)^2, x_L \leq x \leq x_U \tag{44}$$

where, $\hat{f}(x)$ is the surrogate, $f(x)$ is the true function, and x_L and x_U are the bounds within which error is to be maximized.

Some approaches rely on ranking the exploration and exploitation and weighing both as per need. One such approach was recently proposed by Garud et al. [106]. They propose a metric consisting of two separate measures for exploration and exploitation. For exploration, they use the sum of squares of the distance a new sample from all the previous samples. For exploitation, the impact of new sample added near an already sampled location is quantified with the help of a departure function.

$$\Delta_j(x) = \hat{f}(x) - \widehat{f_J}(x), j\epsilon S \tag{45}$$

where, $\hat{f}(x)$ is the surrogate built using all points in the sampled set S, $\hat{f}_j(x)$ is the surrogate built using all points except point j. Estimating the prediction variance of surrogate model using a technique called jackknifing, Eason and Cremaschi [107] propose an adaptive sampling strategy. They use this strategy with a surrogate model built using ANN and choose samples locations that have high prediction variance. Advantage of this type of adaptive sampling is that it is not specific to the choice of surrogate model. A study of space filling sequential design methods is conducted by Crombecq et al. [108]. They propose a set of sequential sampling methods that shows comparable performance with stationary or one-shot experimental design.

2.7 Validation methods

2.7.1 Surrogate models

Assessing the reliability of surrogate model is one of the major concerns because having an inaccurate surrogate model can lead to waste of resources and have a bad effect on optimization, prediction or feasibility analysis. Surrogate model validation is the process of assessing the reliability of the surrogate model. In addition to assessing accuracy, validation techniques can be used to select a surrogate model from a set of candidate models and to tune hyper-parameters (such as correlation model parameters in Kriging). For problems of lower dimensions, a visual comparison between predictions and true value is possible. However, the difficulty in having enough data for visual comparison and inability to visualize predictions for problems over two dimensions necessitates more sophisticated approaches. As surrogate models cannot be validated on the same data with which those were built, surrogate models are built with the help of only a part of the available data. The remaining part of the data is used for testing the accuracy. The data set on which the model is built is referred to as training set and the set on which the model is tested is referred to as test set. The metrics used for quantifying the error on test set are referred to as validation metrics.

One possible approach to tackling this is using resampling strategies such as cross validation and bootstrapping. In cross-validation, available data is divided into k blocks containing an equal number of data points. Data from (k-1) blocks are used as training set and data from the remaining block are used as a test set. The process is repeated for all possible combinations of (k-1) blocks. Finally, an appropriate metric to quantify the error on test set such as the sum of squared errors is evaluated based on the accuracy of the model on test data that can act as an indicator of model adequacy. However, with limited data available, using part of the data for building the surrogate is not always possible. One such approach known as leave one out cross validation was used by Jones et al. [93]. In this approach, the number of subsets k equals the number of data points or observations, thus leaving only one data point each time a surrogate is built. A sampling set is considered inadequate to build a quality surrogate if removal of one data point significantly affects the new model. A similar approach, but with allowing repeated samples in the training set is known as bootstrapping. By allowing repeated samples in the set used to build models, one can have a training set of the size equal to the size of actual data. Usually, number of subsets k chosen for bootstrapping is much higher than that for cross validation. Details on resampling methods for validation of surrogates are provided by Bischl et al. [109].

Validation metrics that are commonly used to quantify the error using the above-mentioned resampling strategies are the explained variance score, the mean absolute error, the mean squared error, the median absolute error, the R^2 score, the relative absolute error, and the relative maximum absolute error. These metrics with their respective mathematical equations are shown in Table 2-5 where y, $\hat{y}$, $n_{samples}$, and $\bar{y}$ denote true value, surrogate predicted value, number of samples and mean predicted value, respectively. Relative maximum absolute error indicates error in one part of feasible space. However, it is not a good indicator of the overall performance. Explained variance score equals R^2 score if mean of prediction error is zero. Kersting et al. [110] use normalized mean squared error as well as average Negative Log estimated Predictive Density (NLPD) for heteroscedastic Gaussian process regression that penalizes over-confident as well as underconfident predictions. In the same area, Boukouvalas and Cornford [111] use Mahalanobis error that utilizes full predictive covariance avoiding the assumption of uncorrelated errors. Yin et al. [112] use mean absolute error as well as maximum absolute error for validation. For the case of multiple surrogates, Viana et al. [113] used prediction sum of squares (PRESS) as an estimator of root mean square error (RMSE) to pick the best surrogate. Their computational results reveal that PRESS becomes more and more useful for identifying the best surrogate as the number of sample points increases. PRESS vector $\tilde{e}$ is the vector of errors obtained from carrying leave one out cross validation. RMSE is predicted using Eq. (46).

$$\mathrm{PRESS_{RMS}} = \sqrt{\frac{1}{n_{samples}}\ \tilde{e}^T\tilde{e}} \tag{46}$$

Table 2-5: Commonly used surrogate validation metrics

Validation metric	**Formula**
Explained variance score	$1 - \frac{Var\{y - \widehat{y}\}}{Var\{y\}}$
Mean absolute error	$\frac{1}{n_{samples}} \sum_{i=0}^{n_{samples}-1} \lvert y_i - \hat{y} \rvert$

Mean squared error	$\frac{1}{n_{samples}} \sum_{i=0}^{n_{samples}-1} (y_i - \hat{y})^2$
Median absolute error	$\text{median}(\lvert y_1 - \widehat{y_1}\rvert, \dots, \lvert y_n - \widehat{y_n}\rvert)$
R^2 score	$1 - \frac{\sum_{i=0}^{n_{samples}-1} (y_i - \hat{y}_i)^2}{\sum_{i=0}^{n_{samples}-1} (y_i - \bar{y})^2}$
Relative average absolute error	$\frac{\sum_{i=1}^{n_{samples}} \lvert y_i - \hat{y}_i\rvert}{n_{samples} * \text{STD}}$
Relative maximum absolute error	$\frac{\max\,(\lvert y_1 - \widehat{y_1}\rvert, \lvert y_2 - \widehat{y_2}\rvert, \dots, \lvert y_n - \widehat{y_n}\rvert)}{\text{STD}}$

2.7.2 Machine learning-based classification models

Before incorporating the classifier into an optimization problem, it is important to verify the quality of the classifier. The metrics explained here are previously used by Dias and Ierapetritou [114]. The goal is having a comprehensive quantitative measure of the prediction of the feasible region, the infeasible region as well as the prediction accuracy. This is achieved by dividing the dataset into four parts named CF (correct feasible), CIF (correct infeasible), ICF (incorrect feasible), and ICIF (incorrect infeasible). Correct or incorrect refers to the prediction by the model whereas feasible or infeasible is based on the true data. For example, correct feasible refers to points that are feasible and that are correctly identified as feasible. Based on this division of model predictions, four metrics are proposed that are CF%, CIF%, NC%, and Total Error. Expressions for calculating these metrics are shown in Eq. (47).

$$\begin{gathered}
\text{CF\%} = \frac{\text{CF}}{\text{CF} + \text{ICIF}} \times 100 \\
\text{CIF\%} = \frac{\text{CIF}}{\text{ICF} + \text{CIF}} \times 100 \\
\text{NC\%} = \frac{\text{ICF}}{\text{ICF} + \text{CF}} \times 100
\end{gathered} \tag{47}$$

$$\text{Total Error} = \frac{\text{ICF} + \text{ICIF}}{\text{CF} + \text{ICF} + \text{CIF} + \text{ICIF}} \times 100$$

Of these metrics, CF% and CIF% represent how well the model represents the feasible region as well as the infeasible region. The metric NC% represents the overprediction of the feasible region. Total Error quantifies the percentage of total misclassifications. Validation in this way provides a true picture of the model quality because all metrics combined can detect special cases such as imbalanced data where the Total Error will be low, but one of the CIF% or CF% will also be low indicating inadequacy of the model. A good quality model has high values of CF% and CIF% and low values for NC% and Total Error. For validating models in this work, the data is randomly divided into training and test data sets with a split of 80% and 20%, respectively. The training data set is used for building the model, whereas the test dataset was used to assess the performance.

3 Supply chain optimization

Abstract

Supply chain simulation models are widely used for assessing supply chain performance and analyzing supply chain decisions. In combination with derivative-free optimization algorithms, simulation models have shown great potential in effective decision-making. Most of the derivative-free optimization algorithms, however, assume continuity of the response, which may not be true in some practical applications. In this work, a supply chain inventory optimization problem is addressed that results in a discontinuous objective function. A derivative-free optimization framework is proposed that addresses the discontinuities in the objective function. The framework employs a sparse grid sampling and support vector machines for identification of discontinuities. Computational comparisons presented show that addressing discontinuity leads to more cost-effective decisions over existing approaches.

3.1 Introduction

Globalization and the sudden increase in the exchange of information, trade, and capital all around the world, driven by technological innovation, has given rise to complex global supply chain networks. Under such networks, the problem of optimal inventory allocation is known to have a significant impact on the service level and the total cost of the supply chain and thereby impacting the profit that an individual enterprise would make [2]. This work considers the optimal inventory allocation problem for multienterprise supply chain networks. The methodology proposed in this work is motivated by three factors. First, the ability of simulation models to represent a complex system accurately. Second, recent advancements in the use of derivative-free optimization for decision-making based on the simulations. Third, the observation that the objective function of such a system, when modeled as a derivative-free optimization problem, may lead to a discontinuous response. Relevant literature from supply chain optimization, derivative-free optimization, and discontinuous optimization that motivates the framework presented in this work are presented in the remainder of this section followed by a brief description of the problem addressed.

Modern supply chain networks consist of entities that usually belong to different enterprises and operate based on their individual goals leading to a decentralized network. Since earlier strategies in the supply chain optimization literature focused mainly on the centralized networks, there is a growing interest towards developing new optimization frameworks that take into account different goals and operating policies of different entities in a supply chain [1]. Competition between buyers and sellers, conflicting interests between different entities, and game theoretic models that describe these interactions are some of the considerations in modeling such frameworks based on the supply chain network under consideration. The optimization approaches for decentralized networks can be broadly classified as analytical approaches and simulation-based approaches. Analytical approaches formulate the supply chain model as an equation-based mixed integer linear or nonlinear programming problem. Ryu et al. [115] present a bilevel programming framework to capture conflicting interests as well as imbalances in the available information at different levels such as distribution network planning and production planning. Zamarripa et al. [116] propose a multi-objective optimization formulation for cooperative or competitive supply chains. Yeh et al. [117] use bilevel optimization for supply allocation using a Stackelberg game which are two player turn based games with a leader and a follower. Both have separate objectives and have a complete understanding of each other's information. More recently, Yue & You [118] discuss the optimization of noncooperative supply chains under stackelberg game using mixed integer bilevel programing. Florensa et al. [119] address capacity planning problem by formulating it as a trilevel optimization problem to capture the dynamics of a duopolistic market. Even though these approaches benefit from efficient optimization techniques, modeling complex interactions in a network with the help of equation-based models is not always possible. As a result, the assumptions necessary for analytical models lead to solutions that are not applicable for real world scenarios [120][121]. Moreover, even the simplified models used in the analytical approaches are computationally expensive for large networks [122].

To overcome these shortcomings, another approach towards addressing the problem of multi-enterprise supply chain optimization is using a simulation model that captures details of the supply chain network. Techniques such as agent-based modeling allow a bottom-up approach towards building a simulation [3], [4] and have been found useful in supply chain networks in various areas. Swaminathan et al. [123] describe a framework for developing supply chain models using a modular approach. Lee & Kim [124] review modeling techniques for multi-agent systems. For supply chains in the process industry, García-Flores & Wang [125] study information flow with the help of agent-based simulation. Using an agent-based framework, Julka et al. [126] show an application where the framework works as a decision support system in a refinery application [127]. Even though agent-based modeling is popular for building accurate simulations, its utility is not limited to analyzing a complex system. Agent-based simulations have also been used for optimization using derivative-free or simulation-based optimization. Singh et al. [128] study a biorefinery supply chain network with agent-based modeling and use genetic algorithms to identify the location and capacity of each biorefinery in the network. Sahay & Ierapetritou [129] consider a supply chain network consisting of entities from multiple enterprises. They consider an auction mechanism where enterprises adapt their strategy based on the outcome of the auction. As opposed to a small network consisting of a single enterprise, a centralized decision cannot be imposed on these type of multi-enterprise networks. Ye & You [121] propose an optimization framework for reducing the total cost of the supply chain under demand uncertainty where an agent-based simulation model is used to represent the inventory system. Because of their flexibility to adapt to different networks structures and demonstrated success in the existing literature, this work uses an agent-based simulation model details of which are provided in section 3.2.

With the increased complexity of simulation models, obtaining a closed form expression or a mathematical model becomes difficult which limits the use of conventional algorithms. For this purpose, derivative-free optimization algorithms are used [10]. Even though a vast amount of literature exists on derivative-free optimization, a majority of the available algorithms and software packages rely on the assumption of continuity of the response. This may not be the case in some applications, especially for the case of supply chain networks where the objective function is dependent on several discrete decisions. For this reason, optimization of problems with discontinuous response needs special attention. Previous works on addressing discontinuities majorly focus on modeling a discontinuous response. The problem of identifying discontinuities using a Bayesian modeling approach is thoroughly analysed by Anderson [130]. Gorodetsky & Marzouk [131] propose an approach for identifying discontinuities and refining them in an adaptive manner. They make use of support vector machines and uncertainty sampling to adaptively sample new points near discontinuity. Jakeman et al. [132] use an adaptive sparse grid approach where new samples are generated at the locations where a discontinuity is suspected using polynomial annihilation, a technique for estimating the size of discontinuities. This approach is later used for multi-element collocation [133]. Archibald et al. [134] discuss polynomial annihilation method for detecting discontinuities and extend it to higher dimensional problems such as stochastic partial differential equations. Caiado & Goldstein [135] use a Bayesian approach to address discontinuities. They use a separate surrogate model for each continuous region for modeling purposes. For optimization, Moreau & Aeyels [136] propose the use of semigradient for optimization of discontinuous functions. Vicente & Custódio [137] analyse the properties of direct search methods for optimizing piecewise continuous functions in the presence of constraints. Recently, a framework for optimizing in the presence of discontinuities to address a structural optimization problem is proposed [138]. They propose a framework where continuous regions are clustered together after identification using polynomial annihilation. These regions are then classified using support vector machines. The discontinuity identification framework proposed in this work is inspired by adaptive sparse-grid algorithm by Jakeman et al. [132]. Although the framework follows a similar flow to that of previous work where discontinuity identification step is followed by classification, there are several differences between. First, the objective of the previous work is modeling discontinuities accurately whereas this work considers an optimization problem where discontinuity identification is a subproblem. Secondly, discontinuity identification in the previous works encompasses the entire feasible or search space whereas in this work it is confined to several local searches. A global surrogate model is used in this work instead which guides the search towards promising local search regions. Finally, a sparse grid is used even for the refinement step whereas the previous work [132] uses 4.3^{d-1} samples for refinement.

This work addresses the problem of optimal inventory allocation using a derivative-free optimization framework. An agent-based model is considered that consists of a supply chain network where entities belong to multiple-enterprises. Enterprises interact with each other through an auction mechanism. The problem of deciding optimal warehouse inventory is found to display a discontinuous behavior. To that end, the optimization framework proposed in this work can handle discontinuous objective functions. The framework iteratively uses a Kriging model for the global search followed by a local search followed that includes discontinuity detection. It uses sparse grid sampling and support vectors classification for identifying discontinuities.

The remainder of this chapter is organized as follows. Section 3.2 describes the details of the simulation model. Section 3.3 describes the optimization framework in detail. Section 3.4 compares the performance of the proposed framework on test problems as well as on the supply chain simulation. Finally, summary and future directions are discussed in section 3.5.

3.2 Supply chain simulation and the problem definition

For modeling a system where each component has an autonomous behavior, a natural choice is a bottom-up approach. This being one of the central ideas behind agent-based simulation, it is a suitable approach for modeling a supply chain network. Broad categories of agents considered in the supply chain simulation in this work are raw material suppliers, production sites, warehouses, retailers, and auctioneer. Each agent has a set of rules according to which it behaves or modifies its behavior. Based on the tasks performed, each agent has a cost associated with it. This cost could be transportation cost, inventory cost, production cost or a combination of these costs. Additionally, for retailer agents, a penalty for unmet demand is considered. The total cost for an enterprise is the combined cost of all entities that belong to that enterprise. There can be multiple agents of the same category (for example, warehouses) and each agent may belong to a different enterprise. Java programming language is used to build an agent-based simulation. A typical schematic of such a supply chain is depicted in Fig. 3-1. The simulation model used in this work is an extension of the model proposed by Sahay & Ierapetritou [139] where more details are provided regarding the simulation model. A brief description of each agent is provided below.

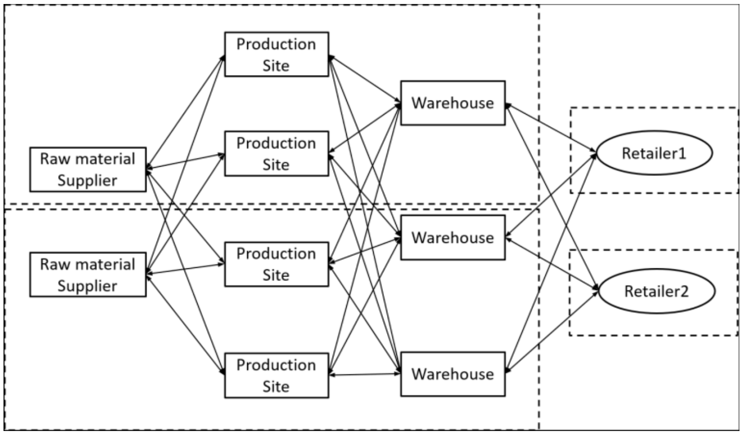


Fig. 3-1 An example of a supply chain network

Retailer

Demand originates at the retailer agent. Tasks associated with this agent are conveying information regarding demands to the auctioneer and submitting the bid to buy products from the warehouse. A bid consists of maximum price that the retailer is willing to pay and the maximum quantity that the retailer is willing to buy. Retailers from different enterprises compete in the form of auctions to receive product from a warehouse. New demand is incurred at the beginning of each period. Based on the results of the auction in the previous period, the retailer may satisfy the demand fully or partially. Unfulfilled demand is considered lost and a penalty cost is enforced. Costs associated with this agent are a penalty for unmet demand and transportation cost from warehouse to retailer.

Warehouse

Warehouse agent is responsible for storing inventory, submitting its asks to the auctioneer, sending material to the retailer, and ordering material from the production sites for replenishing inventory. Ask from a warehouse includes the minimum price at which it is willing to sell a product and the maximum quantity of that product it can deliver. Warehouse receives a response from the auctioneer agent with the amount of product that is to be delivered to a retailer. It updates the inventory based on reorder-level reorder-amount policy with continuous review. After sending the product to a retailer, it requests product from production sites and updates its inventory. Costs associated with this agent are inventory cost and transportation cost. Inventory cost may be different for each product.

Production site

Production site agent stores product and raw material inventory. It receives an order from warehouses and sends shipment accordingly. As production sites and warehouses are assumed to belong to the same enterprise, production sites aim to satisfy maximum demand from a warehouse. It produces a product using its own replenishment policy for maintaining a product inventory. Similarly, it maintains a raw material inventory by ordering raw materials from the raw material supplier. A bill of material relationship is used to express the conversion of raw materials to products. Costs associated with this agent are production cost, transportation cost, and storage cost.

Raw material supplier

Raw material supplier supplies raw materials to the production sites according to demand. There are no costs associated with this agent. It is assumed that the inventory of raw material supplier has no limits.

Auctioneer

Multiple enterprises communicate with each other through an auction mechanism. Auctioneer agent is responsible for conducting auctions. It receives bids and asks from retailers and warehouses respectively. Using this information about asks and bids, auctioneer performs the matching and communicates the final trade quantities and prices to warehouses and retailers respectively. It is assumed that there is no cost associated with this agent.

Auction mechanism

In each planning period, multiple rounds of auctions take place. In each round of auction, a warehouse enters as a seller and multiple retailers enter as buyers. The auction mechanism considered in this work consists of two steps that are matching and arbitration [140]. The matching step starts with the warehouse submitting its ask and retailers submitting their bids. An ask or a bid consists of price and quantity. Warehouse asks and retailer bids depend on a certain bidding strategy adapted from Steiglitz et al. [141]. This strategy allows the warehouse and retailers to adapt their asks to learn from their previous experience. Minimum acceptable price for a warehouse depends on available product amount and its bid in the previous planning period. This relation is given by Eq. (48).

$$\mathrm{bid}(t) = P(t-1)\mathrm{B}(\bar{\mathrm{f}}) \tag{48}$$

where,

$$B(\bar{f}) = 1.2^{1-\bar{f}} \tag{49}$$

$$\bar{f} = \frac{\mathrm{inv}(t)}{\text{target inventory}} \tag{50}$$

where $\mathrm{inv}(t)$ is the product inventory that each warehouse has at the start of planning period t; $\mathrm{bid}(t)$ is the price that warehouse bids in the planning period t; $P(t-1)$ is the bid price by the warehouse in the previous planning period.

Finally, warehouse bid cannot be lower than a product value to make a profit. The product value is determined by including all costs for transportation, production, and holding costs before making the product available at the warehouse. Retailers adjust their bid based on the trading price in the previous planning period. The extent to which retailers adjust their bids depends on the learning factor. Value 1.0 of the learning factor means that the bid will be equal to the trading price of the previous period. The maximum acceptable quantity for retailers is the demand, whereas, for the warehouses, it is the maximum quantity that it can deliver equals the available inventory.

After receiving bids and asks from retailers and the warehouse respectively, the auctioneer matches the warehouse to one retailer by calculating the maximum possible payoff that can be achieved by trading with a retailer. The payoff for a warehouse (Eq. (51)) is the revenue that it receives by selling a product. For retailers, the payoff is the profit gained by selling a product (Eq. (52)).

$$\pi_{\mathrm{wh}} = \left(\mathrm{P_{trade}} - \mathrm{TC_{r,wh}}\right)\mathrm{Q_{trade}} \tag{51}$$

where, π_{wh} represents payoff of a warehouse wh by trading with the retailer r; $\mathrm{P_{trade}}$ is the trading price; $\mathrm{Q_{trade}}$ is the quantity of product supplied to the retailer; $\mathrm{TC_{r,wh}}$ is the transportation cost for delivery of product from warehouse wh to retailer r. Maximum possible payoff for the warehouse is when $\mathrm{P_{trade}}$ is the maximum acceptable price by the retailer and $\mathrm{Q_{trade}}$ is the maximum possible shipment quantity which could be maximum quantity that warehouse can supply or the maximum quantity that a retailer needs.

$$\pi_r = (\mathrm{P_{sell}} - \mathrm{P_{trade}})\mathrm{Q_{trade}} \tag{52}$$

where, π_r is the payoff of a retailer by trading with a warehouse; $\mathrm{P_{sell}}$ is the selling price and $\mathrm{P_{trade}}$ is the trading price; $\mathrm{Q_{trade}}$ is the trading quantity.

Matching step is followed by the arbitration step. In the arbitration step, the actual price and quantity at which the trade will take place are decided. With the help of Nash bargaining mechanism, a fair-trade price and quantity are decided. Status quo point for the warehouse is the maximum payoff that it can achieve by trading with any other retailer. Status quo point for the retailer is zero. It is natural to assume here that the minimum acceptable price for a warehouse is less than the maximum acceptable price for the retailer. If this does not hold, the retailer is eliminated in the matching step. The problem is formulated using equations (53-57).

$$\min(\pi_{\mathrm{wh}} - \pi_{\mathrm{whmin}})(\pi_{\mathrm{r}} - 0) \tag{53}$$

$$\mathrm{s.t.}\ \pi_{\mathrm{wh}} \geq \pi_{\mathrm{minwh}} \tag{54}$$

$$\pi_{\mathrm{r}} \geq 0 \tag{55}$$

$$\mathrm{P}_{\mathrm{min}} \leq \mathrm{P}_{\mathrm{trade}} \leq \mathrm{P}_{\mathrm{max}} \tag{56}$$

$$0 \leq \mathrm{Q}_{\mathrm{trade}} \leq \mathrm{Q}_{\mathrm{max}} \tag{57}$$

where, $\mathrm{P}_{\mathrm{min}}$ is the minimum acceptable price for the warehouse; $\mathrm{P}_{\mathrm{max}}$ is the maximum acceptable price for the retailer; $\mathrm{Q}_{\mathrm{max}}$ is the maximum trade quantity. As payoff of warehouse and retailer depend on trade price and trade quantity from Eq. 51 and Eq. 52 respectively, solving the problem Eq.(53-57) gives the trade quantity and price based at which the trade takes place.

A simple demonstration of this mechanism is provided in Fig. **3-2** where a round of auction between two buyers and one seller is displayed. The horizontal axis represents prices and the vertical axis represents price. For each buyer and seller, a rectangular region marked by price and quantity represents the region in which the buyer or seller is willing to trade. Thus, trade can happen only in overlapping regions between a buyer and a seller. In Fig. 3-2, buyer 1 and seller can trade within regions A and C whereas buyer 2 and seller can trade within region B and C. Since there is only one seller, auctioneer calculates the maximum payoff that the seller can achieve by trading with each buyer. The buyer for which the payoff is maximum wins the trade.

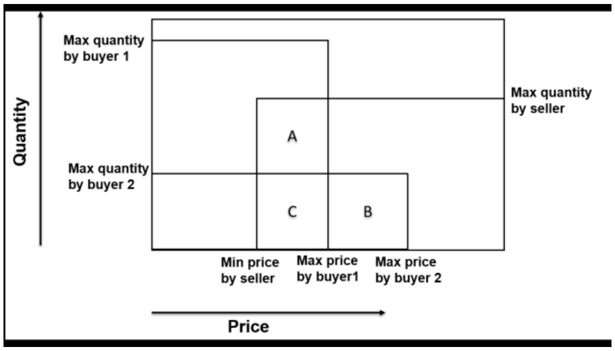


Fig. 3-2 Auction mechanism demonstration

Problem definition

The problem of minimizing the total cost of a supply chain network is considered. The total cost is obtained from the simulation. For the simulation, demand is assumed to be deterministic. Information about warehouse and production site inventories are externally provided to the simulation. Inventory of warehouses as well as production sites are determined by formulating a problem as described in Eq. (58) where Bound constraints on warehouse and production site inventories are considered.

$$\min \text{total cost} \tag{58}$$

$$\text{s.t.}\ \ \mathrm{lb_{wh,p}} \le \mathrm{inv_{wh,p}} \le \mathrm{ub_{wh,p}} \quad \forall \mathrm{p} \epsilon \mathrm{P}, \forall \mathrm{wh} \epsilon \mathrm{WH}$$

$$\mathrm{lb_{ps,p}} \le \mathrm{inv_{ps,p}} \le \mathrm{ub_{ps,p}} \quad \forall \mathrm{p} \epsilon \mathrm{P}, \forall \mathrm{ps} \epsilon \mathrm{PS}$$

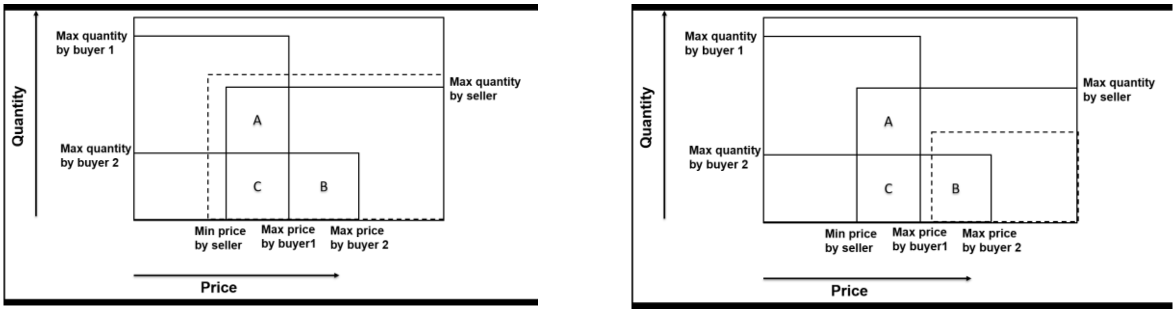
where the total cost is given by Eq. (59)

$$\text{total cost} = \sum_{t \epsilon H} \sum_{a \epsilon A} (\alpha_{TC} Q_{a,t} + \alpha_{PC} Q_{a,t} + \alpha_{IC} Q_{a,t} + \alpha_{UC} Q_{a,t}) \tag{59}$$

The total cost in Eq. (59) is obtained from the simulation and it is the summation of individual costs associated with each agent over the planning horizon H; $\alpha_{TC}Q_{a,t}$, $\alpha_{PC}Q_{a,t}$, $\alpha_{IC}Q_{a,t}$, and $\alpha_{UC}Q_{a,t}$ represent transportation cost, production cost, inventory cost, and unmet demand penalty cost respectively for an agent a and planning period t. $\mathrm{lb}_{\mathrm{i,p}}$ and $\mathrm{ub}_{i,p}$ are the lower and upper bounds on the inventory of product p at the warehouse or production site i; $\mathrm{inv}_{i,p}$ is the capacity to store product p at the warehouse or production site i and total cost depends on it; WH is the set of all warehouses; PS is the set of all production sites; A is the set of all agents; P is the set of all products; α is cost per unit of product, Q is the quantity of the product. Inventories at production sites as well as warehouses for each product are the decision variables in the problem given by Eq. (58) and Eq. (59). Given that different values are allowed for product inventories at each warehouse and production site, number of decision variables in the the problem are $(N_{wh} + N_{ps})N_P$ where, N_{wh} is the number of warehouses, N_{ps} is the number of production sites, and N_p is the number of products. In the context of analytical approaches, it is important to note that the problem defined by Eq. (58) and Eq. (59) does not include any integer variables since simulation handles such decisions internally.

Characteristics of the problem

In the supply chain network considered in this work, discrete decisions are involved at every stage. These decisions include the choice of a production site for delivering product to a warehouse, choice within a production site on the product to produce with priority, and the matching step in the auction. As warehouse inventory plays a significant role in auctions, changing warehouse inventory results in different decisions. To illustrate how auction results could change with a change in warehouse inventory, the matching step in a single round of auctions with one warehouse and two retailers is considered as shown in Fig. 3-2. It is assumed that the transportation cost to both retailers is the same. The maximum possible payoff for the seller by trading with both warehouses is observed as the seller inventory increases. The dotted line in Fig. 3-3(a) illustrates the new bid by the seller after increasing seller inventory. It is assumed that in the current state, buyer 2 wins the auction. Since in the current state, buyer 1 is willing to buy more amount than that offered by the seller, increasing seller inventory will increase the maximum potential payoff for the seller can achieve by trading with buyer 1. However, since the seller is satisfied to fulfill the demand for buyer 2 in the existing scenario, increasing seller inventory will not affect the maximum possible payoff the seller can achieve by trading with buyer 2. Therefore, there could be a point where payoff by trading with buyer 1 will surpass that of buyer 2. As a result, buyer 1 will be matched with the seller. On the other hand, the bid after decreasing the seller inventory is illustrated in Fig. 3-3(b). It is assumed that in the current state, buyer 1 wins the auction. As the inventory decreases, buyer 1 cannot enter auctions because it is eliminated in the matching step. As a result, buyer 2 will be matched with the seller. As transportation cost, inventory holding cost is different from all warehouses and similarly, production costs are different at production sites, it is reasonable to expect a discontinuity in the total cost. Moreover, from Eq. (59) one can observe that individual components of the total cost such as transportation cost, inventory cost, production cost and penalty cost for unmet demand depend linearly on the quantity of product. Since, warehouse bids (Eq. (48), Eq. (49), and Eq. (50)) and reorder amount are continuous functions of warehouse inventory, whenever total cost is continuous, it is safe to assume that it will also be linearly dependent on the warehouse inventory. Therefore, if the discrete decisions mentioned above do not change, the objective function is continuous and linear. Finally, because of the need to solve optimization subproblems such as the one given by Eq. (53) – Eq. (57) multiple times in each simulation run, the simulation is computationally expensive. **Fig**. **3-4** is obtained from a problem containing two warehouses and two products where the inventory of only one warehouse for one product is varied. It is a demonstration of the typical behavior of the objective function considered where discontinuities can be observed. However, in the continuous parts of the objective function, linear behavior can be observed. Because of these two properties, it is usually observed that the optimum lies at the boundary of the feasible region or at the boundary of the continuous region near the discontinuity. The framework proposed in section 3 is developed to address these important problem characteristics.



(a)

(b)

Fig. 3-3 Auction mechanism decision change demonstration (a) increasing seller inventory (b) decreasing seller inventory

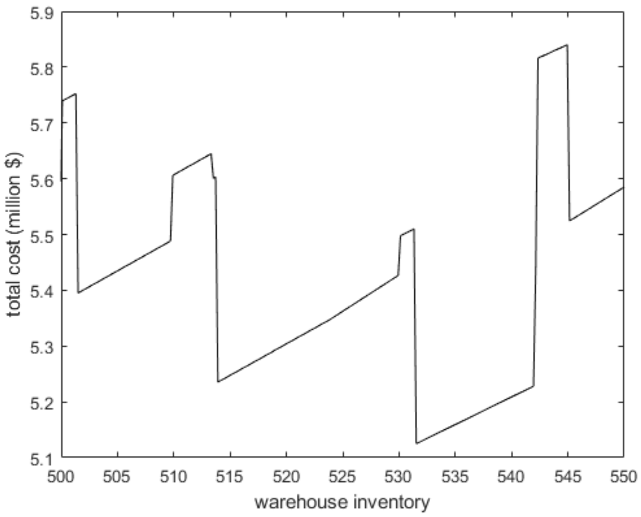


Fig. 3-4 Dependence of total cost on warehouse inventory

3.3 Optimization framework and algorithmic details

The framework presented in this section consists of three phases. The first phase involves a global search where promising regions for local optimization are identified. The second phase consists of discontinuity identification. In this phase, information about discontinuities is obtained. The third phase consists of a local search where discontinuity information from the second phase is utilized to guide the search toward a local optimum. These phases are iteratively carried out as shown in Fig. 3-5 until a budget of maximum function evaluations is reached or maximizing prediction variance fails to obtain an unsampled point. In this section, each phase in the framework is presented in detail followed by methodological details that include Kriging surrogate model, sparse grids, and support vector machines.

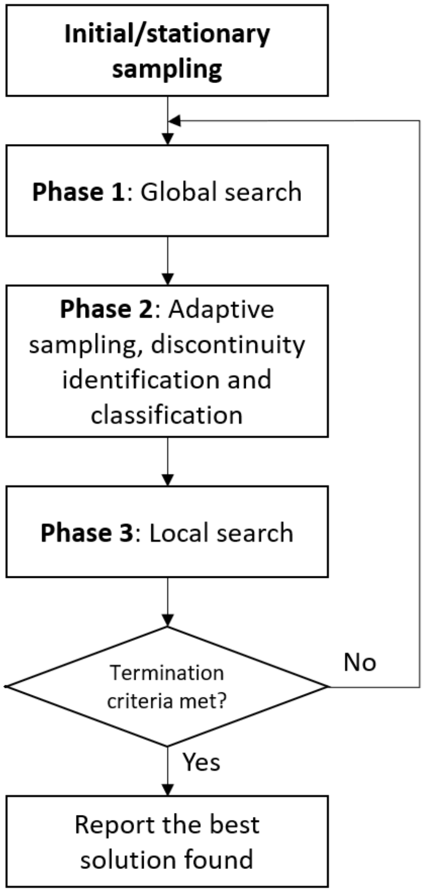

Initial/stationary sampling
Phase 1: Global search
Phase 2: Adaptive sampling, discontinuity identification and classification
Phase 3: Local search
Termination criteria met?
No
Yes
Report the best solution found

Fig. 3-5: Iterative optimization framework flowchart

Phase 1 (global search): Global search relies on a Kriging surrogate model built using a set S of points that is initialized using Latin Hypercube Design (LHD) with maximizing minimum distance criterion in the initial sampling phase [142] and iteratively updated. A squared exponential correlation model with a constant regression term is used for building the model. Details regarding the Kriging model are provided in section 2.1.4. Since the problem is nonconvex in nature, the resulting Kriging model is nonconvex as well. Local optima of the Kriging model decide promising regions for further exploration. To obtain more than one local optima, a multistart local search is used. In this work, MATLAB function 'fmincon' is used with 'Sequential Quadratic Programming' (SQP) solver option. After identifying local optima, the following phases are carried out on a box centered at each of the optima and having a size smaller than the feasible region of the whole problem. After completion of phase 2 and phase 3 in an iteration, an exploratory sample is collected by maximizing the Kriging prediction variance. This helps the framework in identifying new local optima that are discovered as the framework progresses and more samples are collected. With improving the Kriging model in every iteration, the local minima that were initially missed because of initial guesses in the optimization subproblem can be found in the subsequent iterations making sure that the solution quality is not affected. It is important to note that as opposed to traditional expected improvement maximization, not all points are used for building the Kriging model. As described in section 2.1.4, Kriging predictor depends on the correlation matrix R and elements of R depend on the correlation models described in Table 2-3. Since all the correlation models are distance based, two samples close to each other are highly correlated. In the presence of discontinuities, this correlation can be misleading. As a result, quality of the Kriging model after the optimization of hyperparameters may not be reliable. To avoid this issue, the set S of points used to build the model is maintained separately. Only the points that are farther away by a certain distance from all the existing points in S are included to the set S. Moreover, the set S is updated at two steps in the algorithm. First, the local optima of the Kriging model and second, exploratory samples.

