



Modeling and heuristics for production time crashing in supply chain network design

Yi Liao¹ · Ali Diabat^{2,3}  · Chaher Alzaman⁴ · Yiqiang Zhang⁵

Published online: 18 February 2020

© Springer Science+Business Media, LLC, part of Springer Nature 2020

Abstract

Supply chains with shorter lead times can bring their constituents cost reductions, flexibility, and speed. Since manufacturing is a prominent operation within the supply chain, the reduction of its time duration can prove important in reducing the overall supply chain's lead-time. Some works in the area of supply chain network design (SCND) have looked at the crashing of supply chain's lead-time. However, the literature lacks works that explicitly model a crashing cost in SCND. The work formulates a cost model that integrates production, crashing, inventory, transportation, and plant selection. Given the complexity of these elements, the model emerging is a nonlinear and binary in both the objective function and the constraints. The paper introduces a gradient search method to solve the model coupled with efficient search heuristics. The work presents seven search heuristics along with variants to solve the difficult problem at hand. Further, the work looks at different parameters that affect the crashing cost, presents the cost avoidances that can result from crashing, and discusses the operational opportunities to be reaped.

Keywords Supply chain network design · Manufacturing · Resource allocation · Gradient search · Nonlinear mixed integer programming · Local search heuristic

✉ Yiqiang Zhang
zyq@lixin.edu.cn

¹ School of Business Administration, Southwestern University of Finance and Economics, Chengdu 610074, Sichuan, China

² Division of Engineering, New York University Abu Dhabi, Saadiyat Island, Abu Dhabi 129188, United Arab Emirates

³ Department of Civil and Urban Engineering, Tandon School of Engineering, New York University, Brooklyn, NY 11201, United States

⁴ Department of Supply Chain and Business Technology Management, John Molson School of Business, Concordia University, Montréal, QC H3H 0A1, Canada

⁵ Xulun Honor School, Shanghai Lixin University of Accounting and Finance, Shanghai 201209, China

1 Introduction

Beamon (1998) suggests that a supply chain may be defined as an integrated process of various business entities interacting with each other. Supply chain network design (SCND), on the other hand, is a modeling module that aims at optimizing an organization's resources. In a classical SCND problem, attributes such as cost, location, and functionality are lumped in an objective function to be optimized accompanied with operational constraints.

The aim of this work is to reduce the manufacturing or production time (Pt), which is an important component of the overall supply chain lead-time. Lead-time in supply chain operations is an area receiving much attention with numerous works highlighting the importance of its reduction. However, there is a gap in the research for works that look specifically at Pt reduction. The work of Towill (1996) discusses the importance of reducing Pt as a competitive weapon in supply chains. As mentioned, reducing Pt cuts the overall supply chain lead-time. This, in turns, allows firms to be more flexible to demand changes. The advantages of reducing Pt are many. First, with shorter lead-time firms are able to order closer to selling season or consumption point. In illustration, the work of Blackburn (1991) shines light on the operations of the retail giant, Wal-Mart. According to the investigation, if the retailer orders 16 weeks instead of 26 weeks before the selling season, the error in forecasting market demand drops from 40 to 20% (Blackburn 1991). Secondly, reducing lead-time can prove important in industries, which are time sensitive. For instance, in the aerospace industry, aircraft manufacturers pay heavy penalties for late deliveries. Third, since this work looks specifically at Pt , an important benefit of reducing Pt is the attainment of excess capacity. In elaboration, the reduction in Pt frees production resources and hence creates excess capacity from which more output can be produced. Hence, the objective of this work is the reduction of Pt by crashing production time. In this work, crashing cost is the cost incurred for shortening production duration by some time, S . However, crashing Pt requires a deeper understanding of the production cost.

Another important contribution of this work is the formulation of heuristics to search the nonlinear and discrete solution space. Given the surge in papers that present nonlinear and binary mathematical models (Table 1), works that analyze the accuracy of heuristics are necessary. The literature shows a number of works that introduce nonlinear binary models but the body of work lacks work that contests a set of heuristics to solve the corresponding problems. In contrast, this paper devises seven heuristics and analyses the effectiveness and accuracy of each.

The paper is organized in the following manner. Previous relevant works are presented and reviewed in Sect. 2. Section 3 presents important definitions, a general description of the model, and a discussion of the challenges present in it. Section 4 presents the model assumptions and the model itself. Section 5 explains the challenges of solving the nonlinear binary model at hand. It introduces all seven solution heuristics to be used (gradient and binary search heuristics). Section 6 presents comparisons between the different heuristics, validates the accuracy of each of the heuristics, and presents the overall results of the model. At the end of the section, parameter analysis is performed to assess important variations of the model. Section 7 discusses the benefits, limitations, and future prospect of the research.

2 Literature review

The field of SCND has seen a significant number of works. The earliest work in this area can be traced to Geoffrion and Graves (1974). They introduce a multi-commodity logistics network model for optimizing the flow of finished products from plants to distribution centers.

Table 1 Relevant SCND literature

| MP approach (1) | Inventory cost (2) | Transport cost (3) | Shorten (crash) lead time (4) | Production cost (components) (5) | Integer search heuristic (6) | Solution robustness (7) |
|-----------------------------|--------------------|--------------------|-------------------------------|----------------------------------|------------------------------|-------------------------|
| Arntzen et al. (1995) | MIP ✓ | ✓ | X | X | X | X |
| Babazadeh et al. (2017) | MP ✓ | ✓ | X | X | X | X |
| Badria et al. (2017) | SP ✓ | ✓ | X | X | ✓ | ✓ |
| Cakravastia et al. (2002) | MIP ✓ | ✓ | ✓ | X | X | X |
| Coskun et al. (2016) | MuIP ✓ | ✓ | X | X | X | X |
| Diabat (2016) | BP ✓ | X | X | X | ✓ | ✓ |
| Diabat and Al-Salem (2015) | Stoch ✓ | ✓ | X | X | ✓ | ✓ |
| Diaby et al. (2013) | NMP ✓ | X | X | X | ✓ | X |
| Esmailikia et al. (2016) | NMIP ✓ | ✓ | X | X | ✓ | ✓ |
| Fahimnia et al. (2015) | MIP ✓ | ✓ | X | X | ✓ | X |
| Govindan et al. (2014) | MuIP ✓ | ✓ | X | X | ✓ | ✓ |
| Govindan and Fattahi (2017) | Stoch ✓ | ✓ | X | X | ✓ | ✓ |
| Hammani and Frein (2013) | NMIP ✓ | ✓ | X | X | X | ✓ |
| Hasani et al. (2015) | NMP ✓ | ✓ | X | X | ✓ | ✓ |
| Hoque and Goyal (2006) | NA ✓ | X | ✓ | X | X | X |
| Jayaraman and Ross (2003) | MIP ✓ | ✓ | X | X | ✓ | ✓ |

Table 1 continued

| | MP approach (1) | Inventory cost (2) | Transport cost (3) | Shorten (crash) lead time (4) | Production cost (components) (5) | Integer search heuristic (6) | Solution robustness (7) |
|----------------------------|-----------------|--------------------|--------------------|-------------------------------|----------------------------------|------------------------------|-------------------------|
| Jian et al. (2015) | NA | X | X | ✓ | X | X | X |
| Kaya and Urek (2016) | NMIP | ✓ | X | X | X | ✓ | ✓ |
| Keyvanshokoh et al. (2016) | Stoch | ✓ | ✓ | X | X | ✓ | ✓ |
| Lim et al. (2016) | MP | ✓ | ✓ | X | X | X | X |
| Longimidis et al. (2015) | MIP | ✓ | ✓ | X | X | X | X |
| Mangla et al. (2016) | MIP | ✓ | ✓ | X | X | ✓ | ✓ |
| Marti et al. (2015) | MIP | X | ✓ | X | X | ✓ | X |
| Mizgier (2017) | SA | X | X | ✓ | X | X | X |
| Pan and Nagi (2013) | MIP | ✓ | ✓ | X | X | ✓ | ✓ |
| Petridis (2015) | NMIPT | X | ✓ | X | X | ✓ | ✓ |
| Pham and Yenradee (2017) | MuP | X | ✓ | X | X | ✓ | ✓ |
| Pishvae and Torabi (2010) | MIP | X | ✓ | X | X | ✓ | ✓ |
| Ramudhin et al. (2008) | MP | X | ✓ | X | X | X | X |
| Rezapour et al. (2017) | NMP | ✓ | ✓ | X | X | ✓ | ✓ |

Table 1 continued

| | MP approach (1) | Inventory cost (2) | Transport cost (3) | Shorten (crash) lead time (4) | Production cost (components) (5) | Integer search heuristic (6) | Solution robustness (7) |
|--------------------------------|-----------------|--------------------|--------------------|-------------------------------|----------------------------------|------------------------------|-------------------------|
| Sadjady and Davoudpour (2012) | MIP | ✓ | ✓ | X | X | ✓ | ✓ |
| Santos et al. (2005) | SP | X | ✓ | X | X | ✓ | ✓ |
| Sarrafa et al. (2015) | NMIP | ✓ | ✓ | X | X | ✓ | ✓ |
| Vahdani and Mohammadi (2015) | MuP | X | ✓ | X | X | ✓ | ✓ |
| Varsei and Polyakovskiy (2017) | MuIP | ✓ | ✓ | X | X | ✓ | X |
| Yang (2010) | NA | ✓ | X | ✓ | X | X | X |
| Yildiz et al. (2016) | MuP | ✓ | X | X | X | ✓ | X |

MP mathematical programming (no integers), MuP multi-objective integer programming, MuIP multi-objective integer programming, MMP nonlinear mathematical programming, NMIP nonlinear mixed integer programming, SA sensitivity analysis, SP stochastic programming, BP binary programming, Stoch stochastic, MA numerical analysis

Arntzen et al. (1995) provide a deterministic model for supply chain network designs. Table 1 gives a snapshot of the literature, with emphasis on recent publications in SCND. Most of the papers in the table meet the general description of the supply chain with multiple echelons focusing on suppliers, manufacturers, and/or distributors. Overall, the state-of-the-art tends to be populated with models that look at transportation and inventory costs. However, few papers look at the individual components of the production cost. Papers devising solutions that search a discrete solution space are many. However, papers that deal with nonlinear and binary variables are a minority.

Relevant to this work, Cakravastia et al. (2002) present a mixed integer programming (MIP) model to analyze different production scenarios for ultimately performing supplier selection. The work devises a crashing cost where suppliers have two choices: crash production or normal production. The work presents a crashing cost per unit, which can be analogous to the crashing cost presented in this paper. However, the work of Cakravastia et al. (2002) fails to dig into the relationship between crashing and normal costs. The work of Hoque and Goyal (2006) deals with crashing lead-time; here, the crashing pertains to preparation to a series of elements such as time, supplier lead-time, and other elements. This is quite different from the objective of the paper, which looks specifically and exclusively at production time. Jian et al. (2015) look at the crashing cost of production; however, their work focuses on balancing the trade-off between demand forecasting risk and production cost. Yang (2010) looks at crashing the lead-time and not specifically at production time. The work utilizes a polynomial function to model crashing, which is nonlinear. This work also utilizes a nonlinear convex function to model crashing. The work of Diaby et al. (2013) looks at shortening cycle time by reducing setup cost in a capacitated production setting. They present an exponential function, which depicts the capital investment required to shorten production setup time. However, the work looks at the shortening of setup time and not the whole production cycle. Moving to other works, the trade-off between normal costs and crashing costs in the context of SCND is absent. The work of Esmailikia et al. (2016) differentiates between regular time and overtime costs. However, the work does not address the crashing of production time. The work of Mizgier (2017) looks at direct and indirect costs. However, these costs are associated with risk. For instance, his/her work defines the crashing cost as the cost of property damage for a firm due to a given hazard.

