Diagnosis of Linear Programming Supply Chain Optimization Models: Detecting Infeasibilities and Minimizing Changes for New Parameter Value

Sitoshna Jatty[a], Niharika Singh[a], Ignacio Grossmann[a], Leonardo Salsano de Assis[b], Christos Galanopoulos[b], Pablo Garcia-Herreros[b], Bianca Springub[b], Nga Tran[b]

[a] *Department of Chemical Engineering, Carnegie Mellon University, 5000 Forbes Ave, Pittsburgh, 15213, PA, USA*

[b] *Research, Development & Innovation Department, Aurubis AG, Hamburg, 20539, , Germany*

Abstract

This paper addresses two major challenges that arise in linear programming supply chain models: detecting infeasibilities and minimizing changes that are introduced in a model due to new parameter data. In the first part, we use the flexibility test method to quantitatively evaluate the constraints in the model that cause infeasibility. If the parameters in these infeasible constraints were incorrectly entered, we use a machine learning technique to detect outliers that may be the cause of infeasibility. The user of these models can take corrective actions to restore feasibility through the proposed algorithm. In the second part, we provide three formulations to minimize the magnitude of changes, minimize the number of changes, and minimize the weighted sum of both magnitude and number of changes in a model when there is a change in the parameters. We also formulate a bi-criterion optimization model by considering the objectives of minimizing cost and minimizing the weighted sum of the number and magnitude of changes to analyze the trade-off between the two. An ideal compromise solution is also presented. The proposed algorithms are applied to several supply chain test problems to demonstrate their usefulness.

Keywords: Supply Chain, Disruptions, Infeasibility Diagnosis, Flexibility Test, Machine Learning, Multi-Objective Optimization

---

Email addresses: `sjatty@alumni.cmu.edu` (Sitoshna Jatty), `grossmann@cmu.edu` (Ignacio Grossmann)

1. Introduction

Supply chains are formed through complex and varied stakeholder participation where the differences in topology, scale, product, commodities, service, and planning-horizon introduces different complexities in every supply chain network (Zhang et al., 2017; Garcia and You, 2015; Barbosa-Povoa and Pinto, 2020; Bok et al., 2000; Maravelias and Sung, 2009; Harjunkoski et al., 2014). Therefore, management of these systems becomes a very challenging undertaking as it requires an understanding of financial, material and information flows (Barbosa-Póvoa, 2012; Barbosa-Povoa and Pinto, 2020). Thus, many organizations rely on a Decision Support System (DSS) to help optimize their supply chain models for the flow of goods, information and finances (Barbosa-Povoa and Pinto, 2020; Perez et al., 2021; Garcia and You, 2015). To represent strategic and operational problems associated with the design, long-term, midterm, or short-term operations of supply chains a variety of Linear Programming (LP), Mixed Integer Linear Programming (MILP), and Mixed Integer Non-Linear Programming (MINLP) models have been developed (Grossmann, 2005; Harjunkoski et al., 2014). Although these models are useful, some challenges may arise when they are in operation.

In large scale supply chain models, interactions between the different nodes of the supply chain model lead to a complex structure wherein making any changes to parameters can impact the output of the entire model often making the model infeasible (Cafaro and Grossmann, 2014; You and Grossmann, 2008; Garcia and You, 2015). Furthermore, these models can be used by the operators to carry out their day to day operations, or it can be used by the planners in an enterprise to understand and interpret their supply chain. Thus, the models are often used by people who are not experts or well acquainted with the mathematical formulations of optimization, which can lead to difficulties in rectifying infeasibilities or interpreting the results that are obtained.

Therefore, understanding and interpreting results of supply chain models is an increasingly important direction for research. There is a strong motivation to develop specialized algorithms that can provide useful information to assist in the understanding and usage of the optimization models.

This paper contributes to the analysis of linear programming supply chain optimization results by providing tools to identify causes of infeasibility and minimize changes that are introduced due to changes in parameter values. This paper is largely inspired by Harvey Greenberg's "The ANALYZE rulebase for supporting LP analysis" which outlines some methodologies to detect infeasibility in linear programming problems and to interpret static results and dynamic outcomes of the optimization (Greenberg, 1996).

The paper is organized as follows. Section 2 outlines a description of the problems addressed in this paper for a linear programming supply chain optimization problem. In the section 3, a review of prior infeasibility diagnosis methods is presented, followed by a novel comprehensive algorithm to detect infeasibility using the flexibility test and a regression machine learning technique. Section 4 proposes three mathematical formulations to minimize changes along with a bi-criterion optimization model to observe the trade-off between minimizing cost and minimizing changes when the model is subjected to parameter changes. Section 5 draws conclusions for the paper.

2. Problem Statement

Figure 1 represents a typical supply chain network that consists of suppliers, plants, distribution centers and customers. As the material flows from the suppliers to the customers, we want to optimize the flow to ensure coordination between components of the supply chain. On many occasions, disruptions or changes in the model may be introduced (e.g. changes in raw material availability, production capacities etc) which could cause models to yield infeasible solutions or counter intuitive results with many changes. Furthermore, as the size of the model increases, it becomes more difficult to interpret the details of the results that may have deviated from the original solution or may have failed to converge.

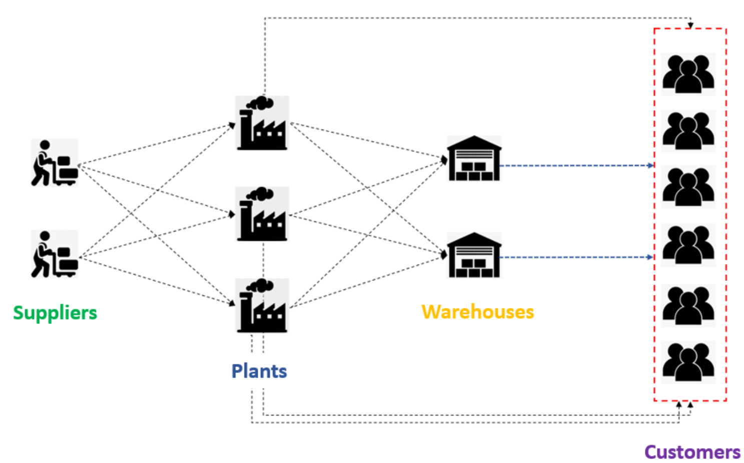

Figure 1: Supply Chain Layout

Two key problems that we address in this paper are the following:

1. **Infeasibility Diagnosis and Resolution** When a large supply chain model becomes infeasible, it is usually difficult for the user or the modeler to pinpoint the cause. It can be the data, the constraints, or a combination of both that might be causing the infeasibility. However, restoring feasibility is not trivial. To tackle the problem, this paper proposes a systematic approach for quantitatively evaluating the infeasibility of a given linear programming planning model. Furthermore, a data-driven approach is presented, which helps to identify the cause of infeasibility given by potentially incorrect specifications.

2. **Minimizing Changes in a parameter-varying linear supply chain optimization** When parameters are changed in a supply chain model such as supplier capacities and customer demands, the resulting difference in the optimization result can be very different as is illustrated through an example in section 4.2. To address this challenge, this paper proposes optimization formulations to reduce such deviations, determine the trade-off between minimizing changes vs minimizing costs via a bi-criterion optimization model that includes the ideal compromise solution.

3. Automated Diagnosis of Infeasibility

Puranik and Sahinidis (2017) reported in their paper that 7% of the models submitted to BARON via the NEOS server over a period of two years (from the beginning of January 2012 to the end of December 2013) were infeasible, which indicates the relevance of this problem in optimization models. Typically, diagnosing the source of infeasibility may not be straightforward or intuitive as it can be caused due to a new specification that leads to infeasibility or due to an error in data.

The research efforts in the area of diagnosing infeasibility have seen several significant contributions and various approaches have been investigated which are outlined next. In the paper "How to Analyze the Results of Linear Programs - Part 3: Infeasibility Diagnosis", Harvey Greenberg describes the isolation of the infeasible part of the model by methods like Price Aggregation, finding the Irreducible Infeasible Subsystems, and Successive Bounds reduction (Greenberg, 1993, 1996). Puranik et al. (2018), in their paper "Infeasibility resolution for multi-purpose batch process scheduling", explore the diagnosis of infeasibility in mixed-integer scheduling problems. Puranik and Sahinidis (2017), in their paper "Deletion Presolve for Accelerating Infeasibility Diagnosis in Optimization Models" discuss the up-grade of the filtering methods for the isolation of IISs (Irreducible Infeasible Sets) using the deletion presolve method for the efficient resolution of infeasibility.

León and Liern (2001) propose the use of the Fuzzy Logic method to repair Infeasibility in LP models. Data Analytic and Neural Network algorithms have also been explored to obtain a feasible region of operation for optimization models. Goyal and Ierapetritou (2002) use simplicial approximations to determine feasible operating envelopes for chemical processes. Dias and Ierapetritou (2019) designed a framework using classification methods like decision trees and support vector machines to define a feasible region in a scheduling problem. This method along with a review on feasibility analysis using surrogate modeling on black-box models is detailed extensively in the review paper by Bhosekar and Ierapetritou (2018).

Work has also been reported in the areas of Flexibility Analysis for better understanding the response of a model in the face of changes in the parameters. Grossmann and Floudas (1987) present the mathematical optimization formulations for performing the Flexibility Test and calculating the Flexibility Index of an uncertain model. This was a comprehensive extension of the work conducted by Swaney and Grossmann (1985) and Halemane and Grossmann (1983) to evaluate flexibility for a set of uncertain parameters involved in a given fixed design. Bansal et al. (2002) presented a design and flexibility optimization model based on parametric programming.

Discussion of the existing methods of infeasibility diagnosis motivates the introduction of our framework in a data-driven context. We have proposed an algorithm that gives a qualitative as well as a quantitative method for the identification and resolution of infeasibility for supply chain models that take into account the data-driven nature of the parameters.

Specifically, we present a novel method that applies the Flexibility Test method and linear regression techniques to find the constraints and the parameters causing infeasibility and give quantitative measures to correct it. The methodology is as follows: determine which constraints in the model are causing infeasibility using the flexibility test. If a parameter is associated with the flagged constraint, then perform a check for possible incorrect data that may be a cause of infeasibility. Else, if the data is correctly entered, adopt corrective actions to regain feasibility.

3.1. Flexibility Test Method

The Flexibility Test represents the backbone of the infeasibility diagnosis algorithm. This method determines the feasibility of operation of an optimization model in a given region of uncertainty (Pistikopoulos and Grossmann, 1988; Zhang et al., 2016). A linear supply chain problem can be modeled by the set of constraints given in Equation 1 :

$$\begin{aligned} Ax &= b \\ Cx &\leq d \end{aligned} \tag{1}$$

Here, the coefficients in the matrices and right-hand sides A, C, b, and d of the variable x correspond to the nominal values (Grossmann and Floudas, 1987; Grossmann et al., 2014). The infeasibility diagnosis can be formulate as: the following min-max problem:

$$\begin{aligned} \psi = & \min_{\text{x}} \ \max_{\text{j}} \ g_j(x) \\ & s.t. \ Ax = b \end{aligned} \tag{2}$$

where $g_j(x) = C_j x - d_j$ is the j^{th} row of the inequalities. Using Equation 2, we minimize over the variable x the maximum value of each constraint. ψ is defined as the feasibility function and its value can be used to determine whether the model is feasible or not. If $\psi \leq 0$, we have a feasible operation, whereas if $\psi > 0$, the operation is not feasible. $\psi = 0$ indicates that we are at the boundary of feasible operation of the supply chain.