Algorithm 1: global search

```
Initialize parameters: m, $\delta$, $\epsilon$, $S$
Build a Kriging surrogate model using the available data in set $S$.
Create a set D using LHD of size m in the search space.
Conduct a local search on Kriging surrogate model starting from points in the set D.
Obtain set P of the local optima.
Filter set P to contain points away from each other at least by the distance of $\epsilon$.
```

```
Create a set B of boxes with bounds [ $\mathbf{p}-\boldsymbol{\delta}, \boldsymbol{p}+\boldsymbol{\delta}$] for all points p in the set P where $\boldsymbol{p}, \boldsymbol{\delta} \in R^{d}$
If any box is partly outside the search space, adjust the bounds of the box to make it feasible
Use **Algorithm 2** to conduct a local search
Maximize prediction variance of the Kriging model to obtain an exploratory sample
```

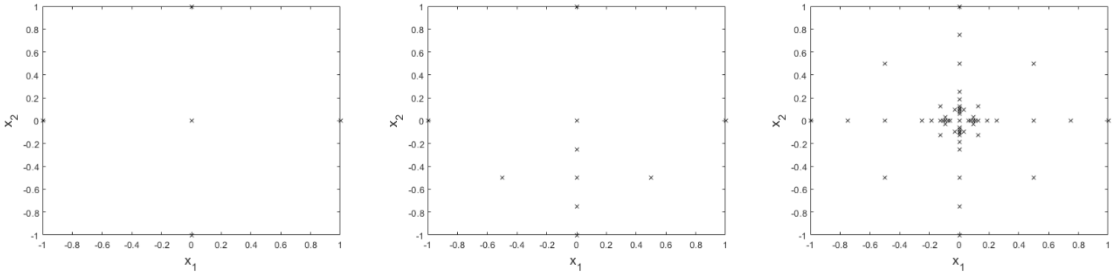
Phase 2 (discontinuity identification): In the second phase, discontinuity identification is carried out in each local region starting from a region that has the lowest prediction in Phase 1. For discontinuity identification, an adaptive sparse grid (section 3.3.1) of the user-specified level is generated. For all the samples and their neighbors, discontinuity detection is carried out to assess the possible presence of a discontinuity. This is done in a heuristic way as follows. A simplex of points is sampled in a small neighborhood at two different locations in the feasible space. Slopes between the points in the simplex are calculated. It is assumed that the maximum value from the calculated slopes provides a reliable estimate of the slope in continuous regions and hence, it is used as a threshold. Given two points, if the absolute value of the slope of the line joining those points exceeds the threshold, a discontinuity is considered to exist. If a discontinuity is present, the space is further divided by creating another sparse grid in a hyper-rectangle centered at the midpoint of two samples that need refinement. This procedure is repeated until no pair of neighboring points exists that has points separated by a distance greater than a predefined tolerance and have a discontinuity between them. Available samples are then labeled based on the continuous region they belong to. For labeling, each sample and its neighboring samples are assessed for discontinuity. Neighboring samples with no discontinuity between them are labeled the same whereas samples with a discontinuity between them are labeled differently. If there is no pair of neighboring points with a possible discontinuity, a line search based local search is triggered according to Algorithm 3. Support vector machines (SVM) is trained using the labels for available data (section 2.2.1). This classifier is used to represent the boundaries of a continuous region. Details of Phase 2 are described in Algorithm 2.

Algorithm 2: Discontinuity identification

```
Initialize parameters: box B, $\epsilon_{tol}$
Create a sparse grid of level 2 inside box B.
Evaluate the objective function at the grid points.
Identify the set N of pairs of neighboring points.
Initialize $\epsilon$ to be the minimum distance between all the pairs in N.
while $\epsilon > \epsilon_{tol}$ and $N \neq \phi$
    For each pair p in set N containing points $p_1$ and $p_2$, conduct discontinuity identification test.
    If discontinuity is not present, remove p from N.
    If discontinuity is present
        Let $d_{refine}$ be the dimension along which the points are neighbors.
        Let $z$ be the midpoint of two points in p.
        Create a hyper-rectangle H centered at $z$.
        Let $\delta_{d1}$, $\delta_{d2}$ as the maximum distance to neighbor along dimension $\mathrm{d}$ from $p_1$ and $p_2$ respectively.
        Define $\delta_d = \max(\delta_{d1}, \delta_{d2})$
        For $d \neq d_{\text{refine}}$ the bounds for H are set to be [$z_d - \delta_d$, $z_d + \delta_d$].
        Create a sparse grid of level 2 in H.
        Evaluate the objective function at the grid points.
        Remove pair p from the set N.
        Identify the neighboring points in H and add to the set N.
    End
End
Label all points such that any given pair of neighboring points should have same labels if discontinuity is not present. It should have different labels, if discontinuity is present.
Let the best sampled point belong to label a. Label all points that do not have label a to label b.
Train SVM using obtained labels.
```

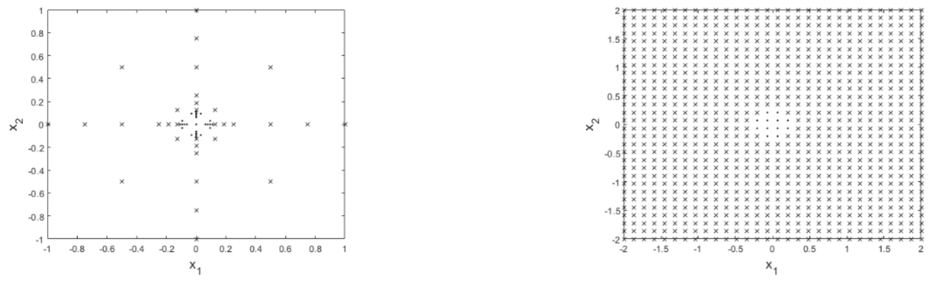
A demonstration of the steps mentioned in Algorithm 2 is provided in Fig. 3-6 and Fig. 3-7. The problem considered in this demonstration is taken from Jakeman et al. [132] and given by Eq. (60).

$$f(x) = \begin{cases} 0, & \sum_{i=1}^{2} x_i^2 < r^2 \\ 1, & \text{otherwise} \end{cases} \tag{60}$$



(a) (b) (c)

Fig. 3-6: Demonstration of sparse grid adaptive sampling (a) shows initial grid (b) shows refinement between a pair of points (c) shows all samples at the end of the adaptive sampling step



(a)

(b)

Fig. 3-7: Sample (a) labeling and (b)prediction using classifier in phase 2 of the framework

Threshold slope of 0 is assumed to be known a priori. Each sample is displayed in the Fig. 3-6 with a cross. The algorithm starts with sampling a level 2 grid in the local region. Two points are called neighbors if they differ only in one dimension and have no point sampled between them. All such pairs of neighboring points are collected. The slope is evaluated, and it is checked if the slope is higher than the known threshold of zero. Each pair is refined by creating another sparse grid of level 2 and evaluating the function at grid points. Refinement of one of the pairs (0,0) and (0, -1) is shown in the second figure. Neighboring points along the dimension x_1 of the point (0, 0) are (-1, 0) and (1, 0) which sets the hyper-rectangle width to 1. Similarly, Neighboring points along the dimension x_2 are (0, 1) which sets the hyper-rectangle length to 1. This refinement is iteratively carried out until no such pair of neighboring points exists that has discontinuity between the points in it or the distance between two points in the pair is larger than a predefined tolerance. The final set of sampled points are shown in Fig. 3-6 (c). As one can see, the framework samples more points near the boundary of discontinuity. Following this step, labeling is carried out by checking all pairs of neighboring points. If the points have a discontinuity between them, they are given separate labels. Whereas, when no discontinuity is detected, they are labelled the same. Since there are two continuous regions in the problem, two separate labels are obtained which are shown in Fig. 3-7 (a). Finally, SVM model is trained using these samples and used for future prediction. SVM predictions for this problem are shown in Fig. 3-7(b).

Phase 3 (local search): In the third phase, starting from the sample having the lowest objective function value, a line search is employed. To determine the direction of descent for line search, a simplex gradient is used that generates (d + 1) samples in a d dimensional space and determines the descent direction. Support vectors classifier developed in phase 2 is used to make sure that the line search stays in the same continuous region. Finally, the best objective function value among all available samples is reported. These steps are schematically shown in Algorithm 3. Line search stops when either small step sizes are reached, or a maximum number of failed evaluations is reached or if the search is stuck at the boundary of the continuous region.

Algorithm 3: local search

```
Initialize parameters: $\delta$,$h_0$,$h_{tol}$
Starting from the best available sample $x_0$, create a simplex within distance $\delta$ from $x_0$
Evaluate the objective function at the simplex
Evaluate simplex gradient $\nabla$.
Initialize a step size $h = h_0$.
While $h$ < $h_{tol}$
    $x_1 = x_0 - \nabla . h$
    if SVM($x_1$) $\neq \mathrm{SVM}(x_0)$
        reduce $h$.
    else
        evaluate the objective function $f$ at $x_1$.
        If $f(x_1) < f(x_0)$
            Increase $h$.
            $x_0 = x_1$
```

```
      Else
        Reduce $h$.
      End
    End
  End
```

Details regarding specific details of the proposed approach including the kriging model, sparse grid, and support vectors classification are provided in Sections 2.1.4, 3.3.1, and 2.2.1, respectively.

3.3.1 Sparse grids

Sparse grids is a discretization method often used in the literature as part of solving differential and integral equations [143] due to their ability to scale well with a number of dimensions and reduce the curse of dimensionality. For interpolation purposes, sparse grids have been shown to provide a good approximation. Detailed analysis of error bounds using interpolation models built from sparse grids has been studied in the literature [144]. With their ability to form a grid structure without exponentially increasing the sampling requirement, it has been applied in multiple other application areas. Recently, sparse grids were used for black-box optimization [145]. Since the objective function is computationally expensive in the black-box optimization, full grids where the sampling requirement grows exponentially with the dimensionality of the problem are impractical. This makes the use of sparse grids more attractive. The typical sampling requirements for forming sparse grids for a given dimension of input space and for a given level of user-specified discretization level are shown in Table 3-1.

Table 3-1 Number of samples with respect to the dimensionality of the problem and level of discretization

No. of variables	**Level 1**	**Level 2**	**Level 3**	**Level 4**	**Level 5**
2	1	5	17	49	129
4	1	9	49	209	769
6	1	13	97	545	2561
8	1	17	161	1121	6401

In this work, sparse grid sampling is used for identification of discontinuities and for adaptive sampling to better approximate discontinuities in the feasible space. Several types of sparse grids exist in the literature. However, since achieving a good interpolation is not the goal of this work, equidistant or trapezoidal sparse grids are used. The samples collected by adaptively sampling inside the local region, are used for training support vector machines.

3.4 Results

In this section, the proposed algorithm is applied to various supply chain optimization problems. The aim is to demonstrate that cost-effective solutions can be achieved with the proposed approach. A comparative study with the existing derivative-free optimization algorithms is provided for the optimal warehouse inventory allocation for three supply chain networks. The application is further extended to find the optimal inventory allocation for the combined warehouse and production site inventory.

For the first comparison, a supply chain network with two warehouses, two products, two markets, and three warehouses is considered as shown in Fig. 3-8. It is assumed that the inventory at production sites has been decided a priori and inventory allocated for both products at the warehouses is the same. For three different sets of values of parameters such as transport cost, inventory cost, and the penalty for unmet demand, three scenarios are generated, and the results are compared. Three other derivative-free optimization solvers are chosen for the comparison. The solvers and their underlying algorithms are presented in Table 3-2.

Table 3-2 Description of solvers used for comparison against the proposed framework

Solver name	Algorithm	Reference
NOMAD	Nonlinear mesh-adaptive search (direct pattern search or model-based)	[146] [147], 'opti-toolbox' [148]
ISRES	Stochastic ranking evolution strategy	[149], 'nlopt' [150]
EGO	Efficient global optimization	'surrogates toolbox' [151] [93]

The algorithms depicted in Table 5 were selected to represent a variety of different approaches that can be used for simulation-based optimization. Existing derivative-free optimization algorithms can be classified as deterministic algorithms which include model-based and direct algorithms and stochastic algorithms. Choice of the solvers under comparison was based on choosing one solver from each category. Out of the solvers presented in Table 3-2, EGO is a model-based algorithm that makes use of a global Kriging model, NOMAD is a direct search algorithm, and ISRES is an evolutionary algorithm. Solvers that need a starting point were initiated starting from the center of the box-bounded feasible region. A Latin Hypercube design was provided for EGO algorithm.

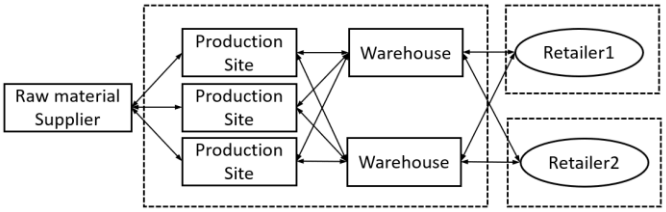


Fig. 3-8 Supply chain considered for comparison

A limit of 1500 function evaluations was given to all the solvers. In case of early termination by a solver, the best value found until the termination is reported. Bounds of 400 units and 600 units are imposed on the warehouse capacities. For the proposed framework, the median value is reported starting from 5 different initial sampling designs. Table 3-3 displays the optimal total cost value reported by solvers under comparison for all scenarios in 500 function evaluations. Respective values for 1000 and 1500 evaluations are shown in Table 3-4 and Table 3-5. The results demonstrate that the proposed algorithm provides the best objective function value for all three scenarios and under varied computational budget.

Table 3-3 Comparison of the objective function value provided by DFO solvers for different scenarios in 500 evaluations

Solver	**Total Cost (million $)**		
	scenario A	**scenario B**	**scenario C**
NOMAD	4.8233	4.7222	4.8336
ISRES	4.8826	4.6250	4.8617
EGO	4.8261	4.6014	4.8698
Proposed framework	4.7892	4.5821	4.7988

Table 3-4: Comparison of the objective function value provided by DFO solvers for different scenarios in 1000 evaluations

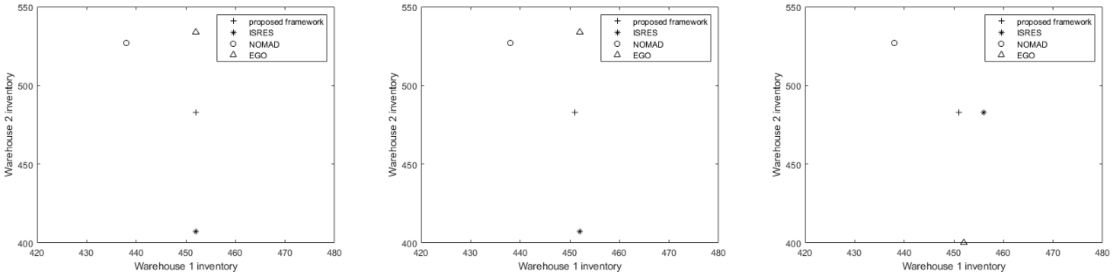
Solver	**Total Cost (million $)**		
	scenario A	**scenario B**	**scenario C**

NOMAD	4.8233	4.7222	4.8336
ISRES	4.8826	4.6024	4.8583
EGO	4.8261	4.6014	4.8575
Proposed framework	4.7802	4.5821	4.7988

Table 3-5: Comparison of the objective function value provided by DFO solvers for different scenarios in 1500 evaluations

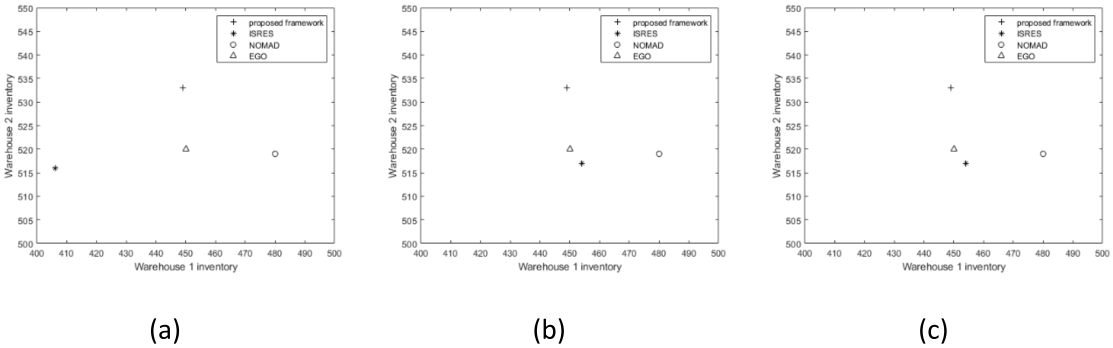
Solver	**Total Cost (million $)**		
	scenario A	**scenario B**	**scenario C**
NOMAD	4.8233	4.7222	4.8336
ISRES	4.8375	4.6024	4.8407
EGO	4.8105	4.6014	4.8575
Proposed framework	4.7802	4.5821	4.7913

In addition to the objective function value, it is also essential to assess the suggested solution to make sure that the superiority in the objective function value is not because of local refinement. To ensure that, suggested warehouse inventories by the proposed framework as well as ISRES, EGO, and NOMAD are reported in Fig. 3-9, Fig. 3-10, and Fig. 3-11. For all scenarios A, B, and C as shown in Fig. 3-9, Fig. 3-10, and Fig. 3-11 the solution suggested by the proposed framework is different from that suggested by the other algorithms. For scenario B as shown in Fig. 3-10, the solution is different for 500 evaluations. For higher computational budget, EGO is finding a solution close to the one suggested by the proposed framework. However, from Table 3-4 and Table 3-5, it can be concluded that the better solution is because of the better refinement achieved by the proposed framework. For scenario C as shown in Fig. 3-11, For 500 evaluations, the solution is different for all algorithms for 500 evaluations. Given a budget of 1000 evaluations, ISRES converges to the same solution as EGO. The solution is different from that of the proposed framework but very close. Here, the difference could be attributed to the local refinement achieved by the proposed framework based on the lower objective function value obtained as shown in Table 3-5. Finally, the proposed framework suggests a better optimal objective function value for all the cases under consideration. This highlights the success achieved by the discontinuity identification step in the local search. The computations are carried out on a PC with Intel® Xeon® CPU E5-1620 v2 @ 3.70GHz and 16.0 GB RAM, running a Windows 7 Professional, 64-bit operating system. The computational time for one simulation run for the network shown in **Fig. 3-8** is 26.71 seconds. The computational time for a bigger network consisting of 10 warehouses, 10 retailers, 7 production sites, and 10 products is found to be 282.76 seconds. These computational times are significantly large for an optimization problem. Since the simulation run is the most computationally expensive part of the algorithm, the number of simulation runs is usually considered to be a direct indicator of computational expense.



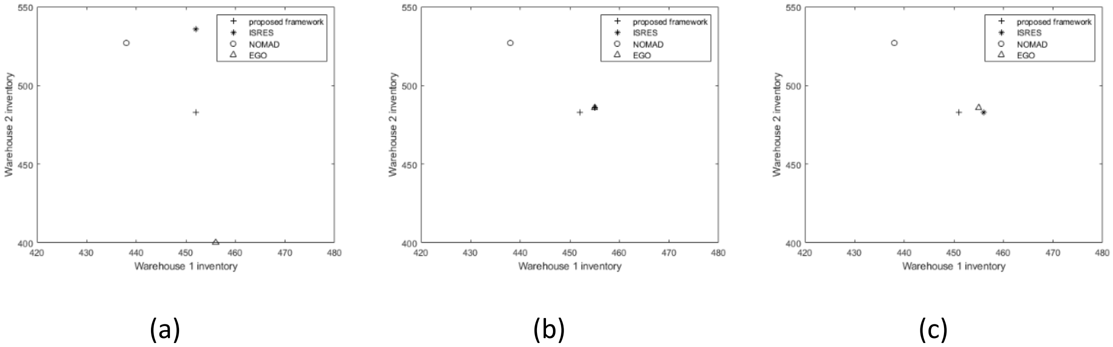
(a) (b) (c)

Fig. 3-9: Solution reported by the algorithms under comparison for scenario A (a) 500 evaluations (b) 1000 evaluations (c) 1500 evaluations



(a) (b) (c)

Fig. 3-10: Solution reported by the algorithms under comparison for scenario B (a) 500 evaluations (b) 1000 evaluations (c) 1500 evaluations



(a) (b) (c)

Fig. 3-11:Solution reported by the algorithms under comparison for scenario C (a) 500 evaluations (b) 1000 evaluations (c) 1500 evaluations

An example of improvement in the objective function values achieved with the number of simulation runs using the proposed framework is shown in Fig. 3-12. The simulation finds the best objective function value and continues exploring the unexplored regions until the computational budget of 1500 simulation runs is exhausted which explains the plateau after a certain number of simulation runs. For the same example, number of discontinuities were studied. The algorithm carried out phase 2 and phase 3 27 number of times and the combined number of continuous regions in the local search regions explored is 109 which is also the same as the number of discontinuities.

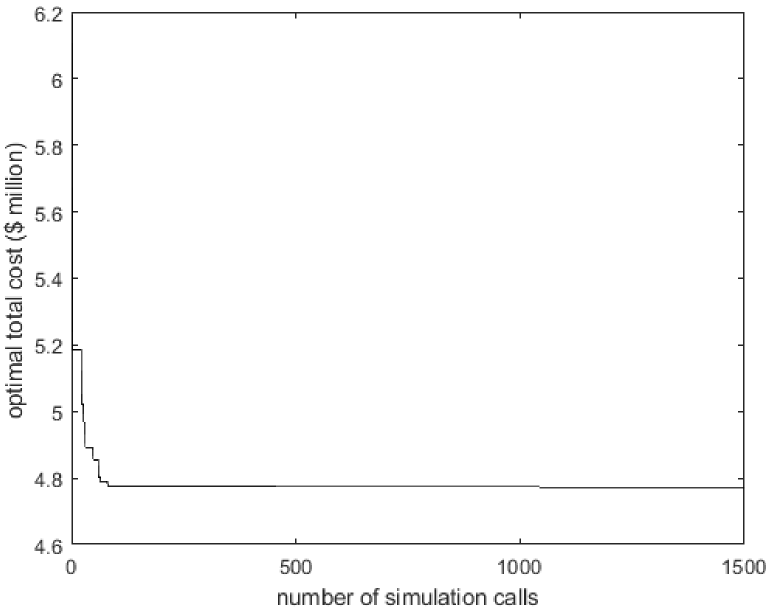


Fig. 3-12: Best objective function vs number of simulation calls for Scenario B

In the next comparison, the same network is considered. However, more decisions are made with varying production site capacities in addition to warehouses and allowing each warehouse and production site to hold a different amount of inventory for each product. The lower and upper bound on the production site capacities is 180 units and 220 units, respectively. Since the problem has two production sites, three warehouses, and two products, it results in a ten-dimensional problem. The problem is solved for scenario A. The optimal total cost reported is $4,743,655 with a computational budget of 1500 evaluations. Suggested warehouse and production site inventories are reported in Table 3-6. Comparing it with the previously obtained total cost for scenario A, including more decision variables leads to a more cost-efficient supply chain network.

Table 3-6 Optimal solution returned by the proposed framework

Unit	**Production site 1**	**Production site 2**	**Production site 3**	**Warehouse 1**	**Warehouse 2**
Product 1	210	204	187	573	503

Product 2	212	204	210	425	501

3.5 Summary

In this work, a multi-enterprise supply chain inventory optimization problem is considered. An agent-based simulation is used to model complex cooperative and competitive interactions between different enterprises. An auction mechanism is presented through which different enterprises interact with each other. It is observed that the problem of minimizing total cost with respect to inventory, has a discontinuous objective function. Since many existing approaches assume continuity of the response, a novel derivative-free optimization framework is proposed that can address discontinuous objective function by identifying and modeling discontinuities. The framework broadly consists of three phases that are global search, discontinuity identification, and local search. To save computational cost and to be able to handle problems of larger size, the framework makes use of adaptive sparse grid refinement. For discontinuity identification and local search, the framework makes use of certain problem characteristics such as linearity of the objective function in continuous regions to save computational cost. A comparison with three other existing derivative-free optimization solvers is made for supply chain networks. Results show that the proposed framework outperforms other existing algorithms in terms of the objective value and offers a solution that may be different from and superior to that obtained from other solvers. Finally, the framework is successfully applied to a higher dimensional problem where production site inventory is considered in as a decision variable in addition to warehouse inventory. The performance of the proposed framework is demonstrated on the resulting ten-dimensional problem. This work highlights the need to assess possible discontinuities prior to choosing a derivative-free method. Even though in this work discontinuities are identified for the purposes of optimization, the approach used in this work for approximating continuous regions with a classifier is more general. It is trivial to generalize this for use in modeling a discontinuous response in other applications. Finally, a more general discontinuity identification technique based on the data and with theoretical guarantees will be a valuable contribution to the literature on derivative-free optimization.

4 Modular design optimization

Abstract

Recent studies on modular and distributed manufacturing have introduced a new angle to the traditional economies of scale that claim that large plants exhibit better efficiencies and lower costs. A modular design has several advantages, including higher flexibility of decisions, lower investment costs, shorter time-to-market, and adaptability to market conditions. While design flexibility is a widely studied concept in the process design, modular design provides an interesting new opportunity to the design optimization problem under demand variability. In this work, a framework for modular design under demand variability is proposed. The framework consists of two steps. First, the feasible region for each module is represented analytically with the help of the historical data or the data from a simulation using a classification technique. In the second step, the optimal design choice is obtained by integrating the classifier models built in the first step as constraints in the design optimization problem. The design optimization problem is first solved considering a single objective, i.e., minimizing the total cost or maximizing the flexibility. These two objectives are then addressed simultaneously using a multiobjective optimization framework that considers the tradeoff between maximizing the flexibility of design and minimizing the cost. Computational studies conducted using a case study of an air separation plant, demonstrate the efficacy of the proposed framework. Several advantages of using a modular design, as well as data-driven methods in the decision-making process in the design step, are discussed.

4.1 Introduction

For years, the design of chemical process facilities has followed a traditional cost reduction paradigm relying on the economy of scale [8]. The 2/3 power law implies that as chemical plants grow bigger (scale-up), the capital cost increases following a 2/3 power low. However, large plants may exhibit better efficiencies and lower costs due to more efficient process integration. Recent studies on modular and distributed manufacturing have introduced a new angle to the economy of scale [11]. While large plants have better efficiency, due to their centralized nature, all raw materials have to be transported to the plant and similarly, all products have to be distributed from the plant. In the usual case where the source of raw materials, as well as the demand, is not geographically close to the large plant, there are significant transportation costs involved. With smaller scale modular plants, one can effectively distribute manufacturing which leads to reduced transportation costs. While large plants demand a large investment at the beginning, small plants require a relatively smaller investment and this significantly reduces the risk for the investors [152]. Finally, construction times for large plants are longer than those for a small plant and therefore, time-to-market is less for a small plant, again, reducing the risk.

Modular design involves the use of small and standardized modules of fixed size in a production process. Multiple identical devices may be assembled to achieve the desired production. Modular and distributed processes may not only contribute to decreases in distribution costs but also provide an alternative to overcome several manufacturing challenges. Small devices offer inherent safety and can be used for on-demand and on-site production of hazardous materials [9]. They provide a fast path to commercialization since challenges related to the scaling up of chemical processes are not substantial. Moreover, the time for construction of manufacturing facilities may be reduced, since modules can be preassembled in a shop and are not subject to delays related to weather and on-site inspections. Because of standardized units, the process of numbering up as a part of plant expansion becomes faster. Economically, as the standardized units or small modular plants are numbered up, vendors, as well as process engineers, gain experience. As a result of the learning curve, the vendors may be able to sell the equipment for a cheaper price and process engineers can reduce the time-to-market. All these factors contribute to a relatively lower risk of investment related to small modular designs. Recent work on modular design quantitatively demonstrates some of these advantages. Arora et al. [17] study the economy of numbers and equipment standardization for capital cost reduction. Yang and You [153] compare modular methanol manufacturing with and without module relocation, and large-scale methanol manufacturing based on the economic as well as the environmental impact. Sánchez and Martín [154] assess the modularization of ammonia plants as the production capacity is varied. Modular processes can also provide additional flexibility to production processes when compared to large-scale plants. With small and flexible modules, business units can carry out production plans and introduce new products independent of each other. The flexibility of modular plants has been investigated by Lier et al. [152], who demonstrated how to adapt capacity by starting up or shutting down the operation of certain modules according to different market developments. While large plants require high-investment decisions, modular plants provide managers with alternatives that are lower in investment costs and can adapt according to better forecasts. While a centralized plant can make use of customized designs, modular plants rely on available standardized modules. Therefore, the key-question in modular design then becomes to define a process based on a limited number of different modules [155]. In the context of this work, modular designs refer to the design and construction of smaller chemical process units or even entire processes of fixed production capacities [14]. It is important to note that this definition includes the possibility of process intensification [15], transportable processing units [16], standardization of equipment modules [17], and even integrated or customized unit operations [8]. Finally, the modularization of a process depends on the process knowledge and expertise of the engineer.

In addition to flexibility with respect to management decisions and market conditions, it is also important to have flexibility with respect to uncertainties. During conceptual design, there are often process parameters that are not well known, such as kinetic rate constants, demand, or product and feedstock prices. Understanding the operability characteristics of a process is therefore crucial at the design stage. It is addressed in the literature under the name of feasibility analysis, flexibility analysis, and operability analysis [156][157]. Analyzing the flexibility of a design is a fundamental concept in process design and it refers to quantifying the ability of a process to maintain feasible operation under variability due to uncertain parameters [13]. Ensuring flexible designs allows one to systematically hedge against exceptional realizations of process parameters [11]. The work on flexibility analysis started over three decades ago and it is still an active area of research [12][13]. This work can be broadly classified in static flexibility analysis and resiliency which deals with dynamic flexibility analysis. Some of the theoretical advances in flexibility analysis started with proposing a quantitative measure of flexibility known as the flexibility index [102]. Initial formulations for obtaining the flexibility index involved solving the problem as a multilevel optimization problem. Later research focused on reducing the computational complexity of this problem with the help of methods such as vertex enumeration [158], active constraint strategy [159], parametric programming [160], and as a global optimization problem [161] for nonconvex cases. More recently, Zhao et al. [13] presented a method of space projection for quantification of flexibility. Goyal and Ierapetritou [162] propose a simplicial approximation approach for obtaining operating envelopes within which a design is feasible. Pulsipher and Zavala [163] propose a mixed integer conic formulation for computing the flexibility index when the uncertainty is characterized using multivariate Gaussian random variables. Ochoa and Grossmann distinguish uncertain parameters as measured and unmeasured uncertain parameters. They propose MINLP reformulations for the resulting multilevel optimization problems [164]. Instead of representing the uncertainty by a hyperrectangle, there are approaches for obtaining flexibility by considering probability distribution functions. Relevant work in this area proposes the stochastic flexibility index [165] or expected stochastic flexibility [166]. Flexibility analysis also has a wide range of applications ranging from product design [167], process design and synthesis [168][169] to supply chain design [170]. For the dynamic systems, Dimitriadis and Pistikopoulos [171] propose a dynamic flexibility index. They extended the flexibility analysis to consider time-varying uncertain parameters and, thus, the feasible region. Moreover, since the dynamic behavior of a system is greatly influenced by the installed control system, resilient designs are proposed to simultaneously consider operational aspects as well as the steady state or economic aspects [172]. Palazaoglu and Arkun [173] address this problem by formulating it as a multiobjective optimization problem. Luyben and Floudas [174] translate this problem as an MINLP problem where alternatives for the process also vary in the control system, thus also addressing controllability. Later approaches for integration of design and control utilized computational advances and proposed dynamic optimization frameworks. These frameworks broadly consist of an iterative procedure implementing dynamic flexibility and feasibility analysis. Sanchez-Sanchez and Ricardez-Sandoval [175] propose an MINLP framework that simultaneously considers dynamic flexibility and feasibility in a single optimization formulation. Swartz and Kawajiri [176] review the applications of dynamic optimization for analyzing the interaction between design and dynamic performance. For a more detailed review on integration of process design and control, the readers are referred to relevant texts [177][178][179] [180]. Even though integration of design and process control is an interesting problem, the scope of this work is restricted to steady-state processes. A review of flexibility analysis and resiliency is provided by Grossmann et al. [181]. While flexibility analysis tries to determine the maximum disturbance from the nominal point in uncertain parameters that can be handled by a design, a similar but not identical concept known as operability aims at finding if the desired output ranges can be achieved by the controller in the presence of disturbances within the available input space [157].

Operability is quantified with the help of operability index and dynamic operability index. In the dynamic operability index, the assessment is done by solving an optimal control problem to find the shortest time for a process to respond to a disturbance and move to a new operating point. However, the optimal control problem has a solution only when there is at least one feasible solution to the final time constraints at steady state [182]. As a result, the feasibility and flexibility methodologies presented in this work also play an important role in considering further extensions to the problem of operability index or dynamic operability index. For more details on dynamic operability, an interested reader is referred to the relevant review paper [182]. Finally, Mohideen et al. [183] propose a way to simultaneously consider flexibility analysis and controllability as well as operability [184].

These research works rely on the closed form expression of the simulation model. A common problem arises when the process information is not available in a closed form, but it is available in the form of data or a computationally expensive simulation. In such cases, the literature is limited to feasibility analysis where the aim is to identify the feasible region where all the relevant constraints are satisfied. Feasibility analysis when a closed form expression of the problem is not available is referred to as black-box feasibility analysis. These methods rely on building a data-driven approximation or a surrogate model using the data generated from the complex simulation [10]. Banerjee and Ierapetritou [167] use high dimensional model representation (HDMR) surrogate model to determine the feasible region. Using a Kriging surrogate model based approach, Boukouvala and Ierapetritou [97] approximate the feasibility function, a metric for feasibility. Zhang et al. [185] propose a convex region surrogate for representing a nonlinear and nonconvex feasible region by a combination of convex regions. They approximate the cost function for each region by a linear approximation. Adi et al. [186] use a random line search for detecting boundary points of the feasible region. With their ability to utilize process data coupled with recent developments in machine learning software, data-driven methods are continually finding new applications for various problems in process systems engineering [187][188].

The problem of process synthesis and design optimization has been widely studied in the process systems engineering literature. The process synthesis problem refers to synthesizing processing systems via simultaneous structural and parameter optimization. In the structural optimization part, the aim is to select a configuration or a topology from available alternatives. In the parameter optimization part, the aim is to choose equipment sizes and operating conditions [189]. Conceptually, this problem leads to an MINLP problem where binary variables refer to the potential existence of units, and continuous variables refer to process operating conditions, flows, pressures, and equipment sizes [190]. For the structural optimization, process synthesis problem may consider different technologies and different connections in the possible set of alternatives. In the design optimization, however, the problem usually refers to optimizing a design after the technology is chosen. In this work, the optimization problem considered addresses the problem of selection between several available designs, and the problem is referred to as a design optimization problem. Even though this problem achieves simultaneous optimization of the structure and operating conditions, mathematical complexity of the resulting optimization model imposes a limitation on its applicability on a large-scale problem. In an attempt to reduce computational complexity, data-driven models were used for addressing these problems. Henao and Maravelias [191] propose a superstructure optimization framework where instead of a detailed process model, a surrogate model is utilized. Wang et al. [192] replace first principle models in a refinery hydrogen network with surrogate models and solve the problem of finding an optimal hydrogen network. Rafiei and Ricardez-Sandoval [193] highlight the potential of novel artificial intelligence (AI) and ML-based techniques such as artificial neural networks (ANN) for utilizing big data in the process of decision-making. For the feasibility analysis of a process flow sheet, the current literature relies on building approximations considering the entire flow sheet as a single unit [194]. A limitation of this approach is that since the majority of the state-of-the-art surrogate models handle continuous input and output variables. As a result, when the problem involves discrete decisions, one needs to build a separate surrogate model for each combination of discrete variables. This is computationally impractical and difficult to utilize in an optimization framework. Another common aspect in the above-mentioned approaches is that the analysis is done after the process or product design is available and the design decision is not directly dependent on the flexibility. In other words, other considerations in the design optimization step such as cost minimization or profit maximization are addressed separately from the flexibility analysis problem. As correctly highlighted in a recent study [193], there is a growing need to address multiple interconnected objectives at the design stage. Modular design provides an interesting new opportunity in the design optimization problem under demand variability in this context since the feasibility analysis for each module can be conducted beforehand, and a simultaneous design optimization and flexibility evaluation can be performed.