In terms of solution procedures, Hammami and Frein (2013) present a model for a global multi-echelon supply chains with lead-time constraints. They devise a mathematical model with nonlinearity in the constraint. The authors linearize the constraints and solve the model using CPLEX. Kaya and Urek (2016) present a closed-loop supply chain that integrates a combination of three sub problems: location, pricing, and inventory. The modeling results in a nonlinear problem. The authors utilize a gradient-based search method to solve the model. Jayaraman and Ross (2003) introduce a global distribution system design that utilizes simulated annealing methodology. Merz and Freisleben (2002) present a greedy heuristic and two local search algorithms, 1-opt local search and k-opt local search. Their work is the basis of the heuristic used in this paper with the incorporation of a feasibility check criteria. Many works (Badria et al. 2017; Diabat and Al-Salem 2015; Diaby et al. 2013; Esmailikia et al. 2016; Fahimnia et al. 2015; Govindan et al. 2014; Govindan and Fattahi 2017; Hasani et al. 2015; Jayaraman and Ross 2003; Kaya and Urek 2016; Keyvanshokoh et al. 2016; Mangla et al. 2016; Marti et al. 2015; Petridis 2015; Pan and Nagi 2013; Pishvae and Torabi 2010; Pham and Yenradee 2017; Rezapour et al. 2017; Sadjady and Davoudpour 2012; Santos et al. 2005; Sarrafha et al. 2015; Vahdani and Mohammadi 2015; Varsei and Polyakovskiy 2017; Yildiz et al. 2016) utilize integer-based search heuristics (designed for binary variables).

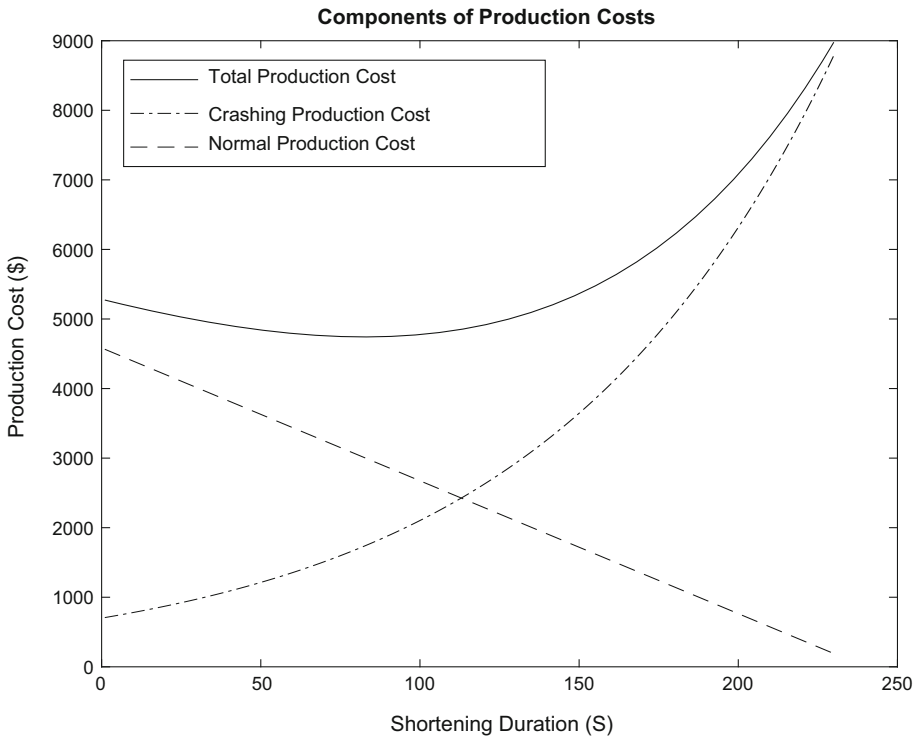


Fig. 1 Crashing and normal production costs

Historically in the SCND literature, components of the production cost have been lumped into one cost. In Table 1, we summarize the literature review we've carried. Column five illustrates a gap in works that look at production cost components. To remedy this, the work looks at two very important components of the production cost: crashing and normal costs. Crashing costs encompass variable costs that are geared towards the shortening of Pt . On the other hand, normal production costs include classical costs such as rent, electricity, and labor. Seeing the production cost in terms of crashing and normal costs allows one to infer a relationship between the crashing of Pt and its consequential overall production cost (Fig. 1). This nonlinear relation highlights the contribution of this work, since it is a novel way of looking at Pt crashing. Figure 3 illustrates the classical crashing cost behavior coming from the Project Management discipline (Stevenson et al. 2007). In the figure, we use the data presented later in Table 3.

3 Problem description

Before getting to the description of the problem at hand, it is important to introduce important definitions. The shortening length (S) is the time a production run is shortened. Production time (Pt) is the time required to produce the planned production quantity (Qo). Qo is the initial quantity a plant produces during a production cycle if there is no shortening (S). Production cycle (PC) is the original production run time before crashing. Whenever a production run is shortened, the production cycle remains the same while an excess capacity forms. The

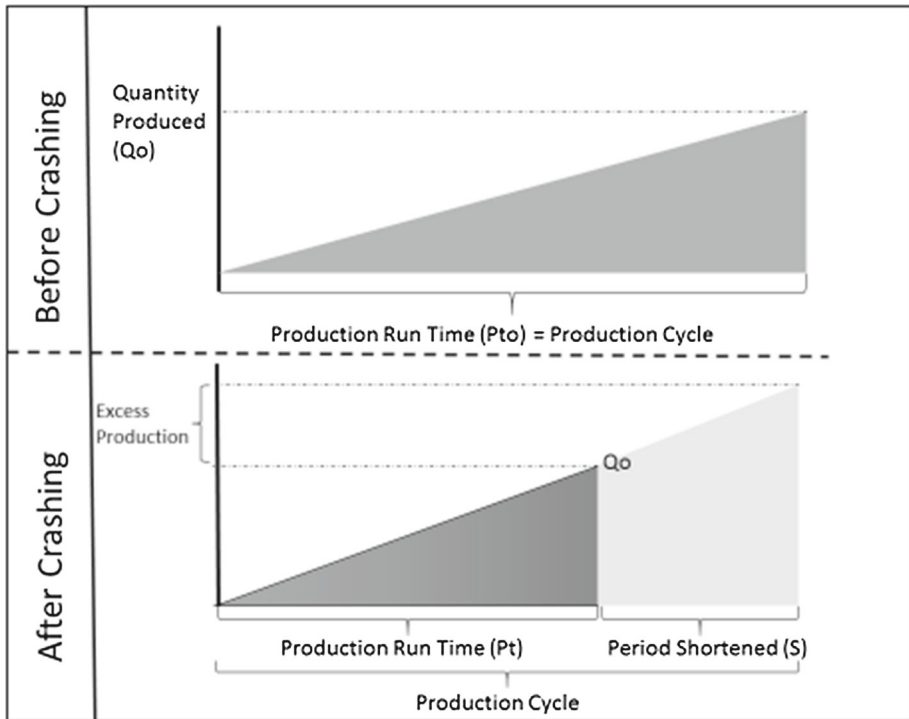


Fig. 2 Pt before and after crashing

demand, D , is the quantity of units demanded during a production cycle. Here, the demand remains constant for each time period regardless of crashing. Figure 2 visually illustrates the production cycle before crashing, where a production cycle is exactly equal the production time (Pto). After crashing, the production cycle remains the same but the production time is shortened (Fig. 2). The demand rate remains the same and it is per production cycle. Notably, the crashing creates an excess capacity, of which a plant can increase output as shown in Fig. 2. In context of the overall supply chain, excess capacity at one plant might alter product assignments. For instance, if a product is assigned to two different plants but one plant crashes production, the output at the other plant might be reduced or even stopped altogether.

For the purpose of this study, normal costs include facility costs (property rental/mortgage/financing costs), electricity costs, labor costs, machining costs (rental costs or depreciation), and supervision costs. Crashing costs include the resources needed to shorten Pt . It includes incremental labor costs due to additional hiring, part-time costs, overtime costs, excess machine costs (additional machining), and acquisition costs of superior production process technologies. Reducing the production time (Pt) results in additional production costs but these costs are offset by a reduction in the overall production costs (crashing plus normal) as the overhead costs is spread over larger production volumes (i.e., more lot sizes per time period). Figure 1 illustrates this, where the total production cost (normal + crashing) is convex and hence has a unique minimum.

The normal production cost (NC) is a function of Pt and is proportional to Pt . Normal costs (NC) can be written as follows.

$$NC = CPt = C(Pto - S) \tag{3.1}$$

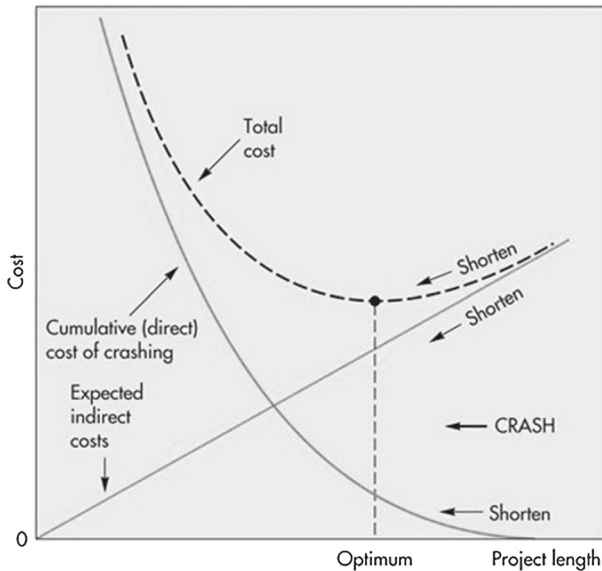


Fig. 3 Cumulative cost of crashing activities in Project Management (taken from Stevenson et al. 2007)

C is an overhead cost parameter that is dependent on workforce costs, property costs, machine rental costs, and production lot size. Pt_o is the length of the production time if no crashing is applied.

The crashing cost of production (CC) is the cost of the additional resources (e.g., additional machines, additional labor, and/or acquisition of superior production process technology) required to shorten Pt by time S . CC is a function of S . This is analogous to other works in the literature. We see a similar phenomenon in Project Management and in particular project crashing (see Fig. 3). When we shorten (i.e., crash) a given activity within a project, the cost of crashing increases exponentially. Elsewhere, the work of Diaby et al. (2013) presents an exponential function to depict the capital investment required to shorten production setup time. Based on their work, the CC can be written as follows.

$$CC = \alpha e^{\beta S} \tag{3.2}$$

α is a costing parameter that is dependent on the production environment. It is a positive number ($\alpha \geq 0$). β is the exponential factor, which is a positive real number between zero and one ($\beta \geq 0$). Both terms were necessary in the work of Diaby et al. (2013) and are used in this work as they construct mathematical functions that mimic the classical behavior of the crashing cost (Figs. 1, 3).

4 Model

The model is a multi-period and multiproduct system. The supply chain has assignable costs (i.e., binary variables). The production at each plant is communicated in terms of lot sizes while the production time, Pt , is a variable. The objective of the work is to reduce the overall production lead-time which, in turn, can reduce the overall supply chain lead-time. The following assumptions are necessary for the conceptualization of the model.

Assumptions:

1. Inventory is held at suppliers and the cost of carrying inventory is visible to all supply chain partners.
2. No inventory, or negligible inventory, is held at the plants.
3. Suppliers procure raw material or components at a procurement cost.
4. Parts arrive from the suppliers to the plants just before the start of each production cycle so no inventory is needed to be held at plants.
5. Products are shipped from plants to distributors at the end of each production cycle.
6. Demand is a constant and is per production cycle for each period.
7. Plants with excess capacity (due to crashing) can increase output.
8. Distributors are the end customers for our network and hence operational parameters (e.g. inventory) at the distributors are outside the boundaries of our model.
9. Before production cycle one, all initial condition are based on Pto .
10. This is a multiproduct supply chain model, which means if a product is not assigned to a plant, the plant is still open for production on other products. This is quite common practice for plants to work on more than one product.
11. The normal costs of production are incurred per a given time period for a set quantity of units. The increase of production throughput (i.e., increase of production quantity), due to the shortening of Pt , results in unitary cost reduction. This assumption is grounded in most manufacturing settings where economy of scale brings unit production savings (Alzaman et al. 2018; Stevenson et al. 2007).
12. More than one production run can occur in a production cycle, if Pt 's shortening is achieved.

The model is as follows.

Sets:

- I = Group of suppliers.
- J = Group of Manufacturing Plants.
- K = Set of Distributors.
- P = Set of all product types.
- R = Set of part types that are predecessors to P .
- T = Set of production cycles.