Equation 2 is a non-differentiable optimization problem. To reduce the complexity of the problem, we reformulate it as a standard optimization problem by introducing a scalar variable (violation term) u that is greater than or equal to each constraint $C_j x - d_j$ for all $j \in J$.

This corresponds to the original Flexibility Test method that uses infinity-Norm of the violation terms by using, a single scalar u to convert from bi-level to single level optimization problem (Grossmann and Floudas, 1987). Since the infinity-Norm determines the largest violation, using this formulation leads to solving multiple iterations of the LP to restore the model's feasibility. Therefore, we use the 1-Norm instead of the infinity-Norm which may provide more useful information in terms of identifying the constraints that cause infeasibility. The 1-Norm has the advantage that the model needs to be solved once to quantify the infeasibility in the model since the violation of all the constraints is determined simultaneously.

$$\begin{aligned} \psi = & \min_{\text{x}} \sum_{\text{j}} u_j \\ s.t. \ & Ax = b \\ & C_j x - d_j \leq u_j \qquad \forall j \in J \\ & u_j \geq 0 \qquad \forall j \in J \end{aligned} \tag{3}$$

Equation 3 represents the minimization of the sum of violations for every inequality constraint. It is important to note that the constraints for which the violation term (u_j) are non-zero are the ones causing the infeasibility. For restoring feasibility, we relax the right hand side of these constraints by a magnitude that is equal to the violation term.

The Flexibility Test method for infeasibility diagnosis helps in narrowing down the constraints causing infeasibility. When implemented in a supply chain model where the constraints have physical significance, it is assumed that all mass balance constraints at the plant and the distribution center as well as the constraint for the conversion of raw material to product ($Ax = b$) are not violated. The idea behind this assumption is that the linear program we consider is an operations optimization model where the production capacities of the plants and the storage capacities of the distribution centers remain unchanged.

3.2. Detecting Error in the Parameters causing Infeasibility using Regression Machine Learning techniques

In the case when the constraints causing infeasibility are indexed over a large set, and there is a vector of parameters associated with the constraint singled out by the Flexibility Test method, it is usually not trivial to identify the set of parameters causing the infeasibility. To assist the model user in detecting infeasibility caused due to an error in the parameter, the option of using a Machine Learning algorithm is explored. The idea is to model the past data of the individual parameters, and compare any changes in the parameter made by the user (which leads to infeasibility) with the prediction made by the machine learning model to check the viability of the new data. The goal is to pinpoint outlining data that may cause infeasibility in the model, assuming that the model is accurately represented. To use a regression machine learning model, an essential building block required is the past data of the parameter.

The methodology for detecting error in the parameter causing infeasibility is as follows. We first generate a model to characterize the parameter from the available historical data. In this application, the data is assumed to have linear relationship. We optimize the weights of the linear regression model using Ordinary Least Squares method and draw a confidence interval around the regression model outlined in Appendix B, sections B.1., B.2., and B.3. Using this model, we can predict the demand of the customer at any time interval in the future. Since we assume our data has seasonal trends, we used the SARIMA model to predict future demand to account for the seasonality trend as detailed in Appendix B section B.4. The theoretical concepts and equations for this method are outlined in the Appendix B and an example of its implementation is presented in section B.5.

3.3. *Comprehensive Infeasibility Detection Algorithm for Linear Supply Chain Models*

In summary, the comprehensive Infeasibility diagnosis algorithm uses the Flexibility Test method, machine learning regression, and forecasting techniques to find the constraints causing infeasibility and pinpoint the flawed data. As shown in Figure 2, the multi-step algorithm is as follows:

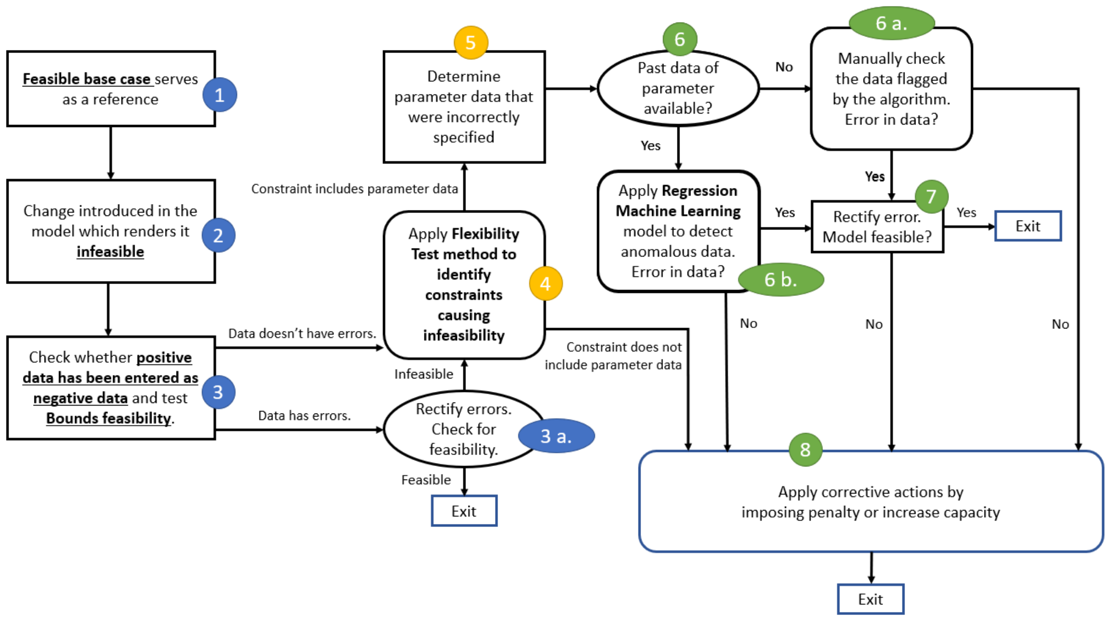

Figure 2: Proposed infeasibility diagnosis algorithm

Step 1: The supply chain model has to have a feasible base case that serves as a reference.

Step 2: A change is made in the input parameter. For example, the operator makes some changes in the model (e.g., changing a parameter such as product/customer demand), which renders the model infeasible.

Step 3: Check whether any negative data has been entered for parameters that were defined as positive. Also check for Bounds Feasibility (that is, no lower bound can be higher than the upper bound).

Step 3. a: If there is an error found while doing trivial checks, rectify the error and perform a feasibility test. If the model becomes feasible, exit the algorithm; else, proceed with the algorithm.

Step 4: To rectify the infeasibility issue in the model, perform the Flexibility Test method for infeasibility diagnosis. The Flexibility Test method will indicate the constraints causing infeasibility.

Step 5: If the constraint indicated by the flexibility test has parameters, determine whether there is an error in the input data for the parameter.

Step 6: Check if past data is available or unavailable for the parameter under inspection.

Step 6.a: If there is no data available, the parameters have to be checked manually. If an error is found, correct it and check for model feasibility.

Step 6.b: If the past data is available, apply the regression and forecasting techniques and check the value of the parameter entered in reference to the predicted value.

Step 7 If an error is found in step 6 a. and 6 b., rectify it and check for model feasibility. At the end of this step, if the model becomes feasible, exit the algorithm.

Step 8: Assuming that the model is correctly represented and the constraints causing infeasibility that are identified in step 5 do not have data, take corrective actions to make the model feasible. Similarly,if the model remains infeasible or if there were no errors found, we know that the data that has been entered is correct, and revisions to the model need to be made to ensure that the model regains feasibility.

In the linear supply chain model, we make an assumption stating that mass balance and conversion constraints are not violable, whereas parameters such as customer demand and supplier capacity can be modified. Hence, the constraints related to these parameters are flexible and in such cases, we can regain model feasibility by imposing a penalty for delay/shortage in delivery or by increasing the capacity of the suppliers.

In the next section, this algorithm is applied to an instance of a supply chain model illustrated in Figure 3.

3.4. Illustrative supply chain model for the infeasibility detection algorithm

3.4.1. Example Problem

Let us consider the supply chain model shown in Figure 3. It is representative of a small supply chain model having two suppliers (S_1, S_2), three plants (P_1, P_2, and P_3), two distribution centers (DC_1, DC_2), and six customers (C_1, C_2, C_3, C_4, C_5, and C_6). The sets S, P, DC and C represent suppliers, plants, distribution centers (warehouses) and customers.

The parameters in this model are ω = Capacity of the supplier, κ = Capacity of the Plant, ϕ = Capacity of the distribution center, λ = conversion rate of materials in the plant and D = Demand of customers. The value of all the parameters and

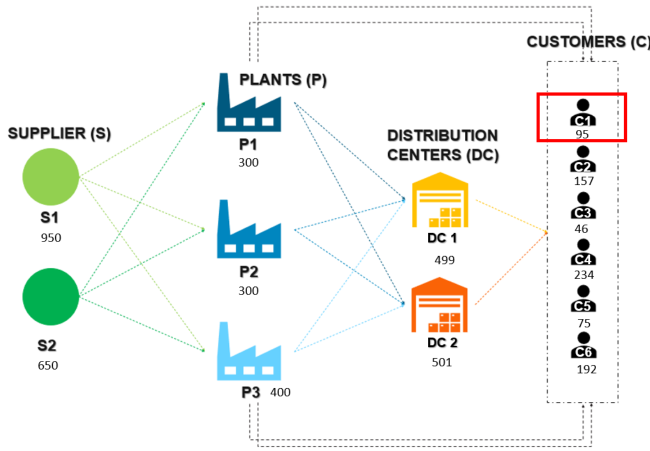

Figure 3: Supply Chain Example Problem; Note that the change is taking place in customer's demand. The numbers under the supplier, plant and distribution center are the capacities of each node. The numbers under the customer represents the customer demand

data in the model are given in Appendix B. There are four kinds of flows f_{ij}; from Suppliers to Plants (S,P), Plants to Customers (P,C), Plants to Distribution Centers (P, DC) and Distribution Centers to Customers (DC, C). The objective function minimizes cost of transportation from various nodes and cost of production at plant (e) when material flows from suppliers to plants.