In this work, a framework for modular design under demand variability is presented. It is assumed that several module options for different equipment are available, that equipment are arranged in a sequential process, and that the sequence of equipment is known and fixed. Then, given a certain demand space, the goal is to determine the optimal selection of module options that minimize investment costs, maximize flexibility, or both while ensuring that the desired demand space can be covered. The framework consists of two basic steps: first, the feasible region of different module options is determined using a data-driven approach. Then, the simultaneous design optimization and flexibility evaluation problem is formulated as a multiobjective optimization problem and solved to optimality. With the help of traditional process synthesis literature, the design optimization problem considered in this work can be easily extended to handle the problem of several process alternatives or the cases where the sequence of the process modules is not unique. This extension also encompasses the possibility of having a module obtained using process intensification. In the comparisons presented in this work, however, the scope is not limited to process synthesis. Therefore, for clarity, the applications demonstrated in this work do not include factors such as process alternatives and process intensification. Moreover, a single series of modular processing equipment is considered since the possibility of numbering up of equipment exists to meet the product demand, and this will not affect the process feasibility or flexibility. Modular designs that rely on process intensification may pose challenges due to the loss of degrees of freedom for control [9]. Some of the key operational challenges in this context arise due to coordination between modules and, therefore, cooperative control strategies, discrete decisions in numbering up or numbering down of equipment, a sudden change in operating conditions requiring coordination between scheduling and control. These interesting problems motivate further research in the area of process control for modular designs.

This work proposes a novel approach for the flexibility analysis problem with the help of classifier models and provides an extension to the feasibility analysis approach proposed by Dias and Ierapetritou [195]. The multiobjective design optimization framework, as well as the flexibility analysis, take advantage of the modular design of the process by analyzing the feasibility of each module separately. In doing so, this work proposes a novel way to combine classifiers for the feasibility of individual modules such that process constraints such as mass and energy balances are implicitly handled. The design optimization framework is generic and can be easily adapted to new classification techniques and different definitions of flexibility. The framework can take advantage of the large amounts of historical data collected by an enterprise and use it to build better models and address the problem of large size.

The chapter is organized as follows. In section 4.2, a detailed description of each component of the proposed framework is provided. The problem addressed in this work is defined along with the specific steps of the proposed framework in Section 4.4. Section 4.5 provides an illustrative example of the framework and demonstrates the results on a small problem of a process containing a reactor and a separator. Section 4.6 presents a case study of an air separation unit (ASU) and resulting optimal design choices using the proposed framework. Finally, the conclusions are provided in section 4.7.

4.2 Background

In this section, a background is provided on some of the key steps in the proposed framework. More specifically, a state-of-the-art literature review is provided on the concepts of feasibility and flexibility analysis, machine learning classification-based feasibility analysis and on quality assessment of the classifier for feasibility analysis. This review is not meant to be exhaustive but intends to highlight some of the key differences between existing methods and the proposed framework. In doing so, detailed mathematical formulations for the relevant subproblems are provided.

4.2.1 Feasibility Analysis

This work deviates from the previous works mentioned in section 2.4 and extends a relatively new approach for the feasibility problem recently proposed by Dias and Ierapetritou [195] to the problem of design optimization. In this approach, the feasibility problem is treated as a classification problem. This way, machine learning algorithms for classification can be used to identify the feasible region of a system. Key advantages of this interpretation are the ability to handle large amounts of process data both in size (number of data points) and dimensions (number of features), and the availability of the sophisticated machine learning software tools.

In the design optimization problem considered, feasibility analysis for each module is conducted using the historical data that contains information about flow rates, temperatures, product demands, and quality requirements as input variables and feasibility labels of the process as the output. The classifier built from the historical data aims to accurately represent the region in the design space for which the module is feasible. It is important to note that historical data naturally contains a larger number of feasible data points and a small number of infeasible data points. This problem is known as the problem of imbalanced classes and has been widely studied in the field of machine learning [196][197]. However, since one of the ways to tackle this problem is by having more samples from the minority or, in this case, infeasible class, simulation models may be used. Moreover, one can increase the number of infeasible data points using process knowledge. The feasibility of the entire process is ensured by assessing the feasibility of classifiers for all the individual modules.

4.2.2 Flexibility analysis

Several approaches for solving the flexibility index problem are available in the literature as shown in section 2.5. In this work, vertex solution method [158] is used where it is assumed that critical points correspond to vertices of extreme values of the parameter set $T(F)$. In a general case, the problem is solved by first obtaining flexibility index in the direction of each vertex and then representing flexibility index as the minimum of all indices. This procedure is shown below:

Step 1: For each vertex direction k, solve the problem given by Eq.(61) and obtain F^k.

$$\begin{gathered} F^k = \max \delta \\ \text{s.t.}\, f_j\left(d, z, \theta^k\right) \leq 0 \\ \theta^k = \theta^N + \delta \Delta \theta^k \end{gathered} \tag{61}$$

Where, $\Delta\theta^{k}$ represents the expected deviation in the direction of vertex k

Step 2: Flexibility index F is calculated as the minimum of all δ^{k}.

$$F = \min\{F^{k}\} \tag{62}$$

However, such an iterative procedure cannot be incorporated into a single optimization problem. This problem is addressed by including a separate constraint for each vertex. This way, the optimization problem naturally selects the minimum value of F^{k} as the flexibility index. The constraints for flexibility index are shown by Eq.(63) and Eq. (64)

$$f_j\left(d, z, \theta^{k}\right) \leq 0 \; \forall k \in V \tag{63}$$

$$\theta^{k} = \theta^{N} + \delta\Delta\theta^{k} \forall k \in V \tag{64}$$

where, V is the set of all vertices.

4.2.3 Feasibility analysis using support vector machine

The output of Eq. (35) is binary where in the context of this work, a value of -1 corresponds to infeasible points whereas a value of +1 corresponds to the feasible points. However, since the output of Eq. (35) is not smooth and the sign operator may lead to computational difficulties in the optimization step, the expression given by Eq. (65) is used instead while modeling SVM models as constraints in the optimization problem. The only difference between Eq. (**35**) and Eq. (65) is the absence of $\mathrm{sign}(.)$ operator. Since a value of +1 for $\hat{f}(x)$ in Eq. (35) corresponds to a feasible point, a positive value of $\bar{f}(x)$ in Eq.(65) represents a feasible point. Similarly, since a value of -1 for $\hat{f}(x)$ in Eq. (35) corresponds to an infeasible point, a negative value of $\bar{f}(x)$ in Eq.(65) represents an infeasible point.

$$\bar{f}(x) = \sum_{i=1}^{n} y_i \alpha_i K(x, x_i) + \beta_0 \tag{65}$$

4.3 Process design optimization

4.3.1 Single objective optimization

In general, the deterministic process design optimization problem involving the selection of process units and their interconnection, as well as the evaluation of the design and operating variables, results in a mixed integer nonlinear programming (MINLP) problem [189]. The problem is formulated as shown by Eq. (66)

$$\begin{gathered} \min C(y,x) \\ \mathrm{s.t.\,h(y,x)} = 0 \\ \mathrm{g(y,x)} \leq 0 \\ \mathrm{y} \in \{0,1\}^{\mathrm{m}}, \mathrm{x} \in \mathrm{R}^{\mathrm{n}} \end{gathered} \tag{66}$$

where, y are binary variables that represent inclusion or exclusion of units with values 1 and 0 respectively; x are continuous variables which corresponds to process variables such as flow rate, composition, and temperature; $C(y,x)$ is the objective function; $h(y,x)$ represent equality constraints and $g(y,x)$ represent inequality constraints. Usually, for the design optimization problem, the objective function considered is the total cost of the process.

Often, at the design stage, a number of data, external to or within the process, may not be fully determined or known with certainty. For example, external conditions such as product demands, economic data, and environmental parameters could typically only be forecasted, given by a range of possible values or some probability distributional form. Therefore, it is clear that some degree of flexibility must be introduced at the design stage to ensure that the plant will be able to handle uncertain parameters during operation. There are several approaches to address this problem based on the description of uncertainty. One of the proposed procedures is the deterministic approach, where the description of uncertainty is provided by specific bounds or via a finite number of fixed parameter values. An alternative approach is a stochastic approach where uncertainty is described by probability distribution functions.

Since the uncertainty considered in this work mainly arises from the product demand for which the range of demand variability is known, a deterministic approach is used. Uncertain range of parameters is bounded by box constraints, i.e., the lower and upper bounds for each demand define the demand variability. Stochastic uncertainty can be addressed using stochastic flexibility analysis, which requires modified cost objective and data sampling. However, it is worth noting that for the cases when the uncertainty is present in the process parameters, considering a deterministic uncertainty may lead to a conservative design [198]. This motivates an interesting extension of the present work for incorporating stochastic flexibility index, and it will be considered in future work.

4.3.2 Multiobjective optimization

In addition to achieving an objective such as minimizing the total cost, it is often of interest to assess other aspects of the optimal design, such as robustness or flexibility. One way to address this tradeoff is to penalize the objective function with the other objectives [199]. However, such optimization leads to only one solution, and the solution is dependent on the penalty function. Therefore, such an approach does not explore the tradeoff between several objectives systematically. To achieve that, a multiobjective optimization problem should be solved. A vast amount of literature is available on multiobjective optimization, and a review can be found in Marler and Arora [200]. This work uses the ϵ- constraint method, and the problem formulation is represented by Eq. (67)

$$\begin{gathered} \min f_1(x) \\ \mathrm{s.t.\,f_2(x) \leq \epsilon} \end{gathered} \tag{67}$$

$$\mathrm{h(x)} = 0$$
$$\mathrm{g(y,x)} \leq 0$$

where, $x \in R^p; y \in \{0,1\}^q$ where p is the number of continuous variables and q is the number of binary variables; f_1 and f_2 are the two objectives to be minimized in the problem; $h(y,x)$ represent equality constraints, and $g(y,x)$ represent inequality constraints. In this work, the objectives considered are the total annualized cost and the flexibility index. Since bigger units are feasible over a wider range, they exhibit higher flexibility. On the downside, these units are more expensive. Therefore, a multiobjective formulation is used to handle the trade-off between cost and flexibility.

4.4 Problem definition and the proposed framework

4.4.1 Problem definition

The motivation for the problem considered in this work stems from the general concept that the development of standardized designs considering the customer demand space can lead to significant economic savings. The relevance of the work can be appreciated from the fact that developing modular designs would be substantially cheaper for a manufacturer than developing customized designs, and beneficial for the customer because various design alternatives would be available to choose from.

This work aims to find the optimal design for a modular process where several standardized options for the process module are available. It is assumed that several module options for different equipment are available, that equipment are arranged in a sequential process, and that the sequence of equipment is known and fixed. For each module, the options are different from each other based on the definition. As an example, if the reactor module is defined based on the reactor volume, the options for the reactor module will include reactors of different volumes.

Moreover, based on the variables associated with options, each option has a feasible range of operation. If an option is selected, it is important to make sure that the solution is feasible for that option. It is assumed that the historical information of the process feasibility data is available in the form of process variables and their corresponding feasibility labels. To write it formally, n instances in the space of process variables $x \in R^p$ and corresponding labels $y \in \{-1,1\}$ (-1 for infeasible and 1 for feasible) are available as $X = [x_1, x_2, x_3, \dots, x_n]$ and $Y = [y_1, y_2, y_3, \dots, y_n]$ where, p is the dimension of the problem, and n is the number of labeled instances.

Based on the objective considered, three different problem formulations can be used. First, the optimal design is obtained for a given demand for products in order to minimize the total cost of the process. In the second case, given the nominal demand and the variability from the nominal values, the objective is to find the design that maximizes flexibility with respect to the demand variability. In most cases, however, both of these objectives of minimizing cost and maximizing flexibility will be of interest. The third problem considers both the objectives and aims to find a number of solutions (pareto) that balance those two objectives. Moreover, since quantitative information is available regarding the feasibility and flexibility of the chosen optimal designs, it will facilitate steady-state and dynamic operability analysis.

4.4.2 Framework for optimization

The framework for modular design under demand variability consists of five steps, as shown in Figure 4-1. First, the historical process data for feasibility is obtained. The data is split between training data $S_{train} = [(x_1, y_1), (x_2, y_2), \dots, (x_{T1}, y_{T1})]$ and testing data $S_{test} = [(x_1, y_1), (x_2, y_2), \dots, (x_{T2}, y_{T2})]$ where $T1$ and $T2$ are the number of instances in the training set and the testing set, respectively. Using the training dataset, a classifier is built that will attempt to act as an accurate representation of the unlabeled data. The performance of the classifier is tested on the testing data, and the predicted labels are compared against the true labels in the data. For comparison, four metrics are used. A detailed explanation of the validation metrics is provided in section 2.7.2. The process is repeated for evaluating the feasible regions for each module option is evaluated. Finally, algebraic expression for the classifier is obtained, and the equation is incorporated in the process design optimization problem. A typical way to add the classifier as a constraint is shown below

$$\mathrm{CF_{om}(x)} \geq -\mathrm{M}(1-\mathrm{y_{om}}) \tag{68}$$

where CF_{om} is the classifier for option o of module m; y_{om} is a binary variable for the selection of option o for module m; M is a positive constant for the big-M type of constraint. Since the positive value of the classifier indicates feasibility, the constraint makes sure that if y_{om} is 1, $CF_{om}(x) \geq 0$.

The feasibility analysis approach used here is proposed by Dias and Ierapetritou [114] for the integration of planning and scheduling problems. Such an approach helps in obtaining an algebraic equation of the feasible production region without the use of detailed dynamic models. The combination of modules and respective feasible regions generates the overall feasible production space of a set of module options. It should be noted that the purpose of this work is not to provide a comparative study between several classifiers from the machine learning literature but to provide a framework for optimization. As SVM has shown a good predictive ability from the analysis of Dias and Ierapetritou [114], SVM is the classifier used in the rest of this work. For the flexibility analysis, this work limits itself to vertex enumeration strategies. However, as the number of uncertain parameters increases, a more computationally efficient active-constraint strategy can be utilized.

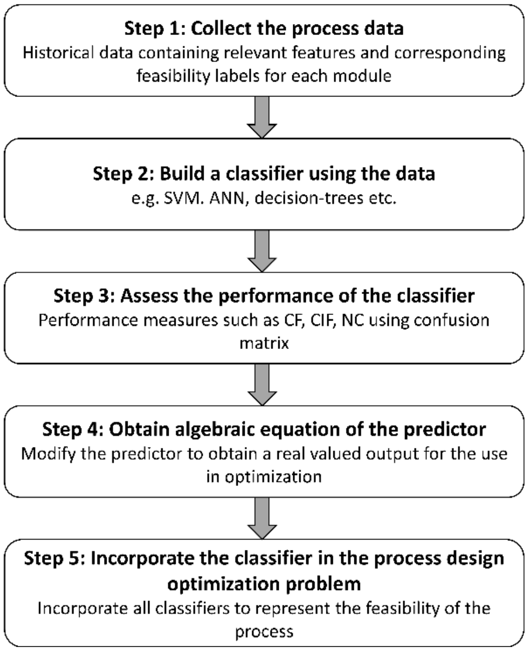

Step 1: Collect the process data
Historical data containing relevant features and corresponding feasibility labels for each module
Step 2: Build a classifier using the data
e.g. SVM. ANN, decision-trees etc.
Step 3: Assess the performance of the classifier
Performance measures such as CF, CIF, NC using confusion matrix
Step 4: Obtain algebraic equation of the predictor
Modify the predictor to obtain a real valued output for the use in optimization
Step 5: Incorporate the classifier in the process design optimization problem
Incorporate all classifiers to represent the feasibility of the process

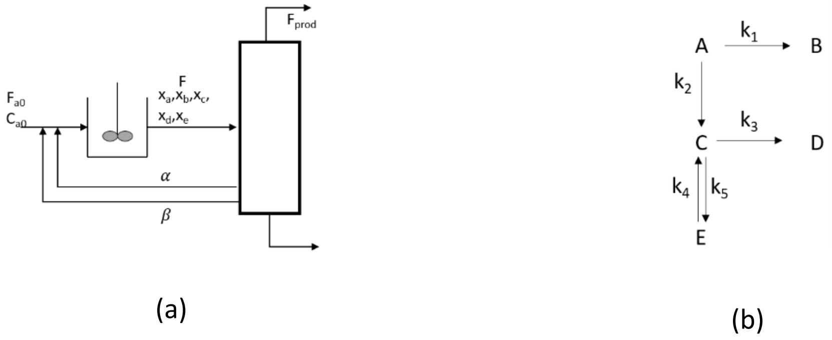
Figure 4-1: Summary of the proposed framework for design optimization

4.5 Illustrative Example

To better illustrate the idea, an example of a process consisting of a continuously stirred tank reactor in series with an ideal separator is chosen. It is assumed that the feasibility of the ideal separator depends only on the inlet flow rate range. The aim is to convert raw material A into two finished products B and E as shown in Figure 4-2. An isothermal liquid-phase reaction is considered following the kinetic mechanism as described in the previous studies of Rooney and Biegler [198] and Goyal and Ierapetritou [169]. The model equations for the process are shown by Eq. (69).

$$\begin{gathered}
F_{A0} - x_A F(1-\alpha) - VC_{A0}(k_1 + k_2)x_A = 0 \\
-Fx_B(1-\alpha) + VC_{A0}k_1 x_A = 0 \\
-Fx_C + VC_{A0}(k_2 x_A - (k_3 + k_4)x_C + k_5 x_E) = 0 \\
-Fx_D(1-\beta) + VC_{A0}k_3 x_C = 0 \\
-Fx_E(1-\beta) + VC_{A0}(k_4 x_C - k_5 x_D) = 0 \\
x_A + x_B + x_C + x_D + x_E - 1 = 0
\end{gathered} \tag{69}$$

where, x_A, x_B, x_C, x_D, and x_E represent the mole fraction of components A, B, C, D, and E, respectively; k_i are the rate constants; V is the volume of the reactor; C_{A0} is the inlet concentration of A; α is the recycle fraction of stream A and B; β is the recycle fraction of D and E; F is the molar flow rate at the outlet of the reactor; F_{A0} is the molar flow rate at the inlet of the reactor. The nominal values of the kinetic constants are $k_1 = 0.0374$, $k_2 = 0.0195$, $k_3 = 0.0165$, $k_4 = 0.2701$, and $k_5 = 0.0261$.



(a)

(b)

Figure 4-2: Reactor separator illustrative example (a) process flow diagram (b) reaction mechanism

The overall annualized cost of the process consists of the capital cost of the reactor as well as the capital cost of the separator. It is assumed that four reactor design options are available based on their volume. Different separator design options are based on the inlet flow rate. The available options and the respective costs for the reactor and the separator are shown in Table 4-1.

Table 4-1: Design options for reactor and separator

Options	**Reactor (m^3)**	$\boldsymbol{C_r}$ **(k\$)**	**Separator** **(F_{A0}mol/h)**	$\boldsymbol{C_s}$ **(k\$)**
Option 1	20	400	40-60	100
Option 2	25	550	50-70	150
Option 3	30	700	60-80	200

Option 4	35	850	80-100	250

Three scenarios for the nominal demand are considered in this work as shown in Table 4-2. The deviations from the nominal demand are 30 mol/h for the product B and 13 mol/h for the product E.

Table 4-2: Three scenarios based on the nominal demands for the products B and E

Scenario	**Nominal Demand B**	**Nominal Demand E**
	(mol/h)	**(mol/h)**
Scenario A	50	35
Scenario B	55	30
Scenario C	45	35

There are three problems considered in this work. In the first problem, the aim is to identify a reactor design and a separator design that can satisfy the known product demand and minimizes the annualized total cost. The second problem differs from the first problem in the objective which is to maximize the flexibility index for the second problem. Finally, both objectives are simultaneously considered in the third problem. A design is considered feasible if certain nominal product demand is satisfied by the design. The flexibility of a design quantifies the deviations in the product demand from the nominal demand for which the chosen design is feasible. The design optimization problem intends to find a set of optimal module designs based on the respective objectives of the three problems. Since the first four steps of the framework presented in Figure 4-1 are the same for all three problems, the steps of the framework are explained next.

Step 1: The first step is to collect the historic feasibility data or data from process simulation. In this example, the data is obtained by running the simulation of the reactor, as shown in Eq. (69). The simulation is developed in GAMS 28.2.0 and solved as a nonlinear program using Baron global optimization solver version 19.7.13. Inputs for the simulation includes four variables that are reactor volume, F_{A0}, F_B, and F_E. The output includes labels (-1 for infeasible and 1 for feasible) that indicate if the set of inputs leads to a feasible process. For each option of the reactor a grid-based sampling approach is used and 1000 data points (10 samples in each F_{A0}, F_B, F_E) are generated, and the output labels are collected.

Step 2: In this step, we train a classifier for each option using the data generated in step 1. In this work, SVM is the chosen classifier and SVM models are trained for the reactor as described in section 4.2.3. Scikit-learn python toolbox is used with default options for training the SVM models. Please note that since the separator is ideal and its feasibility depends only on the flow rate, there is no need to build a classifier for the separator.

Step 3: In this step, the model quality is assessed using the test dataset.

Table 4-3: SVM model validation for the reactor using RBF kernel

Option	**CF%**	**CIF%**	**NC%**	**Total Error**
Option 1	100	98.27	0.7	0.5
Option 2	100	93.94	1.18	1
Option 3	99.45	94.44	0.55	1
Option 4	98.89	95	0.56	1.5

Table 4-4: Confusion matrix for the reactor using RBF kernel

Option	**tn**	**fp**	**fn**	**tp**

Option 1	57	1	0	142
Option 2	31	2	0	167
Option 3	17	1	1	181
Option 4	19	1	2	178

The SVM performance results in Table 4-3 show that all SVM models have CF% and CIF% greater than 90%, and NC% and Total Error are less than 5%. This indicates that the models have acceptable quality for prediction of the feasible region, and we can move to step 4. In the machine learning literature, confusion matrix is a more commonly used metric that quantifies the number of true negatives (tn), false positives (fp), false negatives (fn), and true positives (tp). These metrics are shown in Table 4-4, Table 4-6, and Table 4-8. It is highlighted that the metrics shown in section 2.7.2 quantify the quality of classifiers better in the context of feasibility analysis, where metrics such as NC% are useful.

Table 4-5: SVM model validation for the reactor using a linear kernel

Option	**CF%**	**CIF%**	**NC%**	**Total Error**
Option 1	98.59	89.65	4.11	4
Option 2	100	100	0	0
Option 3	100	100	0	0
Option 4	100	100	0	0

Table 4-6: Confusion matrix for the reactor using a linear kernel

Option	**tn**	**fp**	**fn**	**tp**
Option 1	52	6	2	140
Option 2	33	0	0	167
Option 3	18	0	0	182

Option 4	20	0	0	180

Table 4-7: SVM model validation for the reactor using a sigmoid kernel

Option	CF%	CIF%	NC%	Total Error
Option 1	94.36	77.59	8.84	10.5
Option 2	96.41	60.61	7.47	9.5
Option 3	95.42	52	6.70	10
Option 4	95.55	45	6.01	9.5

Table 4-8: Confusion matrix for the reactor using a sigmoid kernel

Option	tn	fp	fn	tp
Option 1	45	13	8	134
Option 2	20	13	6	161
Option 3	13	12	8	167
Option 4	9	11	8	172

For understanding the effect of classifier parameters such as the choice of kernel functions, we train SVM models using linear and sigmoid kernel functions. It can be seen from Table 4-5 and Table 4-7 that SVM model with the sigmoid kernel function demonstrates the worst performance for all the options. Moreover, as seen from Table 4-4 and Table 4-6, SVM models using linear kernel function show a comparable performance for options 2, 3, and 4. Whereas, for option 1, RBF kernel function shows superior performance. Because of this performance and because of the general practice that RBF kernels are better at classifying nonlinear data, this work implements RBF kernel in solving the optimization problem using steps 4 and 5. It should be noted that if the models in this step do not meet the desired quality, the model performance should be improved by going back to step 1, choosing different SVM choices such as the choice of kernel function, choice of the penalty for misclassification, or choosing a different classification technique such as ANN or decision trees.

Step 4: In this step, we obtain an algebraic equation for the SVM model. This is done by obtaining the intercept and support vectors from the trained classifier from step 3. Since this is the only information required for Eq. (65), algebraic expression for the classifier can be obtained.

Step 5: This is the final step of the framework where the classifier is incorporated into the optimization problem.

The problem formulation for cost minimization is shown by Eq.(70)

$$
\begin{gathered}
\min C_r^T y_r + C_s^T y_s \\
\text{s.t.}\, SVM_r(\boldsymbol{x}) \geq -M_1(1-y_r) \forall r \in R; \forall k \in K \\
V - V_r \leq M(1-y_r)\ \forall r \in R \\
V_r - V \leq M(1-y_r)\ \forall r \in R \\
F_{A0} - ub_s \leq M(1-y_s) \forall s \in S \\
lb_s - F_{A0} \leq M(1-y_s) \forall s \in S \\
\sum_{r \in R} y_r = 1 \\
\sum_{s \in S} y_s = 1
\end{gathered}
\tag{70}
$$

where, C_r and C_s are the vectors of the cost coefficients for the reactor and the separator, respectively; y_r and y_s are the binary variables for the selection of reactor r and the separator s respectively; $SVM_r(.)$ represents the SVM model for the option r for the reactor; M_1 and M are the big constants; V_r is the volume of the reactor option r; R and S are the sets of all reactor and separator options respectively; lb_s and ub_s are the upper and lower bounds on F_{A0} for the option s of the separator.

For the problem of minimizing the total cost, 3600 scenarios are considered by varying the product demands. The MINLP problem is solved using in GAMS 28.2.0 and solved using Baron version 19.7.13 with the time limit of 1000 seconds on a PC with Intel® Xeon® CPU E-2174G @ 3.80GHz and 32.0 GB RAM, running a Windows 10 Enterprise, 64-bit operating system. The results are shown in Figure 4-3. Since the problem only considers capital cost, the total cost for an option is the same for all demand scenarios. The total costs, therefore, are shown in Table 4-9. Each data point in Figure 4-3 corresponds to a demand scenario for product B and the product E. The optimal choice of the option is abbreviated using the following convention. If the first option for the reactor is chosen and the first option for the separator is chosen, the optimal choice will represent option 11. The region shaded in the red color corresponds to all scenarios for which the option 11 was optimal. Similarly, the region shaded in the blue color represents the demand scenarios for which option 24 was optimal. Since the problem is solved to optimality, it can be concluded that the options that are not chosen are either infeasible or more cost-intensive than the ones selected by the optimization framework. The results shown in Figure 4-3 are as expected since the framework chooses option 11 for the lower demands, and as the demand increases, it chooses more expensive options. From Table 4-9, it can be observed that the cost for the option 11 is the least and that of option 24 is the highest. The results demonstrate the proposed formulation favors the cheapest feasible option.

Table 4-9: Total costs for the optimal options for minimizing the total cost for the reactor separator system

Option	11	12	13	14	23	24
Cost (k$)	500	550	600	650	750	800

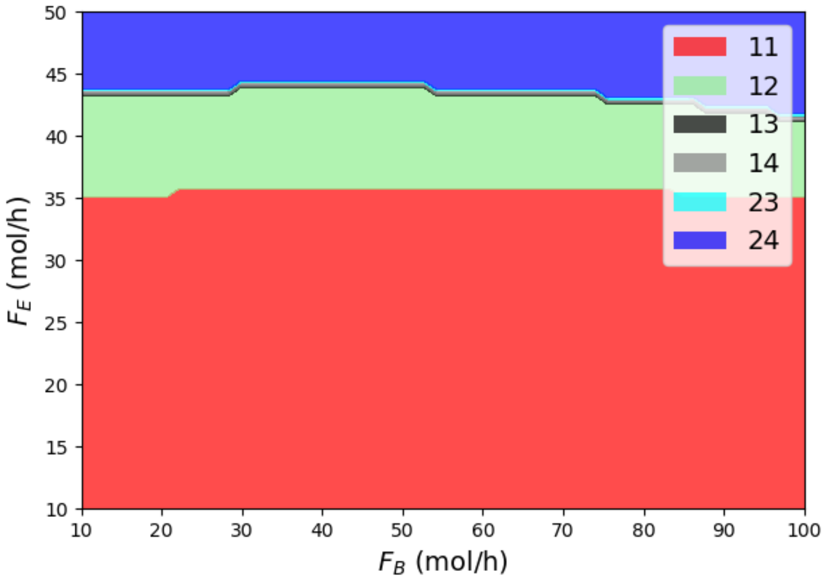


Figure 4-3: Feasible regions of the optimal design choices proposed by the framework for minimizing the total cost of the reactor separator system

The second problem we consider aims to find a design that has maximum flexibility given the nominal demand and the deviation from the nominal demand. The problem formulation for maximizing the flexibility index is shown by Eq. (71).

$$\begin{gathered}\max \delta \\ s.t.\ SVM_r\left(\boldsymbol{x} + \delta\Delta\boldsymbol{\Theta}^{\mathbf{k}}\right) \geq -M_1(1 - y_r)\forall r \in R; \forall k \in K \\ V - V_r \leq M(1 - y_r)\ \forall r \in R \\ V_r - V \leq M(1 - y_r)\ \forall r \in R \\ F_{A0} - ub_s \leq M(1 - y_s)\forall s \in S \\ lb_s - F_{A0} \leq M(1 - y_s)\forall s \in S \\ \sum_{r \in R} y_r = 1\end{gathered} \tag{71}$$

$$\sum_{s \in S} y_s = 1$$

$$\boldsymbol{x} + \delta \Delta \boldsymbol{\Theta}^{\mathbf{k}} \geq \mathbf{lb} \ \forall \mathrm{k} \in \mathrm{K}$$

$$\boldsymbol{x} + \delta \Delta \boldsymbol{\Theta}^{\mathbf{k}} \leq \mathbf{ub} \ \forall \mathrm{k} \in \mathrm{K}$$

where δ is the flexibility index; Θ^{k} is the vector of deviation of product demand from the nominal values in the direction of vertex k of the inscribed hyper-rectangle; K is the set of all vertices of the inscribed hyper-rectangle for the feasibility; $\boldsymbol{x}$ is a vector containing variables V, F_{A0}, F_B, and F_E where F_B and F_E are flow rates of the products B and E respectively; $\mathbf{lb}$ and $\mathbf{ub}$ are the vectors of lower and upper bounds for the variables in $\boldsymbol{x}$; All the remaining variables follow the same convention as that of the previous problem for cost minimization.

Three scenarios for the nominal demand as shown in Table 4-2 are considered for addressing the problem of maximum flexibility. The MINLP problem is solved using GAMS/Baron with a time limit of 1000 seconds. For each demand scenario, the optimal choice proposed by the framework is shown along with the flexibility index obtained for the optimal choice are displayed in Table 4-10.

Table 4-10: Results for maximizing flexibility for three demand scenarios

Scenario	**Demand B**	**Demand E**	**Flexibility Index**	**Optimal Choice**
	(mol/h)	**(mol/h)**		
Scenario A	50	35	1.15	44
Scenario B	55	30	1.5	44
Scenario C	45	35	1.15	44

It can be observed from Table 4-10 that for all the scenarios, the framework finds a solution that has a flexibility index greater than one, indicating that the optimal design is able to satisfy the expected deviations from the nominal demand. It is important to note that when there are multiple designs with the same flexibility index, this formulation for maximizing the flexibility does not use any other criteria to prefer one design option over the other design options. To demonstrate this, the maximum flexibility indices are shown in Figure 4-4 for all the feasible options. Two important observations can be derived from Figure 4-4. First, it can be observed that option 24, 34, 44, and 43 all have the same flexibility index that is greater than one. Moreover, option 33 also has a flexibility index greater than one. Second, option 12 is the cheapest feasible option and it has the least value of the flexibility index. These two observations motivate the need for a framework to address these two objectives together.

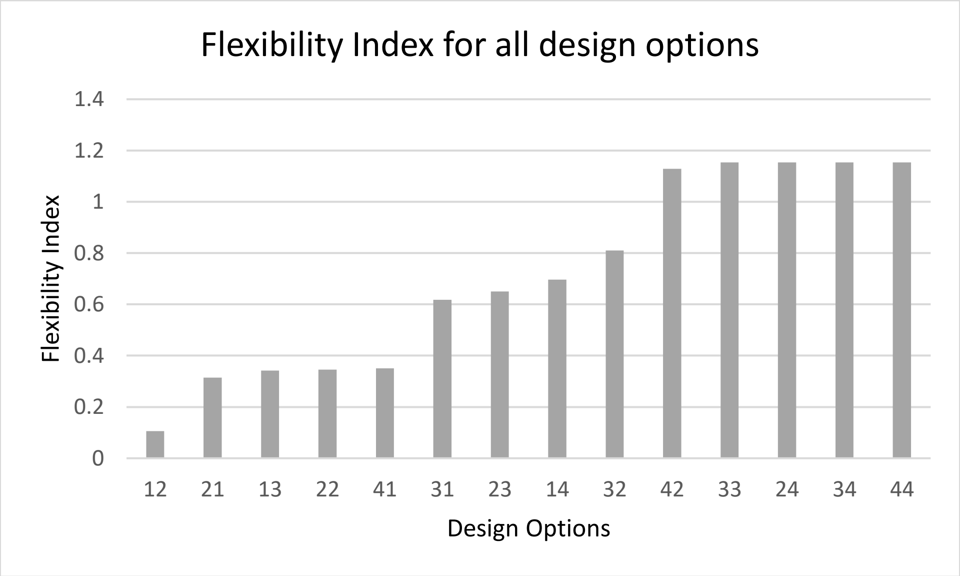


Figure 4-4: Flexibility Indices for all design options for demand scenario A

This trade-off can be better visualized using Figure 4-5, where both the objectives are plotted against each other for each design option. Since we want to minimize the total cost and maximize the flexibility index, options 12, 13, 14, and 24 are the Pareto optimal designs.

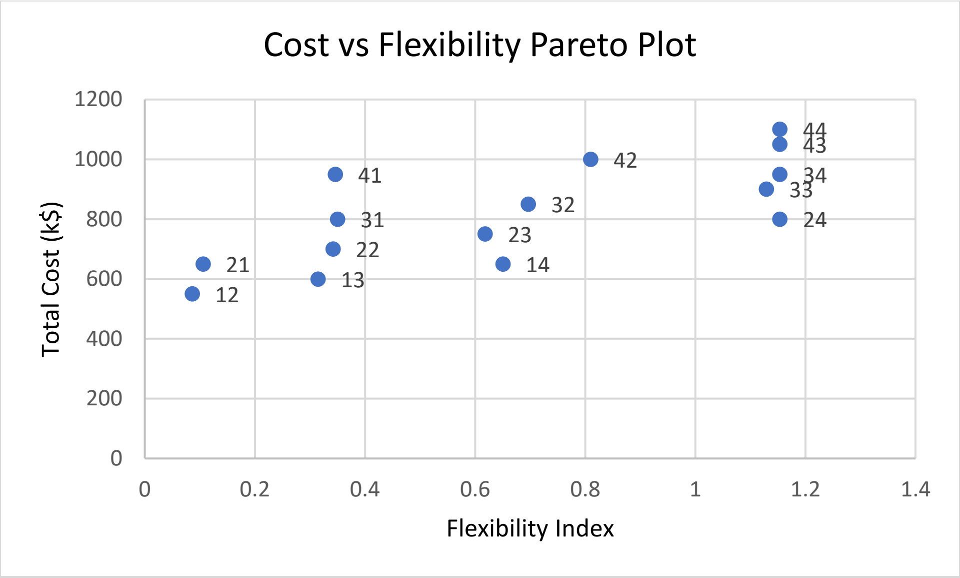


Figure 4-5: Cost vs Flexibility Pareto plot for scenario A

The third problem is a multiobjective problem where the problem of minimizing the total cost as well as maximizing the flexibility is addressed. The problem formulation for multiobjective optimization is shown by Eqs. (72).

$$\min C_r^T y_r + C_s^T y_s \tag{72a}$$

$$\mathrm{s.t.}\ \ \delta \geq \epsilon \tag{24b}$$

$$SVM_r\left(\boldsymbol{x} + \delta\Delta\boldsymbol{\Theta}^{\mathbf{k}}\right) \geq -M_1(1 - y_r)\forall r \in R; \forall k \in K \tag{24c}$$

$$x_1 - V_r \leq M(1 - y_r)\ \forall r \in R \tag{24d}$$

$$V_r - x_1 \leq M(1 - y_r)\ \forall r \in R \tag{24e}$$

$$x_2 - ub_s \leq M(1 - y_s)\forall s \in S \tag{24f}$$

$$lb_s - x_2 \leq M(1 - y_s)\forall s \in S \tag{24g}$$

$$\sum_{r \in R} y_r = 1 \tag{24h}$$

$$\sum_{s \in S} y_s = 1 \tag{24i}$$

$$\boldsymbol{x} + \delta\Delta\mathbf{\Theta}^{\mathbf{k}} \geq \mathbf{lb}\ \forall \mathrm{k} \in \mathrm{K} \tag{24j}$$

$$\boldsymbol{x} + \delta\Delta\mathbf{\Theta}^{\mathbf{k}} \leq \mathbf{ub}\ \forall \mathrm{k} \in \mathrm{K} \tag{24k}$$

The ϵ-constraint method is used for multiobjective optimization where one of the objectives is expressed as a constraint, as shown by Eq. (24b). Additionally, if an option is selected for the reactor or separator, the SVM prediction should ensure feasibility. This is imposed with the help of constraints given by Eq. (24c). The solution obtained by solving the optimization problem should satisfy the operability ranges for the options selected for the reactor and separator. Since the option for the reactor is based on the volume, it is given by Eq. (24d) and Eq. (24e). The options for the separator are based on the flow rate ranges and it is ensured by the constraints given by Eq. (24f) and Eq. (24g)). Finally, only one option must be selected for the reactor as well as the separator, as given by Eq. (24h) and Eq. (24i) and the flexibility of the demand should be within the known bounds for the demand for products B and E as given by Eq. (24j) and Eq.(24k).