Parameters:

- $Pto_{j,p,t}$ = Standard production run time at plant j , for product p , if no crashing is applied, during production cycle t ; $j \in J, p \in P, t \in T$.
- $Q_{j,p,t}$ = Production patch size (Lot size) given $Pto_{j,p,t}$, for product p at plant j during production cycle t ; $j \in J, p \in P, t \in T$.
- $PCr_{i,r,t}$ = Procurement cost at supplier i , for one unit of part r , during production cycle t ; $i \in I, r \in R, t \in T$.
- $IC_{i,r,t}$ = Inventory holding cost for holding inventory at supplier i , for one unit of part r , per production cycle t ; $i \in I, r \in R, t \in T$.
- $Fix_{j,p,t}$ = Fixed cost for assigning product p to plant j , during production cycle t ; $j \in J, p \in P, t \in T$.
- $Tr_{i,j,r,t}$ = Unit transportation cost of transporting a part r from supplier i to plant j during production cycle t ; $i \in I, j \in J, r \in R, t \in T$.
- $Tp_{j,k,p,t}$ = Unit transportation cost of transporting a product p from plant j to distributor k during production cycle t ; $j \in J, k \in K, p \in P, t \in T$.

$SCap_{i,r,t}$ = Allowable capacity of part r that can be held at supplier i , during production cycle t ; $i \in I, r \in R, t \in T$.

$PCap_{j,p,t}$ = Allowable capacity of product p that can be produced at plant j , during production cycle t ; $j \in J, p \in P, t \in T$.

$D_{k,p,t}$ = Number of lot sizes of product p demanded at distributor k during production cycle t ; $k \in K, p \in P, t \in T$.

$I_{i,r,0}$ = Initial Inventory level of parts r , at supplier i , at the beginning of production cycle 1; $i \in I, r \in R$.

$n_{r,p}$ = Number of parts r required to manufacture one unit of product p ; $r \in R, p \in P$.

$C_{j,p}$ = An overhead cost parameter for product p at plant j ; $j \in J, p \in P$.

$\alpha_{j,p}$ = A costing parameter that is dependent on the production setting at plant j for product p . It is a positive number ($\alpha \geq 0$); $j \in J, p \in P$.

$\beta_{j,p}$ = An exponential factor, which is a positive real number between zero and one at plant j for product p ; $j \in J, p \in P$.

Decision variables:

$X_{j,p,t}$ = Number of lots produced for product p at plant j , during production cycle t ; $j \in J, p \in P, t \in T$.

$XP_{i,r,t}$ = Number of parts r procured or produced at supplier i , during production cycle t ; $i \in I, r \in R, t \in T$.

$Z_{j,p,t}$ = Binary variable that takes the value of 1, when output is assigned for product p at plant j , during production cycle t , or zero otherwise; $j \in J, p \in P, t \in T$.

$XT_{r_{i,j,r,t}}$ = Number of parts transported from supplier i to plant j for part r , during production cycle t ; $i \in I, j \in J, r \in R, t \in T$.

$XT_{p_{j,k,p,t}}$ = Number of products transported from plant j to distributor k for product p , during production cycle t ; $j \in J, k \in K, p \in P, t \in T$.

$XH_{i,r,t}$ = Number of parts, r , held at supplier i , during production cycle t ; $i \in I, r \in R, t \in T$

$S_{j,p,t}$ = Time shortened from original $P_{to_{j,p,t}}$ for plant j , for product p , at production cycle t ; $j \in J, p \in P, t \in T$

$P_{t_{j,p,t}}$ = Production run time at plant j , for product p , as a result of crashing the production run by $S_{j,p,t}$ in production cycle t ; $j \in J, p \in P, t \in T$.

$PCF_{j,p,t}$ = Production cost as a function of period shortened, S , for product p and lot size $Q_{j,p,t}$ manufactured at plant j during production cycle t ; $j \in J, p \in P, t \in T$.

Model:

$$\begin{aligned}
 MinZ = & \sum_{i \in I} \sum_{r \in R} \sum_{t \in T} PCr_{i,r,t} XP_{i,r,t} + \sum_{i \in I} \sum_{j \in J} \sum_{r \in R} \sum_{t \in T} Tr_{i,j,r,t} XT_{r_{i,j,r,t}} \\
 & + \sum_{i \in I} \sum_{r \in R} \sum_{t \in T} IC_{i,r,t} XH_{i,r,t} \\
 & + \sum_{j \in J} \sum_{p \in P} \sum_{t \in T} X_{j,p,t} PCF_{j,p,t} + \sum_{j \in J} \sum_{p \in P} \sum_{t \in T} Z_{j,p,t} Fix_{j,p,t} \\
 & + \sum_{j \in J} \sum_{k \in K} \sum_{p \in P} \sum_{t \in T} T_{p_{j,k,p,t}} XT_{p_{j,k,p,t}}
 \end{aligned} \tag{4.1}$$

The objective function (4.1) minimizes the total operational costs. It includes (order from left to right) the procurement costs at suppliers, transportation costs from suppliers to plants, sum of inventory holding costs at suppliers, production costs (crashing and normal) at plants, transportation costs from plants to distributors for all product types. Importantly, the objective

function minimizes the total of normal and crashing production costs and, in turn, optimizes production time at a given plant.

The production time, $Pt_{j,p,t}$, can be written as:

$$Pt_{j,p,t} = Pto_{j,p,t} - S_{j,p,t} \tag{4.2}$$

The production cost term in the function returns a lot-size production cost that is a function of the shortening of production time, $S_{j,p,t}$. The PCF is the total production cost, encompassing both the crashing and normal production costs.

$$PCF_{j,p,t} = \alpha_{j,p} e^{\beta_{j,p} S_{j,p,t}} - C_{j,p} S_{j,p,t} + C_{j,p} Pto_{j,p,t} \tag{4.3}$$

Subject to

Constraint (4.4) ensures that parts held at supplier i do not exceed the capacity limit.

$$XH_{i,r,t} \leq SCap_{i,r,t} \quad \forall i, \forall r, \forall t \tag{4.4}$$

Constraint (4.5) balances the parts procured, and held at supplier i , with the parts to be transported to plants.

$$XP_{i,r,t} - \sum_{j \in J} XTr_{i,j,r,t} - XH_{i,r,t} + XH_{i,r,t-1} = 0 \quad \forall i, \forall r, \forall t : \xrightarrow{t=0} XH_{i,r,t} = I_{i,r,0} \tag{4.5}$$

Constraint (4.6) balances the outbound supply from suppliers with the demanded parts required for production at plants (also ensures that no transportation is performed for plants that are inactive).

$$\sum_{i \in I} XTr_{i,j,r,t} = X_{j,p,t} Z_{j,p,t} Q_{j,p,t} n_{r,p} \quad \forall j, \forall p, \forall r, \forall t \tag{4.6}$$

Constraint (4.7) ensures that demanded products at a given distributor k are met.

$$\sum_j \frac{XTp_{j,k,p,t}}{Q_{j,p,t}} = D_{k,p,t} \quad \forall k, \forall p, \forall t \tag{4.7}$$

Constraint (4.8) balances the number of products transported to distributors with the number of products made at plants.

$$X_{j,p,t} Q_{j,p,t} Z_{j,p,t} - \sum_k XTp_{j,k,p,t} = 0 \quad \forall j, \forall p, \forall t \tag{4.8}$$

Constraint (4.9) restricts the number of products made at a given plant to the plant's capacity plus excess capacity resulting from crashing.

$$Q_{j,p,t} X_{j,p,t} \leq Z_{j,p,t} \left(PCap_{j,p,t} + S_{j,p,t} \frac{Q_{j,p,t}}{Pt_{j,p,t}} \right) \quad \forall j, \forall p, \forall t \tag{4.9}$$

Constraint (4.10) ensures non-negativity and binary representations.

$$XTr_{i,j,r,t}, XTp_{j,k,p,t}, XH_{i,r,t}, X_{j,p,t}, S_{j,p,t}, Pt_{j,p,t} \geq 0; \\ Z_{j,p,t} \in \{0, 1\} \quad \forall i, \forall r, \forall j, \forall p, \forall t \tag{4.10}$$

Fig. 4 Problem P1

Problem P1 :

$$z_1 = \min_{x,y} f(\mathbf{x}, \mathbf{y})$$

$$\text{subject to : } \mathbf{h}(\mathbf{x}, \mathbf{y}) = 0$$

$$\mathbf{g}(\mathbf{x}, \mathbf{y}) \leq 0$$

$$\mathbf{x} \in \mathfrak{R}^{n_x}$$

$$\mathbf{y} \in \{0, 1\}^{n_y}$$

5 Methods

The contribution of this work is emphasized in the following:

- The inclusion of crashing/normal production costs and the conceptualization of the tradeoff between production time crashing and production costs.
- The solving of a nonlinear binary model and the proving of convexity.
- The design of accurate solution heuristics (minimizing the solution gap with respect to the lower bound).

To understand the contribution of this paper, close attention needs to be paid to the special characteristics of the model. The work of Dua (2015) presents a generic formulation of a binary nonlinear mathematical programming problem as P1 (Grossmann 2002).

As shown in Fig. 4, P1 is a function of x and y . Where x is a vector of continuous variables, y is a vector of binary variables, h is an n_h dimensional vector of equality constraints, g is an n_g dimensional vector of inequality constraints and f is the scalar objective function. Solving P1 is NP-hard (Dua 2015). The objective function (4.1) in this work has similar characteristics but is complicated in the fact that two of the decision variables ($X_{j,p,t}$ and $S_{j,p,t}$) are multiplied by each other. The production cost term present in Eq. (4.1) is nonlinear as $PCF_{j,p,t}$ is a function of the decision variable $S_{j,p,t}$ and is exponential. Moreover, the plant assignments are represented by binary variables, $Z_{i,p,t}$. Further, nonlinearities in constraints (4.6), (4.8), and (4.9) are present. Given the complexity of the problem at hand, heuristics need to be devised to solve the model. If the binary variables were fixed (i.e., assigned), then the resulting objective function would only contain continuous variables, which can be solved by a gradient search method. The near-optimal solution can then be found by iteratively searching the solution space using search heuristics. This linearizes constraints (4.6), (4.8), and (4.9). The remains of the section will highlight the convexity proof (Sect. 5.1), gradient search (Sect. 5.2), local search procedures (Sect. 5.3), genetic search (Sect. 5.4), and finally the simulated annealing search (Sect. 5.5).

5.1 Convexity proof

Proposition *The objective function (4.1) is convex.*

The total cost, PCF , of normal and crashing costs is $PCF = CC + NC$. Using definitions and equations introduced in Sect. 3, PCF can be written as:

$$PCF = \alpha e^{\beta * S} + C(Pto - S) \tag{5.1}$$

For any function $f(x)$ to be convex, then a second derivative everywhere must be $\frac{d^2 f(x)}{dx^2} \geq 0$, for all values of x (Shah et al. 2007; Mohri et al. 2018). Therefore, taking the second derivative of the function on the only variable in the equation, S , brings the following expression:

$$\alpha \beta^2 e^{\beta S} \geq 0$$

Since S , α , and β can only take positive values by definition (see Sect. 3), then the whole expression must always be positive and hence the function is convex.

The PCF term in the objective function is also multiplied by X which is a decision variable. Both partial derivatives of $\frac{d^2 f(S)}{dS^2}$ and $\frac{d^2 f(X)}{dX^2}$ are greater or equal to zero rendering the overall expression convex. Given that the sum of convex functions is also convex and the holding of Z as constant for each iterate in the solution procedure (to be discussed in Sects. 5.3–5.5), the whole objective function is convex.

5.2 Gradient search method

Since the objective function (4.1) is convex, a gradient search method can fetch the optimal values of $S_{j,p,t}$ and solve the model effectively. As discussed, the objective function, f , is convex. Thus f is differentiable at x , and a vector $d \in \mathfrak{R}^n$ is a descent direction for f at x if:

$$-\nabla f(x)T^d > 0$$

According to the definition of the derivative (Kolda et al. 2003):

$$f(x + \alpha d) = f(x) + \alpha \nabla f(x)^T d + o(\alpha)$$

If d is a descent direction, and $\alpha > 0$ is sufficiently small, then $x^{k+1} = x^k + \alpha^k d^k$ reduces the value of the objective f . This observation forms the basis of line search methods. At the iterate x^k , a descent direction d^k is chosen and a search is conducted along this direction for a point $x^{k+1} = x^k + \alpha^k d^k$ (with $\alpha^k > 0$) that has a smaller objective value. Choosing the correct α is important to guarantee faster convergence (Kolda et al. 2003).