The optimization model that minimizes the total cost in Equation (4a) is subjected to the following constraints: Equation (4b) states that for all plants $i \in P$, the sum of raw material flows from each supplier ($j \in S$) times plant conversion factor λ_i are equal to the product flows sent to the distribution centers (DC) and customers (C). Equation (4c) states that for all suppliers $i \in S$, the flow of raw materials from each supplier to all plants ($i \in P$) is less than the capacity of the supplier ω_i. Equation (4d) states that for all plants $i \in P$, sum of product flows from each plant sent to the distribution centers (DC) and customers (C) is less than the production capacity of the plant κ_i. Equation (4e) states that the flow of products to all distribution centers $i \in DC$ from all plants $j \in P$ is less than the capacity of the distribution centers ϕ_i.

Equation (4f) ensures that the flow out of a distribution center $i \in DC$ to all customers $j \in C$ is not greater than the flow into the distribution center i from all plants $j \in P$. Equation (4g) is a demand satisfaction constraint that ensures the sum flow of products from all distribution centers $j \in DC$ and all plants $j \in P$

to a customer $i \in C$ is greater than or equal to the demand of the customer D_i. Equation (4h) states that all the feasible flows in the optimization problem should be non-negative.

$$
\begin{aligned}
\min_{f_{ij}} \quad & z = a\sum_{i\in S}\sum_{j\in P} f_{ij} + b\sum_{i\in P}\sum_{j\in DC} f_{ij} + c\sum_{i\in P}\sum_{j\in C} f_{ij} + d\sum_{i\in DC}\sum_{j\in C} f_{ij} \\
& \quad + e(\sum_{i\in P}\sum_{j\in DC} f_{ij} + \sum_{i\in P}\sum_{j\in C} f_{ij}) && \text{(4a)} \\
\text{s.t.} \quad & \sum_{j\in DC} f_{ij} + \sum_{j\in C} f_{ij} = \lambda_i \sum_{j\in S} f_{ji} \qquad \forall\ i \in P && \text{(4b)} \\
& \sum_{j\in P} f_{ij} \le \omega_i \qquad \forall\ i \in S && \text{(4c)} \\
& \sum_{j\in DC} f_{ij} + \sum_{j\in C} f_i \le \kappa_i \qquad \forall\ i \in P && \text{(4d)} \\
& \sum_{j\in P} f_{ji} \le \phi_i \qquad \forall\ i \in DC && \text{(4e)} \\
& \sum_{j\in C} f_{ij} \le \sum_{j\in P} f_{ji} \qquad \forall\ i \in DC && \text{(4f)} \\
& -\sum_{j\in P} f_{ji} - \sum_{j\in DC} f_{ji} \le -D_i \qquad \forall\ i \in C && \text{(4g)} \\
& f_{ij} \ge 0 \qquad \forall\ i,j \in \{(S,P),(P,DC),(P,C),(DC,C)\} && \text{(4h)}
\end{aligned}
$$

3.4.2. Application of algorithm to the example supply chain problem

Let us consider the model outlined in Equation 4. We assume that, during the an update of the data an error is introduced in the model either due to human mistakes or due to a large magnitude change in the data point (could be within the demands, production, and storage parameters), which causes the model to become infeasible.

Consider a case where the demand of the first customer D_1 increases from 95 to 215 in the n^{th} time interval; all the other demands remain the same. However, the operator enters the demand of the first customer as 515 instead of 215.

The model is run and an infeasible solution is detected. We then perform trivial checks with the lower and upper bounds to rectify any errors. However, the model is still rendered infeasible. Therefore, we apply the Flexibility Test method with 1-Norm violations (u_{ij}) to the model (Equation 4). Since constraints (4b), (4d), (4e), and (4f) are mass-balance and conversion constraints, they are inviolable in our example model. Therefore, the violation terms can only be applied to constraints

(4c), (4g) and (4h). These constraints are updated and reflected in Equations (5c), (5g), and (5h). The rest of the constrains in the model have the same topology as the constraints in the model represented using Equation 4.

$$
\begin{aligned}
\min_{f_{ij}, u2_i, u6_i, u7_{ij}} \quad & z = \sum_{i \in S} u2_i + \sum_{i \in C} u6_i + \sum_{i,j \in A \times A} u7_{ij} && && \text{(5a)} \\
\text{s.t.} \quad & \sum_{j \in DC} f_{ij} + \sum_{j \in C} f_{ij} = \lambda_i \sum_{j \in S} f_{ji} && \forall\ i \in P && \text{(5b)} \\
& \sum_{j \in P} f_{ij} \leq \omega_i + u2_i && \forall\ i \in S && \text{(5c)} \\
& \sum_{j \in DC} f_{ij} + \sum_{j \in C} f_i \leq \kappa_i && \forall\ i \in P && \text{(5d)} \\
& \sum_{j \in P} f_{ji} \leq \phi_i && \forall\ i \in DC && \text{(5e)} \\
& \sum_{j \in C} f_{ij} \leq \sum_{j \in P} f_{ji} && \forall\ i \in DC && \text{(5f)} \\
& -\sum_{j \in P} f_{ji} - \sum_{j \in DC} f_{ji} \leq -D_i + u6_i && \forall\ i \in C && \text{(5g)} \\
& - f_{ij} \geq u7_{ij} && \forall\ i,j \in A \times A && \text{(5h)} \\
& u2_i \geq 0 && \forall\ i \in S && \text{(5i)} \\
& u6_i \geq 0 && \forall\ i \in C && \text{(5j)} \\
& u7_{ij} \geq 0 && \forall\ i,j \in A \times A && \text{(5k)} \\
& A \times A \in \{(S,P), (P,DC), (P,C), (DC,C)\}
\end{aligned}
$$

The objective (Equation (5a)) gives a positive value = 219. It is found that the constraint corresponding to the customer demand (Equation (5g)) had non-zero violation terms. This indicates that the infeasibility is being caused due to some problem within this constraint represented in Equation (5g). The violation terms are listed in Table 1.

As per our assumption, we know that the model is correctly represented. This means that there is no problem with the relevance of the model, but there might be some error in the altered data. The identified constraint includes parameters involving customer demand and variables involving flow of material to customers from plants and distribution centers. The only data present within the constraint are the customer demands. Therefore, our next step is to identify if a data-point(s) corresponding to the customer demand are incorrect.

Table 1: Violation terms obtained from constraint (5g). The equation included flow of material from plant to customer (f^P) and flow from plant to distribution center (f^{DC}), where D_i represents the customer demands.

Violation Term	Corresponding Constraint	Value of Violation term
u6(1)	$D_1 - \sum_{j=1}^{3} f_{j1}^{P} - \sum_{j=1}^{2} f_{j1}^{DC} \leq u6(1)$	98
u6(3)	$D_3 - \sum_{j=1}^{3} f_{j3}^{P} - \sum_{j=1}^{2} f_{j3}^{DC} \leq u6(3)$	46
u6(5)	$D_5 - \sum_{j=1}^{3} f_{j5}^{P} - \sum_{j=1}^{2} f_{j5}^{DC} \leq u6(5)$	75

Using the historical data from the past ten years (120 data points) with seasonality and trends, we apply the regression/forecasting technique to predict the customer demand at the n^{th} time interval and check if the data entered by the operator is within the prediction interval given by the model. This is illustrated in Appendix B section B.4. Say the operator has made the changes in the data of the 130^{th} (Oct 2021) time interval.

As per the SARIMA model plotted as Figure B.12, the predicted value for the 130^{th} time interval is 212.28 with the interval being 200.09 and 224.48. Since 515 is outside this interval, the data is flagged as an erroneous one. The flagged data is corrected, and the model is rechecked for feasibility. The model regains feasibility, and we can exit the algorithm.

Through this algorithm, we are able to present a methodology to detect and resolve infeasibility using automated tools. In this example, it is assumed that only one parameter is changed at the n^{th} time step. This assumption is made for simplicity to illustrate the algorithm. The algorithm remains the same for changes introduced in multiple parameters. The user would need to develop the forecast of data for every parameter that needs to be evaluated given that the historical data is available for every parameter.

As we conclude the section on automated infeasibility diagnosis that helps restore the feasibility. In the following section, we cover algorithms used to minimize changes in a data-driven linear supply chain optimization problem when changes are introduced in the model.

4. Minimizing Changes in a parameter-varying linear supply chain optimization

In this section we assume that changes that are introduced in the model lead to feasible solutions of the LP supply chain model; otherwise we apply the algorithm in section 3. There can be many sources of changes that could be introduced in a supply chain problem. For instance, equipment failure by a supplier may result in an increase in transportation costs between supplier and plants. This change can induce a restructuring of network between the nodes, i.e., disrupting the supplier-plant or plant-customer connections, thereby changing the optimization results.

Consider a linear supply chain model (Equation 6), which is formulated as:

$$\begin{aligned} \underset{x}{min} \quad & c^T x \\ s.t. \quad & Ax = b \\ & Cx \leq d \\ & x \geq 0 \end{aligned} \tag{6}$$

The solution of this model is represented as $x = x_A$. Assume variations occur in parameters and data collected, due to changes in supplier operations or unplanned equipment failure. Modifications in the model are made to obtain the new optimal network flow (Equation 7):

$$\begin{aligned} \underset{x}{min} \quad & c^{'T} x \\ s.t. \quad & A^{'} x = b^{'} \\ & C^{'} x \leq d^{'} \\ & x \geq 0 \end{aligned} \tag{7}$$

This model yields the new solution $x = x_B$. When the optimization problem is solved a different configuration may be obtained which could lead to a disruptive sales and operation planning routine. Thus, minimizing the number of changes that occur and the magnitude of those changes are important problems to solve. In this section, we outline three frameworks that will be used to supplement the constraints in the altered supply chain model (Equation 7):

1. A formulation that minimizes the magnitude of changes between the results of the altered (Equation 7) and the pre-existing optimization model (Equation 6)

2. A formulation that minimizes the number of changes in the values of the variables between the altered (Equation 7)) and existing model (Equation 6)

3. A formulation that minimizes a weighted sum of the number as well as the magnitude of changes in the result of the altered (Equation 7) and the existing (Equation 6) optimization model

To determine the trade-off between minimizing changes and minimizing the cost of the supply chain, we consider an additional bi-criterion optimization problem.

Each model will be first explained mathematically, followed by an implementation of Approach 3 on a example problem. This will be followed by a bi-criterion optimization analysis on a small test supply chain problem.

4.1. Mathematical Formulation

4.1.1. Approach 1 : Minimizing magnitude of changes

The approach for minimizing the magnitude of changes consists of sequentially solving two models. First, a supply chain model (Equation 6) is solved to minimize cost. This is followed by solving the second model (Equation 8) which minimizes the changes between the initial model (Equation 6) and the current model (Equation 7) when a change is introduced in the parameters. The mathematical formulation for this model is as follows:

$$\min_{x,\,u} \quad z_{A1} = ||u||_1 \tag{8a}$$

$$\text{s.t.} \quad A^{'}x = b^{'} \tag{8b}$$

$$C^{'}x \leq d^{'} \tag{8c}$$

$$u \geq \ |x_A - x| \tag{8d}$$

$$x, u \geq 0 \tag{8e}$$

Equation (8b) are the given set of constraints for the supply chain model accounting for flows between nodes. x_A are the solutions of the original model (Equation 6) which are used as parameters in Equation 8. x are the variables of the model with altered parameter values. The objective of the model is to minimize $||u||_1 = \sum_i u_i$, where u_i is the slack variable that accounts for the offset between the current optimization variables x and the initial solutions x_A.