For the multiobjective optimization problem, the nominal demand of 50 mol/h for product B and 35 mol/h for the product E, as shown in scenario A in Table 4-10. Flexibility is modeled in the optimization problem with the help of ϵ-constraint method as shown by Eq. (24b). When the value of ϵ is 0, the framework leads to minimization of the cost. However, as the value of ϵ is increased, flexibility constraint becomes strict, and as a result, the framework selects options that are more flexible. The MINLP problem is solved using GAMS/Baron with a time limit of 1000 seconds. The results for the cost and the optimal design choice are shown in Table 4-11. It can be verified from Figure 4-5 that the optimal choices proposed by the multiobjective optimization framework correspond belong to the designs on the Pareto optimal curve.

Table 4-11: Optimal choices for the system of reactor and separator using multiobjective optimization

ϵ	**Cost (k$)**	**Reactor**	**Separator**
0	550	1	2
0.5	650	1	4
1	800	2	4
1.1	800	2	4
1.15	800	2	4

4.6 Air separation unit case study

4.6.1 Process and model description

In this section, the problem of process design optimization and flexibility evaluation of an air separation unit (ASU) is considered. The process shown in Figure 4-6 consists of separating the feed air into oxygen, nitrogen, and argon [201]. This is achieved through liquefaction with the help of products and waste streams in the main heat exchanger. Part of the compressed air stream is withdrawn from the heat exchanger at an intermediate location and passed through the turbine, followed by a low-pressure distillation column. The rest of the compressed air traverses the entire length of the heat exchanger and is fed to the high-pressure column. The columns operate at a pressure that enables heat integration with a common system of a reboiler and a condenser. The ASU model used in this work is based on the previous work of Dias et al. [195] and Sirdeshpande et al. [202]. The reader is referred to previous works in the literature for more details on the simulation model [203][204][205][206][207]. More recently, Caspari et al. [208] worked on the design and optimization of a flexible ASU as well as on optimal operation using economic model predictive control [209]. The models considered in this work are steady-state models, and a standalone simulation is developed for each unit operation. A brief description of the mathematical models describing the behavior of unit operations of the ASU process is provided below.

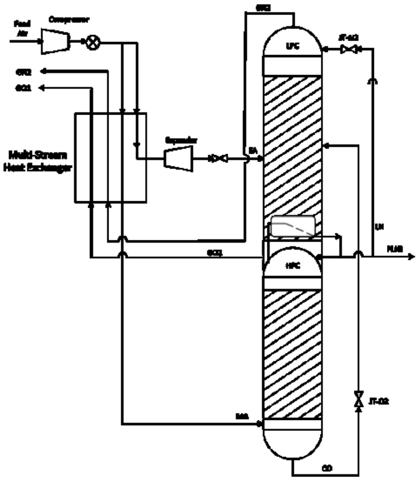


Figure 4-6: Air separation unit case study

Distillation column model. A double column system is considered for the distillation of which high pressure column (HPC) operates at a pressure of 6.5 bar whereas the low pressure column (LPC) operates at a pressure of 1.5 bar. The compressed and cooled air streams are passed to the distillation columns. These columns share a common condenser/reboiler system, and the operating pressures of the columns are chosen accordingly. The main air (MA) enters the bottom of the HPC as a saturated vapor whereas the turbine air is expanded to the pressure of the LPC and is fed to the LPC.

Integrated reboiler/condenser model: The steady-state model for the integrated reboiler/ condenser is adapted from the work of Dias and Ierapetritou [195]. This model consists of two parts, one of which is for the condenser side and one for the reboiler side. The oxygen-rich stream at the bottom of the distillation column is expanded through a valve to 2.5 bar to provide cooling (via Joule-Thomson effect) to the condenser. The reboiler is modeled as an additional equilibrium stage. The equivalent of condenser heat duty is added as an additional input to the energy balance for the reboiler.

Heat exchanger model. The air streams MA and EA are cooled in a brazed plate-fin multistream heat exchanger. The main air traverses the entire length of the heat exchanger, whereas the TA is partially cooled and withdrawn at an intermediate location. Based on the location at which TA is withdrawn, the heat exchanger model is divided into two zones. The remainder of the air, which is MA, is taken out at the outlet of zone 2. As a result, zone 1 corresponds to sensible heat removal from the air stream, whereas zone 2 corresponds to the latent heat removal. The first zone is discretized into 50 segments, while the energy balance is carried out over the second zone as a single unit to simplify the phase transformation calculations. The geometry of the channels within each segment is accounted for when calculating the energy accumulation of each stream in each finite volume.

Compressor/turbine model. Feed air is compressed to a pressure of 6-7 bar using the compressor followed by which the air is split and passed to the heat exchanger. Part of the air that is withdrawn at an intermediate location from the heat exchanger is expanded in the turbine expander to produce a cold exhaust and mechanical work.

4.6.2 Process modularization

Based on the unit operations mentioned, this process may be modularized into the four basic operations of heat exchange, expansion, distillation, and compression [202]. Given different module options for each equipment, varying in size and process specification (e.g., number of columns, pressure, etc.), the goal is to define a set of options that can achieve product specifications at a minimum investment cost while ensuring that the operation remains flexible and the desired demand is met. The options for the heat exchanger and the distillation column are based on the same parameters. The set of available options for each module, together with the capital cost of distillation and heat exchanger options, are shown in Table 4-12 and Table 4-13, respectively. The options for compressor and turbine modules depend on the input flow rates, as shown in Table 4-14 and Table 4-15, respectively. It is important to note that the costs for the compressor module and the turbine module includes the compressor and turbine as well as their respective auxiliary units. From Table 4-12, Table 4-13, Table 4-14, and Table 4-15, it can be observed that a convention is followed that the lower option number represents a smaller unit and lower capital cost. It is important to note that, however, a smaller unit need not have a lower capital cost because of several process intensification strategies [210][15][211]. The proposed framework for multiobjective optimization can readily handle such an alternative in the design optimization stage.

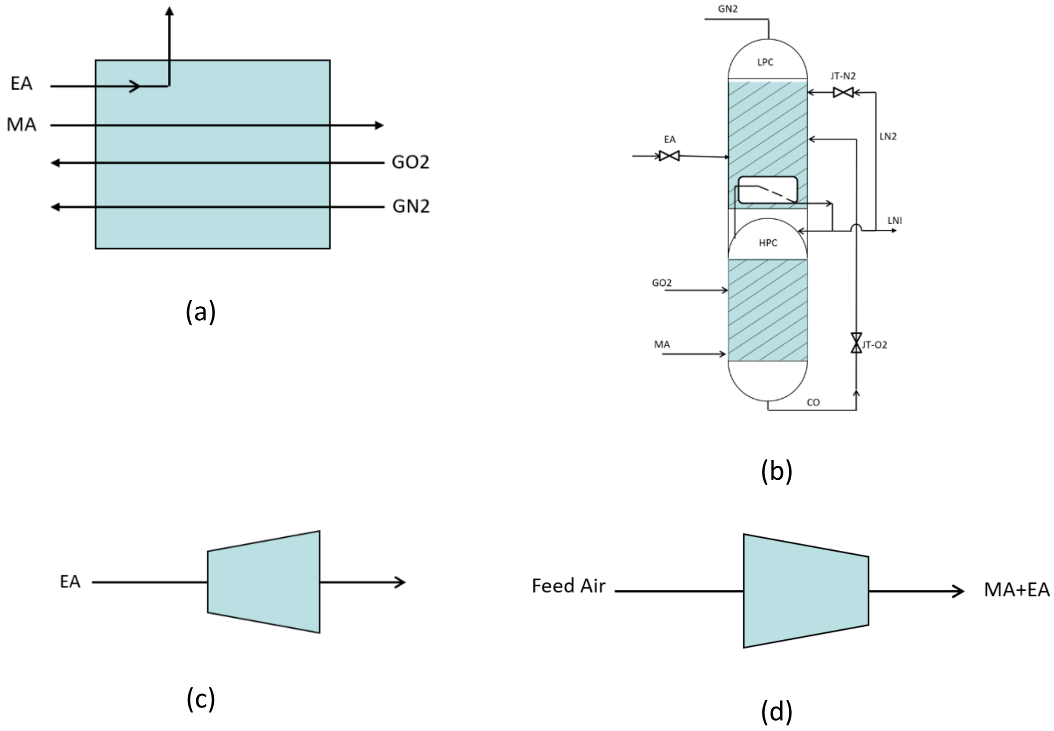


Figure 4-7: ASU modules (a) heat exchanger (b) distillation column (c) compressor (d) turbine

Table 4-12: Distillation module options and the capital cost

Options	**MA** **(mol/s)**	**EA** **(mol/s)**	**NI** **(mol/s)**	**OX** **(mol/s)**	**Capital** **cost(k$)**
Option 1	30-40	0-6	30-45	8-12.2	2382
Option 2	35-45	3-9	34-50	9-13.5	3119
Option 3	40-50	6-12	38-50	10-13.5	3388
Option 4	45-60	10-14	30-50	8-13.5	3768

Table 4-13: Heat exchanger module options and the capital cost

Options	**MA** **(mol/s)**	**EA** **(mol/s)**	**NI** **(mol/s)**	**OX** **(mol/s)**	**Capital** **cost(k$)**
Option 1	30-40	0-6	30-45	8-12.2	1966
Option 2	35-45	3-9	34-50	9-13.5	2164
Option 3	40-50	6-12	38-50	10-13.5	2351
Option 4	45-60	10-14	30-50	8-13.5	2615

Table 4-14: Compressor options

Options	**Compressor options - Air flow rate** **(mol/s)**	**Capital cost** **(k$)**
Option 1	30-46	667
Option 2	38-54	734
Option 3	46-62	798
Option 4	51-74	887

Table 4-15: Turbine options

Options	**turbine options - Air flow rate (mol/s)**	**Capital cost(k$)**
Option 1	0-6	1966
Option 2	3-9	2164
Option 3	6-12	2351
Option 4	10-14	2614

4.6.3 Application of the optimization framework

For the compressor and turbine, it is assumed that given the flow rates in the operable range, sufficient head is produced. As this only requires the operable range and the flow rate, the SVM model need not be built for the compressor and the turbine options. The efficiency and power consumption due to these modules is included in the calculation of the total cost. Moreover, some of the connecting variables, such as the composition of the oxygen and nitrogen streams, are omitted in the analysis because of numerical issues due to a narrow feasible region. Ranges for these variables are provided instead to ensure that feasibility information is preserved. Finally, this work does not consider the storage of the products, and the feasibility is defined accordingly. This means that a bigger design for a particular module may not necessarily be feasible for smaller product demands.

To achieve the goal of choosing an optimal design that minimizes the total cost as well as provides maximum flexibility with respect to the demand variability, the first step is to understand the feasibility of each option related to each module. For this, historical data for the feasibility of each module is obtained. Based on the process considerations, the inputs or features for the classifier include two types of variables. The first type contains the variables inherent to the model, and the other type includes variables that connect two modules. By defining the connecting variables as the inputs to the classifier ensures that the mass and energy balances are implicitly handled. As an example, for the distillation module, the temperature and flow rates of inlet streams and the required purity are inherent variables, whereas the flow rates and temperatures of the outlet streams are connecting variables. Since the same connecting variables will also be included in the classifier for the heat exchanger and because both classifiers are used as constraints to define the feasibility of the process, the framework forces connecting variables to have the same value. Having the same value of connecting variables for the heat exchanger ensures that the mass balance and the energy balance is satisfied. By handling mass and energy balances implicitly, the feasibility of all modules is sufficient to ensure the feasibility of the process. The input variables for the SVM models for the heat exchanger and distillation column are shown in Table 4-16.

Table 4-16: ASU modules feasibility analysis inputs

Module	**Input variables**
Heat exchanger	Flow rates of Nitrogen, Oxygen, Air, Temperature of turbine air
Distillation column	Flow rates of Nitrogen, Oxygen, temperature and flow rate of turbine air and main air, impurities of nitrogen and oxygen

For each option of the distillation column, 1000 samples are generated using the simulation, and the SVM model is built for 7 variables. For the heat exchanger options, 1000 samples are generated for each option, and the SVM model is built for 5 variables. The models are tested on the test data, and the validation results are provided in Table 4-17-Table 4-22, respectively. The tables show a comparison between different kernel functions for analyzing the feasibility of the heat exchanger and distillation column. From Table 4-17-Table 4-19, it can be observed that the SVM models built using a sigmoid kernel show the worst performance. From Table 4-17 and Table 4-18, SVM models built using the RBF kernel are shown to be superior for options 1 and 2 whereas, those built using linear kernel are shown to be better for options 3 and 4. For SVM models built using RBF kernel, it can be observed that all heat exchanger models have less than 5% values of NC% as well as the Total Error, whereas over 95% values for CF% and over 90% values for CIF% except for the option 1.

Table 4-17: Validation results for the heat exchanger SVM models using RBF kernel

Option	**CF%**	**CIF%**	**NC%**	**Total Error**
Option 1	99.52	89.85	3.27	2.89
Option 2	100	91.30	2.91	2.23
Option 3	100	92.31	4.37	2.87
Option 4	97.66	95.86	2.05	2.93

Table 4-18: Validation results for the heat exchanger SVM models using linear kernel

Option	**CF%**	**CIF%**	**NC%**	**Total Error**
Option 1	97.11	82.61	5.61	6.5
Option 2	99	86.95	4.35	4.09
Option 3	100	97.80	1.29	0.82
Option 4	97.67	98.22	0.89	2.14

Table 4-19: Validation results for the heat exchanger SVM models using sigmoid kernel

Option	**CF%**	**CIF%**	**NC%**	**Total Error**
Option 1	88.94	69.56	10.19	15.88
Option 2	95	73.91	8.65	10.41

Option 3	96.08	81.32	10.36	9.43
Option 4	90.96	82.84	8.50	11.71

A similar comparison is carried out between SVM models built using different kernel functions for the distillation module. For all the options, SVM models built using a sigmoid kernel function performed the worst as it can be observed from Table 4-20-Table 4-22. Those built using RBF kernel demonstrated the lowest CIF%, NC%, and Total Error. Moreover, CF% values are higher for the SVM models built using RBF kernels for options 1, 3, and 4. As a result, RBF models are chosen for representing the feasible region. In absolute terms, the SVM models built using RBF kernel for a distillation column, the CF% is above 95% for all options, CIF% is above 90% for all options, and NC% and Total Error values are less than 1%. As a result, it can be concluded that the models provide a good representation of the feasible region for all options for all modules.

Table 4-20: Validation results for the distillation SVM models using RBF kernel

Option	CF%	CIF%	NC%	Total Error
Option 1	97.39	99.51	1.97	0.92
Option 2	99.43	99.80	0.14	0.40
Option 3	99.23	98.48	0.39	0.92
Option 4	99.89	90.32	0.43	0.52

Table 4-21: Validation results for the distillation SVM models using linear kernel

Option	CF%	CIF%	NC%	Total Error
Option 1	99.02	98.76	4.72	1.18
Option 2	99.29	98.05	1.4	1.23
Option 3	99.32	95.83	10.56	1.38

Option 4	99.89	72.58	1.23	1.23

Table 4-22: Validation results for the distillation SVM models using sigmoid kernel

Option	CF%	CIF%	NC%	Total Error
Option 1	62.74	90.81	36.63	14.79
Option 2	93.35	88.11	8.46	8.85
Option 3	92.87	71.59	7.22	11.44
Option 4	97.55	18.55	3.64	5.86

All the models are built-in GAMS 28.2.0 and solved using Baron Version 19.7.13 on a PC with Intel® Xeon® CPU E-2174G @ 3.80GHz and 32.0 GB RAM, running a Windows 10 Enterprise, 64-bit operating system. All computations in this work are carried out with the same specifications. For building SVM models, scikit-learn python toolbox [77] is used with default settings, and the inputs are scaled to have zero mean and unit variance before building the SVM models.

One of the two objectives of this work is minimizing the annualized cost of the process. For this, we consider capital cost and the operating cost, as shown in the objective function of Eq. (73). The values for capital cost are obtained from the recent work of Tesch et al. [212] and scaled following the 0.6 power rule. Moreover, it is assumed that the annualized capital cost is 10 percent of the total capital cost. The operating cost is calculated based on the work of Sirdeshpande and Ierapetritou [202], and the electricity price used the average electricity price for the US energy information administration website [213]. Another objective is the maximization of the flexibility index. In this work, it is assumed that the nominal demand is known, and the variation $\Delta\Theta$ is centered on the nominal demand. The multiobjective formulation can then be written by Eq. (73)

$$\min \text{total cost} = \text{Capital cost} + \text{Operating cost} \tag{73}$$

$$\text{s.t. } \delta \geq \epsilon$$

$$SVM_d(\boldsymbol{x} + \delta\Delta\boldsymbol{\Theta}) \geq -M_1(1 - y_d)\forall d \in D$$

$$SVM_h(\boldsymbol{x} - \delta\Delta\boldsymbol{\Theta}) \leq -M_1(1 - y_h)\forall h \in H$$

$$x_d - ub_d \leq M(1 - y_d))\forall d \in D$$

$$lb_d - x_d \leq M(1 - y_d))\forall d \in D$$

$$x_h - ub_h \leq M(1 - y_h))\forall h \in H$$

$$lb_h - x_h \leq M(1 - y_h))\forall h \in H$$

$$x_c - ub_c \leq M(1 - y_c))\forall c \in C$$

$$lb_c - x_c \leq M(1 - y_c))\forall c \in C$$

$$x_t - ub_t \leq M(1 - y_t))\forall t \in T$$

$$lb_t - x_t \leq M(1 - y_t))\forall t \in T$$

$$\sum_{d \in D} y_d = 1$$

$$\sum_{h \in H} y_h = 1$$

$$\sum_{c \in C} y_c = 1$$

$$\sum_{t \in T} y_t = 1$$

$$\boldsymbol{x} - \delta\Delta\boldsymbol{\Theta} \geq \mathbf{lb}$$

$$\boldsymbol{x} + \delta\Delta\boldsymbol{\Theta} \leq \mathbf{ub}$$

where, the capital cost refers to the total annualized capital cost of the chosen options and operating cost depends on the quantity of products produced, air flow rate, and compressor and turbine efficiency among the other factors; y_d, y_h, y_c and y_t are the binary variables for the selection of the distillation column d, heat exchanger h, compressor c, and the turbine t; δ is the flexibility index; Θ is the scaling factor; $SVM_d(.)$ represents the SVM model for the option d for the distillation column; SVM_h represents the SVM model for the option h for the heat exchanger; M_1 and M are the big-M constants; $\boldsymbol{x}$ is the decision variable vector that contains flow rates of main air, turbine air, nitrogen and oxygen, the temperature of turbine air, and purity of Nitrogen and oxygen in that order; D, H, C and T are the sets of all distillation column, heat exchanger, compressor and turbine respectively; $\mathbf{ub}$ and $\mathbf{lb}$ are upper and lower bounds for the variable $\boldsymbol{x}$. The variable $\boldsymbol{x_i}$ represents continuous variables corresponding to module i.

In multiobjective optimization, both the objectives of cost and flexibility are considered where flexibility is modeled using the ϵ-constraint method. The nominal demand is 40 mol/s for nitrogen and 10 mol/s for oxygen. Results are obtained by varying ϵ from a value of 0 to 1. This range is obtained from the definition of flexibility index which is at least 0 for all the feasible designs, and the desired value of flexibility index is 1.0, where the design is flexible enough to address the expected variability in demand. The expected deviation is 5 mol/s and 1.6 mol/s, from nitrogen and oxygen flowrates, respectively. The problem is solved using GAMS/Baron with a time limit of 1000 seconds. This way, one can obtain designs on the Pareto optimal curve by varying the values of ϵ as shown in Table 4-23. For ϵ equal to 0, 0.2, and 0.5 the optimal design choice was option 2211 with a total annualized costs of \$815,370, \$827,159, and \$838,949, respectively. For values of ϵ greater than 0.5, the framework suggests the optimal design choice option 4433. The annualized costs when the values of ϵ are 0.7, 0.9, and 1.0 are \$983,731, \$985,120, and \$990,086, respectively. It can be observed that as ϵ increases, the cost increases, indicating that more flexible designs tend to be more cost intensive. The difference in costs for the same choice of optimal design is mainly due to the operating costs. Moreover, feasibility of the multiobjective optimization problem for a value of ϵ equal to 1 indicates that there are feasible designs in the available options that can satisfy the expected deviation in the product demand. For a problem where one expects that there are no designs available with flexibility index value of 1, the value of ϵ can be varied between 0 and the value obtained by maximizing the flexibility index. It should also be noted that even though the framework suggests only two optimal designs, there are other designs that are feasible as well. Since this work does not consider the option for holding the products in a storage tank, the only options considered by the framework are the options that are feasible for the nominal demand of 40 mol/s for nitrogen and 10 mol/s for oxygen.

Table 4-23: Optimal choices of modules for ASU using multiobjective optimization

ϵ	**cost (k$)**	**distillation**	**heat exchanger**	**compressor**	**turbine**
0	815.370	2	2	1	1
0.2	834.070	2	2	1	2
0.5	871.189	2	2	2	2
0.7	983.731	4	4	3	3
0.9	985.120	4	4	3	3
1.0	990.086	4	4	3	3

4.7 Conclusions

In this work, a framework for modular design under variability is presented. The framework consists of two steps: a data-driven feasibility analysis, followed by simultaneous design optimization and flexibility evaluation. The concept behind the framework is illustrated first on an example of a reactor and separator system. Further, the performance of the framework is demonstrated in the study of the modular design of ASUs. The problem of choosing between several module options is addressed for the objectives of choosing a design that minimizes cost as well as has the maximum flexibility with respect to the product demand. A data-driven approach is used with the help of support vector machines to represent the feasible region for each module option. In doing so, a quantitative comparison is provided for choosing the kernel function for the SVM model.

The approach presented in this work has several advantages over the methods existing in the current literature. Since the framework does not rely on analytical equations of the process, historical data can be directly used for the analysis. In the absence of analytical equations for the process, the problem of handling discrete choices is known to be extremely difficult. An intuitive approach to building a data-driven model for each combination of discrete choice requires a huge number of data-driven models, which leads to a computationally intractable optimization problem. The proposed framework takes advantage of the modularity of the process, requiring a significantly low number of data-driven models for obtaining an optimal design. In doing so, it maintains the feasibility of the process as a whole by implicitly handling inherent constraints such as the mass and energy balances. In the optimization framework, two different objectives are simultaneously achieved to obtain a set of Pareto optimal designs leading to a more informed decision. For analyzing the feasibility of a design, the examples showed in this work use SVM. However, due to the generic nature of the framework, other classification techniques can be used as well. Ease access to software that supports such techniques makes this framework an attractive choice for industrial use. Moreover, the design optimization considered in this work can be generalized to process synthesis problems where different technologies or even different connections in a flow sheet can be compared. This approach opens new and exciting possibilities for process design in many other chemical engineering processes, considering comprehensive objectives such as the environmental impact and in assessing the applicability of modular design with a more detailed cost estimate in a broader supply chain context.

5 Supply chain optimization with modular processing

Abstract

Modular manufacturing is gaining popularity in a variety of industrial applications due to potential cost-savings as well as process flexibility that can be achieved with the use of small and standardized modules. In this work, a framework for supply chain optimization is proposed that ensures production feasibility with the help of historical process data for individual process modules and machine learning-based feasibility analysis. A supply chain optimization problem is formulated where the binary variables represent facility locations, and the integer variables correspond to the number of modular equipment installed. Results demonstrate that the tradeoff between centralized and distributed manufacturing and the effect of economies of numbers on the cost of the supply chain can be studied by solving the problem.

5.1 Introduction

Globalization and increasing market competition have been a significant impetus for changing the pace and nature of businesses and innovation around the world. More and more customer-orientated products are driving change in many industries, and product cycles are becoming shorter [5]. Modularization, process intensification, and design standardization are increasingly being recognized as critical factors to reduce the time to market for a product [6]. With a wide range of applications in the areas of gas conversion, solid conversion ammonia synthesis, CO_2 conversion, water purification, renewable energy, power generation, and chemical processing along with growing industrial interest, modular manufacturing provides a promising way forward for the process engineering [7][8].

Recent studies on modular and distributed manufacturing have challenged the traditional cost reduction paradigm relying on the economy of scale[11]. The traditionally used 2/3rd power law indicates that the capital cost increases with plant capacity following a 2/3rd power law. This leads to the conclusion that having large plants is more cost efficient. Modular design involves the use of small and standardized modules of fixed size in a production plant that can provide several strategic, manufacturing, as well as economic advantages. Modular designs reduce time-to-market for a product because of preassembled modules, lower construction times due to standardization, and the possibility of numbering up identical devices to achieve the desired production. Shorter time-to-market, combined with the safety of operation[9], reduces the financial risk for the investors[152]. Moreover, centralized large-scale manufacturing, often means high transportation costs for the delivery of raw materials as well as the products whereas small scale, modular manufacturing provides a way to distribute manufacturing effectively to reduce transportation costs. Finally, with repeated production of standardized units, engineers as well as the vendors, gain experience which results in lowering the capital cost and improving production lead time.

However, quantifying the economic viability of such modular processes is a relatively underexplored problem. Recent work on modular design focuses on creating quantitative measures for understanding the benefits of modular manufacturing. Shao and Zavala model a manufacturing system as a graph composed of nodes as equipment units and edges and propose a measure for modularity that considers connectivity, number of modules, and dimensions that can be computed by solving a mixed integer quadratic program[214]. One crucial advantage of standardized units is the cost-savings achieved from mass production, which is studied under the topics of the economy of numbers, the economy of mass production, and experiential learning economics. Arora et al. [17] study the economy of numbers and equipment standardization for capital cost reduction. Patience and Boffito [215] note that the effect of the experience curve should be accounted for while conducting economic analysis. They investigate the effect of scaling down and numbering-up on the cost. Leiberman [216] discusses learning-based cost advantages in processing plants using experiential learning economics.

In addition to analyzing the effect of modular units on the capital cost, it is essential to analyze their overall effect on the cost of the supply chain. Palys et al. [14] study the effect of mass production exponent for a supply chain consisting of modular units. Allman and Zhang propose a framework for determining optimal location and relocation of mobile production modules under time-varying demands [217]. They propose a metric to assess the economic benefits of module mobility. They refer to this problem as dynamic modular and mobile facility location problem. Becker et al. [218] propose a mixed integer linear programming model for the tactical planning of modular production networks with consideration of volume, location, and process flexibility. Lier et al. [219] review transformable production concepts with modular composition. They review challenges on the equipment level, network level, and logistics level. Palys et al. [14] address the problem of a supply chain network consisting of renewable-powered ammonia production where the renewable production units are modular, and the number of modules installed at each production facility needs to be determined. They compare the cost of traditional centralized production with a modular design by varying the mass production exponent that describes the rate of experiential learning. Allen et al. [16] consider the problem of shale gas production planning where, as the resources deplete, there may not be enough throughput for a large plant to operate. In such a case, it is beneficial to have modular designs that can handle lower throughputs. These modules can be moved to a new site when the need arises. They solve a multistage stochastic programming problem. Yang and You [153] provide a model-based study for understanding the effects of plant relocation, centralized manufacturing, and distributed manufacturing for methanol manufacturing. They consider the environmental impact and analyze shale gas case study as well as a methanol production example.

Another common problem in modular and distributed manufacturing at the supply chain level is the problem of facility location. Lara et al. [220] propose an algorithm for finding optimal locations for the distribution centers for centralized and distributed manufacturing networks. While doing this, they also solve the problem of choosing a centralized network of distribution centers or a distributed network. Bowling et al. [221] propose a superstructure-based optimization problem for deciding the facility location for the biomass problem. Their problem is to determine flow rates through each link given a superstructure for maximizing profit. A location is selected if the flow rate is nonzero and not selected otherwise. They approximate the exponential behavior of the capital cost (2/3rd law) using piecewise linear approximation and model the capital cost of using disjunctions where each disjunction approximates each piece of linear approximation. Elia and Floudas [222] provide a comprehensive study of facility location and life cycle analysis of biomass. They consider sources of biomass in the United States and find optimal facility location by discretizing states into octants. Using a modular production, they choose from three different processing capacities for each facility and allow for multiple facilities at each site. Tan and Barton [223] note that mobile, modular plants are possible attractive routes for gas monetization and propose a multi-period optimization framework for solving dynamic plant allocation problem. Bramsiepe et al. [224] discuss distributed manufacturing for biomass processing. They note that biomass processing should be divided in small-scale water separation followed by large-scale processing. They review various economic as well as practical aspects of small-scale manufacturing. It is important to note that, most of the aforementioned work treats the whole process as modular and does not incorporate production level details at the supply chain level by making a usual assumption that the capacity of a modular process is known and fixed.

The key details at the production level relate to defining a process based on a limited number of different modules [155]. In this work, modular designs involve the design and construction of smaller process units or processes of fixed production capacities [14]. This definition allows for possible extensions for process intensification [15], mobile processing units [16], standardized equipment [17], and customized unit operations [8]. In the previous work[225], it has been shown that modular designs have an added advantage of providing a quantitative estimate of the flexibility of process design. One reason for flexibility is the increased degrees of freedom since individual module can be designed instead of designing the process as a whole. The flexibility of a process design can be quantified by obtaining an algebraic approximation of the process feasibility. The approach presented in our previous work uses historical information related to the feasible region of operation for each module in a data-driven feasibility analysis. Process feasibility can then be defined using the feasibility of individual modules. At the supply chain design level, the utilization of modules provides great benefits including the choice of the production facility location, deciding between a centralized or distributed production, quantifying the benefits of design standardization, and assessing flexibility of the supply chain in the presence of demand variability.

In this work, supply chain network design problem is addressed using modular production to determine the optimal process design, and the facility location to minimize the total cost of the supply chain over the desired planning horizon. The economies of numbers are modeled by scaling the capital cost of all installed modules with a coefficient of mass production that reduces the cost per-module as more modules are installed. The approach provided in this work provides a way to integrate production level details in the supply chain optimization problem. The formulation presented in this work uses support vector machine (SVM) models trained using historical process data as an approximation of the feasible region for each process module. The resulting problem is a mixed integer nonlinear programming (MINLP) problem where binary variables represent facility locations, and integer variables represent the number of modular equipment installed. Since the module options vary in cost and the processing capacity, the tradeoff between centralized and distributed manufacturing is implicitly addressed by solving the problem to optimality. It is noted that the coefficient of mass production may vary based on the specific process and the modules under consideration. The results demonstrate the effect of the coefficient of mass production on the optimal network design.

This chapter is organized as follows. Section 5.2 provides the details of the supply chain optimization formulation. Section 5.3 covers the background information on the methods used in the proposed work. Section 5.4 presents the application of the proposed approach in two case studies, whereas section 5.5 provides some concluding remarks.

5.2 Supply chain model

The supply chain network considered in this work is generic and is motivated from our previous work [188]. The network consists of suppliers, production facilities, warehouses, and retailers. The demand is realized at each retailer in each time period. The product or raw material flow is allowed using all possible pathways i.e. all suppliers can send material to all production facilities, all production facilities can send material to all warehouses, and all warehouses can send material to all retailers. It is important to note that the production details are taken into account, and the production is carried out using modular processes. This results in a key difference between traditional supply chain optimization formulations and the formulation presented in this work. Since the problem aims to find the number of modules used at each production facility, integer variables are introduced as opposed to binary variables which limit the information to a yes or no decisions. A typical supply chain network is shown in Figure 5-1.

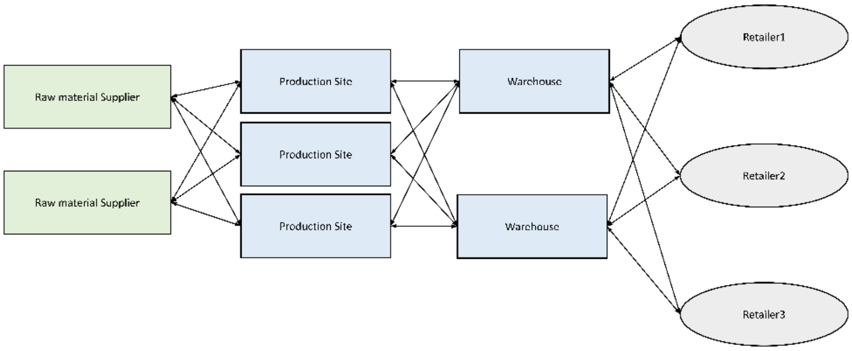


Figure 5-1 A typical supply chain network

The decisions to be made by optimizing this network include the choice of production facility for production in each time period, the number and the type of modules to be installed at each production site, the amount of material transported between suppliers, production facilities, warehouses, and retailers.

5.2.1 Optimization problem formulation

Based on the notations described in Appendix B, we formulate the following MINLP model.

Objective. Minimizing the total cost is the primary objective of this formulation. The total cost consists of transportation cost, inventory cost, capital cost, and operating cost as shown by Eq. (74)

$$\begin{aligned}\text{minimize cost} = \sum_{t\in\mathcal{T}} \Big(&\sum_{f\in\mathcal{F}} \sum_{w\in\mathcal{W}} \sum_{p\in P} Q_{fwpt}\, c_{fp} + \sum_{f\in\mathcal{F}} \sum_{w\in\mathcal{W}} \sum_{p\in P} Q_{fwpt}\, h_{fwp} \\ &+ \sum_{s\in\mathcal{S}} \sum_{f\in\mathcal{F}} \sum_{a\in\mathcal{A}} Q_{sfat}\, h_{sfa} + \sum_{w\in\mathcal{W}} \sum_{r\in\mathcal{R}} \sum_{p\in P} Q_{wrpt}\, h_{wrp} \\ &+ \sum_{w\in\mathcal{W}} \sum_{p\in\mathcal{P}} I_{wpt} g_{wp} \Big) + \sum_{m\in\mathcal{M}} \sum_{o\in\mathcal{O}_m} \zeta_{fmo} \left(\frac{z_{mo}}{\tilde{z}_{mo}} \right)^{\beta} \end{aligned} \qquad (74)$$

where, Q_{fwpt} is the quantity of the product p delivered from production facility f to warehouse w in time period t; , Q_{sfat} is the quantity of the raw material a delivered from supplier s to production facility f in time period t; Q_{wrpt} is the quantity of product p delivered from warehouse w to retailer r in time period t; c_{fp} operating cost per unit of product p at production facility f; h_{fwp} is transportation cost per unit of product p from production facility f to warehouse w; h_{sfa} transportation cost per unit of raw material a from supplier s to production facility f; h_{wrp} is transportation cost per unit of product p from warehouse w to retailer r; g_{wp} is inventory cost for storing a unit of product p for one time period at warehouse w; ζ_{fmo} is capital cost for installing one unit of option o for module m at a production facility f; z_{mo} is the number of units of option $o \in \mathcal{O}_m$ for module $m \in \mathcal{M}$ installed across all production facilities; $\tilde{z}_{mo}$ base number of units for option o for module m; β coefficient of mass production.

The objective function is nonlinear when the economies of numbers are considered, i.e., when β is not equal to 1. To avoid nonlinearity, the capital cost is linearized. It is assumed that Z represents the maximum number of units of option $o \in \mathcal{O}_m$ for module $m \in \mathcal{M}$ installed across all production facilities. The integer variables z_{mo} are re-expressed as a set of binary variables e_{mok} as shown by Eq.(75) and Eq.(78).

$$\sum_{k\in K} e_{mok} = 1 \quad K \in \{0, \dots, Z\} \tag{75}$$

For each integer value in the set {0, ..., Z}, a binary variable e_{mok} is defined. Eq.(76) states that only one of the binary variables can be 1. Finally, the integer variables and the binary variables are related through Eq. (77) which ensures that if e_{mok} is 1, z_{mo} is k.

$$\sum_{k\in K} ke_{mok} = z_{mo} \;\; K \in \{0, \dots, Z\} \tag{78}$$

Using the binary variables e_{mok}, we express the capital cost as shown by Eq.(79)

$$\sum_{m\in\mathcal{M}} \sum_{o\in\mathcal{O}_m} \zeta_{fmo} \left(\frac{z_{mo}}{\tilde{z}_{mo}}\right)^{\beta} = \sum_{m\in\mathcal{M}} \sum_{o\in\mathcal{O}_m} \sum_{k\in K} e_{mok}\zeta_{fmo} \left(\frac{k}{\tilde{z}_{mo}}\right)^{\beta} \quad K \in \{0, \dots, Z\} \tag{79}$$

where variable $\bar{\zeta}_{fmok}$ is defined using Eq. (80).

$$\bar{\zeta}_{fmok} = \zeta_{fmo} \left(\frac{k}{\tilde{z}_{mo}}\right)^{\beta} \quad K \in \{0, \dots, Z\} \tag{80}$$

The key advantage of this reformulation is that the term $\bar{\zeta}$ can be pre-computed for each value of k thus leading to a linear objective function.