Gradient methods are specified in the form:

$$y^{k+1} = y^k + \alpha^k \nabla f(y^k)$$

α^k is the step size and has only a positive value that minimizes: $f(y^k + \alpha^k \nabla f(y^k))$.

$$f(y^k + \alpha^k \nabla f(y^k)) = \min_{\alpha^k} f(y^k + \alpha^k \nabla f(y^k))$$

The expression ‘ $f(y^k + \alpha^k \nabla f(y^k))$ ’ can be alternatively thought of as ‘ $f(x)$ ’ evaluated at ‘ $y^k + \alpha \left(\frac{\partial f}{\partial y^k}\right)$.’ The expression is a function of constants with the exception of: y_i^k . When the ‘ y_i^k ’ expressions are fixed at each iteration, the ‘ $f(x)$ ’ becomes a function of just a single variable: ‘ α ’ (Hillier and Lieberman 1995). The gradient search procedure first starts at y (pertaining to this work, y is S : $S_{j,p,t}$) equals zero and solves the model linearly to produce ‘ $f(y)$ ’. Then the stepsize is calculated by finding a stepsize (α) that minimizes $f(y + \alpha d)$. As the stepsize approaches zero, the value of $f(y + \alpha d)$ approaches that of $f(y)$ and the minimum

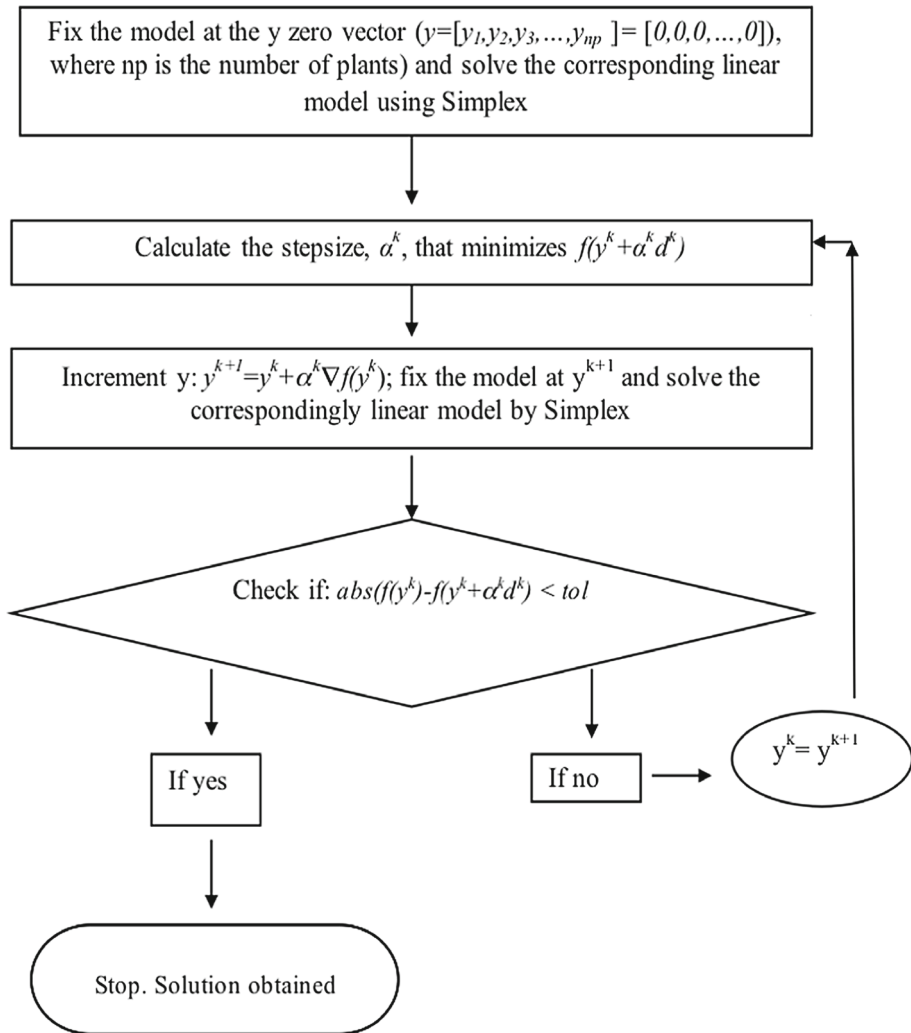


Fig. 5 Gradient-based solution methodology (y is $S_{j,p,t}$)

is intuitively achieved. This minimum is a global minimum, since the objective function has been proven convex. Figure 5 illustrates the steps in the solution methodology where the initial solution of $f(y)$ is a zero vector. The procedure stops when the difference between $f(y + \alpha d)$ and $f(y)$ is very small and less than a tolerance (tolerance is set to 0.03% of objective value).

5.3 Local search heuristics

The paper uses a number of local search heuristics to deal with the set of binary variables in the model, $Z_{j,p,t}$. Starting with an initial solution, the procedure iteratively searches for the best solution vector $Z_{j,p,t}$. Then at each iteration, the gradient search method is invoked to compute

the cost of the solution. Local search (LS) procedures are improvement heuristics that search in the neighborhood of the current solution for a better one until no further improvement can be made.

Five local searches are contested:

1. 1-opt-first procedure
2. k-opt-first procedure
3. 1-opt-best procedure
4. k-opt-best procedure
5. Hybrid k-1-opt procedure.

The *1-opt-first* (Merz and Freisleben 2002) procedure starts from a feasible solution and searches the neighborhood for solutions with a hamming distance of 1 to the current solution (i.e., by flipping a single bit) and the gain calculated as the difference between the objective value prior to the move and post to the move. If no better neighbor solution can be found in a predetermined set of tries, the search immediately stops and returns the best solution found. The *1-opt-best* procedure differs from the *1-opt-first* procedure in that it contests a group of flips and records the best move among them. The binary solution vector in our case is $Z_{j,p,t}$ as defined previously. Similarly, there are two types of *k-opt* move strategies. The first strategy is the first improvement move strategy, which scans solutions in the $N(z)$ space according to pre-specified order, and Z becomes incumbent if the solution Z improves the objective value. The second strategy is a best improvement move strategy, where solution Z is chosen if it is the best cost in the entire candidate set of $N(z)$.

The hybrid solution procedure combines both the *1-opt-best* and *k-opt-best* methodologies. The hybrid solution procedure is inspired by the results obtained from the *1-opt-best* and *k-opt-best* preliminary results. For the *k-opt* solution procedure, when a large number of bits are flipped simultaneously, the chance of breaching feasibility becomes more imminent. We see this in Table 2, where computational time can be really high even for small supply chain networks. The *1-opt* methodology only flips one bit at a time and hence has a lesser chance of breaching feasibility. As shown in Fig. 6, the *k-opt* and *1-opt* moving strategies are altered and each respective gain is recorded. The best move that brings about the fittest solution, among a group of *k-opt* and *1-opt*, is then chosen. In this way, the best move could be a *k-opt* or a *1-opt* move, depending on the solution topography and the proximity of the infeasibility threshold, and would correspond to the highest gain while meeting the feasibility criterion. n_{ni} is the stopping criteria for all the five searches, which counts the number of non-improvement in the solution. This will be discussed in Sect. 6.

5.4 Genetic search

Along with the local search heuristics, the paper utilizes a genetic-based heuristic (GBH). Holland (1975) originally developed the method over the course of the 1960s and 1970s. The heuristic begins by defining the optimization variables and the cost function. GBH defines a chromosome or an array of variables to be optimized. The chromosome has N variables (an N -dimensional optimization problem) given by $v_1, v_2, v_3, \dots, v_N$, which is in our case $Z_{j,p,t}$. It is necessary to define the cost function which generates an output from a set of input variables. The cost function is a function of the chromosomes. The heuristic starts with a group of chromosomes known as the ‘population’; a matrix, $N_{pop} \times N_{bits}$. This results in an N_{pop} chromosomes (i.e., population) and N_{bits} number of bits. The $N_{pop} \times N_{bits}$ matrix gets filled with random ones and zeros. Natural Selection is then applied to improve the fitness of the population of solutions. In a minimization problem as the current, survival of the

Table 2 Comparison between the seven heuristics

| Supply chain network config. | Genetic | | Simulated annealing | | Hybrid (K and L) | | K-opt best | | I-opt-first | | K-opt-first | | | | | |
|------------------------------|----------|----------|---------------------|----------|------------------|----------|------------|----------|-------------|----------|-------------|----------|-------|-------|-------|------|
| | Df (%) | Time (s) | Df (%) | Time (s) | Df (%) | Time (s) | Df (%) | Time (s) | Df (%) | Time (s) | Df (%) | Time (s) | | | | |
| Small networks | 1-2-1 | 7.15 | 22.35 | 20.19 | 320.23 | 7.15 | 14.97 | 7.15 | 12.45 | 7.15 | 14.43 | 7.15 | 3.07 | 7.15 | 2.57 | |
| | 2-3-1 | 8.31 | 25.24 | 8.31 | 146.62 | 8.31 | 8.36 | 8.31 | 9.26 | 8.31 | 10.24 | 8.31 | 2.88 | 8.31 | 3.22 | |
| | 2-4-2 | 5.90 | 35.74 | 5.90 | 43.15 | 5.90 | 20.01 | 5.90 | 39.28 | 5.90 | 20.39 | 5.90 | 4.26 | 5.90 | 5.13 | |
| | 4-8-4 | 5.21 | 43.97 | 13.75 | 15.23 | 6.76 | 24.89 | 8.36 | 19.93 | 5.91 | 24.97 | 8.36 | 8.95 | 5.91 | 6.89 | |
| | 3-5-3 | 6.83 | 37.86 | 10.94 | 29.62 | 6.83 | 39.62 | 6.83 | 20.29 | 6.83 | 47.24 | 6.83 | 3.17 | 6.83 | 11.67 | |
| | 3-6-3 | 6.85 | 37.97 | 6.85 | 57.92 | 5.53 | 29.37 | 5.53 | 18.13 | 6.96 | 7.77 | 5.53 | 9.18 | 6.96 | 7.43 | |
| | 2-5-3 | 5.90 | 30.67 | 6.02 | 31.39 | 5.90 | 16.11 | 6.02 | 18.78 | 5.90 | 23.28 | 6.02 | 3.59 | 5.90 | 5.30 | |
| | 3-7-3 | 8.03 | 37.94 | 9.25 | 55.24 | 6.43 | 26.55 | 6.20 | 17.08 | 6.00 | 19.66 | 6.20 | 6.89 | 6.00 | 4.90 | |
| | Average | 6.77 | 33.97 | 10.15 | 87.42 | 6.60 | 22.49 | 6.79 | 19.40 | 6.62 | 21.00 | 6.79 | 5.25 | 6.62 | 6.62 | 5.89 |
| | 6-11-6 | 6.06 | 50.92 | 6.49 | 27.79 | 6.25 | 49.45 | 7.49 | 27.91 | 5.70 | 18.42 | 7.49 | 7.23 | 5.70 | 6.37 | |
| Medium networks | 6-13-6 | 5.80 | 55.23 | 7.48 | 34.09 | 7.03 | 36.75 | 9.49 | 21.23 | 7.03 | 32.15 | 9.49 | 16.84 | 7.03 | 11.06 | |
| | 6-14-6 | 5.19 | 51.46 | 4.36 | 108.11 | 5.16 | 29.38 | 4.58 | 35.69 | 5.37 | 48.40 | 4.58 | 12.88 | 5.37 | 13.10 | |
| | 7-15-7 | 7.82 | 66.85 | 9.41 | 69.59 | 8.60 | 41.95 | 10.48 | 32.22 | 7.23 | 53.59 | 10.48 | 17.98 | 7.23 | 25.10 | |
| | 7-16-7 | 4.93 | 66.28 | 9.09 | 100.74 | 7.21 | 34.67 | 12.77 | 32.25 | 5.94 | 40.58 | 12.77 | 12.24 | 5.94 | 9.91 | |
| | 7-17-7 | 6.69 | 68.70 | 9.51 | 94.83 | 6.99 | 35.40 | 7.93 | 49.21 | 6.11 | 45.92 | 7.93 | 22.98 | 6.11 | 9.63 | |
| | 8-18-8 | 7.07 | 78.03 | 7.13 | 54.63 | 9.01 | 30.27 | 9.76 | 42.62 | 8.42 | 54.84 | 9.76 | 19.36 | 8.42 | 27.73 | |
| | 8-19-8 | 8.77 | 79.97 | 10.59 | 115.93 | 8.66 | 57.49 | 8.88 | 75.33 | 8.16 | 33.23 | 8.88 | 21.05 | 8.16 | 11.93 | |
| | Average | 6.54 | 64.68 | 8.01 | 75.71 | 7.36 | 39.42 | 8.92 | 39.56 | 6.74 | 40.89 | 8.92 | 16.32 | 6.74 | 14.35 | |
| | 10-25-10 | 5.69 | 129.70 | 8.55 | 85.40 | 7.66 | 88.19 | 8.56 | 81.80 | 8.02 | 60.26 | 8.56 | 61.37 | 8.02 | 21.18 | |
| | 10-26-10 | 6.76 | 131.10 | 13.72 | 276.84 | 7.98 | 71.87 | 10.51 | 99.04 | 8.25 | 58.32 | 10.51 | 32.34 | 8.25 | 15.24 | |
| Large networks | 11-27-10 | 7.87 | 163.92 | 12.81 | 597.85 | 10.39 | 86.60 | 14.51 | 116.98 | 11.45 | 80.27 | 14.51 | 24.81 | 11.45 | 26.04 | |
| | 11-28-10 | 6.74 | 149.85 | 10.03 | 248.60 | 10.52 | 100.63 | 12.83 | 119.70 | 11.33 | 82.96 | 12.83 | 75.07 | 11.33 | 18.13 | |