The algorithm uses the 1-norm method as in Equation 3 to minimize the deviation in every variable within the supply chain model, i.e., every variable x has a corresponding u_i associated with it. This results in constraints with unique u_i variables that capture the deviation between every $x_{i,A}$ and x_i.

Determining the magnitude of the variables makes the model non-differentiable. Therefore, we reformulate it to make the model linear. For example, let p be a scalar such that, $|p| = max\{+p, -p\}$. We introduce a variable τ which captures the magnitude of p as follows:

$$\begin{aligned} \tau &\geq +p \\ \tau &\geq -p \end{aligned} \tag{9}$$

Using Equation 9 as reference, the LP model for Approach 1 model is reformulated as follows:

$$\min_{x_i,\, u_i} \quad z_{A1} = \sum_i u_i \tag{10a}$$
$$\text{s.t.} \quad A^{'}x = b^{'} \tag{10b}$$
$$C^{'}x \leq d^{'} \tag{10c}$$
$$u_i \geq \ x_{A,i} - x_i \qquad \forall\, i \tag{10d}$$
$$u_i \geq \ -(x_{A,i} - x_i) \quad \forall\, i \tag{10e}$$
$$x, u \geq 0 \tag{10f}$$

4.1.2. Approach 2 : Minimizing number of changes

The model for minimizing the number of changes builds on the formulation of Approach 1 (Equation 10). The supply chain model (Equation 6) is first solved to optimize for minimizing cost. This is followed by solving the second model (Equation 11) which minimizes the number of changes between the initial model (Equation 6) and the current model (Equation 7) when a change is introduced in the parameters. This requires the introduction of 0-1 variables (y_i) to track the changes. The MILP formulation for minimizing the number of changes is as follows:

$$\min_{x_i,\, u_i,\, y_i} \quad z_{A2} = \sum_i y_i \tag{11a}$$
$$\text{s.t.} \quad A^{'}x = b^{'} \tag{11b}$$
$$C^{'}x \leq d^{'} \tag{11c}$$
$$u_i \geq \ x_{A,i} - x_i \qquad \forall\, i \tag{11d}$$
$$u_i \geq \ -(x_{A,i} - x_i) \quad \forall\, i \tag{11e}$$
$$u_i \leq M.y_i \qquad \forall\, i \tag{11f}$$
$$y_i \in \{0, 1\} \qquad \forall\, i \tag{11g}$$
$$x, u \geq 0 \tag{11h}$$

Here, the objective of the model is to minimize the $\sum_i y_i$ where y_i are binary variables that indicate through the inequalities (Equation (11d) - (11f)) whether a change in the variables x in model should occur. Extended from the first approach (Equation 10), the deviations between the solution of the variables from the original model (Equation 6) and the modified model (Equation 11) are captured by the u_i variables. If a deviation has occurred, the binary y_i variable is activated to indicate the change that is introduced to the objective.

The objective of the model is to minimize $\sum_i y_i$. Hence the model will try to reduce the number of changes in the variables, while maintaining feasibility. The big-M parameter is set to an appropriate value that is sufficently large.

4.1.3. Approach 3 : Minimize number and magnitude of changes

The model for minimizing the number of changes and magnitude of changes is a further modification to Approach 2. As previously seen, first a supply chain model (Equation 6) is solved to optimize for minimizing cost followed by solving the second model (Equation 12), which minimizes the weighted sum of number of changes and magnitude of changes. The formulation is as follows:

$$\min_{x_i, u_i, y_i} \quad z_{A3} = \alpha \sum_i u_i + \beta \sum_i y_i \tag{12a}$$

$$\text{s.t.} \quad A^{'} x = b^{'} \tag{12b}$$

$$P^{'} x \leq q^{'} \tag{12c}$$

$$u_i \geq \ x_{A,i} - x_i \qquad \forall\, i \tag{12d}$$

$$u_i \geq \ -(x_{A,i} - x_i) \qquad \forall\, i \tag{12e}$$

$$u_i \leq M.y_i \qquad \forall\, i \tag{12f}$$

$$y_i \in \{0, 1\} \qquad \forall\, i \tag{12g}$$

$$x, u \geq 0 \tag{12h}$$

In Equation 12, α and β are relative weights for u and y variables. For simplicity in the illustration presented in section 4.2, we assume that α and β are set to a value of 1. However, the sensitivity to these parameters should be determined when extending the formulation to large scale problems.

4.2. Demonstration of Approach 3 Algorithm on a Supply Chain Problem

As an example, let us consider a supply chain optimization within an enterprise. The contracts drawn between suppliers and plants may leave little room for negotiation if unforeseen changes occur to increase the cost associated with the transportation. This could result in cases where a large number of changes must be made to minimize the overall cost which affects the planning at a plant. Additionally, it must be ensured that the demand of the customer is met without paying back-order costs.

In the next section the results and an analysis of Approach 3 is presented using the supply chain model outlined in section 3.4.1.

4.2.1. Example Problem

The example problem is presented in section 3.4.1 is the base case model used for generating results for optimization. The model has a defined set of parameters associated that is outlined in Appendix C.

Tables 2 and 3 highlight the parameters where the change is made. Specifically, transportation costs associated between the supplier and the plant (parameter a from the example supply chain model) are modified from their original costs to the following new costs.

Table 2: Transportation costs from supplier to plants in base case model (parameter a)

Base	Supplier 1	Supplier 2
Plant 1	0.15	0.11
Plant 2	0.2	0.23
Plant 3	0.3	0.4

Table 3: Transportation costs from supplier to plants in modified case model (parameter a')

Modified	Supplier 1	Supplier 2
Plant 1	1.5	0.11
Plant 2	2	0.23
Plant 3	3	0.4

4.2.2. Optimization of three cases

The following three cases are optimized to yield solutions for the supply chain layout referenced in section 3.4.1.

1. Base Case using Original Model: Base Case model (Equation 6) with costs for transportation of material between supplier and plant outlined in Table 2 and other parameters outlined in Appendix C. The objective of this optimization model is to minimize cost.

2. Modified Case using Original Model: Modified Case model (Equation 7) with new costs for transportation of material between supplier and plant outlined in Table 3 and other parameters outlined in Appendix C. The objective of this model is also to minimize cost.

3. Modified Case using Approach 3 Model: Modified Case model using Approach 3 formulation (Equation 12) with costs for transportation of material between

supplier and plant outlined in Table 3 and other parameters outlined in Appendix C. The objective of this optimization model is to minimize number and magnitude of changes.

Base Case using Original Model

The base case is the supply chain model that is optimized for minimizing cost (Equation 6). The objective is to find a network that would satisfy the demand of the customers yielding the lowest cost.

When the optimization is run with the specified parameters, we get the following cost of 80,245 €. The solution for the optimization is given in Figure 4 on the boxes in the arrows going from one node to another. Each arrow begins at the source node and ends at the sink node indicating the direction of the flow of material. Along with the direction, the arrow also indicates the amount of material that flows from the source to the sink node. For example, the flow of material from the supplier S_1 to plant P_2 is 118.8 units.

For each plant P_1, P_2 and P_3, the amount of material flow from the supplier to the plant is converted to product using a conversion rate. The amount of final product at each plant is written in white within the colored box. As shown in the case of plant P_2 that receives 118.8 units of material from supplier S_1, we observe the conversion of the raw material to 101 units of products is predicted which are distributed to customers and distribution centers. For completion, observe that the products produced at plant P_2 are distributed to customer C_5 with a demand of 75 units and distribution center DC_2. The 26 units of product sent to the distribution center DC_2 is ultimately sent to customer C_6 to satisfy its demand of 192 units by adding to the products sent from plant P_3. It is assumed that the products produced at the plants P_1, P_2 and P_3 are homogeneous and the material distributed from the suppliers S_1 and S_2 to the plants are also miscible.

For added information, the capacities of each nodes are written in grey above or below the representative icons and the parameters associated with cost of transportation of materials or products from the source node are written besides the icons.

Modified Case using Original Model

The modified case is the supply chain model that is optimized for minimizing cost (Equation 7) with changed parameters shown in Table 3. The objective is to find a network that satisfies the demand of the customers yielding the lowest cost with the changed data.

When this optimization is run we notice the following change represented in Figure 5. The network configuration has significantly changed to optimize for minimum cost when compared with Figure 4. We observe at total of 13 changes in the form of new connections and changes in the magnitude of the flow of variables. For example,

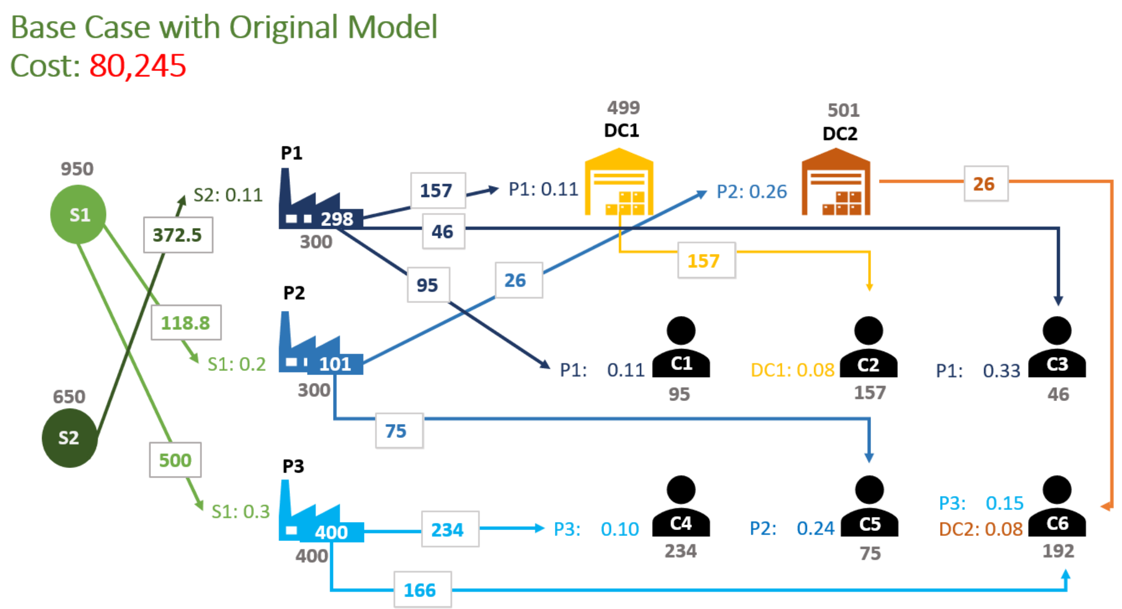

Figure 4: Base Case Optimization result using Original model (Equation 6)

compare the flows from supplier S_1 and S_2 to plants P_1, P_2 and P_3. As highlighted using the red arrows, new connections are added from supplier S_2 to plants P_2 and P_3. Since the capacity of supplier S_2 is reached, the solver has to draw material from S_1 to satisfy the demand. The solver optimizes to reduce the cost of sourcing material from the suppliers to the plants. Furthermore, in certain cases where the connection was already present within Figure 4, a change in the magnitude of the flow is observed as seen in the flow of products from plant P_2 to distribution center DC_2 which changed from 26 units (Figure 4) to 192 units (Figure 5). Finally, the cost has increased to 80,750 € due to the changes observed in the model. The increase in cost is incurred because of the increase in value of the parameter a.