Inventory balance

Inventory balance must be satisfied at each warehouse at each time period. Warehouse inventory at each time period is the difference between the amount of product they ship out and the amount of product they receive added to the warehouse inventory at the end of the previous planning period as shown by Eq. (81). At the beginning of first time period, the initial inventory I_{wp0} is known and introduced as a model parameter.

$$I_{wpt} = I_{wp(t-1)} + \sum_{f \in \mathcal{F}} \sum_{w \in \mathcal{W}} Q_{fwpt} - \sum_{w \in \mathcal{W}} \sum_{r \in \mathcal{R}} Q_{wrpt} \qquad \forall w \in \mathcal{W}, \forall p \in \mathcal{P}, \forall t \in \mathcal{T} \tag{81}$$

where I_{wpt} is the inventory of product $p \in \mathcal{P}$ stored at warehouse w during time period t. It is assumed that there is no inventory at the production facilities. As a result, no reorder policy is incorporated at the warehouses.

Demand satisfaction

The product demand must be satisfied at each planning period for each product. This is enforced by Eq.(82)

$$\sum_{w \in \mathcal{W}} Q_{wrpt} \geq \delta_{rpt} \qquad \forall r \in \mathcal{R}, \forall p \in \mathcal{P}, \forall t \in \mathcal{T} \tag{82}$$

where, δ_{rpt} is the demand for product p at retailer r in time period t. A strict demand constraint is imposed and the design is considered to be infeasible if the demand is not met.

Feasibility constraints

The production feasibility should be ensured at each production facility during each time period. This is accomplished with the help of machine learning-based feasibility models as shown in Eq. (83).

$$SVM_{of}(\boldsymbol{x_{tf}}) \geq -K(1 - y_{tfo}) \qquad \forall o \in \mathcal{O}_m; \forall m \in \mathcal{M}; \forall t \in \mathcal{T}; \forall f \in \mathcal{F} \tag{83}$$

where, $SVM_{of}(\boldsymbol{x_{tf}})$ represents SVM models for option o at production facility f; K is the big-M constant for the feasibility constraints; y_{tfo} are binary variable indicating if option $o \in \mathcal{O}_m$ for module $m \in \mathcal{M}$ during time period t is installed at production facility f; $\boldsymbol{x_{tf}}$ is material flow per production line during time period t at production facility f. The value of K is obtained empirically for each SVM model, since the upper and lower bounds on the variables are known over which a particular SVM model is applicable. A large number of samples are generated and the value of $SVM_{of}(\boldsymbol{x_{tf}})$ is calculated and K is chosen to be larger than the maximum of all values. The constraint indicates that when y_{tfo} is 1, $SVM_{of}(\boldsymbol{x_{tf}})$ should be positive, indicating a feasible point.

Moreover, it is assumed that each processing line at a production facility is subject to equal throughput as shown by Eq. (84)

$$\boldsymbol{x_{tf}} v_{tf} = \sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} Q_{sfat} \qquad \forall t \in \mathcal{T}; \forall f \in \mathcal{F} \tag{84}$$

where, v_{tf} corresponds to the number of production lines during time period t at production facility f; $\boldsymbol{x_{tf}}$ is the raw material flow per production line during time period t at production facility f. Additionally, the process variables should be in agreement with the chosen module option as enforced by Eq. (85-86)

$$lb_o - x_{tf} \leq H(1 - y_{tfo}) \qquad \forall o \in \mathcal{O}_m; \forall m \in \mathcal{M}; \forall t \in \mathcal{T}; \forall f \in \mathcal{F} \tag{85}$$

$$x_{tf} - ub_o \leq H(1 - y_{tfo}) \qquad \forall o \in \mathcal{O}_m; \forall m \in \mathcal{M}; \forall t \in \mathcal{T}; \forall f \in \mathcal{F} \tag{86}$$

where, ub_o is a vector of upper bounds on the variable x for option o for module m; lb_o is vector of lower bounds on the variable x for option o for module m; H is a big-M constant.

It is assumed that production facilities do not have any storage capacity for products and thus all the material produced at production facilities is transported to warehouses for storage. The total amount of products produced at the production facilities is the summation of the product produced at each parallel processing line as shown by Eq. (87)

$$\boldsymbol{x_{tf}} v_{tf} = \sum_{f \in \mathcal{F}} \sum_{p \in P} Q_{fwpt} \qquad \forall t \in \mathcal{T}; \forall w \in \mathcal{W} \tag{87}$$

where, $\boldsymbol{x_{tf}}$ is the product flow per production line during time period t at production facility f.

Logical constraints

The number of active processing lines at a given production facility in each time period are evaluated using Eq. (88). This is required for constraint given by Eq. (84) to model the throughput to each processing line.

$$v_{tf} = \sum_{o \in \mathcal{O}_m} q_{tfo} \qquad \forall t \in \mathcal{T}, f \in \mathcal{F} \tag{88}$$

q_{tfo} is the number of production lines of option $o \in \mathcal{O}_m$ for module $m \in \mathcal{M}$ during time period t at production facility f. The constraints shown by Eq.(89 -90) are needed to evaluate z_{mo} as these variables are not included in any other constraints but are needed for the evaluation of the capital cost in the objective function.

$$\bar{v}_{fo} \geq q_{tfo} \qquad \forall t \in \mathcal{T}, \forall f \in \mathcal{F}, \forall o \in \mathcal{O}_m; \forall m \in \mathcal{M} \tag{89}$$

where, $\bar{v}_{fo}$ is the number of production lines of option $o \in \mathcal{O}_m$ for module $m \in \mathcal{M}$ installed at production facility f.

$$z_{mo} = \sum_{f \in \mathcal{F}} \bar{v}_{fo} \qquad \forall o \in \mathcal{O}_m; \forall m \in \mathcal{M} \tag{90}$$

The relation between integer variables q_{tfo} and binary variables y_{tfo} is established with the help of constraints given by Eq. (91-92).

$$y_{tfo} \leq q_{tfo} \quad \forall t \in \mathcal{T}, f \in \mathcal{F}, \forall o \in \mathcal{O}_m; \forall m \in \mathcal{M} \tag{91}$$

$$q_{tfo} \leq v_{\max} y_{tfo} \qquad \forall t \in \mathcal{T}, f \in \mathcal{F}, \forall o \in \mathcal{O}_m; \forall m \in \mathcal{M} \tag{92}$$

It is assumed that the total number of units of each module active at a production facility are same for all modules. This is modeled using Eq. (93).

$$\sum_{o \in \mathcal{O}_{\bar{m}}, \bar{m} \in \mathcal{M}} q_{tfo} = \sum_{o \in \mathcal{O}_m, m \in \mathcal{M} \setminus \{\bar{m}\}} q_{tfo} \qquad \forall t \in \mathcal{T}, f \in \mathcal{F} \tag{93}$$

It is assumed that in each time period, only one module option can be installed at a production facility as shown by Eq. (94)

$$\sum_{o \in \mathcal{O}_m, m \in M} y_{tfo} \leq 1 \qquad \forall t \in \mathcal{T}, f \in \mathcal{F} \tag{94}$$

where, $Q_{fwpt}, Q_{sfat}, Q_{wrpt}, I_{wpt}, x_{tf} \in \mathbb{R}; z_{mo}, v_{tf}, \bar{v}_{fo}, q_{tfo} \in \mathbb{Z}; y_{tfo} \in \{0,1\}; \quad Q_{fwpt} \geq 0; Q_{sfat} \geq 0, Q_{wrpt} \geq 0; 0 \leq I_{wpt} \leq \bar{I}_{wp}; 0 \leq x_{tf}.$

It is assumed that the optimal production facilities selected by solving the optimization problem are installed at the beginning of the planning horizon.

It is important to note that the constraints given by Eq.(84) and Eq.(87) contain additional bilinear terms. The bilinear terms in this case, are multiplications of continuous variable and integer variables. Since McCormick relaxations are accurate at the boundaries or end points, it is beneficial to express integer variables in terms of binary variables. This is achieved by first rewriting Eq. (84) and Eq. (87) as Eq.(95) and Eq.(96), respectively.

$$\sum_{j\in J} g_{tfj} = \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} Q_{sfat} \tag{95}$$

$$\sum_{j\in J} b_{tfj} = \sum_{f\in\mathcal{F}}\sum_{p\in P} Q_{fwpt} \tag{96}$$

where, g_{tfj} and b_{tfj} are intermediate variables which can be expressed as shown by Eq.(97) and Eq.(98), respectively.

$$g_{tfj} = \boldsymbol{x_{tf}} j u_{tfj} \tag{97}$$

$$b_{tfj} = \boldsymbol{x_{tf}} j u_{tfj} \tag{98}$$

The integer variables v_{tf} are expanded using binary variables u_{tfj}, as shown by Eq.(99) and Eq.(100).

$$\sum_{j\in J} u_{tfj} = 1 \quad J \in \{0, \dots, N\} \tag{99}$$

where, N is the upper bound on integer variables v_{tf}. For each integer value from 0 to N, a binary variable u_{tfj} is defined. Constraint given by Eq.(99) states that only one of the binary variables can be 1. Finally, the integer variables and the binary variables are related through Eq. (100) which ensures that if u_{tfj} is 1, v_{tf} is j.

$$\sum_{j\in J} j u_{tfj} = v_{tf} \quad J \in \{0, \dots, N\} \tag{100}$$

The constraints shown by Eq.(97) and Eq.(98) are then relaxed using McCormick relaxations as discussed in Section 5.3.2. The relaxations for Eq.(97) are shown by Eq. (101-104)

$$g_{tfj} \geq x_{tf}^{L} ju_{tfj} \tag{101}$$

$$g_{tfj} \geq x_{tf}^{U} ju_{tfj} + x_{tf}j - x_{tf}^{U}j \tag{102}$$

$$g_{tfj} \leq x_{tf}^{U} ju_{tfj} \tag{103}$$

$$g_{tfj} \leq x_{tf}j + x_{tf}^{L} ju_{tfj} - x_{tf}^{L}j \tag{104}$$

where, x_{tf}^{L} and x_{tf}^{U} represent lower and upper bounds on the continuous variables that represent the material flow rate of the raw materials respectively. The lower bound for binary variables is 0 and the upper bound is 1. However, since the variable being relaxed is ju_{tfj}, the upper bound is replaced by j. Similarly, the relaxations for Eq.(98) are shown by Eq.(105-108).

$$b_{tfj} \geq x_{tf}^{L} u_{tfj} \tag{105}$$

$$b_{tfj} \geq x_{tf}^{U} u_{tfj} + x_{tf}j - x_{tf}^{U}j \tag{106}$$

$$b_{tfj} \leq x_{tf}^{U} ju_{tfj} \tag{107}$$

$$b_{tfj} \leq x_{tf}j + x_{tf}^{L} ju_{tfj} - x_{tf}^{L}j \tag{108}$$

where, x_{tf}^{L} and x_{tf}^{U} represent lower and upper bounds on the continuous variables that represent the material flow rate of the products respectively. Since the relaxations shown by Eq. (101-108) are with respect to the binary variable u_j, the relaxation will lead to the optimal solution of the original problem.

5.3 Background

5.3.1 Economy of numbers

In manufacturing, when a module is built in large numbers the cost of a module decreases because of the effect of learning or experience. This effect is studied under economies of experiential learning or economies of numbers and it is an important factor to consider for assessing the economic advantages of modular designs. Like the economy of scale where a 2/3rd power law is commonly used to estimate the capital cost across different scales, economies of numbers are modeled using an exponential relation as shown by Eq. (109)

$$C_{eon} = C_0(n)^{\beta} \tag{109}$$

where, C_{eon} represents the capital cost considering the economy of numbers; β is the coefficient of mass production; C_0 is the capital cost for producing one equipment; n is the number of modules produced.

Palys et al. [14] solve an integer programming problem where the number of modules is decided in order to minimize the supply chain cost. They study the effect of exponent β on the supply chain cost using Eq. (109). The economy of numbers as shown by Eq. (109) and economies of scale can be combined to represent the capital cost as shown in Eq. (110) [17]

$$C = C_0\left(\frac{d}{D_0}\right)^{\alpha}(n)^{\beta} \tag{110}$$

where, β is the exponent for economy of scale. n is the actual number of equipment; α_i is the factor for economies of scale; d is the equipment capacity; D_0 represents the base capacity used for evaluating C_0; C_0 is total annualized capital cost.

5.3.2 Convexification of bilinear terms using McCormick relaxation

The presence of bilinear terms in an MINLP problem can significantly affect the computational complexity of solving the problem to optimality by introducing additional nonconvexity which leads to multiple local optima. In such a case, the computational complexity of an MINLP model can be reduced if the problem can be partly or fully convexified. McCormick envelopes provide a way to convexify bilinear terms leading to a convex nonlinear programming model [226][227]. McCormick envelopes reduce the complexity of an MINLP problem at the cost of leading to a solution that does not corresponding to the objective function. Solution of the relaxed problem is a lower bound to the solution of the original problem. Therefore, there have been studies aiming to achieve a tighter relaxation that is closer to the true solution [228][229]. A typical bilinear term is shown by Eq.(111), where variables x and y are continuous variables.

$$xy = q \tag{111}$$

Because of the term contains products of two variables, the resulting constraint is a nonconvex constraint. This constraint can be relaxed using McCormick relaxation as shown by Eq.(112).

$$\begin{gathered} q \geq x^L y + x y^L - x^L y^L \\ q \geq x^U y + x y^U - x^U y^U \\ q \leq x^U y + x y^L - x^U y^L \\ q \leq x y^U + x^L y - x^L y^U \end{gathered} \tag{112}$$

It should be noted that the McCormick relaxation is exact when one of the variables x and y in Eq.(112) is binary. In such a case, solving the relaxed problem leads to an optimal solution to the original MINLP.

In the context of the problem considered in this work, the bilinear term appears as a product of continuous and integer variables. To adapt the McCormick relaxation for integer variables while obtaining an exact solution, first the integer variables need to be expressed as binary variables. To achieve this, binary variables z_j are introduced where each j corresponds to an integer value of j that the variable y can take. By definition, only one of the variables z_j can be 1. This is enforced by Eq. (113).

$$\sum_{j \in J} z_j = 1 \tag{113}$$

The relation between binary variables z_j and the integer variables is established using Eq.(114).

$$\sum_{j} j z_j = y \tag{114}$$

It can be observed that if the value of y is j, z_j is 1 and vice versa. Finally, intermediate variables p are used such that the relation shown by Eq. (115)

$$\sum_{j \in J} p_j = q \tag{115}$$

where, each p_j is expressed using Eq. (116). Variables p_j are intermediate variables that do not hold any physical significance but are required to write McCormick relaxation for the bilinear terms.

$$p_j = xjz_j \quad J \in \{0, \dots, N\} \tag{116}$$

It can be observed that due to multiplication of continuous and binary variables in Eq.(116), the constraint is bilinear. McCormick relaxations can now be written for Eq.(116) that lead to an exact solution as that of the original problem.

5.4 Results

In this section, a case study of a supply chain network consisting of a modular process that includes a system of reactor and separator is considered. A detailed explanation of the proposed solution methodology and results are first shown on a problem consisting of the planning horizon of six months and two planning periods of three months. The aim is to study the effect of mass production exponent on the optimal facility location, as well as on the optimal process design. Further, the framework is applied to another scenario consisting of eight planning periods of 3 months with the planning horizon of two years.

Problem definition

To demonstrate the idea, a sample superstructure of the supply chain network is considered, as shown in Figure 5-1. The network consists three retailers and two warehouses and potential two suppliers and three production facilities to choose from. The planning horizon consists of two planning periods. In each time period, demand is realized by the retailers. The arrows in Figure 5-1 show the allowable material flow in the supply chain network. The problem is to minimize the total cost of the supply chain, as shown by Eq.(74). The parameters such as distances between all possible combinations of entities in the supply chain superstructure, inventory holding costs at warehouses, product demands at the retailers, and operating costs at the production sites are known a priori. It is important to note that the network shown in Figure 5-1 represents a superstructure of all possible options for the optimal design of the supply chain network. The optimal design choice will also select the location of the entities in the supply chain network.

At the production facilities, the production is carried out through a process consisting of a continuously stirred tank reactor in series with an ideal separator. The process is modularized into reactor and separator modules. The aim is to convert raw material A into two finished products B and E, as shown in Figure 4-2. An isothermal liquid-phase reaction is considered following the kinetic mechanism as described in the previous studies of Rooney and Biegler [198] and Goyal and Ierapetritou [169]. The model equations for the process are shown by Eq. (117)

$$\begin{gathered}
F_{A0} - x_A F(1-\alpha) - VC_{A0}(k_1 + k_2)x_A = 0 \\
-Fx_B(1-\alpha) + VC_{A0}k_1x_A = 0 \\
-Fx_C + VC_{A0}(k_2x_A - (k_3 + k_4)x_C + k_5x_E) = 0 \\
-Fx_D(1-\beta) + VC_{A0}k_3x_C = 0 \\
-Fx_E(1-\beta) + VC_{A0}(k_4x_C - k_5x_D) = 0 \\
x_A + x_B + x_C + x_D + x_E - 1 = 0
\end{gathered} \tag{117}$$

where, x_A, x_B, x_C, x_D, and x_E represent the mole fraction of components A, B, C, D, and E, respectively; k_i are the rate constants; V is the volume of the reactor; C_{A0} is the inlet concentration of A; α is the recycle fraction of stream A and B; β is the recycle fraction of D and E; F is the molar flow rate at the outlet of the reactor; F_{A0} is the molar flow rate at the inlet of the reactor. The nominal values of the kinetic constants are $k_1 = 0.0374$, $k_2 = 0.0195$, $k_3 = 0.0165$, $k_4 = 0.2701$, and $k_5 = 0.0261$.

It is assumed that four reactor design options are available based on their volume. Different separator design options are available depending on the throughput that they can handle. The available options and the respective costs for the reactor and the separator are shown in Table 5-1.

Table 5-1: Design options for reactor and separator

Options	**Reactor (m^3)**	**C_r (k$)**	**Separator** **(F_{A0}mol/h)**	**C_s (k$)**
Option 1	5	400	30-50	300

Option 2	20	850	40-70	720
Option 3	35	1200	60-100	980
Option 4	50	1650	90-140	1210

The problem is to determine the optimal supply chain network that minimizes the total cost of the supply chain over the planning horizon. The decisions include the following. i) location of production facilities ii) selection of module options to define a process iii) optimal material flow iv) optimal selection of suppliers v) Feasibility of satisfying the product demand. In general, it is assumed the coefficient of mass production will be known for the modules under consideration. However, in this problem, we study the effect of the coefficient of mass production on the optimal design of the network.

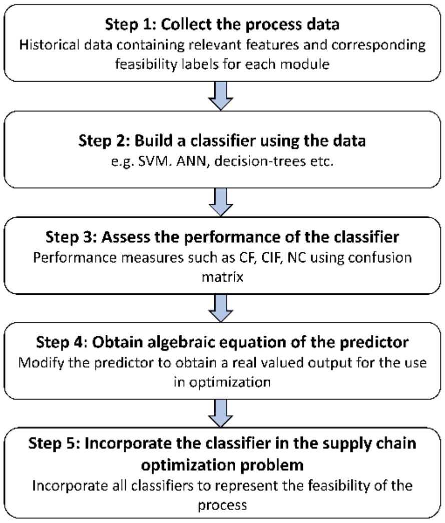

Step 1: Collect the process data
Historical data containing relevant features and corresponding feasibility labels for each module
Step 2: Build a classifier using the data
e.g. SVM. ANN, decision-trees etc.
Step 3: Assess the performance of the classifier
Performance measures such as CF, CIF, NC using confusion matrix
Step 4: Obtain algebraic equation of the predictor
Modify the predictor to obtain a real valued output for the use in optimization
Step 5: Incorporate the classifier in the supply chain optimization problem
Incorporate all classifiers to represent the feasibility of the process

Figure 5-2 Framework for integration of process data in the supply chain optimization formulation

Solution methodology

For solving this problem, the approach first obtains the feasibility constraints, followed by incorporating the constraints in the optimization formulation presented in Section 5.2. For obtaining the feasibility constraints, the methodology used by Bhosekar and Ierapetritou [225] is used. The methodology is shown in Figure 5-2, and the details of the individual steps are described as follows.

Step 1: The first step is to collect the historical feasibility data or data from process simulation. In this example, the data is obtained by running the simulation of the reactor, as shown in Eq. (69). The simulation is developed in GAMS 31.1.0 and solved as a nonlinear program using Baron mixed integer nonlinear programming solver version 20.4.14. Inputs for the simulation include four variables that are reactor volume, F_{A0}, F_B, and F_E. The output includes labels (-1 for infeasible and 1 for feasible) that indicate if the set of inputs leads to a feasible process. For each option of the reactor a grid-based sampling approach is used and 512 data points (10 samples in each F_{A0}, F_B, F_E) are generated, and the output labels are collected. It is assumed that the feasibility of the ideal separator depends only on the inlet flow rate range.

Step 2: In this step, we train a classifier for each option using the data generated in step 1. In this work, SVM is the chosen classifier, and SVM models are trained for the reactor as described in section 4.2.3. The scikit-learn python toolbox is used with default options for training the SVM models. It should be noted that since the separator is ideal and its feasibility depends only on the flow rate, there is no need to build a classifier for the separator.

Step 3: In this step, the model quality is assessed using the test dataset. Because of the simplicity of the classifier equation and satisfactory performance, this work implements a linear kernel for solving the optimization problem using steps 4 and 5. The model performance needs to be improved by choosing a different kernel, more data, or different set of parameters, or a different classifier such as neural network if the model built in this step does not meet the desired quality.

The validation results for reactor options are shown in Table 5-2 where it can be observed that the models have satisfactory values for validation metrics.

Table 5-2: SVM model validation for the reactor using a linear kernel

Option	**CF%**	**CIF%**	**NC%**	**Total Error**
Option 1	100	100	0	0
Option 2	86.36	97.53	9.52	4.85
Option 3	81.82	95.71	10	8.73
Option 4	85.36	88.7	16.66	12.62

Step 4: In this step, we obtain an algebraic equation for the SVM model. This is done by obtaining the intercept and support vectors from the trained classifier from step 3. Since this is the only information required for Eq. (65), algebraic expression for the classifier can be obtained.

Step 5: This is the final step of the framework where the classifier is incorporated into the optimization problem. The problem is obtained by replacing constraints shown by Eq. (117) with Eq. (118).

$$
\begin{gathered}
SVM_r(\boldsymbol{x}) \geq -M_1(1-y_r) \quad \forall r \in R; \forall k \in K \\
V - V_r \leq M(1-y_r) \quad \forall r \in R \\
V_r - V \leq M(1-y_r) \quad \forall r \in R \\
F_{A0} - ub_s \leq M(1-y_s)) \quad \forall s \in S \\
lb_s - F_{A0} \leq M(1-y_s)) \quad \forall s \in S
\end{gathered}
\tag{118}
$$

To study the effect of mass production exponent β, four different values of β are considered. The problem is solved using GAMS 31.1.0 with CPLEX 12.10.0.0. The critical optimality tolerance is set to 0. The maximum allowable time limit for solving the problem to optimality is set to 10000 seconds. First, a value of 1.0 is considered for the exponent of mass production. This value indicates an absence of economy of numbers, as can be observed by Eq. (109). The optimal reactor and separator options, as well as optimal production facility location for this case, are shown in Table 5-3 along with the corresponding results for lower values of β such as 0.7, 0.8, and 0.9. The total number of equipment for the chosen reactor and separator module options that are installed in the supply chain network for all values of β under comparison are shown in Table 5-4. The units Ri and Si represent ith option for the reactor and separator respectively. For example, R1 represents the first option for the reactor and S1 represents the first option for the separator. The number of lines in Table 5-3 correspond to the respective production facilities. For example, for β value of 0.7, there are production facilities 1, 2, and 3 have 1, 2, and 2 units of reactor option 2 (R2) installed.

Table 5-3 Optimal reactor and separator options as β is changed

	$\boldsymbol{\beta = 0.7}$		$\boldsymbol{\beta = 0.8}$		$\boldsymbol{\beta = 0.9}$		$\boldsymbol{\beta = 1.0}$	
Unit	Production facility	Number of lines	Production facility	Number of lines	Production facility	Number of lines	Production facility	Number of lines
R1	-	-	-	-	-	-	-	-
R2	1,2,3	1,2,2	1,2,3	1,2,2	1	1	-	-
R3	-	-	-	-	3	1	1	1
R4	-	-	-	-	-	-	3	1
S1	1,2,3	1,2,2	1,2,3	1,2,2	-	-	1	1
S2	-	-	-	-	-	-	-	-
S3	-	-	-	-	-	-	-	-
S4	-	-	-	-	1,3	1,1	3	1

Table 5-4 The total number of equipment for the reactor and separator

	$\boldsymbol{\beta = 0.7}$	$\boldsymbol{\beta = 0.8}$	$\boldsymbol{\beta = 0.9}$	$\boldsymbol{\beta = 1.0}$
Unit	Total number of lines	Total number of lines	Total number of lines	Total number of lines
R1	-	-	-	-
R2	5	5	1	-
R3	-	-	1	1
R4	-	-	-	1
S1	5	5	-	1
S2	-	-	-	-
S3	-	-	-	-
S4	-	-	2	1

It can be observed from Table 5-3 that in the absence of the economy of numbers (the case with $\beta = 1$), the optimal solution includes one unit each of two reactor options and one unit each of two separator options. The options selected are options 3 and 4 for the reactor and options 1 and 4 for the separator. As β is reduced to 0.9, effect of economy of numbers can be observed as illustrated in Table 5-3, the optimal solution includes two units of separator option 4 and reactor options 2 and 3. This shows that the optimal solution slightly favors standardized and smaller units. This effect is more profound for lower values of β where the reactor option 1 and separator option 2 leads to the most cost-effective solutions. Since the options shown in the first two columns of Table 5-3 represent smaller and less expensive units than those shown in the next two columns, it can be observed that in the presence of the economy of numbers, smaller modules are cost-efficient. For the results related to β values of 0.9 and 1.0 in Table 5-3, it can be observed that the optimal solution includes production facilities 1 and 3, whereas production facility 2 is excluded. On the other hand, for the values of β of 0.7 and 0.8, all production facilities are selected, signifying a more distributed production. It should also be noted that in the absence of the economy of numbers, the framework prefers bigger and decentralized production, indicating that the tradeoff between capital cost savings and transportation costs is addressed. The individual costs for all the values of β are reported in Table 5-5. For lower values of β, the framework finds a solution that reduces transportation cost, as well as the capital cost. In the absence of the economy of numbers, this solution is not preferred. This is because the savings in the capital costs dominate the savings in the transportation costs.

These trends are better depicted in Figure 5-3, Figure 5-4, and Figure 5-5. Optimal total cost increases as the exponent of mass production is increased, as shown in Figure 5-3. This is intuitive since the solution obtained with the consideration of the economy of numbers will be at least as cost-effective as the solution obtained with the consideration of the economy of numbers since the only difference in two cases is the capital cost, which will be lower. However, Figure 5-4 demonstrates the difference in the transportation cost, which indicates that a different solution is preferred for the cases where β is 0.7 and 0.8. Figure 5-5 shows the capital cost corresponding to the optimal solutions as the coefficient of mass production is varied from 0.7 to 1. It can be observed the capital cost increases as β is increased. However, the second data series presented in Figure 5-5 shows the capital cost for the optimal solution if β was equal to 1. This shows that for the case when β is 1, the gain achieved by centralized and relatively larger scale process modules dominates the advantages of the solutions preferred when β is 0.7.

Table 5-5 Cost comparison for different values of β

Cost	$\boldsymbol{\beta = 0.7}$	$\boldsymbol{\beta = 0.8}$	$\boldsymbol{\beta = 0.9}$	$\boldsymbol{\beta = 1.0}$
Total cost(mil $)	3.815304	4.434842	4.622018	4.678892

Transportation cost (mil $)	0.194348	0.194348	0.241078	0.245892
Capital cost (mil $)	3.547945	4.167483	4.307939	4.360000
Operating cost (k$)	73.000	73.000	73.000	73.000
Inv cost ($)	10.6	10.6	0	0

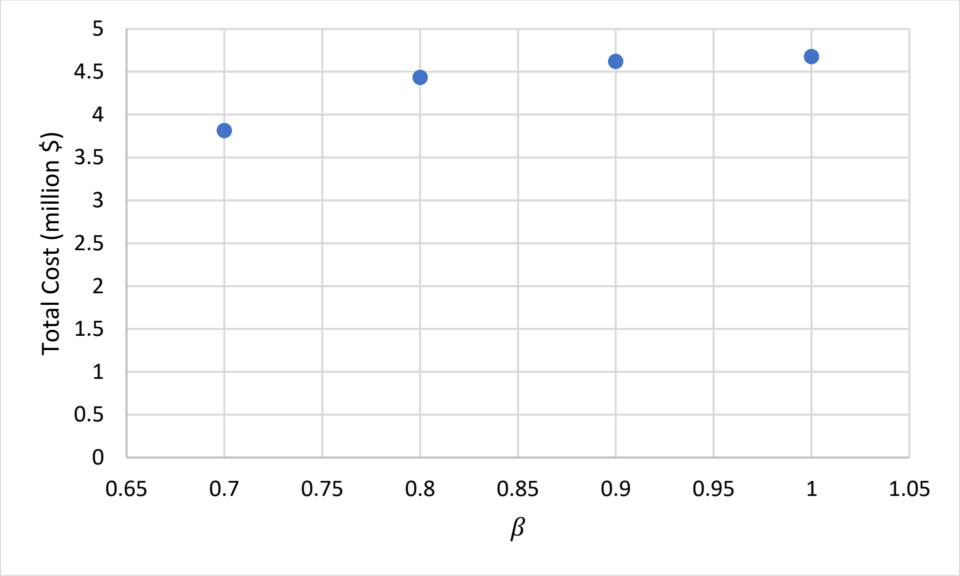


Figure 5-3 Optimal total cost as mass production exponent is varied

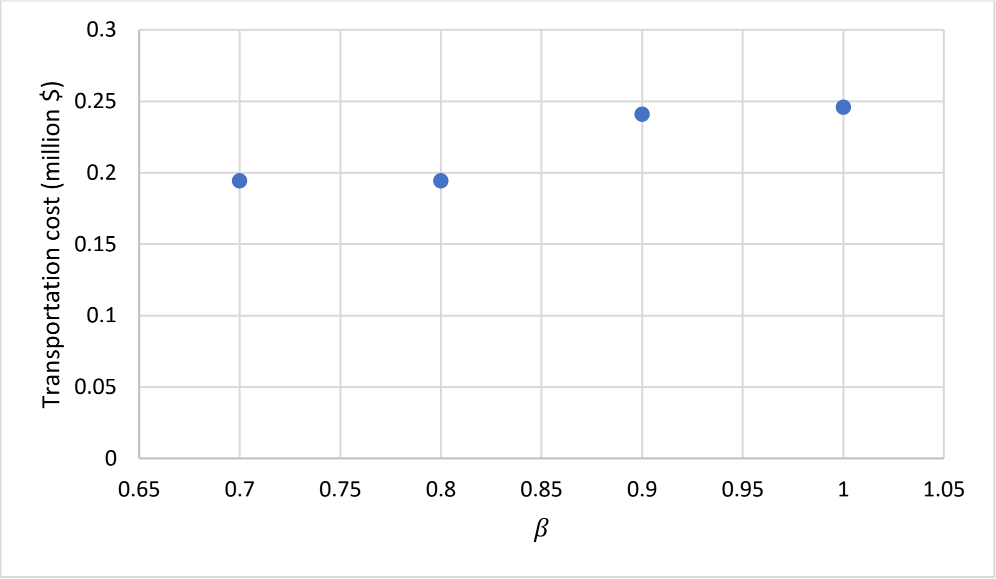


Figure 5-4 Optimal transportation cost as the mass production exponent is varied

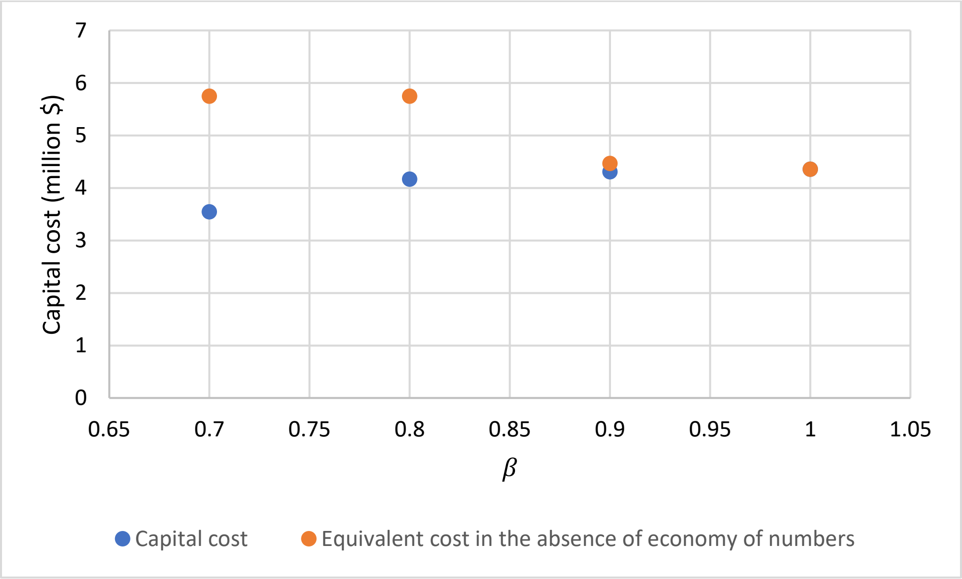


Figure 5-5 Optimal capital cost as the mass production exponent is varied

In the next study, we consider a supply chain network consisting of 4 suppliers, 4 production sites, 3 warehouses, and 5 retailers. The planning horizon considered consists of 8 planning periods. The parameters such as distances, holding costs, demands, and operating costs are considered to be known. Due to a longer planning horizon and more number of supply chain entities this problem is computationally more demanding than the previous case study.

The problem is solved using the formulation proposed in section 5.2 using the mixed integer linear programming solver CPLEX through GAMS with a time limit of 10000 seconds. The critical optimality tolerance is set to 2 percent. The problem has 1056 discrete variables, and the problem is solved to optimality in 9510 seconds on a PC with Intel® Xeon® CPU E-2174G @ 3.80GHz and 32.0 GB RAM, running a Windows 10 Enterprise, 64-bit operating system when the value of β is 0.7. The optimal total cost obtained is $7.74 million, and the optimal solution is presented in Table 5-6 and Table 5-7. The same problem in the absence of economy of numbers leads to a different solution in 3970 seconds with an optimal cost of $49.07 million, as shown in the columns corresponding to $\beta = 1$ in Table 5-6 and Table 5-7. The key difference is that the reactor and separator options proposed by the optimal solution are bigger than those suggested for the previous case with $\beta = 0.7$. The results for β value of 1 are shown in the columns corresponding to $\beta = 1$ in Table 5-6 and Table 5-7. It can be observed that only three units of reactor and separator option 4 are included in the optimal solution. This again shows that with the consideration for the economy of numbers, distributed and smaller-scale manufacturing is preferred.

Table 5-6 Optimal reactor and separator options for two values of β

	$\boldsymbol{\beta = 0.7}$		$\boldsymbol{\beta = 1}$	
Unit	Production facility	Number of lines	Production facility	Number of lines
R1	-	-	-	-

R2	1,2,3,4	2,4,3,1	-	-
R3	-	-	-	-
R4	-	-	1,2,3	1,1,1
S1	1,2,3,4	2,4,3,1	-	-
S2	-	-	-	-
S3	-	-	-	-
S4	-	-	1,2,3	1,1,1

Table 5-7 The total number of equipment for the reactor and separator

	$\boldsymbol{\beta = 0.7}$	$\boldsymbol{\beta = 1}$
Unit	Total number of lines	Total number of lines
R1	-	-
R2	10	-
R3	-	-
R4	-	3
S1	10	-
S2	-	-
S3	-	-
S4	-	3

Finally, it should be noted that the optimal solutions across all the scenarios considered in the two case studies consist of different options for the reactor and separator. For β value of 1, the optimal solution includes reactor option 4 and separator option 4. On the other hand, for β value of 0.7, the optimal solution includes reactor option 2 and separator option 1. The ability to pick the most appropriate modules for the different unit operations involved in the production facility can provide an additional flexibility to supply chain design optimization compared to the case where the entire processing line is considered as one module.