Table 2 continued

| Supply chain network config. | Genetic | | Simulated annealing | | Hybrid (K and L) | | 1-opt best | | K-opt best | | 1-opt-first | | K-opt-first | |
|------------------------------|---------|----------|---------------------|----------|------------------|----------|------------|----------|------------|----------|-------------|----------|-------------|----------|
| | Df (%) | Time (s) | Df (%) | Time (s) | Df (%) | Time (s) | Df (%) | Time (s) | Df (%) | Time (s) | Df (%) | Time (s) | Df (%) | Time (s) |
| 12-29-11 | 9.25 | 174.25 | 11.34 | 305.77 | 9.80 | 90.96 | 13.29 | 116.73 | 10.02 | 88.89 | 13.29 | 43.34 | 10.02 | 21.84 |
| 12-30-11 | 8.31 | 182.07 | 10.05 | 555.84 | 9.32 | 114.22 | 14.26 | 131.28 | 9.29 | 100.63 | 14.26 | 56.05 | 9.29 | 35.06 |
| 13-32-12 | 6.55 | 211.75 | 11.20 | 652.24 | 9.00 | 131.50 | 13.10 | 165.80 | 7.67 | 127.49 | 13.10 | 95.09 | 7.67 | 27.21 |
| 14-33-13 | 8.35 | 245.48 | 15.25 | 573.42 | 8.56 | 131.12 | 13.10 | 164.95 | 8.11 | 164.18 | 13.10 | 95.85 | 8.11 | 36.16 |
| Average | 7.44 | 173.52 | 11.62 | 412.00 | 9.15 | 101.89 | 12.52 | 124.53 | 9.27 | 95.37 | 12.52 | 60.49 | 9.27 | 25.11 |

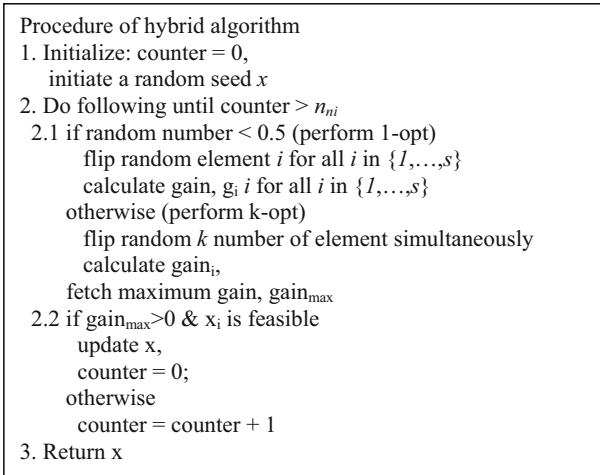


Fig. 6 Hybrid solution procedure

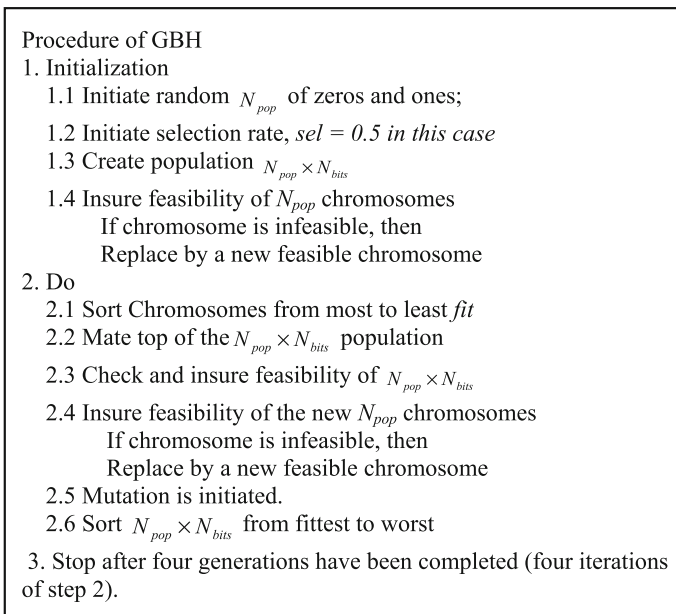


Fig. 7 Genetic-based heuristic schematic

fittest translates into discarding the chromosomes with the highest cost. The overall solution procedures are highlighted in Fig. 7.

As illustrated in Fig. 7, we start with a population of random feasible solutions (i.e., chromosomes). The number of chromosomes is set to 30. Figure 8 illustrates preliminary runs where we increase the number of chromosomes from 10 to 200. The figure shows 30 chromosomes to be a good cutoff point (Df curve becomes less steep after 30 chromosomes). Here, Df is the solution gap [see Eq. (6.1)].

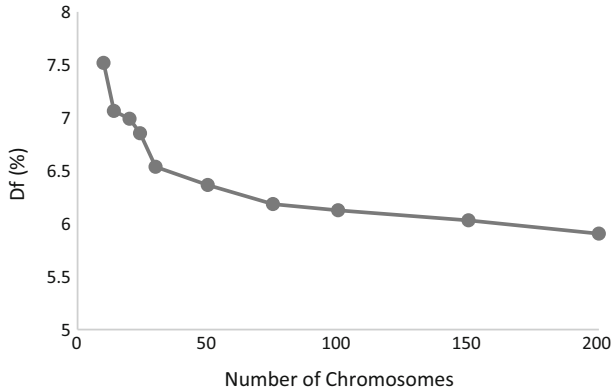


Fig. 8 Average Df (%) of 100 runs at each respective number of chromosomes

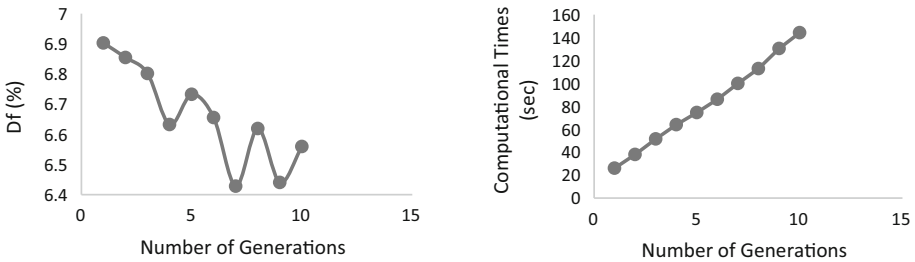


Fig. 9 Results from varying the number of generations in the genetic procedure

Next, the feasibility is insured in step 1.4 (see Fig. 7). Then, we sort chromosomes from best to worst fit. Afterward, we introduce mutations to the population at a rate of 1% for each generation. The procedure’s last step is the stopping criterion. Using a non-improvement count as a stopping criterion is quite common in the literature, especially for local search methods. However, for genetic search procedures, the number of generations is usually a stopping criterion of choice (Michalewicz 1996). In our case, we have studied deeply the behavior of the genetic procedure. We ran hundreds of iterations to identify the best stopping criterion based on accuracy and computational time. In Fig. 9, we vary the number of generations in the genetic search from 1 to 10 generations. At each generation iterate, we conduct 100 runs to assure good representation. We find that the steepest decrease in the solution gap occurs between one and four generations. After four generations, the decrease in DF is smaller while the computational time is steadily higher. Hence, we choose four generations to be the stopping criterion.

5.5 Simulated annealing search procedure

Last of the heuristics is the simulated annealing (SA) procedure, which is based on multiple annealing processes starting from an initial annealing temperature. The first annealing temperature starts at a higher temperature and then is gradually lowered (Kattayama and Narihisa 2001). Each iteration of the search process moves from a current trial solution to an immediate neighbor in the local neighborhood of the solution. The immediate neighbor is

Fig. 10 Simulated annealing search procedure

| |
|--|
| Procedure of simulated annealing 1. Initialize: counter = 0 initiate a random seed x 2. Do following until counter > n_{ni} flip random element i for all i in $\{1, \dots, s\}$ calculate gain, g_i for all i in $\{1, \dots, s\}$ fetch maximum gain, $gain_{max}$ if $gain_{max} > 0$ & x_i is feasible update x , $x_i = 1 - x_i$ counter = 0; if $gain_{max} < 0$ & x_i is feasible if random number < probability & x_i is feasible update x , counter = 0; otherwise counter = counter+1 3. Return x |
|--|

then selected to be the next trial solution. Now among all immediate neighbors of the current trial solution, a solution is accepted if it improves the objective function. If the solution does not improve the objective function, it will be compared to a probability criterion. Here, a random number is compared to the probability; if the probability is greater than the random number, the solution will be accepted or rejected if otherwise. The SA procedure is shown in Fig. 10. Like the other solution procedures, the search converges if n_{ni} reaches four.

6 Results

In order to measure the accuracy and the performance of each of the heuristics, a lower bound is constructed on the objective function (4.1). It is obtained by relaxing the binary variables, $Z_{j,p,t}$'s, associated with the assignable costs, to become continuous variables with a lower bound of zero and an upper bound of one ($0 < Z < 1$). Once the binary variables are relaxed, the model is solved using the gradient procedure and the solution becomes the lower bound to the objective function (4.1). Then, the original model is solved using all seven heuristics discussed and the solutions are then contrasted against the lower bound. The gap difference between the solution of the heuristic and the lower bound, Df , is calculated as follows.

$$Df = \frac{Hv - Lv}{Lv} \times 100 \quad (6.1)$$

Hv = Objective value of respective heuristic.

Lv = Objective value of lower bound.

To introduce the results, we first discuss the stopping criterion (Sect. 6.1), and then we vary the supply chain configurations (Sect. 6.2). We analyze the effect of crashing in Sect. 6.3 and we contest larger runs in Sect. 6.4.