Modified Case using Approach 3 Model

The optimization is performed with the modified transportation costs between the supplier and the plants (Table 3) using Approach 3 model (Equation 12), where a simultaneous minimization in the number and changes in magnitude occurs. The objective here is to minimize changes, for simplicity assuming $\alpha = \beta = 1$ in Equation 12.

The network configuration in Figure 6 resembles the original base case optimization (Figure 4), indicating that it is feasible to obtain a solution without making any

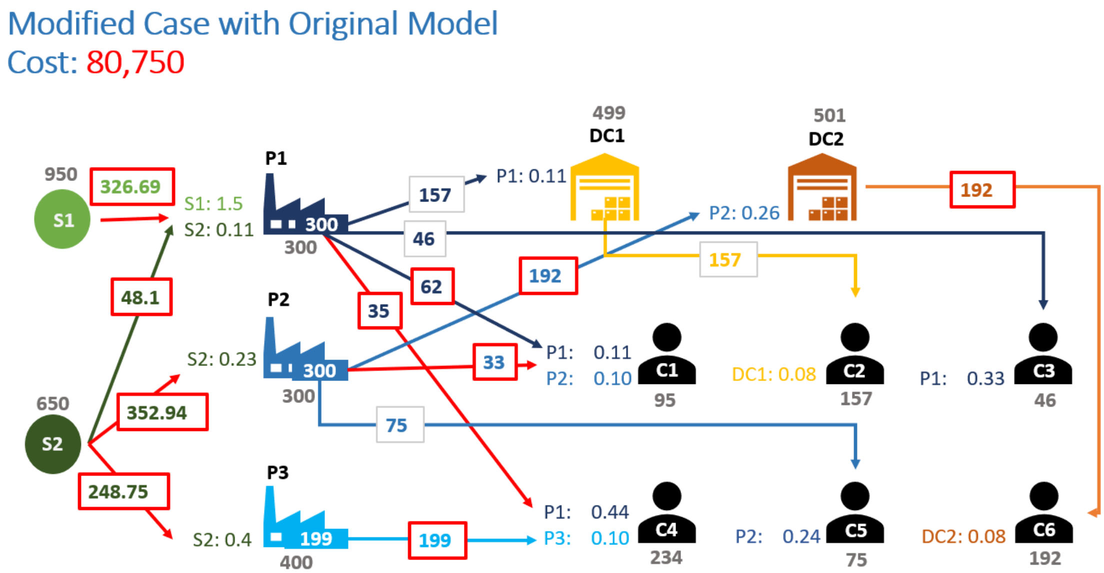

Figure 5: Modified Case Optimization result using Original model (Equation 7)

changes to the orginal configuration (Figure 4). In other words, the original network flow obtained in Figure 4 is still feasible, even when the changes are made to the parameters referenced in section 4.2.1.

When the results of the three cases are compared, it is observed that the penalty associated with keeping the configuration the same using Approach 3 model (Figure 6) is significantly higher than allowing for changes to take place in the system (Figure 5). The cost of the optimization using Approach 3 is 81,800 €, which is greater than 80,245 € that is obtained from the base case optimization. This is consistent with the analysis as we are paying a penalty for trying to minimize changes in the results.

The next question we address is, since there is a motivation to minimize changes and minimize cost, are there solutions where there is a trade-off between the two objectives?

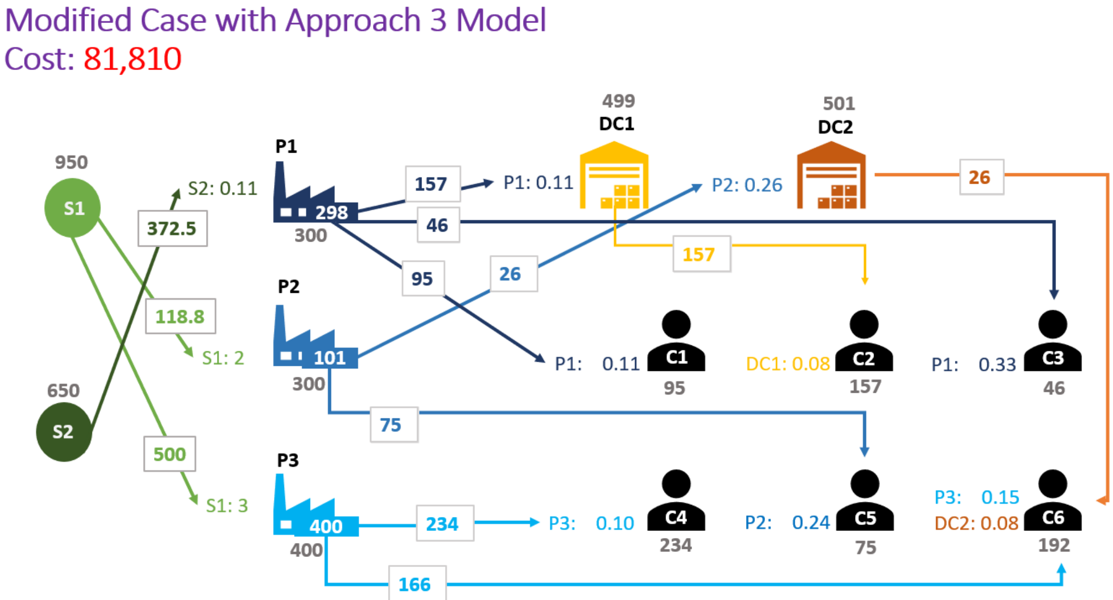

Figure 6: Modified Case Optimization result using Approach 3 model (Equation 12)

4.3. Bi-Criterion Optimization

In the example illustration, we determined the penalty in cost that we pay when keeping the same operation layout as the base case i.e. when the number of changes are minimized. Alternately, we observe that allowing for the changes to take place when the data is varied lead to a lower cost. In this section, we explore the trade-off between the two objectives of our problem: minimizing changes and minimizing cost. Using the ϵ-constraint method we optimize one of the objective functions and specifying the other objective function a constraint bounded by an ϵ value (Mavrotas, 2009). By parametrically varying the RHS of the ϵ-constrained objective functions, the trade-off or Pareto optimal solutions of the problem are obtained.

4.4. Application of bi-criterion optimization on linear supply chain problem and trade-off analysis

In this method, the two objectives: Minimizing Changes (NC) and Minimizing Cost (C) are introduced as a primary objective and a second objective. The constraints (NC and C) are added to the Approach 3 (Equation 12) model for illustration.

$$
\begin{aligned}
\min_{x_i,\, u_i,\, y_i} \quad & \sum_i u_i + \sum_i y_i && \text{(NC)} \\
\text{s.t.} \quad & c^{'T} x \le \epsilon && \text{(C)} \\
& A^{'} x = b^{'} \\
& C^{'} x \le d^{'} \\
& u_i \ge \; x_{A,i} - x_i \qquad \forall\, i \\
& u_i \ge \; -(x_{A,i} - x_i) \qquad \forall\, i && (13) \\
& u_i \le M.y_i \qquad \forall\, i \\
& y_i \in \{0, 1\} \qquad \forall\, i \\
& x, u \ge 0
\end{aligned}
$$

The primary objective (NC) is entered as the objective function of the optimization, whereas the cost function (C) is formulated as an ϵ-constraint. This constraint is bounded by an ϵ value that restricts the variables to stay within the bounds imposed.

4.4.1. Example Linear Supply Chain Problem Analysis

In the example problem, the ϵ value is varied between 81,800 € (Approach 3 cost) to 80,750 € (modified case with original model cost), to obtain a Pareto-curve plotted in Figure 7. The curve represents the points that show the number of changes in the model when the cost is allowed to increase to a value ϵ. Hence the values are enumerated through discrete points plotted on the Pareto curve. Since the supply chain model that minimizes the changes, is non-convex due to the presence of integer variables, we can safely apply the epsilon constraint to generate a Pareto curve. The horizontal axis represents the cost of the supply chain network. The vertical axis represents the number of changes ($\sum y_i$). Note that the formulation of Approach 3 is used to generate the plot which simultaneously minimize the magnitude and number of changes ($\sum y_i + \sum u_i$) and yields a unique objective value for every ϵ-value. However, for easier interpretation, we present only the number of changes on the plot i.e. $\sum y_i$. Hence, the graph shows multiplicity in the number of changes for some associated costs (ϵ-value).

Introducing changes to the parameters in the original model results in 13 changes with the minimum cost of 80,750 €. When approach 3 is utilized to restrict the number and magnitude of changes, the cost 81,809 € is obtained with no changes in the model as noted in section 4.2.2. Imposing a constraint on the cost increase such that it is not greater than 80,800 € and solving for minimizing the changes in the model using Approach 3 model, yields a set of feasible solutions. For example,

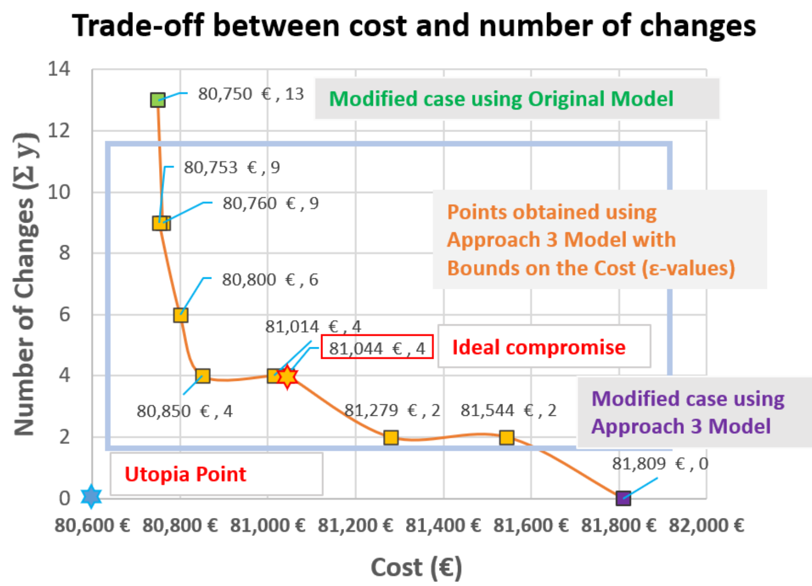

Figure 7: Pareto Curve depicting minimizing changes and cost. Ideal compromise solution and Utopia point plotted on the curve.

allowing 4 instead of 13 changes decreases the cost from 81,809 € to 81,044 € which is in contrast to decreasing the cost to 80,750 € with 13 changes. Hence, the trade-off for allowing an increase in cost is fewer number of changes.