5.5 Conclusions

In this work, a modular supply chain optimization formulation is presented. The formulation considers multiperiod optimization of the supply chain network and simultaneously addresses the problems of facility location, process design, production feasibility analysis, and the effect of mass production on the capital cost of the process. For finding an optimal facility location, the formulation chooses an optimal solution from the set of potential locations. For production feasibility analysis, the model uses SVM classifiers trained using historical production data to approximate the feasible region of the available standardized process modules. For modeling the effect of mass production, an exponential relation is used with the help of the coefficient of mass production that reduces the cost per-module as more modules are installed. The resulting problem consists of binary variables that represent facility locations, and integer variables represent the number of modular equipment installed. This results in an MINLP problem, and the optimal solution addresses the tradeoff between centralized vs. distributed manufacturing. Since the coefficient of mass production depends on the specific modules considered in the problem of interest, a study is provided for different values of the coefficient of mass production.

The resulting problem is reformulated with the help of McCormick relaxations that help reduce the computational complexity of the formulation. Further, the nonlinearity in the capital cost resulting from the economies of numbers is resolved with the help of binary expansion of integer variables converting the optimization problem to an MILP problem. The formulation is applied to a case study of a process consisting of a reactor and separator, where the process can be modularized in two modules based on the unit operations. The process design is obtained by choosing from multiple options for each module, varying in cost as well as capacity. The supply chain optimization problem is solved for two scenarios. In the first scenario, a two-period planning horizon is assumed, and in the second scenario, a five-period planning horizon is considered. Studying the effect of exponent of mass production, it is noted that considering the economy of numbers is essential in truly assessing the economic viability of a small scale, modular, and distributed manufacturing. This work also demonstrates a way using which historical data about a process can be utilized in supply chain network design. As opposed to traditional studies where processing capacity is assumed, the accurate representations of process feasible regions can help in understanding the flexibility of a supply chain network with respect to uncertain parameters such as the product demands. Finally, the formulation presented in this work can be extended in multiple directions. In terms of optimization, for larger-scale problems, decomposition algorithms for improving the computational complexity will be investigated. The formulation can be extended to conduct a systematic flexibility analysis of the supply chain network when subject to product demand uncertainty.

6 Modular supply chain optimization under uncertainty

Abstract

Due to increased competition, globalization, and market volatility, supply chain design under demand uncertainty has been a challenging problem. With the flexibility in strategic decisions as well as in the processing capacity that the modular designs provide, modular manufacturing is a promising way forward for addressing supply chain design under uncertainty. It has been shown that modular designs provide several advantages such as flexible processing capacity, lower costs due to standardization. In this work, a supply chain network consisting of modular processes is considered. The problem of simultaneously designing the process as well as the supply chain is addressed under product demand uncertainty for a multiperiod planning horizon. A mixed integer two-stage stochastic linear programming model is formulated with integer variables indicating the process design and continuous variables represent the material flow in the supply chain network. The problem is solved using a rolling horizon approach. Benders decomposition is used to reduce the computational complexity of the optimization problem. In addition to minimizing the expected cost of the supply chain network, a downside risk measure is incorporated that helps risk-averse decision making. The results demonstrate the flexibility of modular designs to achieve the desired throughputs and, at the same time, yield a pareto optimal curve for minimizing the objectives of expected cost and downside risk.

6.1 Introduction

The increasing popularity of modular design can be attributed to several manufacturing and strategic advantages that modular designs provide. Modular designs reduce time-to-market for a product because of preassembled modules, lower construction times due to standardization, and the possibility of numbering up identical devices to achieve the desired production. Shorter time-to-market, combined with the safety of operation[9], reduces the financial risk for the investors[152]. Moreover, centralized large-scale manufacturing often means high transportation costs for the delivery of raw materials as well as the products. In contrast, small scale, modular manufacturing provides a way to distribute manufacturing effectively to reduce transportation costs. Finally, with repeated production of standardized units, engineers, as well as the vendors, gain experience, which results in lowering the capital cost and improving production lead time [17][215][216]. As the benefits of modular processes are better appreciated at the supply chain level, recent literature focuses on the supply chain consisting of modular processes. Palys et al. [14] consider ammonia supply chain optimization where modular, renewable ammonia production using the new technology can be added to the existing supply chain network. In their study, the cost of traditional centralized production is compared with a modular design by varying the mass production exponent that quantifies the cost-savings due to standardization. Allman and Zhang [217] address the possibility of mobile production units and determine their optimal location and relocation under time-varying demands. Becker et al. [218] demonstrate increased flexibility due to modular manufacturing on the network cost for the problem of tactical planning of modular production networks. The formulations proposed in their work determine the location, process, and capacity of modular plants. Lier et al. [219] provide a review of challenges on the equipment level, network level, and logistics level for transformable production with modular composition. Yang and You [153] study the economic and environmental aspects of plant relocation, centralized manufacturing, and distributed manufacturing for methanol manufacturing using a model-based study. The aforementioned work considers the deterministic supply chain optimization problem.

However, in a practical supply chain network, several uncertainties exist in the form of product demand, production delays, and delivery times to list a few. Due to market volatility, supply chain design under uncertainty has been a challenging problem. You and Grossmann [230] consider the problem of the design of a responsive supply chain addressing the objectives of cost and expected lead times. They propose optimization models and algorithms to solve those models to address the integrated problem of network design and inventory management[231]. Sahay and Ierapetritou [232] propose a hybrid simulation and optimization framework for supply chain optimization under demand uncertainty, where the downside risk and profit objectives are simultaneously optimized. They show that supply chain design considering only the periods in the near future may lead to infeasible supply chain network for future periods when product demand increases. In such a case, desired service levels will not be maintained. Since modular manufacturing provides greater flexibility than a centralized plant, it provides a promising way in this respect. With modular manufacturing, inefficiencies due to overdesign can be avoided at the same time maintaining desired service levels and low inventory costs. However, for the case of supply chains with modular processing, uncertainty considerations are relatively less explored. Allen et al. [16] consider the problem of shale gas production planning, where uncertainties are associated with the depletion of resources. Considering the mobility of modular equipment, they solve a multistage stochastic programming problem. They note that modular plants are beneficial in such a case due to the possibility of operating at lower throughputs as the resources deplete. As shown by our recent work [188], considering production feasibility in design optimization as well as in supply chain optimization (Chapter 5) can provide an additional level of flexibility where a process can be defined using the appropriate standardized modules of choice.

The critical details at the production level relate to defining a process based on a limited number of different modules [155]. Previously[225], it has been shown that modular designs have an added advantage of providing a quantitative estimate of the flexibility of process design. One reason for flexibility is the increased degrees of freedom since individual modules can be designed instead of designing the process as a whole. The approach presented in our previous work uses historical information related to the feasible region of operation for each module in a data-driven feasibility analysis framework. Process feasibility can then be defined using the feasibility of individual modules. Consideration for feasibility in the optimization process can have several advantages, and this approach has been proven useful in the past on a variety of applications as discussed in section 2.4). In this work, it is assumed that data-driven approximation of the feasible region is available.

In this work, the supply chain network design problem is addressed using modular production to determine the optimal process design, and the facility location to minimize the total cost of the supply chain over the desired planning horizon. The formulation presented in this work uses support vector machine (SVM) models trained using historical process data as an approximation of the feasible region for each process module. The feasibility analysis approach used in this work provides a way to integrate production level details in the supply chain optimization problem. The product demand is considered to be uncertain with a known probability distribution. The resulting problem is a two-stage stochastic mixed integer programming problem where binary variables represent facility locations, integer variables represent the number of modular equipment installed, and continuous variables represent the material flow in the network. The problem is reformulated as a large-scale mixed integer linear programming model by representing probability distribution in terms of a finite number of scenarios. The problem of computational complexity for the large-scale problem is reduced by using Benders decomposition. Multiperiod decisions are implemented using a rolling horizon optimization approach. The formulation also accounts for risk-averse decision-making with the help of downside risk. The resulting problem is a multiobjective optimization problem where the objectives of minimizing the total cost as well as minimizing the downside risk are simultaneously achieved. A set of pareto optimal solutions is achieved by solving the problem to optimality. The results also demonstrate that modular designs provide flexibility to adjust processing capacity as the product demand increases over the planning horizon.

The chapter is organized as follows. The supply chain optimization model is presented in section 6.2. The relevant background regarding various details of the formulation is provided in section 6.3. The formulation is solved for two case studies, and the results are presented in section 6.4. Section 6.5 provides a summary and concluding remarks.

6.2 Supply chain model

The supply chain network considered in this work is generic and is motivated by our previous work [188]. The network consists of suppliers, production facilities, warehouses, and retailers. The demand is realized at each retailer in each time period. The product or raw material flow is allowed using all possible pathways i.e. all suppliers can send material to all production facilities, all production facilities can send material to all warehouses, and all warehouses can send material to all retailers. It is important to note that the production details are taken into account, and the production is carried out using modular processes. This results in a key difference between traditional supply chain optimization formulations and the formulation presented in this work. Since the problem aims to find the number of modules used at each production facility, integer variables are introduced as opposed to binary variables, which limit the information to a yes or no decision. A typical supply chain network is shown in Figure 5-1.

The decisions to be made by optimizing this network include the choice of a production facility for production in each time period, the number and the type of modules to be installed at each production site, the amount of material transported between suppliers, production facilities, warehouses, and retailers.

6.2.1 Optimization problem formulation

Based on the notations described in Appendix B an MILP model is formulated.

Objective. Minimizing the total cost is the primary objective of this formulation. The total cost consists of transportation cost, inventory cost, capital cost, and operating cost as shown by Eq. (119)

$$
\begin{aligned}
\text{minimize cost} = &\sum_{m\in\mathcal{M}}\sum_{o\in\mathcal{O}_m}\zeta_{fmo}z_{mo} \\
&+\mathbb{E}(\sum_{t\in\mathcal{T}}(\sum_{f\in\mathcal{F}}\sum_{w\in\mathcal{W}}\sum_{p\in P}Q_{fwpti}\,c_{fp} \\
&+\sum_{f\in\mathcal{F}}\sum_{w\in\mathcal{W}}\sum_{p\in P}Q_{fwpti}\,h_{fwp} \\
&+\sum_{s\in\mathcal{S}}\sum_{f\in\mathcal{F}}\sum_{a\in\mathcal{A}}Q_{sfati}\,h_{sfa}+\sum_{w\in\mathcal{W}}\sum_{r\in\mathcal{R}}\sum_{p\in P}Q_{wrpti}\,h_{wrp} \\
&+\sum_{w\in\mathcal{W}}\sum_{p\in\mathcal{P}}I_{wpti}g_{wp}+\sum_{r\in\mathcal{R}}\sum_{p\in\mathcal{P}}B_{rpti}b_{rp}))
\end{aligned}
\qquad (119)
$$

where, Q_{fwpti} is the quantity of the product p delivered from production facility f to warehouse w in time period t of scenario i; , Q_{sfati} is the quantity of the raw material a delivered from supplier s to production facility f in time period t of scenario i; Q_{wrpti} is the quantity of product p delivered from warehouse w to retailer r in time period t of scenario i; c_{fp} operating cost per unit of product p at production facility f; h_{fwp} is transportation cost per unit of product p from production facility f to warehouse w; h_{sfa} transportation cost per unit of raw material a from supplier s to production facility f; h_{wrp} is transportation cost per unit of product p from warehouse w to retailer r; g_{wp} is inventory cost for storing a unit of product p for one time period at warehouse w; ζ_{fmo} is the capital cost for installing one unit of option o for module m at a production facility f; z_{mo} is the number of units of option $o \in \mathcal{O}_m$ for module $m \in \mathcal{M}$ installed across all production facilities; I_{wpti} is the inventory of product $p \in \mathcal{P}$ stored at warehouse w during time period t of scenario i; B_{rpti} is the unmet demand for product p at retailer r in time period t of scenario i; b_{rp} is the penalty per unit of unmet demand of product p at retailer r.

Inventory balance

Inventory balance must be satisfied at each warehouse at each time period. Warehouse inventory at each time period is the difference between the amount of product they ship out and the amount of product they receive added to the warehouse inventory at the end of the previous planning period, as shown by Eq. (120). At the beginning of the first time period, the initial inventory I_{wp0} is known and introduced as a model parameter.

$$I_{wpti} = I_{wp(t-1)i} + \sum_{f \in \mathcal{F}} \sum_{w \in \mathcal{W}} Q_{fwpti} - \sum_{w \in \mathcal{W}} \sum_{r \in \mathcal{R}} Q_{wrpti} \tag{120}$$

$$\forall w \in \mathcal{W}, \forall p \in \mathcal{P}, \forall t \in \mathcal{T}, \forall i \in SC$$

where I_{wpti} is the inventory of product $p \in \mathcal{P}$ stored at warehouse w during time period t of scenario i. It is assumed that there is no inventory at the production facilities. As a result, no reorder policy is incorporated at the warehouses.

Demand satisfaction

The difference between the product demand and the amount of product delivered to the retailer is unmet demand. This is formulated by Eq.(121)

$$\delta_{rpti} - \sum_{w \in \mathcal{W}} Q_{wrpti} = B_{rpti} \qquad \forall r \in \mathcal{R}, \forall p \in \mathcal{P}, \forall t \in \mathcal{T}, \forall i \in SC \tag{121}$$

where, δ_{rpti} is the demand for product p at retailer r in time period t of scenario i; B_{rpti} is the unmet demand of product p in time period t in scenario i at retailer r. The service level is then calculated using Eq. (122).

$$SL_i = 1 - \frac{\sum_t \sum_p \sum_r B_{rpti}}{\sum_t \sum_p \sum_r \delta_{rpti}} \tag{122}$$

Downside risk

The downside risk is defined using Eq. (123-125).

$$\psi_i = \Omega - SL_i; \psi_i \geq 0 \tag{123}$$

$$DR = \sum_i \pi_i \psi_i \tag{124}$$

$$DR \leq \epsilon \tag{125}$$

where, Ω is the target service level; ψ_i is the difference in service level for scenario i and the target service level; DR is the expected downside risk; ϵ is the risk tolerance parameter specified by the user; π_i is the probability of scenario i.

Feasibility constraints

The production feasibility should be ensured at each production facility during each time period. This is accomplished with the help of machine learning-based feasibility models, as shown in Eq. (126).

$$SVM_{of}(\boldsymbol{x_{tfi}}) \geq -K(1 - y_{fo}) \qquad \forall o \in \mathcal{O}_m; \forall m \in \mathcal{M}; \forall t \in \mathcal{T}; \forall f \in \mathcal{F} \tag{126}$$

where, $SVM_{of}(\boldsymbol{x_{tfi}})$ represents SVM models for option o at production facility f; K is the big-M constant for the feasibility constraints; y_{fo} are binary variables indicating if option $o \in \mathcal{O}_m$ for module $m \in \mathcal{M}$ is installed at production facility f; $\boldsymbol{x_{tfi}}$ is material flow per production line during time period t at production facility f in scenario i. The value of K is obtained empirically for each SVM model, since the upper and lower bounds on the variables are known over which a particular SVM model is applicable. A large number of samples are generated and the value of $SVM_{of}(\boldsymbol{x_{tf}})$ is calculated, and K is chosen to be larger than the maximum of all values. The constraint indicates that when y_{fo} is 1, $SVM_{of}(\boldsymbol{x_{tf}})$ should be positive, indicating a feasible point.

Moreover, it is assumed that each processing line at a production facility is subject to equal throughput as shown by Eq. (127)

$$\boldsymbol{x_{tfi}} v_f = \sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} Q_{sfati} \qquad \forall t \in \mathcal{T}; \forall f \in \mathcal{F}, \forall i \in SC \tag{127}$$

where, v_f corresponds to the number of production lines at production facility f; $\boldsymbol{x_{tfi}}$ is the raw material flow per production line during time period t at production facility f for scenario i. Additionally, the process variables should be in agreement with the chosen module option as enforced by Eq. (128-129)

$$lb_o - x_{tfi} \leq H(1 - y_{fo}) \qquad \forall o \in \mathcal{O}_m; \forall m \in \mathcal{M}; \forall t \in \mathcal{T}; \forall f \in \mathcal{F} \tag{128}$$

$$x_{tfi} - ub_o \leq H(1 - y_{fo}) \qquad \forall o \in \mathcal{O}_m; \forall m \in \mathcal{M}; \forall t \in \mathcal{T}; \forall f \in \mathcal{F} \tag{129}$$

where, ub_o is a vector of upper bounds on the variable x for option o for module m; lb_o is vector of lower bounds on the variable x for option o for module m; H is a big-M constant.

It is assumed that production facilities do not have any storage capacity for products, and thus all the material produced at production facilities is transported to warehouses for storage. The total amount of products manufactured at the production facilities is the summation of the product manufactured at each parallel processing line as shown by Eq. (130)

$$\boldsymbol{x_{tfi}} v_f = \sum_{f \in \mathcal{F}} \sum_{p \in P} Q_{fwpti} \qquad \forall t \in \mathcal{T}; \forall w \in \mathcal{W}, \forall i \in SC \tag{130}$$

where, $\boldsymbol{x_{tfi}}$ is the product flow per production line during time period t at production facility f for scenario i.

Logical constraints

The number of active processing lines at a given production facility in each time period is evaluated using Eq. (131). This is required for constraint given by Eq. (127) to model the throughput to each processing line.

$$v_f = \sum_{o \in \mathcal{O}_m} q_{fo} \qquad f \in \mathcal{F} \tag{131}$$

q_{fo} is the number of production lines of option $o \in \mathcal{O}_m$ for module $m \in \mathcal{M}$ at production facility f. The constraint shown by Eq.(132) is needed to evaluate z_{mo}.

$$z_{mo} = \sum_{f \in \mathcal{F}} q_{fo} \qquad \forall o \in \mathcal{O}_m; \forall m \in \mathcal{M} \tag{132}$$

The relation between integer variables q_{fo} and binary variables y_{fo} is established with the help of constraints given by Eq. (133-134).

$$y_{fo} \leq q_{fo} \quad f \in \mathcal{F}, \forall o \in \mathcal{O}_m; \forall m \in \mathcal{M} \tag{133}$$

$$q_{fo} \leq v_{\max} y_{fo} \qquad f \in \mathcal{F}, \forall o \in \mathcal{O}_m; \forall m \in \mathcal{M} \tag{134}$$

It is assumed that the total number of units of each module active at a production facility are the same for all modules. This is modeled using Eq. (135).

$$\sum_{o \in \mathcal{O}_{\bar{m}}, \bar{m} \in \mathcal{M}} q_{fo} = \sum_{o \in \mathcal{O}_m, m \in \mathcal{M} \backslash \{\bar{m}\}} q_{fo} \qquad f \in \mathcal{F} \tag{135}$$

It is assumed that in each time period, only one module option can be installed at a production facility as shown by Eq. (136)

$$\sum_{o \in \mathcal{O}_m, m \in M} y_{fo} \leq 1 \qquad \forall t \in \mathcal{T}, f \in \mathcal{F} \tag{136}$$

where, $Q_{fwpti}, Q_{sfati}, Q_{wrpti}, I_{wpti}, x_{tfi} \in \mathbb{R}; z_{mo}, v_f, q_{fo} \in \mathbb{Z}; y_{fo} \in \{0,1\};$ $Q_{fwpti} \geq 0; Q_{sfati} \geq 0, Q_{wrpti} \geq 0; 0 \leq I_{wpti} \leq \bar{I}_{wp}; 0 \leq x_{tfi}.$

It is assumed that the optimal production facilities selected by solving the optimization problem are installed at the beginning of the planning horizon.

It is important to note that the constraints given by Eq.(127) and Eq.(130) contain additional bilinear terms. The bilinear terms, in this case, are multiplications of a continuous variable and integer variables. Since McCormick relaxations are accurate at the boundaries or endpoints, it is beneficial to express integer variables in terms of binary variables. This is achieved by first rewriting Eq. (127) and Eq. (130) as Eq.(137) and Eq.(138), respectively.

$$\sum_{j \in J} g_{tfji} = \sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} Q_{sfati} \tag{137}$$

$$\sum_{j \in J} b_{tfji} = \sum_{f \in \mathcal{F}} \sum_{p \in P} Q_{fwpti} \tag{138}$$

where, g_{tfji} and b_{tfji} are intermediate variables which can be expressed as shown by Eq.(139) and Eq.(140), respectively.

$$g_{tfji} = \boldsymbol{x_{tfi}} j u_{fj} \tag{139}$$

$$b_{tfji} = \boldsymbol{x_{tfi}} j u_{fj} \tag{140}$$

The integer variables ν_{tf} are expanded using binary variables u_{tfj}, as shown by Eq.(141) and Eq.(142).

$$\sum_{j \in J} u_{fj} = 1 \quad J \in \{0, \dots, N\} \tag{141}$$

where, N is the upper bound on integer variables ν_f. For each integer value from 0 to N, a binary variable u_{fj} is defined. Constraint, given by Eq. (141), states that only one of the binary variables can be 1. Finally, the integer variables and the binary variables are related through Eq. (142) which ensures that if u_{fj} is 1, ν_f is j.

$$\sum_{j \in J} j u_{fj} = \nu_f \quad J \in \{0, \dots, N\} \tag{142}$$

The constraints shown by Eq.(139) and Eq.(140) are then relaxed using McCormick relaxations as discussed in Section 5.3.2. The relaxations for Eq.(139) are shown by Eq. (143-146)

$$g_{tfji} \geq x_{tf}^{L} j u_{fj} \tag{143}$$

$$g_{tfji} \geq x_{tf}^{U} j u_{fj} + x_{tfi} j - x_{tf}^{U} j \tag{144}$$

$$g_{tfji} \leq x_{tf}^{U} ju_{fj} \tag{145}$$

$$g_{tfji} \leq x_{tfi}j + x_{tf}^{L} ju_{fj} - x_{tf}^{L} j \tag{146}$$

where, x_{tf}^{L} and x_{tf}^{U} represent lower and upper bounds on the continuous variables that represent the material flow rate of the raw materials, respectively. The lower bound for binary variables is 0, and the upper bound is 1. However, since the variable being relaxed is ju_{fj}, the upper bound is replaced by j. Similarly, the relaxations for Eq.(140) are shown by Eq.(147-150).

$$b_{tfji} \geq x_{tf}^{L} u_{fj} \tag{147}$$

$$b_{tfji} \geq x_{tf}^{U} u_{fj} + x_{tfi}j - x_{tf}^{U} j \tag{148}$$

$$b_{tfji} \leq x_{tf}^{U} ju_{fj} \tag{149}$$

$$b_{tfji} \leq x_{tfi}j + x_{tf}^{L} ju_{fj} - x_{tf}^{L} j \tag{150}$$

where, x_{tf}^{L} and x_{tf}^{U} represent lower and upper bounds on the continuous variables that represent the material flow rate of the products, respectively. Since the relaxations shown by Eq. (143-150) are with respect to the binary variable u_j, the relaxation will lead to the optimal solution of the original problem.

6.3 Background

6.3.1 Risk measures

The general stochastic optimization problem minimizes the total expected cost, which results in a solution that is optimal on the average of all scenarios. As a result, this solution is risk-neutral. In the case of a real problem, however, decision-makers are usually risk-averse and are interested in simultaneously minimizing the risk. To address this problem, literature consists of several risk metrics as well as extensions of existing stochastic programming approaches to the problem minimizing risk. Metrics based on variance, variability index, probabilistic risk, and downside risk exist in the literature [233]. Since variance and variability index shift the solution towards higher expected cost, and probabilistic risk results in a computationally demanding problem due to the presence of binary variables [233], downside risk is used in this work. The downside risk is defined as shown by Eq. (151)

$$\min DR(x,\Omega) = \sum_{s \in SC} p_s \psi_s \tag{151}$$

where, $\psi_{sc} \geq \Omega - SL_{sc}; \psi_{sc} \geq 0 \;\; \forall sc \in SC.$

Downside risk in this work is defined in terms of the service level as opposed to profit or total cost as has been used in the existing literature. As a result, Ω is the target service level; SL_{sc} is the service level for scenario sc.

This is considered as a separate objective in the optimization problem considered in this work ϵ-constrained method is used.

6.3.2 Two-stage stochastic optimization

In this work, a two-stage stochastic optimization approach is used for addressing product demand uncertainties in the supply chain. In a two-stage optimization framework, decisions related to raw material transportation, production, product distribution, and inventory for the current planning period are made, whereas decisions associated with the future planning periods are postponed until uncertain demands are revealed. Decisions for the current planning period, also known as 'here and now' decisions are made in the first stage, whereas the decisions for future periods, also known as ‘wait and see’ decisions are incorporated in the second stage. The classical two-stage programming problem is described as shown by Eq. (152) and Eq. (153)

$$\min_{x\in X}\{g(x) \coloneqq c^T x + \mathbb{E}[Q(x,\xi)]\}$$

$$Ax \geq b \tag{152}$$

where, $Q(x,\xi)$ is the optimal value of the second stage problem which is represented as

$$\min_{y} q^T y \text{ subject to } Wy \geq h - Tx. y \in \mathcal{Y} \tag{153}$$

where, x represents first-stage decision vector, X is a polyhedral set, defined by a finite number of linear constraints,y is the second-stage decision vector, and $\xi = (q, T, W, h)$ is the second stage data. The two-stage optimization problem given by Eq. (152) and Eq. (153) can be equivalently represented by a large-scale linear programming problem where the random vector with a known probability distribution can be represented by a finite number of scenarios, as shown by Eq. (154).

$$\min c^T x + \sum_{s\in SC} \pi_s q_s^T y_s$$

$$Ax \geq b \tag{154}$$

$$T_s x + W_s y_s \geq h \;\; \forall s \in SC$$

$$x \in X, y \in \mathcal{Y}$$

The number of scenarios is chosen to address the tradeoff between the accuracy of the distribution, and computational complexity of the problem. If too few scenarios are included to approximate the underlying demand distribution, the representation may not be accurate. On the other hand, if a large number of scenarios are included, the resulting optimization problem is a computationally expensive problem. Even though including constraints with respect to each scenario increases the size of the problem significantly, the resulting problem has a nice structure that allows decomposition into subproblems that are small, and that can be solved independently [234].

The problem considered in this work involves integer variables and, thus, results in a stochastic mixed integer programming problem. The solution approach for stochastic mixed integer programming problems broadly depends on the type of integer variables (general integers or binary) as well as the stage in which integer variables are present [234]. For the problem considered in this work, all the general integer and binary variables are present only in the first stage, and therefore, classical Benders decomposition[235] can be applied.

6.3.3 Benders decomposition

Even though the problem presented in Eq. (154) is large in size, it has a very nice structure that can be utilized. It can be noted that for a given first-stage vector x, the problem given by Eq.(154) can be decomposed for each scenario in the set SC as shown by Eq. (155).

$$\begin{aligned} &\eta_s(x) := \min q_s^T y_s \\ &W_s y_s \geq h_s - T_s x \\ &y_s \geq 0 \end{aligned} \tag{155}$$

In Eq.(155), since the right-hand side in the constraints can be parametrized by x, the problems given by η_S are parametric linear programs. Therefore, the problem given by Eq.(154) can be reformulated, as shown by Eq. (156).

$$\min c^T x + \sum_{s \in SC} \pi_s \eta_s(x)$$

$$\text{s.t.} Ax \geq b \tag{156}$$

$$x \in X$$

As the feasible region of the problem given by Eq. (156) changes with x, we make use of the dual form, as shown by Eq. (157).

$$\eta_s(x) = \max(h_s - T_s x)^T \psi_s$$

$$W_s^T \psi_s \leq q_s \tag{157}$$

$$\psi_s \geq 0$$

With this formulation, it is ensured that only the objective function of the problem changes with changing x, and the feasible region stays the same. Thus we can obtain Benders reformulation as shown by Eqs.(158-162)

$$\min c^T x + \eta \tag{158}$$

$$Ax \geq b \tag{159}$$

$$\eta \geq \sum_{s} \pi_s (h_s - T_s x)^T \psi_s^i \quad \forall i \in I_s, s \in SC \tag{160}$$

$$0 \geq (h_s - T_s x)^T \left(s_s^j\right) \forall j \in J_s, s \in SC \tag{161}$$

$$x \in X \tag{162}$$

where, ψ_s^i is the set of extreme points of the polyhedron formed by the constraints in problem given by Eq. (157); s_s^j is the set of extreme rays of that polyhedron. It can be noted that I_s and J_s are finite. Eq. (160) and Eq.(161) are known as optimality cuts and feasibility cuts, respectively. The resulting problem has many constraints, but fewer variables, as the expected second-stage cost is captured by a single variable η. This formulation is solved by first adding a subset of inequalities and then gradually adding constraints as they are violated. In this work, the implementation of Benders decomposition provided with the CPLEX mixed integer linear programming solver is used.

6.3.4 Rolling horizon optimization

As product demands are important input parameters for a multiperiod supply chain optimization problem, a common approach is to make full use of available information, even if the information in future planning periods is currently uncertain. Moreover, uncertainty in the product demand decreases as the future periods come closer. Rolling horizon optimization is an approach to benefit from this effect where decisions are made dynamically over the stochastic programming problem. The idea is to make decisions at each discrete decision points given a fixed planning horizon. The decisions are implemented from the current decision point to the next decision point. The state of the system at the end of the first optimization problem is passed on to the next optimization problem that starts from the next decision point. This procedure is repeated until the end of the planning horizon. Rolling horizon optimization approach provides significant computational advantages as the problem is temporally decomposed into problems with less number of planning periods and therefore, fewer scenarios. Rolling horizon optimization approach is not new and has been applied to multiperiod optimization problems in a wide variety of supply chain optimization problems [236] [237]. A schematic of the rolling horizon approach is provided in Figure 6-1.

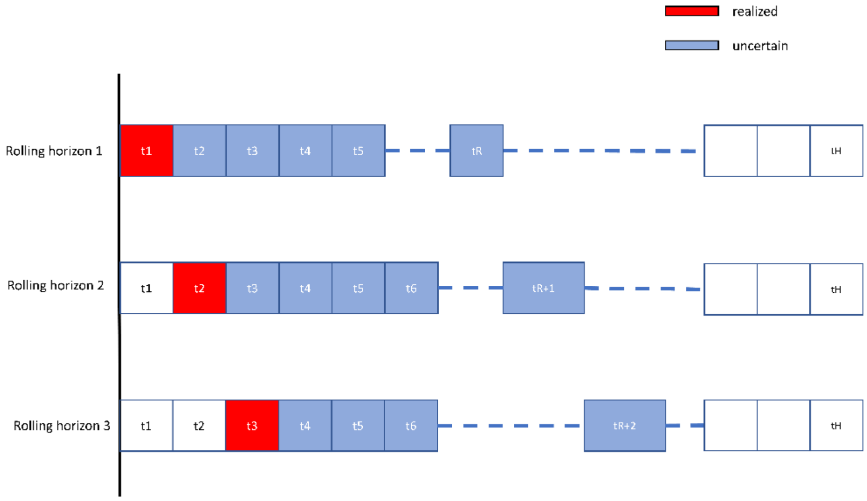


Figure 6-1 Rolling horizon optimization approach

At the beginning of the planning horizon, demand information is available for the first planning period, and the demand till the end of the rolling horizon is uncertain and available only in the form of a random distribution. The expectation and variance of the distribution are predicted using the demand forecast. The first stage problem is solved, and the decisions for the current period are implemented. At the beginning of the second time period, the demand for the second time period is revealed, and the rolling horizon shifts accordingly. The first stage problem is solved again, and the procedure is repeated until the end of the planning horizon. In this work, since the number of modules installed are general integer variables, the implemented solution for the number of modules installed till the current planning period is used as the lower bounds in the subsequent rolling horizon.

6.4 Results

In this section, a case study of a supply chain network consisting of a modular process that includes a system of reactor and separator is considered. The optimization model shown in section 6.2.1 is solved for two cases. The aim is to demonstrate the rolling horizon optimization approach for the problem of optimization under demand uncertainty. In this respect, the optimal decisions regarding optimal facility location, material flows, and process design are demonstrated. Moreover, the multiobjective optimization problem of minimizing the downside risk is solved using the ϵ- constrained method. The results show a set of pareto optimal designs that simultaneously minimize the supply chain total cost as well as the downside risk. Finally, the advantage of modular designs in allowing the possibility of numbering-up of process units is demonstrated.

Problem definition

To demonstrate the idea, a sample superstructure of the supply chain network is considered, as shown in Figure 5-1. The network consists of three retailers and two warehouses and two suppliers and three production facilities to choose from. In each time period, demand is realized by the retailers. The arrows in Figure 5-1 show the allowable material flow in the supply chain network. The problem is to minimize the total cost of the supply chain, as shown by Eq. (119). The parameters such as distances between all possible combinations of entities in the supply chain superstructure, inventory holding costs at warehouses, and operating costs at the production sites are known a priori. It is important to note that the network shown in Figure 5-1 represents a superstructure of all possible options for the optimal design of the supply chain network. The optimal design choice will also select the location of the entities in the supply chain network.

At the production facilities, the production is carried out through a process consisting of a continuously stirred tank reactor in series with an ideal separator. The process is modularized into reactor and separator modules. The aim is to convert raw material A into two finished products B and E, as shown in Figure 4-2. An isothermal liquid-phase reaction is considered following the kinetic mechanism as described in the previous studies of Rooney and Biegler [198] and Goyal and Ierapetritou [169]. The model equations for the process are shown by Eq. (163).

$$\begin{gathered}
F_{A0} - x_A F(1-\alpha) - VC_{A0}(k_1 + k_2)x_A = 0 \\
-Fx_B(1-\alpha) + VC_{A0}k_1 x_A = 0 \\
-Fx_C + VC_{A0}(k_2 x_A - (k_3 + k_4)x_C + k_5 x_E) = 0 \\
-Fx_D(1-\beta) + VC_{A0}k_3 x_C = 0 \\
-Fx_E(1-\beta) + VC_{A0}(k_4 x_C - k_5 x_D) = 0 \\
x_A + x_B + x_C + x_D + x_E - 1 = 0
\end{gathered} \tag{163}$$

where, x_A, x_B, x_C, x_D, and x_E represent the mole fraction of components A, B, C, D, and E, respectively; k_i are the rate constants; V is the volume of the reactor; C_{A0} is the inlet concentration of A; α is the recycle fraction of stream A and B; β is the recycle fraction of D and E; F is the molar flow rate at the outlet of the reactor; F_{A0} is the molar flow rate at the inlet of the reactor. The nominal values of the kinetic constants are $k_1 = 0.0374$, $k_2 = 0.0195$, $k_3 = 0.0165$, $k_4 = 0.2701$, and $k_5 = 0.0261$.

It is assumed that four reactor design options are available based on their volume. Different separator design options are available depending on the throughput that they can handle. The available options and the respective costs for the reactor and the separator are shown in Table 6-1.

Table 6-1: Design options for reactor and separator

Options	**Reactor (m^3)**	C_r **(k$)**	**Separator** **(F_{A0}mol/h)**	C_s **(k$)**
Option 1	5	400	30-50	300
Option 2	20	850	40-70	720

The problem is to determine the optimal supply chain network that minimizes the total cost of the supply chain over the planning horizon. The decisions include the following. i) location of production facilities ii) selection of module options to define a process iii) optimal material flow iv) optimal choice of suppliers v) Feasibility of satisfying the product demand. It is assumed that the coefficient of mass production is known for the modules under consideration. Since the problem involves optimization under demand uncertainty, the objective is also to minimize the downside risk for risk-averse decisions. Optimal solutions are obtained for different values of risk tolerances in order to demonstrate the tradeoff between the conflicting objectives of minimizing the total cost of the supply chain and minimizing the downside risk.

Solution methodology

For solving this problem, it is assumed that the feasibility constraints are available a priori in the form of support vector machine models. These constraints are incorporated in the optimization framework, as shown in section 6.2.1. For model details on the approach to obtain the feasibility constraints, the reader is directed to previous work in the literature [225]. In this work, the data is generated using simulations in GAMS 31.1.0 and solved as a nonlinear program using Baron mixed integer nonlinear programming solver version 20.4.14. SVM models are built using the data from simulations is used. Scikit-learn python toolbox is used to build the classifiers. The validation results for the classifiers are shown in Table 6-2.

Table 6-2: SVM model validation for the reactor using a linear kernel

Option	CF%	CIF%	NC%	Total Error
Option 1	100	100	0	0
Option 2	86.36	97.53	9.52	4.85

A general supply chain network, as shown in Figure 5-1, is used to demonstrate the efficacy of the proposed approach. Two case studies varying in network size, as well as the planning horizon, are used.

6.4.1 Case study 1

The supply chain consists of 2 suppliers, 3 production sites, 3 warehouses, and 3 retailers. Product demand is realized at the beginning of each time period. The demand in future periods is unknown. However, a forecast of the demand distribution is available in the form of expected demand and the demand variability. In general, the product demand in the periods close to the current planning period is less uncertain, and therefore, the demand variability is less. However, since there is more uncertainty related to the demands in the periods much later than the current period, the demand variability is more. The effect of time-dependent demand variability is modeled using a linear model to predict future demand[238]. In a real scenario, the parameters for the linear model are obtained using historical data. In this work, it is assumed that the parameters are available. Expected product demand is considered to be increasing at a constant rate at each planning period. The rate of increase in demand is assumed to be 10% for this case study. In the first study, a planning horizon of 6 periods is considered with each period consisting of 4 months and the total planning horizon of 2 years.