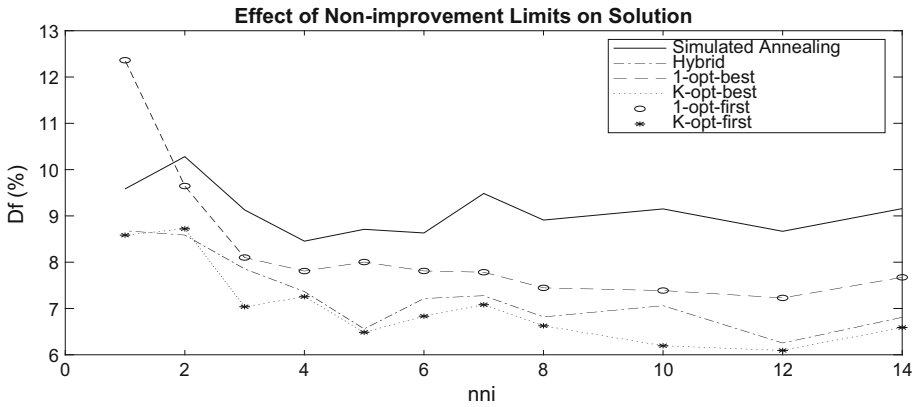


Fig. 11 Effect of non-improvement limits on the solution

6.1 Stopping criterion

The input data for the supply chain is simulated to give a higher number of instances and the results are shown in Table 2. The stopping criterion, n_{ni} , for all seven solution procedures is the same, where n_{ni} is the maximum number of consecutive non-improvement moves allowed (except for the genetic search where it's four generations). An infeasible move is considered a non-improving move as it does not improve the solution. Therefore, the time duration till the stopping criterion is a measurement of how efficient and successful the solution procedure is in reaching the solution without breaching feasibility. It's important to note that the stopping criterion for all the heuristics (except for the genetic search: see Sect. 5.4) is the maximum number of non-improvements, n_{ni} . n_{ni} is a limit set by the user to signal the arrival at the minimum solution. This constructs a fair comparison between all the heuristics since arriving at the solution is tied to n_{ni} . To set the limit value of n_{ni} , we carried numerous runs over different n_{ni} values. We ran over 200 runs. In the runs, we varied the value of n_{ni} from 1 to 14. Figure 11 shows the corresponding solution gaps (averaged over 20 runs each) when varying the limits (n_{ni}). Initially, when we increase the limits (from 1 to 3), the solution gaps (Df) drastically drop. The drop then stabilizes above three. To elaborate, increasing the value of the limits will initially produce high gains then stagnate. In addition, we see higher computational times at higher limits (see Fig. 12). In all, at $n_{ni} = 4$, the benefits (accuracy and efficiency) of increasing n_{ni} diminishes. For this, we chose to run the model at $n_{ni} = 4$.

6.2 Varying supply chain configuration

In Table 2, we note the time duration of convergence for each of the seven solution procedures. To assure consistency and fair representation, the coding, compilation, and execution are done using a MATLAB platform on a fixed computational system (Intel R, Core TM, i5-2400 CPU @3.10 GHz). Table 2 presents the simulation of twenty-four runs. The runs are for a small supply chain, medium supply chain, and large supply chain networks to ensure better representation of practical implications. Local search heuristics, specifically *K-opt* (first and best), tend to do well and outperform the simulated annealing and genetic heuristics for small supply chain networks. However, for larger networks, the genetic heuristic tends to do best in

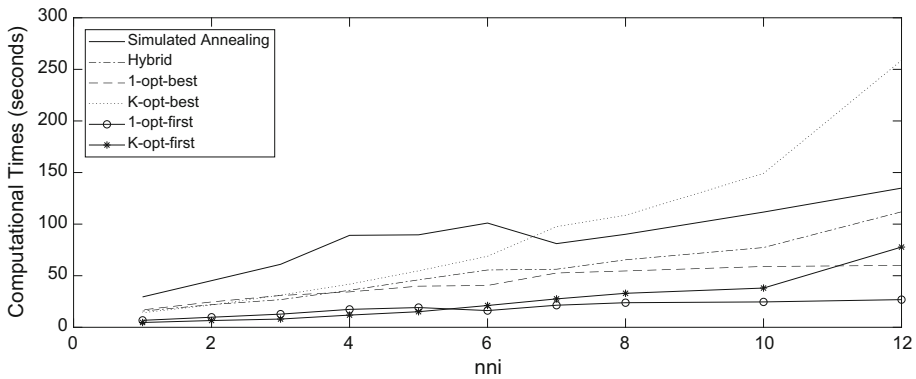


Fig. 12 Computational times for the six different heuristics

terms of accuracy. In terms of computational times, both hybrid and *K-opt* tend to be faster than the genetic search but at two points lower than the genetic in terms of accuracy (Table 2).

6.3 Effect of crashing

Aside from the heuristics, we start with a network of two suppliers, seven potential plants for product assignments, and two distributors (Fig. 13) to infer the monetary impact of crashing the production time (*Pt*). A leading firm in the Aerospace industry has provided us with data. However, the data has been masked to assure the privacy of the operations as per the request of the firm (Table 3). The production cost parameters α , β , and C are extrapolated from the crashing data in Table 4, using exponential data fitting where the error of fitting is very small and negligible (0.001). Notably, the scaling does not affect the results especially given the fact that our aim is for comparative analysis.

All the results, presented in this paper, are comparative between the crashing of *Pt* case and the null case (where no crashing is allowed). This is to answer the main question of this work: Does the crashing of *Pt* minimize the overall production costs, reduce supply chain costs, and/or increase supply chain throughput? To answer this question, two models are contested: one where crashing *Pt* is not possible and one where it is. If there is no crashing of *Pt*, the problem becomes linear as the $S_{j,p,t}$ terms in the model becomes zero. Hence, no gradient search is required to solve the model. Solving the model produces an objective function value, which we should call: Obj_{vnull} . While the objective value of the model where crashing is possible is termed, Obj_{vcrash} . Accordingly, the percent difference or objective function reduction can be expressed as *Acc*.

$$Acc = 100 \times \frac{Obj_{vnull} - Obj_{vcrash}}{Obj_{vnull}}$$

Based on the costing parameters (α , β , and C) shown in Fig. 13, the reduction in the objective function, *Acc*, is approximately 25%. This validates the model since the two plants with the lower cost parameters (α , β , and C) are chosen in the model.

In Table 5, the values of the costing parameters are varied (using MATLAB simulations) to produce eight different instances. Here, the *Acc*'s range is between 12 and 27%. Further, the work devises a measure for supply chain throughput, which is the total time a product (or components to be assembled to be part of a given product) spends in inventory and production,

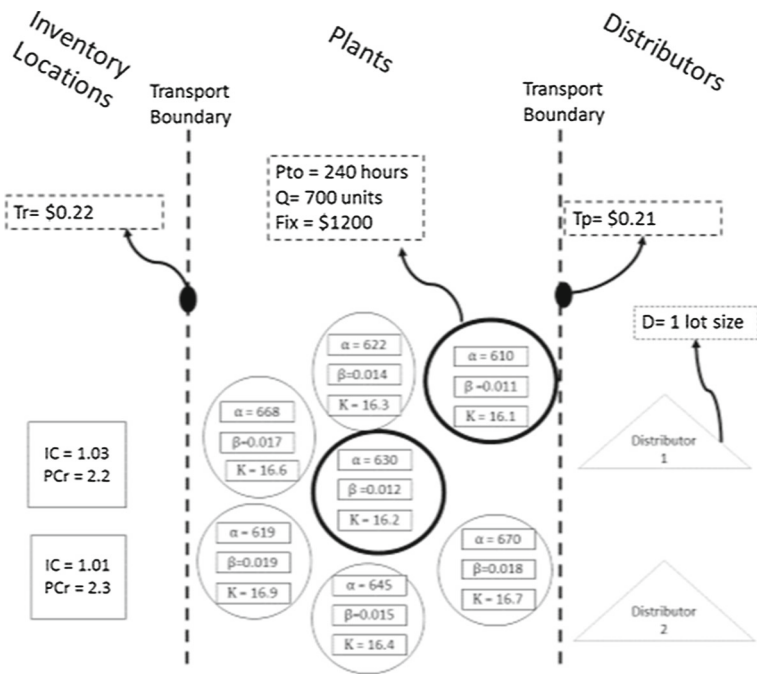


Fig. 13 Example of a supply chain network solved by the model

PDT. Given this definition, the percent increase of throughput, due to the allowance of crashing, is approximately 24% on average. Moreover, savings in inventory costs result, since products are moving faster through the supply chain and spending less time in inventory. The average percent savings in inventory cost, *PDI*, is approximately 23%. The average time shortened at the plants, which are open for production, is reported in Table 5 as *PTS*. *PTS* hovers around 58 h on average. It’s important to note that the time shortened provides a plant with excess capacity and opens up the plant for more output. To elaborate, the allowance for crashing enables plants with advantageous cost structure to fulfill more of the requested demand. For instance, in a three plants network, each plant might fulfill one-third of the overall demand. Due to the allowance of crashing, a plant that has an advantageous cost structure (α , β , C) would get higher than a third of the demand and hence brings the overall supply chain production cost.

Vitaly, the runs show that the model is highly sensitive to the values of the production cost parameters (α , β , C), and inventory cost parameter (IC). Hence, it is imperative to carry a large volume of runs to represent different variations of those parameters. Notably, these parameters are user dependent and reflect the costing structure of the operational environment at hand. Therefore, higher number of runs would allow us to cover a variety of implications. The runs should simulate the high and low production parameter values presented in Table 6. In Table 4, real production data is used to infer realistic cost parameters. Feasible variations are conducted on the values in Table 4 and extrapolated to form limits on the high and lows (Table 6) while remaining feasible. Table 7 illustrates the parameters’ variant of each run where the first column designates the magnitude of the parameters. Each parameter is tested at two instances: One at a high value and another at a low value. For α (designated as A in the first column of Table 7), the high value is 700 while the low value is 600 (See Table 6). For instance, if the run is contesting a high value, then a value of one is placed next to A, while a

Table 3 Example of base data for the production cost parameters (α , β , K)

| Pt (h) | Normal costs (\$) | S (h) | Crashing cost (\$) | Total cost (crashing cost + normal production costs) (\$) | Pt (h) | Normal costs (\$) | S (h) | Crashing costs (\$) | Total cost (crashing cost + normal production costs) (\$) |
|--------|-------------------|-------|--------------------|---|--------|-------------------|-------|---------------------|---|
| 239 | 4302 | 1 | 708 | 5010 | 215 | 3870 | 25 | 945 | 4815 |
| 238 | 4284 | 2 | 717 | 5001 | 214 | 3852 | 26 | 956 | 4808 |
| 237 | 4266 | 3 | 726 | 4992 | 213 | 3834 | 27 | 968 | 4802 |
| 236 | 4248 | 4 | 734 | 4982 | 212 | 3816 | 28 | 980 | 4796 |
| 235 | 4230 | 5 | 743 | 4973 | 211 | 3798 | 29 | 991 | 4789 |
| 234 | 4212 | 6 | 752 | 4964 | 210 | 3780 | 30 | 1003 | 4783 |
| 233 | 4194 | 7 | 761 | 4955 | 209 | 3762 | 31 | 1015 | 4777 |
| 232 | 4176 | 8 | 771 | 4947 | 208 | 3744 | 32 | 1028 | 4772 |
| 231 | 4158 | 9 | 780 | 4938 | 207 | 3726 | 33 | 1040 | 4766 |
| 230 | 4140 | 10 | 789 | 4929 | 206 | 3708 | 34 | 1053 | 4761 |
| 229 | 4122 | 11 | 799 | 4921 | 205 | 3690 | 35 | 1065 | 4755 |
| 228 | 4104 | 12 | 808 | 4912 | 204 | 3672 | 36 | 1078 | 4750 |
| 227 | 4086 | 13 | 818 | 4904 | 203 | 3654 | 37 | 1091 | 4745 |
| 226 | 4068 | 14 | 828 | 4896 | 202 | 3636 | 38 | 1104 | 4740 |
| 225 | 4050 | 15 | 838 | 4888 | 201 | 3618 | 39 | 1118 | 4736 |
| 224 | 4032 | 16 | 848 | 4880 | 200 | 3600 | 40 | 1131 | 4731 |
| 223 | 4014 | 17 | 858 | 4872 | 199 | 3582 | 41 | 1145 | 4727 |
| 222 | 3996 | 18 | 869 | 4865 | 198 | 3564 | 42 | 1159 | 4723 |
| 221 | 3978 | 19 | 879 | 4857 | 197 | 3546 | 43 | 1173 | 4719 |
| 220 | 3960 | 20 | 890 | 4850 | 196 | 3528 | 44 | 1187 | 4715 |
| 219 | 3942 | 21 | 901 | 4843 | 195 | 3510 | 45 | 1201 | 4711 |
| 218 | 3924 | 22 | 911 | 4835 | 194 | 3492 | 46 | 1216 | 4708 |
| 217 | 3906 | 23 | 922 | 4828 | 193 | 3474 | 47 | 1230 | 4704 |
| 216 | 3888 | 24 | 934 | 4822 | 192 | 3456 | 48 | 1245 | 4701 |

value of zero is placed if a low value is used. Similarly, B represents the β , C represents the C, and H represents the inventory cost ($IC_{i,r,t}$) parameter.