An additional piece of information can be inferred from the trade-off curve is the ideal compromise solution. As outlined in the paper by Grossmann et al. (1982), we use the infinity-norm formulation to obtain the minimum distance on the Pareto front from the utopia point. The formulation for the ideal compromise is given in Equation 14.

$$
\begin{aligned}
\min_{\rho, x_i, u_i, y_i} \quad & \rho && && (14a)\\
\text{s.t.} \quad & \rho \geq f_1' && && (14b)\\
& \rho \geq f_2' && && (14c)\\
& A^{'}x = b^{'} && && (14d)\\
& C^{'}x \leq d^{'} && && (14e)\\
& u_i \geq \ x_{A,i} - x_i && \forall\, i && (14f)\\
& u_i \geq \ -(x_{A,i} - x_i) && \forall\, i && (14g)\\
& u_i \leq M.y_i && \forall\, i && (14h)\\
& y_i \in \{0,1\} && \forall\, i && (14i)\\
& x, u \geq 0 && && (14j)
\end{aligned}
$$

Here, our two objectives are to minimize cost and minimize the weighted sum of changes which are represented in constraints constraints (14b) and (14c).The two objectives represented in Equation 16, constraints (NC) and (C) are scaled using Equation 18.

$$
f_i' = \frac{f_i - f_i^L}{f_i^U - f_i^L} \qquad i = 1, 2 \tag{15}
$$

where f^L and f^U are the lower and upper bounds of each f^i. In the given example, the ideal compromise solution is 81,044 € with a value of $\sum_i u_i + \sum_i y_i = 701.17$ and $\sum_i y_i = 4$, representing 4 changes that take place in the model. The ideal compromise solution is plotted on Figure 7 with respect to other solutions on the Pareto front.

4.5. Overview of implementation of the algorithm on a large scale model

The four formulations (Approach 1, Approach 2, Approach 3 and Bi-Criterion model) were also implemented in the AIMMS software for a large scale supply chain model from Aurubis AG's supply Chain model for copper processing. The original model is a linear supply chain model that is a multi-period, multi-commodity model with the objective of maximizing revenue. Components of the Linear supply chain model involves 12 Raw Materials, 8 Suppliers, 4 Plants, 2 Distribution Centers, 16 Products, 120 Customers, and 12 Time periods.The three approaches discussed along with the bi-criterion optimization were successfully implemented on the large-scale model involving thousands of variables and constraints. These models were solved using CPLEX solver on a Dell XPS 15 7th Gen i7 Intel Core Processor with 8 GB

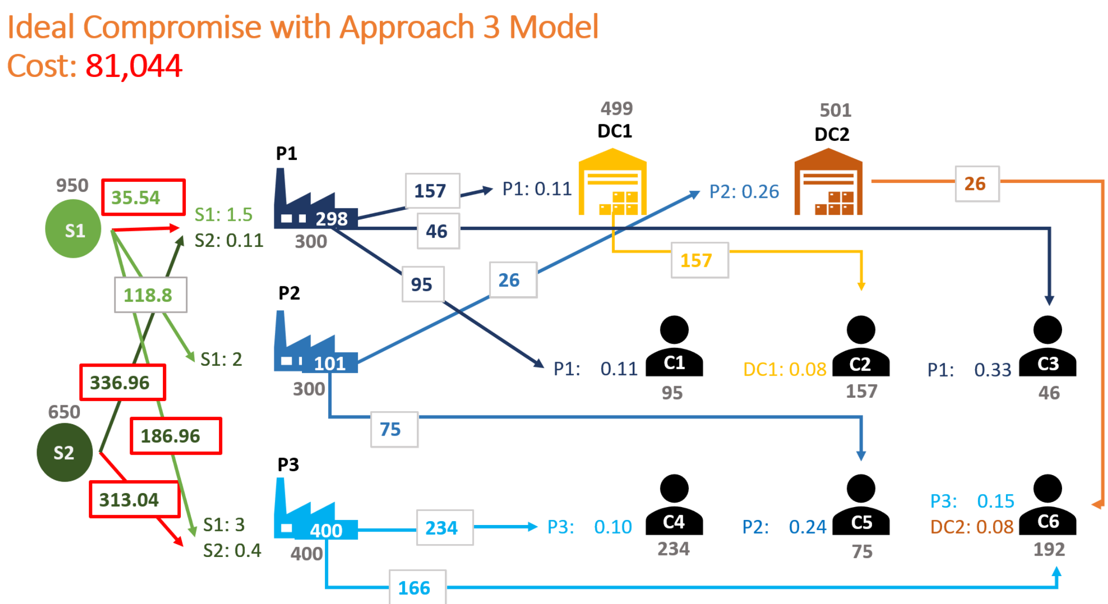

Figure 8: Ideal Compromise Solution obtained using (Equation 14)

RAM and 4 Cores. The computational times reported in table 4 is an average 5 runs each for running 12 different scenarios by perturbing different parameters in the model. We attribute the smaller solver times to the pre-solve capabilities of the solver.

For confidentiality reasons, we can not disclose the details of the model in this paper. However, the specifications of the models are outlined in the table 4.

Table 4: Model specifications for minimizing changes algorithm run on AIMMS

Model Formulation	Constraints	Variables	Binary Variables	Solver time [sec]
Original LP	6,121	15,953	0	<1
Approach 1	36,493	195,089	0	1.1
Approach 2	543,529	374,225	179,136	5.9
Approach 3	543,529	374,225	179,136	14.5
Bi-criterion	543,530	374,225	179,136	159.4

5. Conclusion and Summary

In this paper, we have addressed two major problems that arise in the implementation and interpretation of supply chain optimization models. The first problem is to diagnose and resolve infeasibilities. To resolve infeasibility, we provide a multi-step infeasibility diagnosis algorithm using the Flexibility Test method and the Regression Machine Learning techniques that finds the constraints or parameters causing infeasibility. It further provides quantitative prompts to resolve the infeasibility, without any help from the users or modeler.

The second problem considers the case when parameter changes are introduced to the supply chain network (e.g., raw materials availability). We propose and analyze three strategies, based on alternated formulations of the original supply chain that minimize the impact of the changes. We also show the trade-off using a bi-criterion problem for minimizing changes and minimizing the cost of the supply chain, which gives the decision-maker a tool to negotiate a plan of operation with their stakeholders. We conclude the section by providing the ideal compromise solution in the illustration problem. A future research direction is to classify the types of changes that occur in a supply chain layout and to differentiate between those changes beyond the cost as an objective.

Author Affiliations

Niharika Singh was affiliated with Carnegie Mellon University when she contributed to this paper. Pablo Garcia-Herreros was affiliated with Aurubis AG when he contributed to this paper.

Acknowledgement

The authors gratefully acknowledge the support from Aurubis AG and the Center for Advanced Process Decision-Making (CAPD).

Appendix A. Flexibility Test Method for diagnosing infeasibility in linear supply chain models

To appreciate the usefulness of the Flexibility Test method in analyzing the feasibility of an optimization model, we apply the algorithm to a linear programming model as in Equation 6 but involving only inequalities. Here, we have considered a bi-variate optimization problem .The constraints and the graphical representation of the model are presented in the Equations A.1 and Figure A.9 respectively.

$$
\begin{aligned}
min \quad & x_1 + x_2 \\
s.t. \quad & x_1 + 2x_2 \leq 8 \\
& 3x_1 + 2x_2 \leq 12 \\
& x_1 + x_2 \geq 7 \\
& x_1 + 0.3x_2 \geq 8 \\
& 5x_1 + x_2 \leq 1 \\
& x_1, x_2 \geq 0
\end{aligned}
\tag{A.1}
$$

This model is an instance of an infeasible optimization model and we apply the Flexibility Test method to find out which constraints are responsible for the Infeasibility.

$$
\begin{aligned}
min \quad & u_1 + u_2 + u_3 + u_4 + u_5 \\
s.t. \quad & x_1 + 2x_2 \leq 8 + u_1 \\
& 3x_1 + 2x_2 \leq 12 + u_2 \\
& - x_1 - x_2 \leq -7 + u_3 \\
& - x_1 - 0.3x_2 \leq -8 + u4 \\
& 5x_1 + x_2 \leq 1 + u_5 \\
& x_1, x_2 \geq 0 \\
& u_1, u_2, u_3, u_4, u_5 \geq 0
\end{aligned}
\tag{A.2}
$$

1. When solved, the violation terms u_3, u_4 and u_5 corresponding to constraints 3, 4 and 5 (c3, c4 and c5) were found to be non-zero (with magnitude 3, 6.8 and 3 respectively). The objective value $\sum_i u_i = \psi = 12.8$ was found to be positive, which indicates that the model is infeasible.

2. This indicates that the three corresponding constraints are responsible for the infeasibility and form the optimum combination of constraints for which the changes should be made to regain feasibility.

3. If we add slacks to the RHS of constraints 3, 4 and 5 (c3, c4 and c5) that are equal to the magnitude of u_3, u_4 and u_5 we can gain feasibility.

Appendix B. Review of Regression Models

B.1. Linear Regression

Linear regression is a linear approach to model a relationship between the dependent and independent variables (Montgomery et al., 2012). In machine learning, it

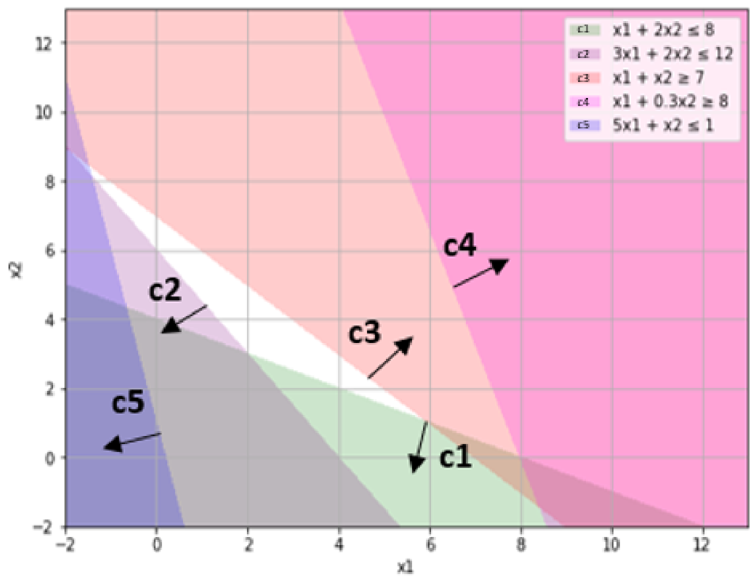

Figure A.9: Constraints graphically represented on x_1 and x_2 plane - no overlapping regions. As noted constraints c_3, c_4 and c_5 had non-zero u_j (violation terms)

is a supervised learning algorithm that is used to predict the value of a dependent variable based on the nature of past data of the independent variables. Given a data set $\{y, x_{i1}, \ldots, x_{ip}\}$ of m statistical units, a linear regression model assumes that the relationship between the dependent variable y and the p-vector of x is linear.

$$\begin{aligned}
Training\ \ Set: \\
[x_1, x_2, ...x_m] \\
[y_1, y_2, ...y_m] \\
\\
Predictive\ \ hypothesis: \\
h(x) = \theta^T x \\
y = \theta^T x + \epsilon
\end{aligned} \tag{B.1}$$

The aim is to find the optimal weights (θ) based on the values of the dependent (y) and the independent variables (x) using methods like Ordinary Least Square method (De Souza and Junqueira, 2005; Pavelescu, 2004), the Gradient Descent Method (Baldi, 1995), and Regularization (Bickel et al., 2006). In this paper, ordinary least square method has been used for regression analysis.