The optimization is carried out using a rolling horizon optimization approach with the rolling horizon period of 6 periods. The rolling horizon optimization framework is implemented in Matlab, and the optimization problems are solved with 0 optimality gap in GAMS using CPLEX mixed integer linear optimization solver on a PC with Intel® Xeon® CPU E-2174G @ 3.80GHz and 32.0 GB RAM in an average time of 169 s. Latin Hypercube Sampling is carried out to design the scenarios. Latin hypercube sampling has a space-filling property that allows covering the probability distribution accurately with a fewer number of scenarios. In this case study, 50 scenarios are generated. The problem is solved for six different values of the risk tolerance ϵ. The results for the expected supply chain cost and risk tolerance are presented in Figure 6-2. It can be observed that as ϵ increases, total cost reduces. In this case, ϵ is the tolerance towards downside risk with respect to the service level. A higher value of ϵ indicates the possibility of sustaining the supply chain network at a lower service level. As a result, the solutions with higher values of ϵ can include lower capital, operating, and transportation costs. For the risk tolerances of 0.1 and 0.15, the total cost is the same which can be attributed to the tradeoff between risk tolerance and backorder cost. If the risk tolerance is high, lower service levels can be maintained. However, this will result in increased value of backorder cost. Finally, based on the desired level of risk tolerance, an optimal design can be chosen that minimizes the total cost.

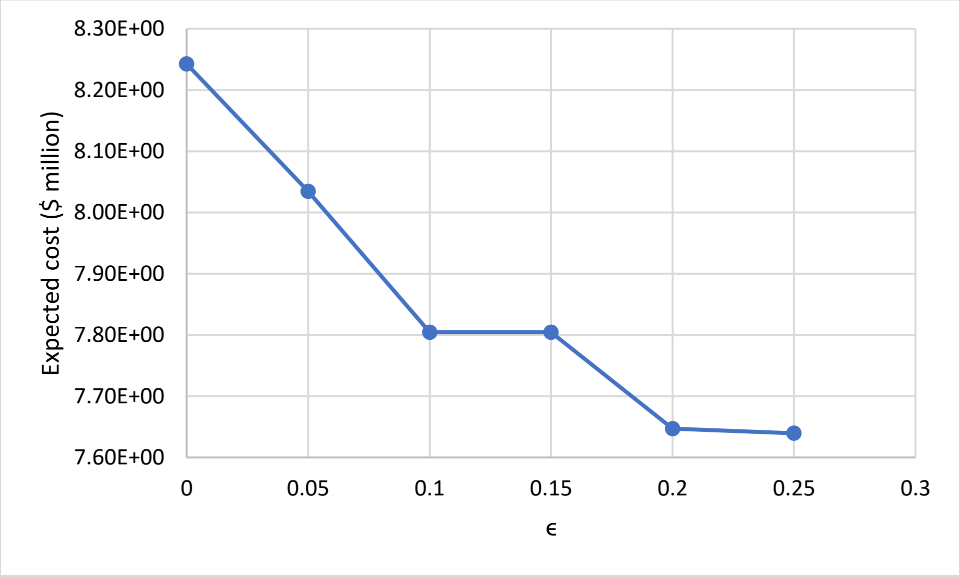


Figure 6-2 Pareto optimal curve for total cost and downside risk

This tradeoff can be observed by looking at the solutions at the extreme values of $\epsilon = 0$ and that for $\epsilon = 0.25$. For $\epsilon = 0$, the total cost value obtained is $8.24 million, whereas, for $\epsilon = 0.25$, the total cost is $7.64 million. When the risk tolerance is lower, service levels need to be kept high, and as a result, capital cost for $\epsilon = 0$ is $6.9 million, which is higher than the capital cost of $4.6 million for $\epsilon = 0.25$. On the other hand, the worst-case scenario for the service level for $\epsilon = 0.25$ is 0.73, whereas that for $\epsilon = 0$ is 0.86 indicating that the solution for $\epsilon = 0$ is risk averse.

For the value of $\epsilon = 0.1$, the complete rolling horizon solution at the end of the planning horizon is reported in Table 6-3. The optimal costs at the end of the planning horizon are reported in Table 6-4. It can be observed that all production facilities are included in the optimal solution. However, different number of production lines are installed at each production facility.

Table 6-3 optimal choice of production sites and modules at the end of planning horizon for $\epsilon = 0.2$

Production site	Reactor module		Separator module	
	Module	Number of units	Module	Number of units

1	2	1	1	1
2	2	2	1	2
3	2	2	1	2

Table 6-4 Optimal cost at the end of rolling horizon optimization

Cost	**$**
Total cost	5907033
Transportation cost	684079
Capital cost	5750000
Operating cost	242823
Inventory cost	2529
Backorder cost	56242

6.4.2 Case study 2

It should be noted that considering the rolling horizon of the same length as the planning horizon may not always be practical. For the problems with a longer planning horizon, this approach could lead to a significantly computationally expensive problem. Moreover, the product demands in the periods far ahead in the future may not be accurate. In the second case study presented in this work, a larger supply chain network consisting of 4 suppliers, 4 production sites, 3 warehouses, and 5 markets. The planning horizon, in this case, is 5 years consisting of 20 planning periods of 3 months each. It is assumed that demand increases by 3 percent in every time period. The rolling horizon length is considered to consist of three time periods. The target service level is set as 0.95, and the downside risk tolerance is set to be 0.1. The problem is solved with 2% optimality gap in GAMS using CPLEX mixed integer linear optimization solver on a PC with Intel® Xeon® CPU E-2174G @ 3.80GHz and 32.0 GB RAM. The optimal solution is reported in Table 6-5.

Table 6-5 optimal choice of production sites and modules at the end of planning horizon for $\epsilon = 0.1$

Production site	Reactor module		Separator module	
	Module	Number of units	Module	Number of units
1	2	2	1	2
2	1	3	1	3
3	2	4	1	4
4	1	1	1	1

As opposed to centralized large-scale manufacturing, modular design provides the possibility of numbering-up of equipment to meet the desired demand. As a result, more units can be installed at a later planning period saving on the initial investment as well as providing additional flexibility. This can be better illustrated with the help of Figure 6-3, where an increase in the capital cost can be observed at multiple periods in the planning horizon. The optimal solution at the end of the first planning period for $\epsilon = 0.1$ is reported in Table 6-5. It can be observed that all four production sites are chosen for production in the optimal solution. Both reactor options and the separator option 1 is chosen. The total number of processing lines of the chosen optimal module options in the supply chain is 10. The total cost at the end of the planning horizon is reported in Table 6-6.

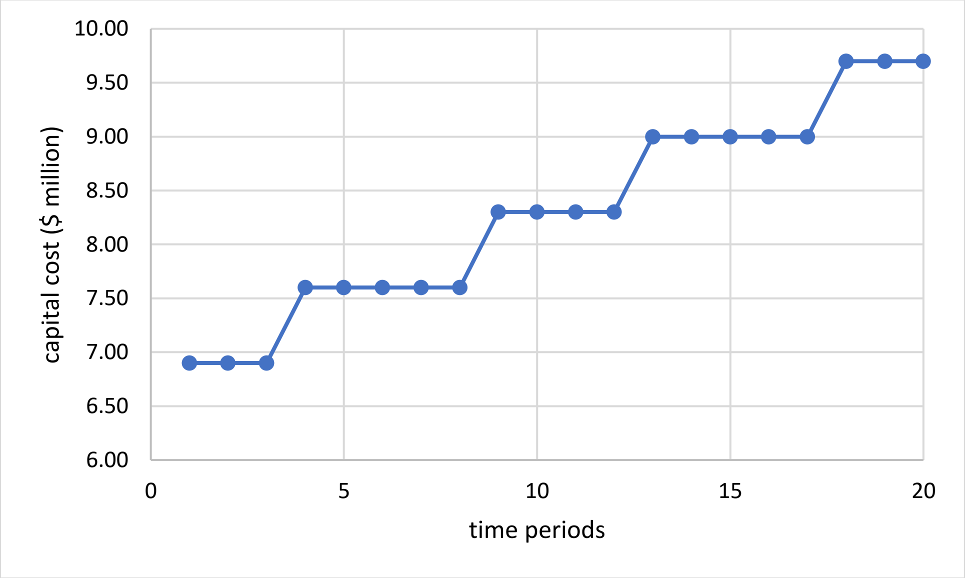


Figure 6-3 Capital cost vs. time periods for $\epsilon = 0.2$

Table 6-6 Optimal cost at the end of the planning horizon

Cost	**$**
Total cost	16083529
Transportation cost	3048872
Capital cost	9700000
Operating cost	1210436
Inventory cost	1234131
Backorder cost	211879

6.5 Conclusions

In this work, the problem of modular supply chain optimization under uncertainty is considered. A stochastic mixed integer linear programming formulation is first presented where general integers represent the number of process units, binary variables represent facility locations, and continuous variables represent material flow in the supply chain. Since the problem is a multiperiod problem and the uncertainties associated with the product demands are not resolved in the present planning period, the resulting problem is a two-stage mixed integer stochastic programming problem. The second stage problem is represented in the form of a finite number of scenarios that represent the demand distribution resulting in a large-scale mixed integer linear programming problem. The resulting computational complexity is addressed with the help of Benders decomposition. The objectives of downside risk and supply chain total costs are minimized simultaneously using multiobjective optimization. The optimization formulation presented in this work allows for simultaneous process design and feasibility analysis. Data-driven approximations of the feasible region are considered to be available that are incorporated in the optimization formulation. With respect to process design, the optimal solution aims at choosing from a small set of standardized module options varying in production capacity and the capital cost. The formulation is applied to a case study of a supply chain consisting of a modular process consisting of two modules. The solution leads to the optimal selection of the type and number of process modules, production facility location, and at the same time, minimizing downside risk. Multiobjective optimization framework consisting of ϵ-constraint for the downside risk is incorporated, using which a set of solutions on the pareto optimal curve can be obtained. It is noted that as opposed to a centralized large-scale processing module, modular designs provide flexibility to number-up production units in order to achieve the desired throughput. The results demonstrate that as the product demand increases in the future, more processing units are installed. This characteristic of the modular processes provides a promising way to reduce initial investment and to maintain a flexible supply chain with respect to market volatility. In the future, this approach provides an interesting opportunity for supply chains where the raw material availability, as well as the product demand, is geographically distributed. Due to reduced costs from design standardization as well as from the flexibility of increasing or reducing the production capacity, new supply chain designs can quantitatively assess the benefits of modular processes.

7 Summary and future work

In this dissertation, machine learning-based frameworks are proposed to address the problems in supply chain optimization. In the first problem, data obtained from the supply chain simulations are used to determine optimal inventory allocation in a multienterprise supply chain network. It was noted that the resulting problem has a particular characteristic, which is the discontinuity of the objective function. To optimize in the presence of discontinuities, SVM models are used first to isolate distinct continuous regions. The results demonstrate that considering discontinuities leads to better results in terms of computational cost as well as the solution quality when compared to a few of the existing state-of-the-art algorithms.

Further, the problems related to modular manufacturing are considered where historical data about individual process modules is available. This data is used to build a classification model that is later incorporated into the design optimization problem as well as the supply chain optimization model. While considering the design optimization, the objectives of minimizing the total cost as well as maximizing the design flexibility are considered, and a pareto set of solutions is obtained. For the modular supply chain optimization, it is demonstrated that integrating process-level details in the supply chain optimization can provide added flexibility to the decision-maker. Using the proposed formulation, the problem of optimal facility location, optimal process design, and optimal material flow in the supply chain network can be obtained. At the same time, tradeoffs between centralized vs. distributed manufacturing as well as capital cost benefits due to design standardization can be quantified. Finally, the problem is extended to the case of demand uncertainty. A two-stage mixed integer stochastic programming problem is formulated and solved using a rolling horizon optimization approach to address the problem of multiperiod planning. While designing the supply chain under demand uncertainty, risk measures are incorporated that facilitate risk-averse decision making based on the risk tolerance of the decision-maker. The resulting solution simultaneously minimizes risk as well as the total expected cost of the supply chain network.

This work has the potential to be extended in several directions.

- For the multienterprise supply chain optimization problem, future work can consider the environmental impact of the supply chain as well as use the agent-based modeling framework to study the effect of environmental policies on the cost and the environmental impact of the supply chain. In addition to optimizing inventory allocation in the supply chain, future work can address the optimization of inventory policy where decisions such as the reorder amount and the reorder policy will be made. The optimization framework presented in this work is not specific to the supply chain application. In the future, more problems such as parameter estimation can be considered. Moreover, the approach presented for isolation of continuous regions can be readily used to model a discontinuous response.
- The design optimization framework considered in this work can be generalized to process synthesis problems where different technologies or even different connections in a flow sheet can be compared. Additional objectives, such as environmental impact, can be considered in the design stage using the proposed approach.
- For the supply chain optimization problem, the investigation of computationally efficient decomposition techniques will help the problem to be scaled to larger supply chain networks and longer planning horizons.
- In the future, the frameworks presented in this work provide an exciting opportunity for supply chains where the raw material availability, as well as the product demand, is geographically distributed, such as biomass supply chains.

- For the problem of optimization under demand uncertainty, future work can consider the possibility of mobile production units that can be moved to different production facilities based on the geographic distribution of the demand and the availability of the raw materials.
- The approach presented for integration of process design and supply chain optimization can be broadened to consider product design where the optimization framework can choose from an array of product portfolio based on the composition and availability of the raw materials as well as the demand for products.
- The integration of feasibility analysis and optimization presented in this work opens new and challenging problems from an optimization perspective. Tailor-made mixed integer nonlinear programming formulations that can handle nonlinear classifiers as constraints will improve the applicability of this approach.

8 Acknowledgement of previous publications

Several chapters of the thesis are published or being prepared for publications. The publications are modified to avoid redundancy and improve readability. The following publications are acknowledged.

Chapter 2 has been partly extracted from the following publication

Bhosekar, A., Ierapetritou, M.: Advances in surrogate based modeling, feasibility analysis, and optimization: A review. Comput. Chem. Eng. 108, 250–267 (2018). doi:10.1016/j.compchemeng.2017.09.017

Chapter 3 has been published under the citation

Bhosekar, A., Ierapetritou, M.: A discontinuous derivative-free optimization framework for multi-enterprise supply chain. Optim. Lett. (2019). doi:10.1007/s11590-019-01446-5

Chapter 4 has been published under the citation

Bhosekar, A., Ierapetritou, M.: Modular Design Optimization using Machine Learning-based Flexibility Analysis. J. Process Control. 90, 18–34 (2020).

Chapter 5 is under review under the citation

Bhosekar, A., Ierapetritou, M.: A framework for supply chain optimization for modular manufacturing with production feasibility analysis. Comput. Chem. Eng.

Chapter 6 is being prepared under the citation

Bhosekar, A., Ierapetritou, M.: Supply chain optimization for modular manufacturing with production feasibility analysis under uncertainty.

Bibliography

1. Garcia, D.J., You, F.: Supply chain design and optimization: Challenges and opportunities. Comput. Chem. Eng. 81, 153–170 (2015). doi:10.1016/j.compchemeng.2015.03.015

2. Grossmann, I.: Enterprise-wide optimization: A new frontier in process systems engineering. AIChE J. 51, 1846–1857 (2005). doi:10.1002/aic.10617

3. Hicks, C., Hines, S.A., Harvey, D., McLeay, F.J., Christensen, K.: An Agent Based Model of Supply Chains. In: Proceedings of the 12th European Simulation Multiconference on Simulation - Past, Present and Future. pp. 609–613. SCS Europe (1998)

4. Manataki, A., Chen-Burger, Y.-H., Rovatsos, M.: Towards Improving Supply Chain Coordination through Agent-Based Simulation. In: Demazeau, Y., Dignum, F., Corchado, J.M., Bajo, J., Corchuelo, R., Corchado, E., Fernández-Riverola, F., Julián, V.J., Pawlewski, P., and Campbell, A. (eds.) Trends in Practical Applications of Agents and Multiagent Systems. pp. 217–224. Springer Berlin Heidelberg, Berlin, Heidelberg (2010)

5. Aglave, R., Lusty, J., Nixon, J.: Using Simulation and Digitalization for Modular Process Intensification. Chem. Eng. Prog. 115, 45–49 (2019)

6. Weber, T.: Modular Plants by modularization and standardization –. 36 (2016)

7. Vlachos, D., Ierapetritou, M., Dauenhauer, P., Hock, A.: Nsf Modular Manufacturing Workshop. (2017)

8. Kim, Y., Park, L.K., Yiacoumi, S., Tsouris, C.: Modular Chemical Process Intensification: A Review. Annu. Rev. Chem. Biomol. Eng. 8, 359–380 (2017). doi:10.1146/annurev-chembioeng-060816-101354

9. Baldea, M., Edgar, T.F., Stanley, B.L., Kiss, A.A.: Modular manufacturing processes: Status, challenges, and opportunities. AIChE J. 63, 4262–4272 (2017). doi:10.1002/aic.15872

10. Bhosekar, A., Ierapetritou, M.: Advances in surrogate based modeling, feasibility analysis, and optimization: A review. Comput. Chem. Eng. 108, 250–267 (2018). doi:10.1016/j.compchemeng.2017.09.017

11. Chen, Q., Grossmann, I.E.: Recent Developments and Challenges in Optimization-Based Process Synthesis. (2017)

12. Halemane, K.P., Grossmann, I.E.: Optimal process design under uncertainty. AIChE J. 29, 425–433 (1983). doi:10.1002/aic.690290312

13. Zhao, F., Zheng, C., Zhang, S., Zhu, L., Chen, X.: Quantification of process flexibility via space projection. AIChE J. 1–15 (2019). doi:10.1002/aic.16706

14. Palys, M.J., Allman, A., Daoutidis, P.: Exploring the Benefits of Modular Renewable-Powered Ammonia Production: A Supply Chain Optimization Study. Ind. Eng. Chem. Res. 58, 5898–5908 (2019). doi:10.1021/acs.iecr.8b04189

15. Tian, Y., Demirel, S.E., Hasan, M.M.F., Pistikopoulos, E.N.: An overview of process systems engineering approaches for process intensification: State of the art. Chem. Eng. Process. - Process Intensif. 133, 160–210 (2018). doi:10.1016/j.cep.2018.07.014

16. Allen, R.C., Allaire, D., El-halwagi, M.M.: Capacity Planning for Modular and Transportable Infrastructure for Shale Gas Production and Processing. Ind. Eng. Chem. Res. 58, 5887–5897 (2018). doi:10.1021/acs.iecr.8b04255

17. Arora, A., Li, J., Zantye, M.S., Hasan, M.M.F.: Design standardization of unit operations for reducing the capital intensity and cost of small-scale chemical processes. AIChE J. 1–14 (2019). doi:10.1002/aic.16802

18. Furnival, G.M., Wilson, R.W.: Regressions by Leaps and Bounds. Technometrics. 16, 499–511 (1974)

19. Konno, H., Yamamoto, R.: Choosing the best set of variables in regression analysis using integer programming. J. Glob. Optim. 44, 273–282 (2009). doi:10.1007/s10898-008-9323-9

20. Bertsimas, D., King, A.: OR Forum—An Algorithmic Approach to Linear Regression. Oper. Res. 64, 2–16 (2016). doi:10.1287/opre.2015.1436

21. Liu, H., Motoda, H.: Computational Methods of Feature Selection (Chapman & Hall/Crc Data Mining and Knowledge Discovery Series). Chapman & Hall/CRC (2007)

22. Konno, H., Yamamoto, R.: Choosing the best set of variables in regression analysis using integer programming. J. Glob. Optim. 44, 273–282 (2009). doi:10.1007/s10898-008-9323-9

23. Park, Y.W., Klabjan, D.: Subset Selection for Multiple Linear Regression via Optimization. 1–27 (2013)

24. Mallows, C.L.: Some Comments on C_P. Technometrics. 15, 661–675 (1973). doi:10.2307/1267380

25. Akaike, H.: A New Look at the Statistical Model Identification. IEEE Trans. Automat. Contr. 19, 716–723 (1974). doi:10.1109/TAC.1974.1100705

26. Schwarz, G.: Estimating the Dimension of a Model. Ann. Stat. 6, 461–464 (1978)

27. Hannan, E.J., Quinn, B.G.: The Determination of the Order of an Autoregression. J. R. Stat. Soc. Ser. B. 41, 190–195 (1979)

28. Foster, D.P., George, E.I.: The Risk Inflation Criterion for Multiple Regression. Ann. Stat. 22, 1947–1975 (1994)

29. Wilson, Z.T., Sahinidis, N. V.: The ALAMO approach to machine learning. Comput. Chem. Eng. (2017). doi:10.1016/j.compchemeng.2017.02.010

30. Cozad, A., Sahinidis, N. V, Miller, D.C.: Learning Surrogate Models for Simulation-Based Optimization. 60, (2014). doi:10.1002/aic

31. Linhart, H., Zucchini, W.: Model Selection. John Wiley &amp; Sons, Inc., New York, NY, USA (1986)

32. Hurvich, C.M., Tsai, C.-L.: Regression and Time Series Model Selection in Small Samples. Biometrika. 76, 297–307 (1989). doi:10.2307/2336663

33. Efron, B., Hastie, T., Johnstone, I., Tibshirani, R.: Least Angle Regression. Ann. Stat. 32, 407–

499 (2004)

34. Candes, E., Tao, T.: The Dantzig selector: Statistical estimation when p is much larger than n. Ann. Stat. 35, 2313–2351 (2007). doi:10.1214/009053606000001523

35. Fan, J., Lv, J.: Sure independence screening for ultrahigh dimensional feature space. J. R. Stat. Soc. Ser. B Stat. Methodol. 70, 849–911 (2008). doi:10.1111/j.1467-9868.2008.00674.x

36. Cadima, J., Cerdeira, J.O., Minhoto, M.: Computational aspects of algorithms for variable selection in the context of principal components. Comput. Stat. Data Anal. 47, 225–236 (2004). doi:10.1016/j.csda.2003.11.001

37. Chen, Z.: Extended Bayesian Information Criteria for Model Selection with Large Model Spaces model selection Extended Bayesian information criteria for with large model spaces. 95, 759–771 (2014). doi:10.1093/biomet/asnO34

38. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning. Elements. 1, 337–387 (2009). doi:10.1007/b94608

39. Seber, G.A.F., Lee, A.J.: Linear regression analysis. Wiley-Interscience, (2003)

40. Zou, H.: The Adaptive Lasso and Its Oracle Properties. J. Am. Stat. Assoc. 101, 1418–1429 (2006). doi:10.1198/016214506000000735

41. Breiman, L.: Better subset regression using the nonnegative garrote, http://amstat.tandfonline.com/doi/abs/10.1080/00401706.1995.10484371, (1995)

42. Smola, A.J., Schölkopf, B.: A tutorial on support vector regression. Stat. Comput. 14, 199–222 (2004). doi:10.1023/B:STCO.0000035301.49549.88

43. Clarke, S.M., Griebsch, J.H., Simpson, T.W.: Analysis of Support Vector Regression for Approximation of Complex Engineering Analyses. J. Mech. Des. 127, 1077 (2005). doi:10.1115/1.1897403

44. Forrester, A.I.J., Keane, A.J.: Recent advances in surrogate-based optimization. Prog. Aerosp. Sci. 45, 50–79 (2009). doi:10.1016/j.paerosci.2008.11.001

45. Björkman, M., Holmström, K.: Global Optimization of Costly Nonconvex Functions Using Radial Basis Functions. Optim. Eng. 1, 373–397 (2000). doi:10.1023/A:1011584207202

46. Gutmann, H.-M.: A Radial Basis FunctionMethod for Global Optimization. J. Glob. Optim. 19, 201–227 (2001). doi:10.1023/A:1011255519438

47. Regis, R.G., Shoemaker, C.A.: Improved strategies for radial basis function methods for global optimization. J. Glob. Optim. 37, 113–135 (2007). doi:10.1007/s10898-006-9040-1

48. Chen, H., Loeppky, J.L., Sacks, J., Welch, W.J.: Analysis Methods for Computer Experiments: How to Assess and What Counts? Stat. Sci. 31, 40–60 (2013). doi:10.1214/15-STS531

49. Joseph, V.R., Hung, Y., Sudjianto, A.: Blind kriging: a new method for developing metamodels. J. Mech. Des. 130, 1–8 (2008). doi:10.1115/1.2829873

50. Huang, H.-C., Chen, C.-S.: Optimal Geostatistical Model Selection. J. Am. Stat. Assoc. 102, 1009–1024 (2007). doi:10.1198/016214507000000491

51. Fan, J., Li, R.: Variable Selection via Nonconcave Penalized. J. Am. Stat. Assoc. 96, 1348–1360 (2001). doi:10.1198/016214501753382273

52. Zou, H., Li, R.: One-step sparse estimates in nonconcave penalized likelihood models, (2008)

53. Chu, T., Zhu, J., Wang, H.: Penalized maximum likelihood estimation and variable selection in geostatistics. Ann. Stat. 39, 2607–2625 (2011). doi:10.1214/11-AOS919

54. Furrer, R., Genton, M.G., Nychka, D.: Covariance Tapering for Interpolation of Large Spatial Datasets. J. Comput. Graph. Stat. 15, 502–523 (2006). doi:10.1198/106186006X132178

55. Kaufman, C.G., Schervish, M.J., Nychka, D.: Covariance Tapering for Likelihood-Based Estimation in Large Spatial Data Sets. J. Am. Stat. Assoc. 103, 1545–1555 (2008). doi:10.1198/016214508000000959

56. Ranjan, P., Haynes, R., Karsten, R.: A Computationally Stable Approach to Gaussian Process Interpolation of Deterministic Computer Simulation Data. Technometrics. 53, 366–378 (2011). doi:10.1198/TECH.2011.09141

57. Toal, D.J.J., Bressloff, N.W., Keane, A.J., Holden, C.M.E.: The development of a hybridized particle swarm for kriging hyperparameter tuning. Eng. Optim. 43, 675–699 (2011). doi:10.1080/0305215X.2010.508524

58. Toal, D.J.J., Forrester, A.I.J., Bressloff, N.W., Keane, A.J., Holden, C.: An adjoint for likelihood maximization. Proc. R. Soc. A Math. Phys. Eng. Sci. 465, 3267–3287 (2009). doi:10.1098/rspa.2009.0096

59. Rasmussen, C.E., Williams, C.: Gaussian Processes for Machine Learning. MIT Press (2006)

60. Lophaven, S., Nielsen, H., Søndergaard, J.: Aspects of the matlab toolbox DACE. Tech. Rep. (2002)

61. Bouhlel, M., Bartoli, N., Otsmane, A., Bouhlel, M., Bartoli, N., Otsmane, A., Morlier, J.: Improving kriging surrogates of high-dimensional design models by Partial Least Squares dimension reduction To cite this version : Improving kriging surrogates of high-dimensional design models by Partial Least Squares dimension reduction. Struct. Multidiscip. Optim. 935–952 (2015). doi:10.1007/s00158-015-1395-9

62. Bachoc, F.: Cross validation and maximum likelihood estimations of hyper-parameters of Gaussian processes with model misspecification. Comput. Stat. Data Anal. 66, 55–69 (2013). doi:10.1016/j.csda.2013.03.016

63. Cressie, N., Johannesson, G.: Fixed rank kriging for very large spatial data sets. J. R. Stat. Soc. Ser. B (Statistical Methodol. 70, 209–226 (2008). doi:10.1111/j.1467-9868.2007.00633.x

64. Nychka, D., Bandyopadhyay, S., Hammerling, D., Lindgren, F., Sain, S.: A Multiresolution Gaussian Process Model for the Analysis of Large Spatial Datasets. J. Comput. Graph. Stat. 24, 579–599 (2015). doi:10.1080/10618600.2014.914946

65. Banerjee, S., Gelfand, A.E., Finley, A.O., Sang, H.: Gaussian predictive process models for large spatial data sets. J. R. Stat. Soc. Ser. B Stat. Methodol. 70, 825–848 (2008).

doi:10.1111/j.1467-9868.2008.00663.x

66. Liang, F., Cheng, Y., Song, Q., Park, J., Yang, P.: A resampling-based stochastic approximation method for analysis of large geostatistical data. J. Am. Stat. Assoc. 108, 325–339 (2013). doi:10.1080/01621459.2012.746061

67. Snelson, E., Ghahramani, Z.: Local and global sparse Gaussian process approximations. In: Meila, M. and Shen, X. (eds.) Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics (AISTATS-07). pp. 524–531. Journal of Machine Learning Research - Proceedings Track (2007)

68. Sang, H., Huang, J.: A full scale approximation of covariance functions for large spatial data sets. J. R. Stat. Soc. …. 111–132 (2012). doi:10.1111/j.1467-9868.2011.01007.x

69. Tajbakhsh, S., Aybat, N., Castillo, E. Del: Sparse Precision Matrix Selection for Fitting Gaussian Random Field Models to Large Data Sets. arXiv Prepr. arXiv1405.5576. 25, 1–18 (2014). doi:10.1007/s10107-014-0826-5.

70. Zerpa, L.E., Queipo, N. V., Pintos, S., Salager, J.-L.: An optimization methodology of alkaline–surfactant–polymer flooding processes using field scale numerical simulation and multiple surrogates. J. Pet. Sci. Eng. 47, 197–208 (2005). doi:10.1016/j.petrol.2005.03.002

71. Goel, T., Haftka, R.T., Shyy, W., Queipo, N. V.: Ensemble of surrogates. Struct. Multidiscip. Optim. 33, 199–216 (2007). doi:10.1007/s00158-006-0051-9

72. Müller, J., Piché, R.: Mixture surrogate models based on Dempster-Shafer theory for global optimization problems. J. Glob. Optim. 51, 79–104 (2011). doi:10.1007/s10898-010-9620-y

73. Viana, F.A.C., Haftka, R.T., Watson, L.T.: Efficient global optimization algorithm assisted by multiple surrogate techniques. J. Glob. Optim. 56, 669–689 (2013). doi:10.1007/s10898-012-9892-5

74. Vapnik, V.N.: The Nature of Statistical Learning Theory. Springer-Verlag, Berlin, Heidelberg (1995)

75. Cristianini, N., Shawe-Taylor, J.: An Introduction to Support Vector Machines: And Other Kernel-based Learning Methods. Cambridge University Press, New York, NY, USA (2000)

76. Kressel, U.H.-G.: Advances in Kernel Methods. Presented at the (1999)

77. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine Learning in {P}ython. J. Mach. Learn. Res. 12, 2825–2830 (2011)

78. Jebara, T.: Multi-task feature and kernel selection for SVMs. Proceedings, Twenty-First Int. Conf. Mach. Learn. ICML 2004. 433–440 (2004). doi:10.1145/1015330.1015426

79. Hooke, R., Jeeves, T.A.: " Direct Search " Solution of Numerical and Statistical Problems *. 212–229 (1960)

80. Nelder, J.A., Mead, R.: A Simplex Method for Function Minimization. Comput. J. 7, 308–313 (1965). doi:10.1093/comjnl/7.4.308

81. Jones, D.R., Perttunen, C.D., Stuckman, B.E.: Lipschitzian optimization without the Lipschitz constant. J. Optim. Theory Appl. 79, 157–181 (1993). doi:10.1007/BF00941892

82. Rios, L.M., Sahinidis, N. V.: Derivative-free optimization: A review of algorithms and comparison of software implementations. J. Glob. Optim. 56, 1247–1293 (2013). doi:10.1007/s10898-012-9951-y

83. Powell, M.J.D.: A Direct Search Optimization Method That Models the Objective and Constraint Functions by Linear Interpolation. In: Gomez, S. and Hennart, J.-P. (eds.) Advances in Optimization and Numerical Analysis. pp. 51–67. Springer Netherlands, Dordrecht (1994)

84. Conn, a. R., Scheinberg, K., Toint, P.L.: Recent progress in unconstrained nonlinear optimization without derivatives. Math. Program. 79, 397–414 (1997). doi:10.1007/BF02614326

85. Powell, M.J.D.: The NEWUOA software for unconstrained optimization without derivatives. In: Di Pillo, G. and Roma, M. (eds.) Large-Scale Nonlinear Optimization. pp. 255–297. Springer US, Boston, MA (2006)

86. Oeuvray, R., Bierlaire, M.: Boosters: A Derivative-Free Algorithm Based on Radial Basis Functions. Int. J. Model. Simul. 29, 26–36 (2009). doi:10.1080/02286203.2009.11442507

87. Wild, S.M., Regis, R.G., Shoemaker, C.A.: ORBIT: Optimization by Radial Basis Function Interpolation in Trust-Regions. SIAM J. Sci. Comput. 30, 3197–3219 (2008). doi:10.1137/070691814

88. Regis, R.G., Wild, S.M.: CONORBIT : Constrained optimization by radial basis function interpolation in trust regions 1 CONORBIT : Constrained optimization by radial basis function interpolation in trust regions. (2015)

89. Wild, S.M., Shoemaker, C.: Global Convergence of Radial Basis Function Trust-Region Algorithms for Derivative-Free Optimization. SIAM Rev. 55, 349–371 (2013). doi:10.1137/120902434

90. Regis, R.G.: Trust regions in Kriging-based optimization with expected improvement. Eng. Optim. 48, 1037–1059 (2016). doi:10.1080/0305215X.2015.1082350

91. Tawarmalani, M., Sahinidis, N. V.: A polyhedral branch-and-cut approach to global optimization. Math. Program. 103, 225–249 (2005). doi:10.1007/s10107-005-0581-8

92. Misener, R., Floudas, C.A.: ANTIGONE: Algorithms for coNTinuous / Integer Global Optimization of Nonlinear Equations. J. Glob. Optim. 59, 503–526 (2014). doi:10.1007/s10898-014-0166-2

93. Jones, D.R., Schonlau, M., Welch, W.J.: Efficient Global Optimization of Expensive Black-Box Functions. J. Glob. Optim. 13, 455–492 (1998). doi:10.1023/A:1008306431147

94. Holmström, K., Quttineh, N.H., Edvall, M.M.: An adaptive radial basis algorithm (ARBF) for expensive black-box mixed-integer constrained global optimization. Optim. Eng. 9, 311–339 (2008). doi:10.1007/s11081-008-9037-3

95. Regis, R.G., Shoemaker, C.A.: Constrained global optimization of expensive black box

functions using radial basis functions. J. Glob. Optim. 31, 153–171 (2005). doi:10.1007/s10898-004-0570-0

96. Boukouvala, F., Floudas, C.A.: ARGONAUT : AlgoRithms for Global Optimization of coNstrAined grey-box compUTational problems. Optim. Lett. (2016). doi:10.1007/s11590-016-1028-2

97. Boukouvala, F., Ierapetritou, M.G.: Feasibility analysis of black-box processes using an adaptive sampling Kriging-based method. Comput. Chem. Eng. 36, 358–368 (2012). doi:10.1016/j.compchemeng.2011.06.005

98. Caballero, J., Grossmann, I.E.: surrogate models in modular flowsheet optimization. AICHE J. 61, 857–866 (2015). doi:10.1002/aic

99. Wang, Z., Ierapetritou, M.: A novel feasibility analysis method for black-box processes using a radial basis function adaptive sampling approach. AIChE J. n/a-n/a (2016). doi:10.1002/aic.15362

100. Banerjee, I., Pal, S., Maiti, S.: Computationally efficient black-box modeling for feasibility analysis. Comput. Chem. Eng. 34, 1515–1521 (2010). doi:10.1016/j.compchemeng.2010.02.016

101. Boukouvala, F., Ierapetritou, M.G.: Derivative-free optimization for expensive constrained problems using a novel expected improvement objective function. AIChE J. 60, 2462–2474 (2014). doi:10.1002/aic.14442

102. Swaney, R.E., Grossmann, I.E.: An index for operational flexibility in chemical process design. Part I: Formulation and theory. AIChE J. 31, 621–630 (1985). doi:10.1002/aic.690310412

103. 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, 239–245 (1979). doi:10.1080/00401706.1979.10489755

104. Sobol, I.M.: The distribution of points in a cube and the approximate evaluation of integrals. Zh. Vychisl. Mat. i Mat. Fiz. 7. 7, 784–802 (1967). doi:10.1016/0041-5553(67)90144-9

105. Provost, F., Jensen, D., Oates, T.: Efficient Progressive Sampling. In: Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 23–32. ACM, New York, NY, USA (1999)

106. Garud, S.S., Karimi, I.A., Kraft, M.: Smart Sampling Algorithm for Surrogate Model Development. Comput. Chem. Eng. 96, 103–114 (2017). doi:10.1016/j.compchemeng.2016.10.006

107. Eason, J., Cremaschi, S.: Adaptive sequential sampling for surrogate model generation with artificial neural networks. Comput. Chem. Eng. 68, 220–232 (2014). doi:10.1016/j.compchemeng.2014.05.021

108. Crombecq, K., Laermans, E., Dhaene, T.: Efficient space-filling and non-collapsing sequential design strategies for simulation-based modeling. Eur. J. Oper. Res. 214, 683–696 (2011). doi:10.1016/j.ejor.2011.05.032