Analyzing Table 7, one can infer the impact of having a lower crashing production cost structure on the overall objective function's cost reduction. With lower crashing cost structure, the objective function reductions are more magnified. Hence, lower values of α , β , and C result in higher production cost savings when crashing. However, β tends to be the most influential of the three. This is quite logical since β is an exponential term. With higher production cost parameters (α , β , C), the cost reductions shrink due to the high costs of additional resources required to crashing Pt . However, the numbers still show around 10% overall supply chain cost savings, which is still quite significant. With favorable values of the production cost parameters, the cost reductions in the objective function reach 29%. It's important to note that these cost avoidances are open to interpretation since the size of the supply chain network and the magnitude of the production costs are relative to overall supply

Table 4 Production settings

| S | Pt (h) | Regular costs (\$) | Crashing costs (\$) | Total cost (\$) | S | Pt (h) | Regular cost (\$) | Crashing costs as a function of Pt (\$) | Total cost (\$) |
|----|--------|--------------------|---------------------|-----------------|----|--------|-------------------|---|-----------------|
| 1 | 239 | 4302 | 708.45 | 5010.45 | 17 | 222 | 3996 | 868.77 | 4864.77 |
| 2 | 238 | 4284 | 717.00 | 5001.00 | 18 | 221 | 3978 | 879.26 | 4857.26 |
| 3 | 237 | 4266 | 725.66 | 4991.66 | 19 | 220 | 3960 | 889.87 | 4849.87 |
| 4 | 236 | 4248 | 734.42 | 4982.42 | 20 | 219 | 3942 | 900.62 | 4842.62 |
| 5 | 235 | 4230 | 743.29 | 4973.29 | 21 | 218 | 3924 | 911.49 | 4835.49 |
| 6 | 234 | 4212 | 752.26 | 4964.26 | 22 | 217 | 3906 | 922.49 | 4828.49 |
| 7 | 233 | 4194 | 761.34 | 4955.34 | 23 | 216 | 3888 | 933.63 | 4821.63 |
| 8 | 232 | 4176 | 770.53 | 4946.53 | 24 | 215 | 3870 | 944.90 | 4814.90 |
| 9 | 231 | 4158 | 779.83 | 4937.83 | 25 | 214 | 3852 | 956.31 | 4808.31 |
| 10 | 230 | 4140 | 789.25 | 4929.25 | 26 | 213 | 3834 | 967.85 | 4801.85 |
| 11 | 229 | 4122 | 798.78 | 4920.78 | 27 | 212 | 3816 | 979.54 | 4795.54 |
| 12 | 228 | 4104 | 808.42 | 4912.42 | 28 | 211 | 3798 | 991.36 | 4789.36 |
| 13 | 227 | 4086 | 818.18 | 4904.18 | 29 | 210 | 3780 | 1003.33 | 4783.33 |
| 14 | 226 | 4068 | 828.06 | 4896.06 | 30 | 209 | 3762 | 1015.44 | 4777.44 |
| 15 | 225 | 4050 | 838.05 | 4888.05 | 31 | 208 | 3744 | 1027.70 | 4771.70 |
| 16 | 224 | 4032 | 848.17 | 4880.17 | 32 | 207 | 3726 | 1040.11 | 4766.11 |
| 17 | 223 | 4014 | 858.41 | 4872.41 | 33 | 206 | 3708 | 1052.67 | 4760.67 |

Table 5 Eight instances of two inventory locations, seven plants, and two distributors

| | Acc (%) | PTS (h) | Average Pt (with crashing) (h) | Average Pt (without crashing) (h) | PDT (%) | PDI (%) |
|---------|---------|---------|--------------------------------|-----------------------------------|---------|---------|
| | 11.79 | 38.84 | 230.54 | 240 | 19.42 | 3.94 |
| | 29.50 | 83.76 | 156.24 | 240 | 27.81 | 34.90 |
| | 15.60 | 49.75 | 190.25 | 240 | 26.60 | 20.73 |
| | 26.89 | 77.96 | 162.04 | 240 | 26.18 | 32.48 |
| | 14.89 | 47.72 | 192.28 | 240 | 23.00 | 19.88 |
| | 20.53 | 62.66 | 177.34 | 240 | 21.66 | 26.11 |
| | 12.59 | 41.29 | 198.71 | 240 | 21.45 | 17.20 |
| | 20.09 | 61.67 | 178.33 | 240 | 24.16 | 25.70 |
| Average | 18.99 | 57.96 | 185.72 | 240 | 23.79 | 22.62 |

Table 6 High and low values of production and inventory cost parameters

| | High (1) | Low (0) |
|----------|----------|---------|
| α | 700 | 600 |
| β | 0.015 | 0.012 |
| C | 20 | 16 |
| H | 9 | 1 |

Table 7 Runs for variant production cost parameters

| Description | Acc (%) | Average S (h) | Average Pt | Pto | PDT (%) | PDI (%) | PDH (%) | PDL (%) |
|-------------|---------|-----------------|--------------|--------|-----------|-----------|-----------|-----------|
| A0B0C0H0 | 11.79 | 38.84 | 230.54 | 240.00 | 19.60 | 3.94 | 24.48 | 7.88 |
| A1B1C0H0 | 6.44 | 22.37 | 217.63 | 240.00 | 15.02 | 9.32 | 9.59 | 9.59 |
| A1B0C1H1 | 30.48 | 84.60 | 155.40 | 240.00 | 33.06 | 35.25 | 36.15 | 36.15 |
| A0B0C0H0 | 15.60 | 49.75 | 190.25 | 240.00 | 20.31 | 20.73 | 24.44 | 24.44 |
| A1B0C0H1 | 25.71 | 74.08 | 165.92 | 240.00 | 17.73 | 30.87 | 32.30 | 32.30 |
| A0B0C0H0 | 14.89 | 47.72 | 192.28 | 240.00 | 11.26 | 19.88 | 22.87 | 22.87 |
| A1B0C0H0 | 11.77 | 38.87 | 228.99 | 240.00 | 16.71 | 4.59 | 23.22 | 9.17 |
| A0B1C0H0 | 5.65 | 19.75 | 233.67 | 240.00 | 4.39 | 2.64 | 11.18 | 5.28 |
| A1B0C0H0 | 23.07 | 69.18 | 170.82 | 240.00 | 17.76 | 28.83 | 28.96 | 28.96 |
| A0B1C0H0 | 9.10 | 30.80 | 225.25 | 240.00 | 7.69 | 6.14 | 13.38 | 12.29 |
| A1B0C0H0 | 4.54 | 16.05 | 223.95 | 240.00 | 11.36 | 6.69 | 11.55 | 11.55 |
| A1B1C0H1 | 6.51 | 22.14 | 217.86 | 240.00 | 5.61 | 9.23 | 9.23 | 9.23 |
| A1B1C1H1 | 11.88 | 38.50 | 201.50 | 240.00 | 25.40 | 16.04 | 16.23 | 16.23 |
| A1B0C0H0 | 7.22 | 25.00 | 235.21 | 240.00 | 21.73 | 1.99 | 16.85 | 3.99 |
| A0B0C1H0 | 29.32 | 83.43 | 156.57 | 240.00 | 22.17 | 34.76 | 38.82 | 38.82 |
| A0B1C0H0 | 13.65 | 99.51 | 140.49 | 240.00 | 24.07 | 41.46 | 76.60 | 76.60 |
| A1B1C0H1 | 6.77 | 22.97 | 217.03 | 240.00 | 22.02 | 9.57 | 9.84 | 9.84 |
| A1B0C0H0 | 8.09 | 27.65 | 223.97 | 240.00 | 23.20 | 6.68 | 13.36 | 13.36 |
| A0B0C1H0 | 28.98 | 82.74 | 157.26 | 240.00 | 24.54 | 34.48 | 35.46 | 35.46 |
| A0B0C1H0 | 34.94 | 95.33 | 144.67 | 240.00 | 31.45 | 39.72 | 41.04 | 41.04 |
| A1B1C1H0 | 10.47 | 35.03 | 204.97 | 240.00 | 20.86 | 14.60 | 15.01 | 15.01 |
| A1B0C0H0 | 17.77 | 55.64 | 184.36 | 240.00 | 21.13 | 23.18 | 31.07 | 31.07 |
| A0B0C0H1 | 21.89 | 65.07 | 174.93 | 240.00 | 30.54 | 27.11 | 29.62 | 29.62 |
| A0B0C1H0 | 40.53 | 105.98 | 194.23 | 240.00 | 59.54 | 19.07 | 50.17 | 38.14 |
| A0B0C0H1 | 29.05 | 81.53 | 158.47 | 240.00 | 66.99 | 33.97 | 40.30 | 40.30 |

chain costs and are highly variable. Nevertheless, these reductions are significant from the perspective of plant owners/managers and more importantly would bring other benefits to the supply chain such as lead-time reduction, excess capacity, increased throughput, and higher inventory turnover. For instance, percent reduction in the production lead-time can reach 38% (Table 7), which implies that products are moving faster through the supply chain. This can be quite important in cases where lead-time is vital or even more important in the case when customers might penalize producers for late deliveries. In addition, the cost savings in inventory become quite pronounced when the value of $IC_{i,p,t}$ is set at a high value. This could be the case when dealing with products with high holding cost characteristics (i.e., space costs, opportunity cost, and obsolescence costs). For this case, the cost reductions in inventory can reach as high as 35%. To demonstrate other benefits of crashing Pt , Table 7 devises new measures. The percent difference between the average time a product spends in production, at the plant with highest crashing duration (S), when crashing is allowed, to the average time when crashing is not allowed is termed, PDH . The same measure is applied but for the plant with the lowest crashing duration and is termed, PDL . These two measures exemplify the competition between plants in terms of crashing opportunities. PDH

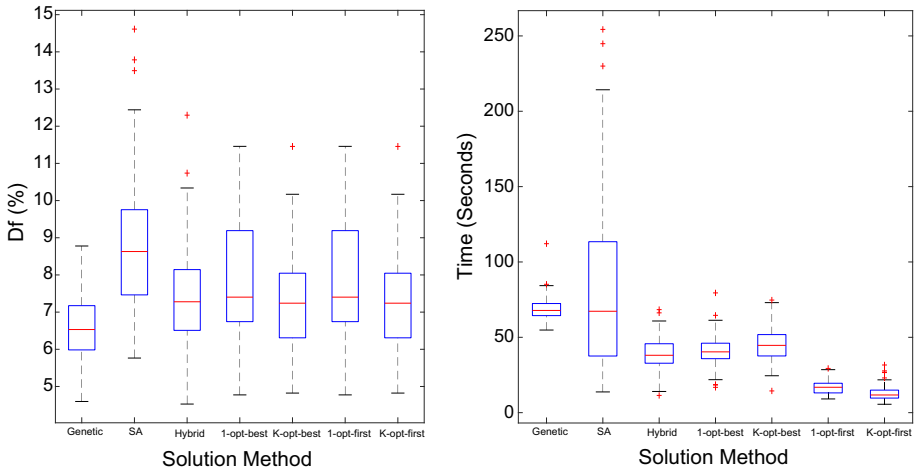


Fig. 14 Heuristic performance for a (7-15-7) supply chain network (Df is to the left)

demonstrates the ability of plants that have advantageous costing parameters (or have access to superior manufacturing processes that can bring the cost of crashing down) to aggressively reduce Pt . Importantly, these plants are able to win more production quota. Moreover, these two measures show the benefits of crashing from the plants' owner/manager point of view, regardless of the magnitude of the supply chain operations.

6.4 Large scale runs

Given the complexity of the work and more importantly the vulnerability to variation, we need to compare the heuristics at more instances. To assure consistency and fair representation, the coding, compilation, and execution are done using a MATLAB platform on a fixed computational system (Intel R, Core TM, i5-2400 CPU @3.10 GHz). In Fig. 14, we contest one hundred runs for a supply chain network of medium size (7-15-17). The figure presents a box plot of the 100 runs. Here, the genetic search tends to give the lowest Df average of approximately 6.5%, followed by the K -opt-best, Hybrid, and K -opt-first. The genetic search also gives the lowest variation and least outliers (i.e., no outliers). However, the computational times average at around 65 s which is the highest among the heuristics. The 1-opt procedures tend to give the lowest time. The Hybrid heuristic shows a good performance in terms of solution gap and computational time.