B.2. Ordinary least-squares method

For the linear model defined as follows:

$$Y = \beta_0 + \beta_1 X \tag{B.2}$$

β_0 and β_1 are the parameters (weights), X is the vector of the independent variables, and Y is the vector of dependent variables. Using ordinary least squares method, we are to determine the values of 0 and 1 such that they minimize the sum square of errors.

$$SSE = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 X) \tag{B.3}$$

In Equation B.3, y_i is the actual value of the dependent variable, and $\hat{y}_i$ is the value of the dependent variable as predicted by the machine learning model. We minimize the SSE while finding its partial derivative with respect to β_0 and β_1.

B.3. Confidence Interval of the predicted value

We need to quantify the confidence of the predicted value. A confidence interval is a bound on the estimate of a population variable (Myers, 1990). It is an interval statistic used to quantify the uncertainty on an estimate. The confidence interval (CI) was calculated using Equation B.4 :

$$CI = \hat{y} \pm t^*_{n-2} S_y \sqrt{\frac{1}{n} + \frac{(x^* - x)^2}{(n-1)S_x^2}} \tag{B.4}$$

where, $\hat{y}$ = predicted value, x^* = sample of x, $t^*_{n-2} = t$ value at 95% confidence, x = mean of S_y = Standard deviation of the residual S_x^2 = standard deviation of x, n = number of data points.

B.4.Predicting demand for data having seasonality and trends

Data in real-life are often erratic. They are usually non-linear, with several factors coming into play like uncertainties, trends, and seasonality. In addition, some data also have time-dependent variation. There are various techniques to model seasonality using techniques such as Time Series Decomposition, ARIMA (Autoregressive Integrated Moving Average), SARIMA (Seasonal Autoregressive Integrated Moving Average) and Exponential Smoothing. Since the data pertaining supply chain models have seasonality associated with them, SARIMA model is a suitable model to predict a parameter like demand (Vagropoulos et al., 2016; Martinez et al., 2011).

The SARIMA model has three hyperparameters, namely the autoregression order (p), the difference order (d), and the moving average order (q) for the trend component of the data. In addition to these hyperparameters, it has hyperparameters to specify the autoregression (AR), differencing (I), and moving average (MA) to account for the seasonal component of the series, as well as an additional parameter for the period of the seasonality.

The SARIMA model can be explained using a "cascading" model, where a non-stationary model forms the first component that is embedded into a larger stationary model which forms the second component.

$$\begin{aligned} Non-stationary: \\ Y_t = (1-L)^d X_t \\ Stationary: \\ (1-\sum_{i=1}^{p}\phi_i L^i)Y_t = (1+\sum_{i=1}^{q}\theta_i L^i)\epsilon_t \end{aligned} \tag{B.5}$$

Where, Y_t = Predicted forecast, p =Trend auto-regression order, L_i = Lag operator for seasonality, d =Trend difference order, θ_i = Moving average part for seasonality, q =Trend moving average order, ϕ_i = Auto-regressive part for seasonality and ϵ_t = noise.

SARIMA model is ideal for modeling a seasonal non-stationary time series using relatively few model parameters. However, because the mathematical definition does not contain model parameters that explicitly account for different means and variances in each season, the SARIMA model is not suitably designed for describing series having stationarity of second-order moments within each season across the years.

B.5. Implementing the machine learning technique on example linear supply chain parameter data

An instance is presented, which shows the utility of the regression machine learning model in predicting the customers' demands. This algorithm will be used to detect the problem in the data causing infeasibility.

Let us consider that there are six customers in a supply chain model (layout in Figure 3) with the past data of their product demand for 12 months. We need to predict what the demand would look like at a future time interval. We first draw the relationship between demand of customer over time based on the given data for the 12 month period using ordinary least squares method and determine the parameters β_0 and β_1 (see Appendix B). The linear regression model is formed as represented in Equation B.6, where we have X as the time period and Y as the demand of the customer. Solving for the parameters in Equation B.6 for customer 1, the following parameters and resultant graph with the 95% confidence interval were obtained:

$$\begin{aligned} Y &= 8.4091 \cdot X - 0.0757 \\ \beta_0 &= -0.0757 \\ \beta_1 &= 8.4091 \end{aligned} \tag{B.6}$$

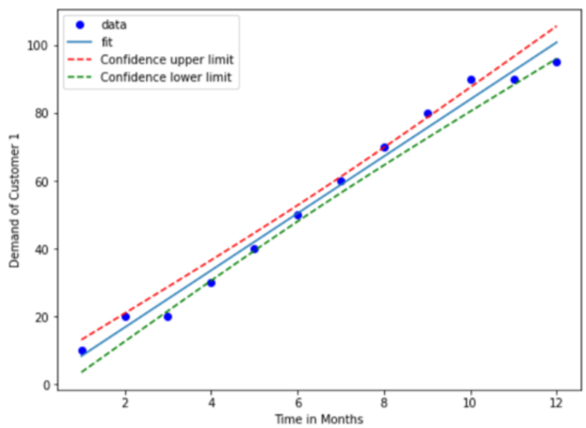

Figure B.10: Randomly generated data for six customers in a supply chain model over a period of 12 months on which regression analysis would be done with the Confidence interval

Data for 12 months was helpful in determining short term-operation parameters. However, our goal is to optimize over a period of a few years time. To do so, we model and make a forecast for demand similarly over monthly data for a period of 10 years. We utilized the data having seasonality and trend for the customer 1 over a period of 10 years to forecast demand using the SARIMA model.

The SARIMA model was implemented on Python without using any predefined machine learning package following the equations given in section Appendix B. When the SARIMA model was applied, conclusive interpretations were obtained concerning the seasonality and trend of the data depicted. It tells us that over a period of 10 years, the overall trend is that the customer demand is increasing in an almost linearly, and during each year and that the demand peaks during the summer months. To obtain the forecast, SARIMA model was solved to obtain a set of hyperparameters of the demand model that provides predictions for the future. Figure B.12 shows the forecast of customer demands for the next two years with a confidence of 95%.

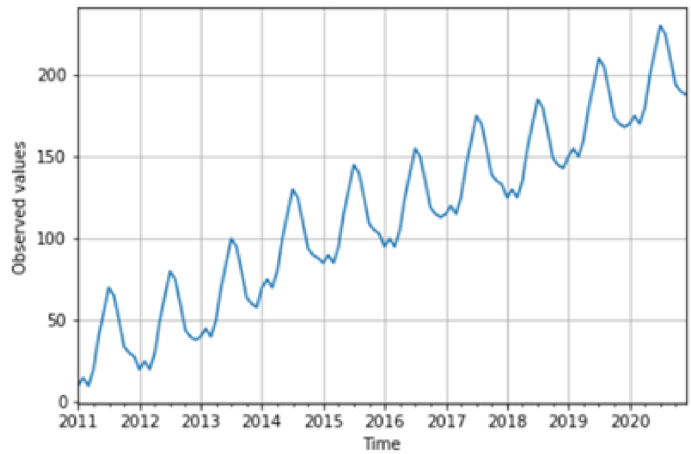

Figure B.11: Customer Demand Data over a period of 10 years

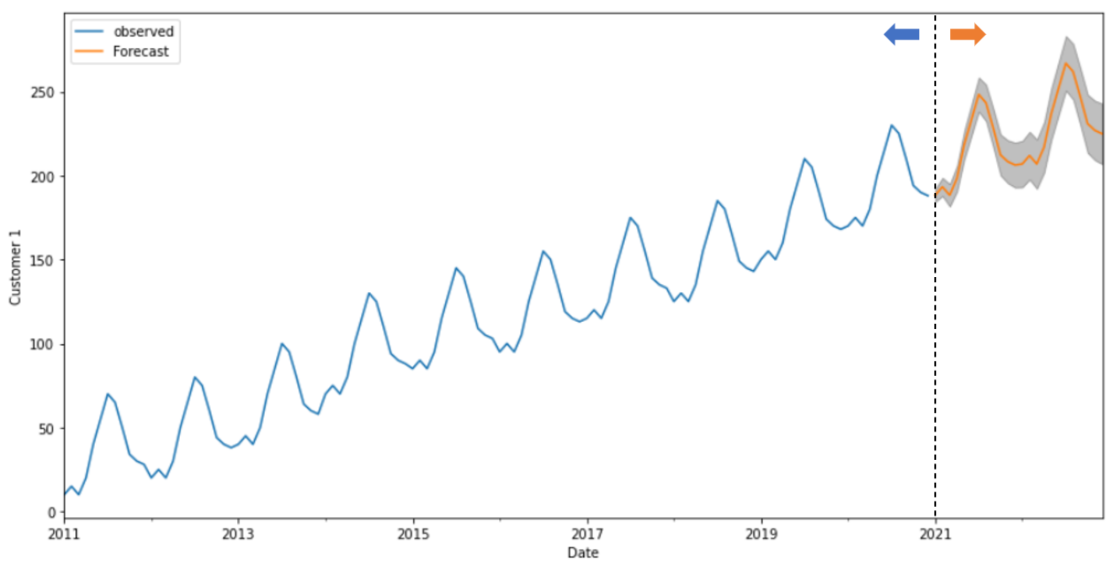

Figure B.12: SARIMA model used to predict the demand for customer C1. The graph to the right of the dotted line represents the predicted demand based on the historical data simulated using linear regression which is plotted to the left of the dotted line

Appendix C. Parameters and Data for the example supply chain problem

The parameters are ω = Capacity of the supplier, κ = Capacity of the Plant, ϕ = Capacity of the distribution center, λ = conversion rate of materials in the plant and D = Demand of customers. The sets S, P, DC and C represent suppliers, plants, distribution centers (warehouses) and customers. There are four kinds of flows f_{xy} that represent flows from Suppliers to Plants (S,P), Plants to Customers (P,C), Plants to Distribution Centers (P, DC) and Distribution Centers to Customers (DC, C) represented in the problem layout. Each flow has a transportation cost associated with it that represent the cost to transport the material from one node to another. The parameters a, b, c, and d correspond the to transportation costs for the flow of material from Suppliers to Plants (S,P), Plants to Customers (P,C), Plants to Distribution Centers (P, DC) and Distribution Centers to Customers (DC, C) respectively. Parameter e represents the cost of production at the plants.