109. Bischl, B., Mersmann, O., Trautmann, H., Weihs, C.: Resampling Methods for Meta-Model Validation with Recommendations for Evolutionary Computation. Evol. Comput. 20, 249–275 (2012). doi:10.1162/EVCO_a_00069

110. Kersting, K., Plagemann, C., Pfaff, P., Burgard, W.: Most Likely Heteroscedastic Gaussian Process Regression. 24th Int. Conf. Mach. Learn. (ICML 2007). 393–400 (2007). doi:10.1145/1273496.1273546

111. Boukouvalas, A., Cornford, D.: Learning Heteroscedastic Gaussian Processes for Complex Datasets. Group. 44, (2009)

112. Yin, J., Ng, S.H., Ng, K.M.: Kriging metamodel with modified nugget-effect: The heteroscedastic variance case. Comput. Ind. Eng. 61, 760–777 (2011). doi:10.1016/j.cie.2011.05.008

113. Viana, F.A.C., Haftka, R.T., Steffen, V.: Multiple surrogates: How cross-validation errors can help us to obtain the best predictor. Struct. Multidiscip. Optim. 39, 439–457 (2009). doi:10.1007/s00158-008-0338-0

114. Dias, L.S., Ierapetritou, M.G.: Data-driven feasibility analysis for the integration of planning and scheduling problems. Springer US (2019)

115. Ryu, J.H., Dua, V., Pistikopoulos, E.N.: A bilevel programming framework for enterprise-wide process networks under uncertainty. Comput. Chem. Eng. 28, 1121–1129 (2004). doi:10.1016/j.compchemeng.2003.09.021

116. Zamarripa, M.A., Aguirre, A.M., Méndez, C.A., Espu, A.: Chemical Engineering Research and Design Mathematical programming and game theory optimization-based tool for supply chain planning in cooperative / competitive environments. 1, 1588–1600 (2013). doi:10.1016/j.cherd.2013.06.008

117. Yeh, K., Whittaker, C., Realff, M.J., Lee, J.H.: Two stage stochastic bilevel programming model of a pre-established timberlands supply chain with biorefinery investment interests. Comput. Chem. Eng. 73, 141–153 (2015). doi:10.1016/j.compchemeng.2014.11.005

118. Yue, D., You, F.: Stackelberg-game-based modeling and optimization for supply chain design and operations: A mixed integer bilevel programming framework. Comput. Chem. Eng. 102, 81–95 (2017). doi:10.1016/j.compchemeng.2016.07.026

119. Florensa, C., Garcia-Herreros, P., Misra, P., Arslan, E., Mehta, S., Grossmann, I.E.: Capacity planning with competitive decision-makers: Trilevel MILP formulation, degeneracy, and solution approaches. Eur. J. Oper. Res. 262, 449–463 (2017). doi:10.1016/j.ejor.2017.04.013

120. Köchel, P., Nieländer, U.: Simulation-based optimisation of multi-echelon inventory systems. Int. J. Prod. Econ. 93–94, 505–513 (2005). doi:10.1016/j.ijpe.2004.06.046

121. Ye, W., You, F.: A computationally efficient simulation-based optimization method with region-wise surrogate modeling for stochastic inventory management of supply chains with general network structures. Comput. Chem. Eng. 87, 164–179 (2016). doi:10.1016/j.compchemeng.2016.01.015

122. Sahay, N., Ierapetritou, M.: Supply chain management using an optimization driven

simulation approach. AIChE J. 59, 4612–4626 (2013). doi:10.1002/aic.14226

123. Swaminathan, J., Smith, S.F., Sadeh, N.M.: Modeling Supply Chain Dynamics : A Multiagent Approach. Decis. Sci. 29, 607–632 (1998). doi:10.1111/j.1540-5915.1998.tb01356.x

124. Lee, J.H., Kim, C.O.: Multi-agent systems applications in manufacturing systems and supply chain management: A review paper. Int. J. Prod. Res. 46, 233–265 (2008). doi:10.1080/00207540701441921

125. García-Flores, R., Wang, X.Z.: A multi-agent system for chemical supply chain simulation and management support. OR Spectr. 24, 343–370 (2002). doi:10.1007/s00291-002-0099-x

126. Julka, N., Srinivasan, R., Karimi, I., Srinivasan, R.: Agent-based supply chain management*/ 1: framework. Comput. Chem. Eng. 26, 1755–1769 (2002). doi:10.1016/S0098-1354(02)00150-3

127. Julka, N., Karimi, I., Srinivasan, R.: Agent-based supply chain management—2: a refinery application. Comput. Chem. Eng. 26, 1771–1781 (2002). doi:10.1016/S0098-1354(02)00151-5

128. Singh, A., Chu, Y., You, F.: Biorefinery supply chain network design under competitive feedstock markets: An agent-based simulation and optimization approach. Ind. Eng. Chem. Res. 53, 15111–15126 (2014). doi:10.1021/ie5020519

129. Sahay, N., Ierapetritou, M.: Multienterprise supply chain: Simulation and optimization. AIChE J. 62, 3392–3403 (2016). doi:10.1002/aic.15399

130. Anderson, J.M.: Modelling Step Discontinuous Functions using Bayesian emulation ., (2017)

131. Gorodetsky, A.A., Marzouk, Y.M.: Efficient Localization of Discontinuities in Complex Computational Simulations. (2014). doi:10.1137/140953137

132. Jakeman, J.D., Archibald, R., Xiu, D.: Characterization of discontinuities in high-dimensional stochastic problems on adaptive sparse grids. J. Comput. Phys. 230, 3977–3997 (2011). doi:10.1016/j.jcp.2011.02.022

133. Jakeman, J.D., Narayan, A., Xiu, D.: Minimal multi-element stochastic collocation for uncertainty quantification of discontinuous functions. J. Comput. Phys. 242, 790–808 (2013). doi:10.1016/j.jcp.2013.02.035

134. Archibald, R., Gelb, A., Saxena, R., Xiu, D.: Discontinuity detection in multivariate space for stochastic simulations. J. Comput. Phys. 228, 2676–2689 (2009). doi:10.1016/j.jcp.2009.01.001

135. Caiado, C.C.S., Goldstein, M.: Bayesian uncertainty analysis for complex physical systems modelled by computer simulators with applications to tipping points. Commun. Nonlinear Sci. Numer. Simul. 26, 123–136 (2015). doi:10.1016/j.cnsns.2015.02.006

136. Moreau, L., Aeyels, D.: Optimization of discontinuous functions: a generalized theory of differentiation. SIAM J. Control Optim. 11, 53–69 (2000). doi:10.1137/S1052623499354679

137. Vicente, L.N., Custódio, A.L.: Analysis of direct searches for discontinuous functions. Math. Program. 133, 299–325 (2012). doi:10.1007/s10107-010-0429-8

138. Boursier Niutta, C., Wehrle, E.J., Duddeck, F., Belingardi, G.: Surrogate modeling in design optimization of structures with discontinuous responses. Struct. Multidiscip. Optim. 57, 1857–1869 (2018). doi:10.1007/s00158-018-1958-7

139. Sahay, N., Ierapetritou, M.: Multienterprise supply chain: Simulation and optimization. AIChE J. 62, 3392–3403 (2016). doi:10.1002/aic.15399

140. Gimpel, H., Mäkiö, J., Weinhardt, C.: Multi-attribute double auctions in financial trading. Proc. - Seventh IEEE Int. Conf. E-Commerce Technol. CEC 2005. 2005, 366–369 (2005). doi:10.1109/ICECT.2005.61

141. Steiglitz, K., Honig, M.L., Cohen, L.M.: A Computational Market Model Based on Individual Action. In: Market-based Control. pp. 1–27. World Scientific Publishing Co., Inc., River Edge, NJ, USA (1996)

142. Morris, M.D., Mitchell, T.J.: Exploratory designs for computational experiments. J. Stat. Plan. Inference. 43, 381–402 (1995). doi:10.1016/0378-3758(94)00035-T

143. Gerstner, T., Griebel, M.: Numerical integration using sparse grids. Numer. Algorithms. 18, 209–232 (1998). doi:10.1023/A:1019129717644

144. Barthelmann, V., Novak, E., Ritter, K.: High dimensional polynomial interpolation on sparse grids. Adv. Comput. Math. 12, 273–288 (2000). doi:10.1023/A:1018977404843

145. Kieslich, C.A., Boukouvala, F., Floudas, C.A.: Optimization of black-box problems using Smolyak grids and polynomial approximations. J. Glob. Optim. (2018). doi:10.1007/s10898-018-0643-0

146. Le Digabel, S.: Algorithm 909: NOMAD: Nonlinear Optimization with the MADS Algorithm. ACM Trans. Math. Softw. 37, 44:1--44:15 (2011). doi:10.1145/1916461.1916468

147. Abramson, M.A., Audet, C., Couture, G., Dennis, J.E., Digabel, S. Le: The Nomad project. http://www.gerad.ca/nomad, (2009)

148. Currie, J., Wilson, D.I.: OPTI: Lowering the Barrier Between Open Source Optimizers and the Industrial MATLAB User. In: Foundations of Computer-Aided Process Operations. , Savannah, Georgia, USA (2012)

149. Runarsson, T.P., Yao, X.: Search biases in constrained evolutionary optimization. IEEE Trans. Syst. Man Cybern. Part C Appl. Rev. 35, 233–243 (2005). doi:10.1109/TSMCC.2004.841906

150. Johnson, S.G.: The NLopt nonlinear-optimization package, https://github.com/stevengj/nlopt

151. Viana, F.A.C.: SURROGATES Toolbox User's Guide", version 2.1, (2010)

152. Lier, S., Grünewald, M.: Net Present Value Analysis of Modular Chemical Production Plants. Chem. Eng. Technol. 34, 809–816 (2011). doi:10.1002/ceat.201000380

153. Yang, M., You, F.: Modular methanol manufacturing from shale gas: Techno-economic and

environmental analyses of conventional large-scale production versus small-scale distributed, modular processing. AIChE J. 64, 495–510 (2018). doi:10.1002/aic.15958

154. Sánchez, A., Martín, M.: Scale up and scale down issues of renewable ammonia plants: Towards modular design. Sustain. Prod. Consum. (2018). doi:10.1016/j.spc.2018.08.001

155. Seifert, T., Sievers, S., Bramsiepe, C., Schembecker, G.: Small scale, modular and continuous: A new approach in plant design. Chem. Eng. Process. Process Intensif. 52, 140–150 (2012). doi:10.1016/j.cep.2011.10.007

156. Floudas, C. a., Gümüş, Z.H.: Global Optimization in Design under Uncertainty: Feasibility Test and Flexibility Index Problems. Ind. Eng. Chem. Res. 40, 4267–4282 (2001). doi:10.1021/ie001014g

157. Lima, F. V, Jia, Z., Ierapetritou, M., Georgakis, C.: Similarities and differences between the concepts of operability and flexibility: The steady-state case. AIChE J. 56, 702–716 (2010). doi:10.1002/aic.12021

158. Swaney, R.E., Grossmann, I.E.: An index for operational flexibility in chemical process design. Part II: Computational algorithms. AIChE J. 31, 631–641 (1985). doi:10.1002/aic.690310413

159. Grossmann, I.E., Floudas, C.A.: Active constraint strategy for flexibility analysis in chemical processes. Comput. Chem. Eng. 11, 675–693 (1987). doi:10.1016/0098-1354(87)87011-4

160. Bansal, V., Perkins, J.D., Pistikopoulos, E.N.: Flexibility analysis and design of linear systems by parametric programming. AIChE J. 46, 335–354 (2000). doi:10.1002/aic.690460212

161. Floudas, C.A., Gümüş, Z.H.: Global Optimization in Design under Uncertainty: Feasibility Test and Flexibility Index Problems. Ind. Eng. Chem. Res. 40, 4267–4282 (2001). doi:10.1021/ie001014g

162. Goyal, V., Ierapetritou, M.G.: Determination of operability limits using simplicial approximation. AIChE J. 48, 2902–2909 (2002). doi:10.1002/aic.690481217

163. Pulsipher, J.L., Zavala, V.M.: A mixed-integer conic programming formulation for computing the flexibility index under multivariate gaussian uncertainty. Comput. Chem. Eng. (2018). doi:10.1016/j.compchemeng.2018.09.005

164. Ochoa, M.P., Grossmann, I.E.: Novel MINLP Formulations for Flexibility Analysis for Measured and Unmeasured Uncertain Parameters. Comput. Chem. Eng. 106727 (2020). doi:10.1016/J.COMPCHEMENG.2020.106727

165. Straub, D.A., Grossmann, I.E.: Design optimization of stochastic flexibility. Comput. Chem. Eng. 17, 339–354 (1993). doi:10.1016/0098-1354(93)80025-I

166. Terrazas-Moreno, S., Grossmann, I.E., Wassick, J.M., Bury, S.J.: Optimal design of reliable integrated chemical production sites. Comput. Chem. Eng. (2010). doi:10.1016/j.compchemeng.2010.07.027

167. Banerjee, I., Ierapetritou, M.G.: Design optimization under parameter uncertainty for general black-box models. Ind. Eng. Chem. Res. 41, 6687–6697 (2002). doi:10.1021/ie0202726

168. Bakar, S.H.A., Hamid, M.K.A., Alwi, S.R.W., Manan, Z.A.: Flexible and operable heat exchanger networks. Chem. Eng. Trans. 32, 1297–1302 (2013). doi:10.3303/CET1332217

169. Goyal, V., Ierapetritou, M.G.: Deterministic Framework for Robust Modular Design with Integrated-Demand Data Analysis. Ind. Eng. Chem. Res. 43, 6813–6821 (2004). doi:10.1021/ie049771s

170. Wang, H., Mastragostino, R., Swartz, C.L.E.: Flexibility analysis of process supply chain networks. Comput. Chem. Eng. 84, 409–421 (2016). doi:10.1016/j.compchemeng.2015.07.016

171. Dimitriadis, V.D., Pistikopoulos, E.N.: Flexibility Analysis of Dynamic Systems. Ind. Eng. Chem. Res. 34, 4451–4462 (1995). doi:10.1021/ie00039a036

172. Lenhoff, A.M., Morari, M.: Design of resilient processing plants-I Process design under consideration of dynamic aspects. Chem. Eng. Sci. (1982). doi:10.1016/0009-2509(82)80159-0

173. Palazoglu, A., Arkun, Y.: A multiobjective approach to design chemical plants with robust dynamic operability characteristics. Comput. Chem. Eng. (1986). doi:10.1016/0098-1354(86)85036-0

174. Luyben, M.L., Floudas, C.A.: Analyzing the interaction of design and control-1. A multiobjective framework and application to binary distillation synthesis. Comput. Chem. Eng. (1994). doi:10.1016/0098-1354(94)E0013-D

175. Sánchez-Sánchez, K., Ricardez-Sandoval, L.: Simultaneous process synthesis and control design under uncertainty: A worst-case performance approach. AIChE J. 59, 2497–2514 (2013). doi:10.1002/aic.14040

176. Swartz, C.L.E., Kawajiri, Y.: Design for dynamic operation - A review and new perspectives for an increasingly dynamic plant operating environment, (2019)

177. Ricardez-Sandoval, L.A., Budman, H.M., Douglas, P.L.: Integration of design and control for chemical processes: A review of the literature and some recent results, (2009)

178. Yuan, Z., Chen, B., Sin, G., Gani, R.: State-of-the-art and progress in the optimization-based simultaneous design and control for chemical processes. AIChE J. 58, 1640–1659 (2012). doi:10.1002/aic.13786

179. Burnak, B., Diangelakis, N.A., Pistikopoulos, E.N.: Towards the Grand Unification of Process Design, Scheduling, and Control—Utopia or Reality? Processes. 7, 461 (2019). doi:10.3390/pr7070461

180. Sakizlis, V., Perkins, J.D., Pistikopoulos, E.N.: Recent advances in optimization-based simultaneous process and control design. Comput. Chem. Eng. (2004). doi:10.1016/j.compchemeng.2004.03.018

181. Grossmann, I.E., Calfa, B.A., Garcia-Herreros, P.: Evolution of concepts and models for quantifying resiliency and flexibility of chemical processes. Comput. Chem. Eng. 70, 22–34 (2014). doi:10.1016/j.compchemeng.2013.12.013

182. Georgakis, C., Uztürk, D., Subramanian, S., Vinson, D.R.: On the operability of continuous

processes. Control Eng. Pract. (2003). doi:10.1016/S0967-0661(02)00217-4

183. Mohideen, M.J., Perkins, J.D., Pistikopoulos, E.N.: Optimal synthesis and design of dynamic systems under uncertainty. Comput. Chem. Eng. 20, 2251–2272 (1996). doi:10.1016/0098-1354(96)00157-3

184. Mohideen, M.J., Perkins, J.D., Pistikopoulos, E.N.: Robust stability considerations in optimal design of dynamic systems under uncertainty. J. Process Control. (1997). doi:10.1016/S0959-1524(97)00014-0

185. Zhang, Q., Grossmann, I.E., Sundaramoorthy, A., Pinto, J.M.: Data-driven construction of Convex Region Surrogate models. Springer US (2016)

186. 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. Chem. Eng. Sci. 147, 137–149 (2016). doi:10.1016/j.ces.2016.03.028

187. Ning, C., You, F.: Data-driven decision making under uncertainty integrating robust optimization with principal component analysis and kernel smoothing methods. Comput. Chem. Eng. 112, 190–210 (2018). doi:10.1016/j.compchemeng.2018.02.007

188. Bhosekar, A., Ierapetritou, M.: A discontinuous derivative-free optimization framework for multi-enterprise supply chain. Optim. Lett. (2019). doi:10.1007/s11590-019-01446-5

189. Grossmann, I.E.: Minlp Optimization Strategies and Algorithms for Process Synthesis. Found. Comput. Process Des. 105–132 (1990)

190. Duran, M.A., Grossmann, I.E.: A mixed-integer nonlinear programming algorithm for process systems synthesis. AIChE J. 32, 592–606 (1986). doi:10.1002/aic.690320408

191. Henao, C.A., Maravelias, C.T.: Surrogate-Based Superstructure Optimization Framework. IFAC Proc. Vol. 7, 405–410 (2010). doi:10.1002/aic

192. Wang, S., Zhou, L., Ji, X., Karimi, I.A., He, G., Dang, Y., Xu, X.: A Surrogate-Assisted Approach for the Optimal Synthesis of Refinery Hydrogen Networks. Ind. Eng. Chem. Res. 58, 16798–16812 (2019). doi:10.1021/acs.iecr.9b03001

193. Rafiei, M., Ricardez-Sandoval, L.A.: New frontiers, challenges, and opportunities in integration of design and control for enterprise-wide sustainability. Comput. Chem. Eng. (2020). doi:10.1016/j.compchemeng.2019.106610

194. Boukouvala, F., Ierapetritou, M.G.: Surrogate-based optimization of expensive flowsheet modeling for continuous pharmaceutical manufacturing. J. Pharm. Innov. 8, 131–145 (2013). doi:10.1007/s12247-013-9154-1

195. Dias, L.S., Pattison, R.C., Tsay, C., Baldea, M., Ierapetritou, M.G.: A Simulation-based Optimization Framework for Integrating Scheduling and Model Predictive Control, and its Application to Air Separation Units. Comput. Chem. Eng. 113, 139–151 (2018). doi:10.1016/j.compchemeng.2018.03.009

196. Krawczyk, B.: Learning from imbalanced data: open challenges and future directions. Prog. Artif. Intell. 5, 221–232 (2016). doi:10.1007/s13748-016-0094-0

197. Fernández, A., García, S., Herrera, F., Chawla, N.V.: SMOTE for Learning from Imbalanced

Data: Progress and Challenges, Marking the 15-year Anniversary. J. Artif. Intell. Res. 61, 863–905 (2018)

198. Rooney, W.C., Biegler, L.T.: Incorporating joint confidence regions into design under uncertainty. Comput. Chem. Eng. 23, 1563–1575 (1999). doi:10.1016/S0098-1354(99)00311-7

199. Mulvey, J.M., Vanderbei, R.J., Zenios, S.A.: Robust Optimization of Large-Scale Systems. Oper. Res. 43, 264–281 (1995). doi:10.1287/opre.43.2.264

200. Marler, R.T., Arora, J.S.: Survey of multi-objective optimization methods for engineering. Struct. Multidiscip. Optim. 26, 369–395 (2004). doi:10.1007/s00158-003-0368-6

201. Dias, L.S., Ierapetritou, M.G.: Integration of planning, scheduling and control problems using data-driven feasibility analysis and surrogate models. Comput. Chem. Eng. 134, 106714 (2020). doi:10.1016/j.compchemeng.2019.106714

202. Sirdeshpande, A.R., Ierapetritou, M.G., Andrecovich, M.J., Naumovitz, J.P.: Process synthesis optimization and flexibility evaluation of air separation cycles. AIChE J. 51, 1190–1200 (2005). doi:10.1002/aic.10377

203. Huang, R.: Nonlinear Model Predictive Control and Dynamic Real Time Optimization for Large-scale Processes. 164 (2010). doi:d

204. Johansson, T.: Integrated Scheduling and control of Air Separation Unit Subject to Time-Varying Electricity Price. (2015)

205. Cao, Y., Swartz, C.L.E., Baldea, M., Blouin, S.: Optimization-based assessment of design limitations to air separation plant agility in demand response scenarios. J. Process Control. 33, 37–48 (2015). doi:10.1016/j.jprocont.2015.05.002

206. Cao, Y., Swartz, C.L.E., Baldea, M.: Design for dynamic performance: Application to an air separation unit. Proc. Am. Control Conf. 2683–2688 (2011). doi:10.1109/acc.2011.5990942

207. Pattison, R.C., Touretzky, C.R., Johansson, T., Harjunkoski, I., Baldea, M.: Optimal Process Operations in Fast-Changing Electricity Markets: Framework for Scheduling with Low-Order Dynamic Models and an Air Separation Application. Ind. Eng. Chem. Res. 55, 4562–4584 (2016). doi:10.1021/acs.iecr.5b03499

208. Caspari, A., Offermanns, C., Schäfer, P., Mhamdi, A., Mitsos, A.: A flexible air separation process: 1. Design and steady-state optimizations. AIChE J. 1–12 (2019). doi:10.1002/aic.16705

209. Caspari, A., Offermanns, C., Schäfer, P., Mhamdi, A., Mitsos, A.: A flexible air separation process: 2. Optimal operation using economic model predictive control. AIChE J. 1–14 (2019). doi:10.1002/aic.16721

210. Hassan, M.M., Ruthven, D.M., Raghavan, N.S.: Air separation by pressure swing adsorption on a carbon molecular sieve. Chem. Eng. Sci. (1986). doi:10.1016/0009-2509(86)87106-8

211. Manenti, F., Rossi, F., Croce, G., Grottoli, M.G., Altavilla, M.: Intensifying air separation units. Chem. Eng. Trans. 35, 1249–1254 (2013). doi:10.3303/CET1335208

212. Tesch, S., Morozyuk, T.: Comparative Evaluation of Cryogenic Air Separation Units from the Exergetic and Economic Points of View. Presented at the (2019)

213. EIA - Electricity Data, https://www.eia.gov/electricity/monthly/epm_table_grapher.php?t=epmt_5_6_a

214. Shao, Y., Zavala, V.M.: Modularity Measures: Concepts, Computation, and Applications to Manufacturing Systems. 2020 Am. Inst. Chem. Eng. 287–288 (2020). doi:10.1016/s0740-5472(96)90021-5

215. Patience, G.S., Boffito, D.C.: Distributed production: Scale-up versus Experience. J. Adv. Manuf. Process. 1–8 (2020). doi:10.1002/amp2.10039

216. Lieberman, M.B.: The Learning Curve and Pricing in the Chemical Processing Industries. RAND J. Econ. 15, 213–228 (1984). doi:10.2307/2555676

217. Allman, A., Zhang, Q.: Dynamic location and relocation of modular manufacturing facilities. Eur. J. Oper. Res. (2020). doi:10.1016/j.ejor.2020.03.045

218. Becker, T., Lier, S., Werners, B.: Value of modular production concepts in future chemical industry production networks. Eur. J. Oper. Res. (2019). doi:10.1016/j.ejor.2019.01.066

219. Lier, S., Wörsdörfer, D., Grünewald, M.: Transformable Production Concepts: Flexible, Mobile, Decentralized, Modular, Fast. ChemBioEng Rev. 3, 16–25 (2016). doi:10.1002/cben.201500027

220. Lara, C.L., Bernal, D.E., Li, C., Grossmann, I.E.: Global optimization algorithm for multi-period design and planning of centralized and distributed manufacturing networks. Comput. Chem. Eng. 127, 295–310 (2019). doi:10.1016/J.COMPCHEMENG.2019.05.022

221. Bowling, I.M., Ponce-Ortega, J.M., El-Halwagi, M.M.: Facility location and supply chain optimization for a biorefinery. Ind. Eng. Chem. Res. 50, 6276–6286 (2011). doi:10.1021/ie101921y

222. Elia, J.A., Baliban, R.C., Xiao, X., Floudas, C.A.: Optimal energy supply network determination and life cycle analysis for hybrid coal, biomass, and natural gas to liquid (CBGTL) plants using carbon-based hydrogen production. Comput. Chem. Eng. 35, 1399–1430 (2011). doi:10.1016/j.compchemeng.2011.01.019

223. Tan, S.H., Barton, P.I.: Optimal dynamic allocation of mobile plants to monetize associated or stranded natural gas, part I: Bakken shale play case study. Energy. 93, 1581–1594 (2015). doi:10.1016/J.ENERGY.2015.10.043

224. Bramsiepe, C., Sievers, S., Seifert, T., Stefanidis, G.D., Vlachos, D.G., Schnitzer, H., Muster, B., Brunner, C., Sanders, J.P.M., Bruins, M.E., Schembecker, G.: Low-cost small scale processing technologies for production applications in various environments—Mass produced factories. Chem. Eng. Process. Process Intensif. 51, 32–52 (2012). doi:10.1016/J.CEP.2011.08.005

225. Bhosekar, A., Ierapetritou, M.: Modular Design Optimization using Machine Learning-based Flexibility Analysis. J. Process Control. 90, 18–34 (2020). doi:https://doi.org/10.1016/j.jprocont.2020.03.014

226. Castro, P.M.: Tightening piecewise McCormick relaxations for bilinear problems. Comput. Chem. Eng. 72, 300–311 (2015). doi:10.1016/j.compchemeng.2014.03.025

227. Nagarajan, H., Lu, M., Yamangil, E., Bent, R.: Tightening McCormick relaxations for nonlinear programs via dynamic multivariate partitioning. Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics). 9892 LNCS, 369–387 (2016). doi:10.1007/978-3-319-44953-1_24

228. Hasan, M.M.F., Karimi, I.A.: Piecewise linear relaxation of bilinear programs using bivariate partitioning. AIChE J. 56, 1880–1893 (2010). doi:10.1002/aic.12109

229. Karuppiah, R., Grossmann, I.E.: Global optimization for the synthesis of integrated water systems in chemical processes. Comput. Chem. Eng. 30, 650–673 (2006). doi:10.1016/j.compchemeng.2005.11.005

230. You, F., Grossmann, I.E.: Design of responsive supply chains under demand uncertainty. Comput. Chem. Eng. 32, 3090–3111 (2008). doi:10.1016/j.compchemeng.2008.05.004

231. You, F., Grossmann, I.E.: Integrated multi-echelon supply chain design with inventories under uncertainty: MINLP models, computational strategies. AIChE J. 56, 419–440 (2010). doi:10.1002/aic.12010

232. Sahay, N., Ierapetritou, M.: Flexibility assessment and risk management in supply chains. AIChE J. 61, 4166–4178 (2015). doi:10.1002/aic.14971

233. You, F., Wassick, J.M., Grossmann, I.E.: Risk management for a global supply chain planning under uncertainty: Models and algorithms. AIChE J. 55, 931–946 (2009). doi:10.1002/aic.11721

234. Küçükyavuz, S., Sen, S.: An Introduction to Two-Stage Stochastic Mixed-Integer Programming. In: The Operations Research Revolution. pp. 1–27. INFORMS (2017)

235. Benders, J.F.: Partitioning procedures for solving mixed-variables programming problems. Numer. Math. 4, 238–252 (1962). doi:10.1007/BF01386316

236. Kopanos, G.M., Pistikopoulos, E.N.: Reactive scheduling by a multiparametric programming rolling horizon framework: A case of a network of combined heat and power units. Ind. Eng. Chem. Res. 53, 4366–4386 (2014). doi:10.1021/ie402393s

237. Li, Z., Ierapetritou, M.G.: Rolling horizon based planning and scheduling integration with production capacity consideration. Chem. Eng. Sci. 65, 5887–5900 (2010). doi:10.1016/j.ces.2010.08.010

238. Wang, L., Lu, Z., Ren, Y.: A rolling horizon approach for production planning and condition-based maintenance under uncertain demand. J Risk Reliab. 233, 1014–1028 (2019). doi:10.1177/1748006X19853671

Appendix A

Sets

$\mathcal{T}$ set of time periods

$\mathcal{F}$ set of production facilities

$\mathcal{S}$ set of suppliers

$\mathcal{W}$ set of warehouses

$\mathcal{P}$ set of products

$\mathcal{R}$ set of retailers

$\mathcal{A}$ set of raw materials

$\mathcal{M}$ set of modules

$\mathcal{O}_m$ set of options for module $m \in \mathcal{M}$

Parameters

c_{fp} operating cost per unit of product $p \in \mathcal{P}$ at production facility $f \in \mathcal{F}$

h_{fwp} transportation cost per unit of product $p \in \mathcal{P}$ from production facility $f \in \mathcal{F}$ to warehouse $w \in \mathcal{W}$

h_{sfa} transportation cost per unit of raw material $a \in \mathcal{A}$ from supplier $s \in \mathcal{S}$ to production facility $f \in \mathcal{F}$

h_{wrp} transportation cost per unit of product $p \in \mathcal{P}$ from warehouse $w \in \mathcal{W}$ to retailer $r \in \mathcal{R}$

g_{wp} inventory cost for storing a unit of product $p \in \mathcal{P}$ for one time period at warehouse $w \in \mathcal{W}$

ζ_{fmo} capital cost for installing one unit of option $o \in \mathcal{O}_m$ for module $m \in \mathcal{M}$at a production facility $f \in \mathcal{F}$

$\tilde{z}_{mo}$ base number of units for option $o \in \mathcal{O}_m$ for module $m \in \mathcal{M}$

β coefficient of mass production

I_{wp0} initial inventory of product $p \in \mathcal{P}$ at warehouse $w \in \mathcal{W}$

$\bar{I}_{wp}$ storage capacity for product $p \in \mathcal{P}$ at warehouse $w \in \mathcal{W}$

δ_{rpt} demand for product $p \in \mathcal{P}$ at retailer $r \in \mathcal{R}$ in time period $t \in \mathcal{T}$

$\nu_{\max}$ maximum number of lines that can be installed at a production facility

K big-M constant for the feasibility constraints

H big-M constant for the constraints defining material flow for each module option

ub_o vector of upper bounds on the variable x for option $o \in \mathcal{O}_m$ for module $m \in \mathcal{M}$

lb_o vector of lower bounds on the variable x for option $o \in \mathcal{O}_m$ for module $m \in \mathcal{M}$

Decision variables

Q_{fwpt} Quantity of product $p \in \mathcal{P}$ delivered from production facility $f \in \mathcal{F}$ to warehouse $w \in \mathcal{W}$ in a time period $t \in \mathcal{T}$

Q_{sfat} Quantity of raw material $a \in \mathcal{A}$ from supplier $s \in \mathcal{S}$ to production facility $f \in \mathcal{F}$ in a time period $t \in \mathcal{T}$

Q_{wrpt} Quantity of product $p \in \mathcal{P}$ delivered from warehouse $w \in \mathcal{W}$ to retailer $r \in \mathcal{R}$

I_{wpt} Inventory of product $p \in \mathcal{P}$ stored at warehouse $w \in \mathcal{W}$ during time period $t \in \mathcal{T}$

z_{mo} Number of units of option $o \in \mathcal{O}_m$ for module $m \in \mathcal{M}$ installed across all production facilities

v_{tf} Number of production lines during time period $t \in \mathcal{T}$ at production facility $f \in \mathcal{F}$

$\bar{v}_{fo}$ Number of production lines of option $o \in \mathcal{O}_m$ for module $m \in \mathcal{M}$ installed at production facility $f \in \mathcal{F}$

q_{tfo} Number of production lines of option $o \in \mathcal{O}_m$ for module $m \in \mathcal{M}$ during time period $t \in \mathcal{T}$ at production facility $f \in \mathcal{F}$

y_{tfo} Binary variable indicating if option $o \in \mathcal{O}_m$ for module $m \in \mathcal{M}$ during time period $t \in \mathcal{T}$ is installed at production facility $f \in \mathcal{F}$

x_{tf} Real variable indicating material flow per production line during time period $t \in \mathcal{T}$ at production facility $f \in \mathcal{F}$

SVM_{of} SVM model for option $o \in \mathcal{O}_m$ at production facility $f \in \mathcal{F}$

Appendix B

Sets

$\mathcal{T}$ set of time periods

$\mathcal{F}$ set of production facilities

$\mathcal{S}$ set of suppliers

$\mathcal{W}$ set of warehouses

$\mathcal{P}$ set of products

$\mathcal{R}$ set of retailers

$\mathcal{A}$ set of raw materials

$\mathcal{M}$ set of modules

$\mathcal{O}_m$ set of options for module $m \in \mathcal{M}$

SC set of scenarios

Parameters

c_{fp} operating cost per unit of product $p \in \mathcal{P}$ at production facility $f \in \mathcal{F}$

h_{fwp} transportation cost per unit of product $p \in \mathcal{P}$ from production facility $f \in \mathcal{F}$ to warehouse $w \in \mathcal{W}$

h_{sfa} transportation cost per unit of raw material $a \in \mathcal{A}$ from supplier $s \in \mathcal{S}$ to production facility $f \in \mathcal{F}$

h_{wrp} transportation cost per unit of product $p \in \mathcal{P}$ from warehouse $w \in \mathcal{W}$ to retailer $r \in \mathcal{R}$

g_{wp} inventory cost for storing a unit of product $p \in \mathcal{P}$ for one time period at warehouse $w \in \mathcal{W}$

b_{rp} penalty per unit of unmet demand of product p at retailer r.

ζ_{fmo} capital cost for installing one unit of option $o \in \mathcal{O}_m$ for module $m \in \mathcal{M}$at a production facility $f \in \mathcal{F}$

$\tilde{z}_{mo}$ base number of units for option $o \in \mathcal{O}_m$ for module $m \in \mathcal{M}$

β coefficient of mass production

I_{wp0} initial inventory of product $p \in \mathcal{P}$ at warehouse $w \in \mathcal{W}$

$\bar{I}_{wp}$ storage capacity for product $p \in \mathcal{P}$ at warehouse $w \in \mathcal{W}$

δ_{rpt} demand for product $p \in \mathcal{P}$ at retailer $r \in \mathcal{R}$ in time period $t \in \mathcal{T}$

$\nu_{\max}$ maximum number of lines that can be installed at a production facility

K big-M constant for the feasibility constraints

H big-M constant for the constraints defining material flow for each module option

ub_o vector of upper bounds on the variable x for option $o \in \mathcal{O}_m$ for module $m \in \mathcal{M}$

lb_o vector of lower bounds on the variable x for option $o \in \mathcal{O}_m$ for module $m \in \mathcal{M}$

Decision variables

Q_{fwpti} Quantity of product $p \in \mathcal{P}$ delivered from production facility $f \in \mathcal{F}$ to warehouse $w \in \mathcal{W}$ in a time period $t \in \mathcal{T}$ for scenario i

Q_{sfati} Quantity of raw material $a \in \mathcal{A}$ from supplier $s \in \mathcal{S}$ to production facility $f \in \mathcal{F}$ in a time period $t \in \mathcal{T}$ for scenario i

Q_{wrpti} Quantity of product $p \in \mathcal{P}$ delivered from warehouse $w \in \mathcal{W}$ to retailer $r \in \mathcal{R}$ in a time period $t \in \mathcal{T}$ for scenario i

I_{wpti} Inventory of product $p \in \mathcal{P}$ stored at warehouse $w \in \mathcal{W}$ during time period $t \in \mathcal{T}$ for scenario i

B_{rpti} Unmet demand for product p at retailer r in time period t of scenario i

z_{mo} Number of units of option $o \in \mathcal{O}_m$ for module $m \in \mathcal{M}$ installed across all production facilities

v_f Number of production lines at production facility $f \in \mathcal{F}$

q_{fo} Number of production lines of option $o \in \mathcal{O}_m$ for module $m \in \mathcal{M}$ at production facility $f \in \mathcal{F}$

y_{fo} Binary variable indicating if option $o \in \mathcal{O}_m$ for module $m \in \mathcal{M}$ is installed at production facility $f \in \mathcal{F}$

x_{tfi} Real variable indicating material flow per production line during time period $t \in \mathcal{T}$ at production facility $f \in \mathcal{F}$ for scenario i

SVM_{of} SVM model for option $o \in \mathcal{O}_m$ at production facility $f \in \mathcal{F}$