7 Conclusion

The main contribution of this paper is the inclusion of crashing cost in supply chain network design (SCND). The vast majority of papers in SCND do not attempt to look at the opportunity of crashing production time (Pt). There is a gap in the research for works that minimize the overall supply chain costs via the crashing of Pt . This work integrates the cost of crashing among other prominent supply chain costs such as inventory, production, and transportation. The practical implications of this paper are the ability to shorten Pt in a manner that would

reduce the overall production costs and to increase the throughput of the supply chain. However, the benefits of reducing the overall production cost and decreasing lead-time are highly dependent on the characteristics of the production cost parameters discussed. It is also dependent on the topography of the supply chain. Conversely, regardless of the topography of the supply chain or the production environment, the work brings many benefits such as lead-time shortening, throughput improvement, and capacity enhancements. Alternatively, other practical implications such as penalties (for late deliveries) might have companies invest more aggressively in optimizing their production resources by crashing Pt . Further, shortening Pt might prove important for companies that are adopting new process technologies or are still in route to optimality.

Importantly, the work's strong contribution is in the heuristics used to solve the model. Nonlinear binary models are difficult to solve (Dua 2015; Katayama and Narihisa 2001). Binary quadratic programming is said to be NP-Hard (Katayama and Narihisa 2001). Given the complexity of the model, near-optimal solution are obtained within small tolerance of the lower bound. This work pushes the envelope in terms of complexity as the objective function has mathematical terms that include the multiplication of two different decision variables and nonlinearity in the constraints. To solve the model, a combination of gradient and search heuristics are supplied. The work presents seven different heuristics to solve the model. From the results presented, we can sense the complexity of the solution especially with larger networks. The work contests the seven solution heuristics and shows the effectiveness of a genetic-based heuristic in arriving at the solution. This is especially true in the case where supply chain networks are large.

This work paves the road ahead for other works to follow. For instance, future works can integrate a penalty cost where companies incurring charges for late deliveries can lessen the financial impact by crashing Pt . In addition, companies in highly competitive environments where arriving first to the market is pivotal might benefit from a model that integrates the opportunity cost of time. Pt in this work has been shortened without attaching a dollar value for the time saved. For instance, the time reduction of producing a given product can be seen as an opportunity to beat a competitor to market and hence can have a dollar value attached to it. This can be the subject of future research.

References

- Alzaman, C., Zhang, Z. H., & Diabat, A. (2018). Supply chain network design with direct and indirect production costs: Hybrid gradient and local search based heuristics. *International Journal of Production Economics*, 203, 203–215.
- Arntzen, B. C., Brown, G. G., Harrison, T. P., & Trafton, L. L. (1995). Global supply chain management at digital equipment corporation. *Interfaces*, 25(1), 69–93.
- Babazadeh, R., Razmi, J., Rabbani, M., & Pishvae, M. S. (2017). An integrated data envelopment analysis–mathematical programming approach to strategic biodiesel supply chain network design problem. *Journal of Cleaner Production*, 147, 694–707.
- Badria, H., Ghomia, S. F., & Hejazib, T. H. (2017). A two-stage stochastic programming approach for value-based closed-loop supply chain network design. *Transportation Research Part E*, 105, 1–17.
- Beamon, B. M. (1998). Supply chain design and analysis: Models and methods. *International Journal of Production Economics*, 55, 281–294.
- Blackburn, J. D. (1991). *Time-based competition: The next battle ground in American manufacturing* (p. 1991). Homewood: Business One Irwin.
- Cakravastia, A., Toha, I. S., & Nakamura, N. (2002). A two-stage model for the design of supply chain networks. *International Journal of Production Economics*, 80(3), 231–248.
- Coskun, S., Ozgur, L., Polat, O., & Gungor, A. (2016). A model proposal for green supply chain network design based on consumer segmentation. *Journal of Cleaner Production*, 110, 149–157.

- Diabat, A. (2016). A capacitated facility location and inventory management problem with single sourcing. *Optimization Letters*, 10(7), 1577–1592.
- Diabat, A., & Al-Salem, M. (2015). An integrated supply chain problem with environmental considerations. *International Journal of Production Economics*, 164, 330–338.
- Diaby, M., Cruz, J. M., & Nsakanda, A. L. (2013). Shortening cycle times in multi-product, capacitated production environments through quality level improvements and setup reduction. *European Journal of Operational Research*, 228(3), 526–535.
- Dua, V. (2015). Mixed integer polynomial programming. *Computers & Chemical Engineering*, 72, 387–394.
- Esmailikia, M., Fahimnia, B., Sarkis, J., Govindan, K., Kumar, A., & Mo, J. (2016). A tactical supply chain planning model with multiple flexibility options: An empirical evaluation. *Annals of Operations Research*, 244(2), 429–454.
- Fahimnia, B., Sarkis, J., & Eshragh, A. (2015). A tradeoff model for green supply chain planning: A leanness-versus-greenness analysis. *Omega*, 54, 173–190.
- Geoffrion, A. M., & Graves, G. (1974). Multi-commodity distribution system design by Benders decomposition. *Management Science*, 29(5), 822–844.
- Govindan, K., & Fattahi, M. (2017). Investigating risk and robustness measures for supply chain network design under demand uncertainty: A case study of glass supply chain. *International Journal of Production Economics*, 183, 680–699.
- Govindan, K., Jafarian, A., Khodaverdi, R., & Devika, K. (2014). Two-echelon multiple-vehicle location–routing problem with time windows for optimization of sustainable supply chain network of perishable food. *International Journal of Production Economics*, 152, 9–28.
- Grossmann, I. E. (2002). Review of nonlinear mixed-integer and disjunctive programming techniques. *Optimization and engineering*, 3(3), 227–252.
- Hammami, R., & Frein, Y. (2013). An optimisation model for the design of global multi-echelon supply chains under lead time constraints. *International Journal of Production Research*, 51(9), 2760–2775.
- Hasani, A., Zegordi, S. H., & Nikbaksh, E. (2015). Robust closed-loop global supply chain network design under uncertainty: The case of the medical device industry. *International Journal of Production Research*, 53(5), 1596–1624.
- Hillier, F. S., & Lieberman, G. J. (1995). *Introduction to mathematical programming*. McGraw-Hill.
- Holland, J. H. (1975). *Adaptation in natural and artificial systems*. Ann Arbor: University of Michigan Press.
- Hoque, M. A., & Goyal, S. K. (2006). A heuristic solution procedure for an integrated inventory system under controllable lead-time with equal or unequal sized batch shipments between a vendor and a buyer. *International Journal of Production Economics*, 102(2), 217–225.
- Jayaraman, V., & Ross, A. (2003). A simulated annealing methodology to distribution network design and management. *European Journal of Operational Research*, 144(3), 629–645.
- Jian, M., Fang, X., Jin, L. Q., & Rajapov, A. (2015). The impact of lead time compression on demand forecasting risk and production cost: A newsvendor model. *Transportation Research Part E: Logistics and Transportation Review*, 84, 61–72.
- Katayama, K., & Narihisa, H. (2001). On fundamental design of parthenogenetic algorithm for the binary quadratic programming problem. In *Proceedings of the 2001 Congress on evolutionary computation* (Vol. 1, pp. 356–363). IEEE.
- Kattayama, K., & Narihisa, H. (2001). Performance of simulated annealing-based heuristic for the unconstrained binary quadratic programming problem. *European Journal of Operational Research*, 134, 103–119.
- Kaya, O., & Urek, B. (2016). A mixed integer nonlinear programming model and heuristic solutions for location, inventory and pricing decisions in a closed loop supply chain. *Computers & Operations Research*, 65, 93–103.
- Keyvanshokoh, E., Ryan, S. M., & Kabir, E. (2016). Hybrid robust and stochastic optimization for closed-loop supply chain network design using accelerated Benders decomposition. *European Journal of Operational Research*, 249(1), 76–92.
- Kolda, T. G., Lewis, R. M., & Torczon, V. (2003). Optimization by direct search: New perspectives on some classical and modern methods. *SIAM Review*, 45(3), 385–482.
- Lim, M. K., Mak, H. Y., & Shen, Z. J. M. (2016). Agility and proximity considerations in supply chain design. *Management Science*, 63(4), 1026–1041.
- Longinidis, P., Georgiadis, M. C., & Kozanidis, G. (2015). Integrating operational hedging of exchange rate risk in the optimal design of global supply chain networks. *Industrial and Engineering Chemistry Research*, 54(24), 6311–6325.
- Mangla, S. K., Kumar, P., & Barua, M. K. (2016). An integrated methodology of FTA and fuzzy AHP for risk assessment in green supply chain. *International Journal of Operational Research*, 25(1), 77–99.

- Marti, J. M. C., Tancrez, J. S., & Seifert, R. W. (2015). Carbon footprint and responsiveness trade-offs in supply chain network design. *International Journal of Production Economics*, *166*, 129–142.
- Merz, P., & Freisleben, B. (2002). Greedy and local search heuristics for unconstrained binary quadratic programming. *Journal of Heuristics*, *8*, 197–213.
- Michalewicz, Z. (1996). *Genetic algorithms + data structures = evolution programs*. New York: Springer.
- Mizgier, K. J. (2017). Global sensitivity analysis and aggregation of risk in multi-product supply chain networks. *International Journal of Production Research*, *55*(1), 130–144.
- Mohri, M., Rostamizadeh, A., & Talwalkar, A. (2018). *Foundations of machine learning*. MIT Press.
- Pan, F., & Nagi, R. (2013). Multi-echelon supply chain network design in agile manufacturing. *Omega*, *41*(6), 969–983.
- Petridis, K. (2015). Optimal design of multi-echelon supply chain networks under normally distributed demand. *Annals of Operations Research*, *227*(1), 63–91.
- Pham, T., & Yenradee, P. (2017). Optimal supply chain network design with process network and BOM under uncertainties: A case study in toothbrush industry. *Computers & Industrial Engineering*, *108*, 177–191.
- Pishvae, M. S., & Torabi, S. A. (2010). A possibilistic programming approach for closed-loop supply chain network design under uncertainty. *Fuzzy Sets and Systems*, *161*(20), 2668–2683.
- Ramudhin, A., Alzaman, C., & Bulgak, A. (2008). Incorporating the cost of quality in supply chain design. *Journal of Quality in Maintenance Engineering*, *14*, 71–86.
- Rezapour, S., Farahani, R. Z., & Pourakbar, M. (2017). Resilient supply chain network design under competition: A case study. *European Journal of Operational Research*, *259*(3), 1017–1035.
- Sadjady, H., & Davoudpour, H. (2012). Two-echelon, multi-commodity supply chain network design with mode selection, lead-times and inventory costs. *Computers & Operations Research*, *39*(7), 1345–1354.
- Santos, T., Ahmed, S., Goetschalckx, M., & Shapiro, A. (2005). A stochastic programming approach for supply chain network design under uncertainty. *European Journal of Operational Research*, *167*(1), 96–115.
- Sarrafha, K., Rahmati, S. H. A., Niaki, S. T. A., & Zaretalab, A. (2015). A bi-objective integrated procurement, production, and distribution problem of a multi-echelon supply chain network design: A new tuned MOEA. *Computers & Operations Research*, *54*, 35–51.
- Shah, N. H., Gor, R. M., & Soni, H. (2007). *Operations research*. New Delhi: PHI Learning Pvt. Ltd.
- Stevenson, W. J., Hojati, M., & Cao, J. (2007). *Operations management*. McGraw-Hill Irwin.
- Towill, D. R. (1996). Time compression and supply chain management—A guided tour. *Supply Chain Management: An International Journal*, *1*(1), 15–27.
- Vahdani, B., & Mohammadi, M. (2015). A bi-objective interval-stochastic robust optimization model for designing closed loop supply chain network with multi-priority queuing system. *International Journal of Production Economics*, *170*, 67–87.
- Varsei, M., & Polyakovskiy, S. (2017). Sustainable supply chain network design: A case of the wine industry in Australia. *Omega*, *66*, 236–247.
- Yang, M. F. (2010). Supply chain integrated inventory model with present value and dependent crashing cost is polynomial. *Mathematical and Computer Modelling*, *51*(5), 802–809.
- Yildiz, H., Yoon, J., Talluri, S., & Ho, W. (2016). Reliable supply chain network design. *Decision Sciences*, *47*(4), 661–698.