C.1. Parameters for supply chain example problem

Base case parameter values for the example supply chain problem used in Section 3 and section 4.

C.1.1. Transportation Costs

Table C.5: Transportation costs from supplier to plants in base case model (a)

Base Case	Supplier 1	Supplier 2
Plant 1	0.15	0.11
Plant 2	0.2	0.23
Plant 3	0.3	0.4

Table C.6: Transportation costs from plants to distribution centers in base case model (b)

Base Case	Plant 1	Plant 2	Plant 3
Distribution Center 1	0.11	0.21	0.31
Distribution Center 2	0.41	0.26	0.21

Table C.7: Transportation costs from plants to customers in base case model (c)

Base Case	Plant 1	Plant 2	Plant 3
Customer 1	0.11	0.10	0.40
Customer 2	0.22	0.25	0.50
Customer 3	0.33	0.35	0.20
Customer 4	0.44	0.45	0.10
Customer 5	0.55	0.24	0.22
Customer 6	0.66	0.45	0.15

Table C.8: Transportation costs from distribution centers to customers in base case model (d)

Base Case	Distribution Center 1	Distribution Center 2
Customer 1	0.04	2.88
Customer 2	0.08	1.32
Customer 3	0.36	1.04
Customer 4	0.88	0.52
Customer 5	1.52	0.12
Customer 6	3.36	0.08

C.1.2. Capacity parameters

Table C.9: Capacity of Suppliers (ω)

Base	Supplier 1	Supplier 2
Capacity	950	650

Table C.10: Capacity of Plants (κ)

Base Case	Plant 1	Plant 2	Plant 3
Capacity	300	300	400

Table C.11: Capacity of Distribution Centers (ϕ)

Base Case	Distribution Center 1	Distribution Center 2
Capacity	499	501

C.1.3. Other Parameters

Table C.12: Cost of converting raw-materials to products at each plant (λ)

Base Case	Conversion
Plant 1	100
Plant 2	100
Plant 3	100

Table C.13: Rate of conversion of raw-materials to products at each plant (λ)

Base Case	Conversion
Plant 1	0.8
Plant 2	0.85
Plant 3	0.8

Table C.14: Customer Demand D

Base Case	Demand
Customer 1	95
Customer 2	157
Customer 3	46
Customer 4	234
Customer 5	75
Customer 6	132

References

Baldi, P., 1995. Gradient Descent Learning Algorithm Overview: A General Dynamical Systems Perspective. IEEE Transactions on Neural Networks 6, 182–195. doi:10.1109/72.363438.

Bansal, V., Perkins, J.D., Pistikopoulos, E.N., 2002. Flexibility analysis and design using a parametric programming framework. AIChE Journal 48, 2851. doi:10.1002/aic.690460212.

Barbosa-Póvoa, A.P., 2012. Progresses and challenges in process industry supply chains optimization. Current Opinion in Chemical Engineering 1, 446–452. doi:10.1016/J.COCHE.2012.09.006.

Barbosa-Povoa, A.P., Pinto, J.M., 2020. Process supply chains: Perspectives from academia and industry. Computers and Chemical Engineering 132. doi:10.1016/J.COMPCHEMENG.2019.106606.

Bhosekar, A., Ierapetritou, M., 2018. Advances in surrogate based modeling, feasibility analysis, and optimization: A review. Computers Chemical Engineering 108, 250–267. doi:10.1016/j.compchemeng.2017.09.017.

Bickel, P., Li, B., Tsybakov, A., Geer, S., Yu, B., Valdés, T., Rivero, C., Fan, J., Vaart, A., 2006. Regularization in statistics. TEST: An Official Journal of the Spanish Society of Statistics and Operations Research 15, 271–344. doi:10.1007/BF02607055.

Bok, J.K., Grossmann, I.E., Park, S., 2000. Supply chain optimization in continuous flexible process networks. Industrial Engineering Chemistry Research 39, 1279–1290. doi:10.1021/ie990526w.

Cafaro, D.C., Grossmann, I.E., 2014. Strategic planning, design, and development of the shale gas supply chain network. AIChE Journal 60, 2122–2142. doi:10.1002/AIC.14405.

De Souza, S.V., Junqueira, R.G., 2005. A procedure to assess linearity by ordinary least squares method. Analytica Chimica Acta 552, 25–35. doi:10.1016/J.ACA.2005.07.043.

Dias, L.S., Ierapetritou, M.G., 2019. Data-driven feasibility analysis for the integration of planning and scheduling problems. Optimization and Engineering 20, 1029–1066. doi:10.1007/S11081-019-09459-W.

Garcia, D.J., You, F., 2015. Supply chain design and optimization: Challenges and opportunities. Computers & Chemical Engineering 81, 153–170. doi:10.1016/J.COMPCHEMENG.2015.03.015.

Goyal, V., Ierapetritou, M.G., 2002. Determination of operability limits using simplicial approximation. AIChE Journal 48, 2902–2909. doi:10.1002/aic.690481217.

Greenberg, H.J., 1993. How to Analyze the Results of Linear Programs—Part 3: Infeasibility Diagnosis. INFORMS Journal on Applied Analytics 23, 120–139. doi:10.1287/inte.23.6.120.

Greenberg, H.J., 1996. The ANALYZE rulebase for supporting LP analysis. Annals of Operations Research 65, 91–126. doi:10.1007/BF02187328.

Grossmann, I.E., 2005. Enterprise-wide optimization: A new frontier in process systems engineering. AIChE Journal 51, 1846–1857. doi:10.1002/AIC.10617.

Grossmann, I.E., Calfa, B.A., Garcia-Herreros, P., 2014. Evolution of concepts and models for quantifying resiliency and flexibility of chemical processes. Computers Chemical Engineering 70, 22–34. doi:10.1016/j.compchemeng.2013.12.013.

Grossmann, I.E., Drabbant, R., Jain R., K., 1982. Incorporating Toxicology in the Synthesis of Industrial Chemical Complexes. Chemical Engineering Communications 17, 151–170. doi:10.1080/00986448208911622.

Grossmann, I.E., Floudas, C.A., 1987. Active constraint strategy for flexibility analysis in chemical processes. Computers & Chemical Engineering 11, 675–693. doi:10.1016/0098-1354(87)87011-4.

Halemane, K.P., Grossmann, I.E., 1983. Optimal process design under uncertainty. AIChE Journal 29. doi:10.1016/0098-1354(87)87011-4.

Harjunkoski, I., Maravelias, C.T., Bongers, P., Castro, P.M., Engell, S., Grossmann, I.E., Hooker, J., Méndez, C., Sand, G., Wassick, J., 2014. Scope for industrial applications of production scheduling models and solution methods. Computers & Chemical Engineering 62, 161–193. doi:10.1016/J.COMPCHEMENG.2013.12.001.

León, T., Liern, V., 2001. A fuzzy method to repair infeasibility in linearly constrained problems. Fuzzy Sets and Systems 122, 237–243. doi:10.1016/S0165-0114(00)00010-5.

Maravelias, C.T., Sung, C., 2009. Integration of production planning and scheduling: Overview, challenges and opportunities. Computers & Chemical Engineering 33, 1919–1930. doi:10.1016/J.COMPCHEMENG.2009.06.007.

Martinez, E.Z., da Silva, E.A.S., dal Fabbro, A.L., 2011. A SARIMA forecasting model to predict the number of cases of dengue in Campinas, State of São Paulo, Brazil. Revista da Sociedade Brasileira de Medicina Tropical (Journal of the Brazilian Society of Tropical Medicine) 44, 436–440. doi:10.1590/s0037-86822011000400007.

Mavrotas, G., 2009. Effective implementation of the ϵ-constraint method in Multi-Objective Mathematical Programming problems. Applied Mathematics and Computation 213, 455–465. doi:10.1016/J.AMC.2009.03.037.

Montgomery, D.C., Peck, E.A., Vining, G.G., 2012. Introduction to linear regression analysis. Wiley series in probability and statistics. 5th ed., Wiley, Hoboken, NJ.

Myers, R.H., 1990. Classical and modern regression with applications. 2nd ed. ed., PWS-KENT.

Pavelescu, F.M., 2004. Features Of The Ordinary Least Square (Ols) Method. Implications For The Estimation Methodology. Journal for Economic Forecasting 1, 85–101.

Perez, H.D., Amaran, S., Erisen, E., Wassick, J.M., Grossmann, I.E., 2021. A Digital Twin Framework for Business Transactional Processes in Supply Chains. Computer Aided Chemical Engineering 50, 1755–1760. doi:10.1016/B978-0-323-88506-5.50272-2.

Pistikopoulos, E.N., Grossmann, I.E., 1988. Optimal retrofit design for improving process flexibility in linear systems. Computers & Chemical Engineering 12, 719–731. doi:10.1016/0098-1354(88)80010-3.

Puranik, Y., Sahinidis, N.V., 2017. Deletion presolve for accelerating infeasibility diagnosis in optimization models. INFORMS Journal on Computing 29, 754–766. doi:10.1287/IJOC.2017.0761.

Puranik, Y., Samudra, A., Sahinidis, N.V., Smith, A.B., Sayyar-Rodsari, B., 2018. Infeasibility resolution for multi-purpose batch process scheduling. Computers and Chemical Engineering 116, 69–79. doi:10.1016/J.COMPCHEMENG.2018.03.005.

Swaney, R.E., Grossmann, I.E., 1985. An index for operational flexibility in chemical process design. part i: Formulation and theory. AIChE Journal 31, 621–630. doi:10.1002/aic.690310412.

Vagropoulos, S.I., Chouliaras, G.I., Kardakos, E.G., Simoglou, C.K., Bakirtzis, A.G., 2016. Comparison of SARIMAX, SARIMA, modified SARIMA and ANN-based models for short-term PV generation forecasting. IEEE International Energy Conference (ENERGYCON) 2016 doi:10.1109/ENERGYCON.2016.7514029.

You, F., Grossmann, I.E., 2008. Design of responsive supply chains under demand uncertainty. Computers & Chemical Engineering 32, 3090–3111. doi:https://doi.org/10.1016/j.compchemeng.2008.05.004.

Zhang, Q., Grossmann, I.E., Lima, R.M., 2016. On the relation between flexibility analysis and robust optimization for linear systems. AIChE Journal 62, 3109–3123. doi:10.1002/AIC.15221.

Zhang, Q., Sundaramoorthy, A., Grossmann, I.E., Pinto, J.M., 2017. Multiscale production routing in multicommodity supply chains with complex production facilities. Computers & Operations Research 79, 207–222. doi:10.1016/J.COR.2016.11.001.