Mohammadi, Maryam; Musa, Siti Nurmaya; Omar, Mohd Bin

Optimization of multi-plant capacitated lot-sizing problems in an integrated supply chain network using calibrated metaheuristic algorithms

Published in:
International Journal of Operational Research

Published: 01/05/2020

Document Version
Peer reviewed version

Published under the following license:
Unspecified

Please cite the original version:
Optimisation of multi-plant capacitated lot-sizing problems in an integrated supply chain network using calibrated metaheuristic algorithms

Maryam Mohammadi
Process Control and Automation Research Group, School of Chemical Technology, Aalto University, Espoo, Finland
Email: maryam.mohammadi@aalto.fi
Email: maryam.mohammadi1987@gmail.com

S. Nurmaya Musa*
Centre of Advanced Manufacturing and Material Processing, University of Malaya, Kuala Lumpur, Malaysia
and
Department of Mechanical Engineering, Faculty of Engineering, University of Malaya, Kuala Lumpur, Malaysia
Email: nurmaya@um.edu.my
*Corresponding author

Mohd Bin Omar
Institute of Mathematical Sciences, Faculty of Science, University of Malaya, Kuala Lumpur, Malaysia
Email: mohd@um.edu.my

Abstract: In this paper, a mathematical model for a multi-item multi-period capacitated lot-sizing problem in an integrated supply chain network composed of multiple suppliers, plants and distribution centres is developed. The combinations of several functions such as purchasing, production, storage, backordering and transportation are considered. The objective is to simultaneously determine the optimal raw material order quantity, production and inventory levels, and the transportation amount, so that the demand can be satisfied with the lowest possible cost. Transfer decisions between plants are made when demand at a plant can be fulfilled by other production sites to cope with the under-capacity and stock-out problems of that plant. Since the proposed model is NP-hard, a genetic algorithm is used to solve the model. To validate the results, particle swarm optimisation and imperialist competitive algorithm are applied to solve the model as well. The results show that genetic algorithm offers better solution compared to other algorithms.
Keywords: capacitated lot-sizing; multi-plant; production and distribution planning; integrated supply chain; optimisation; metaheuristic algorithms; genetic algorithm; GA; particle swarm optimisation; PSO; imperialist competitive algorithm; ICA.


Biographical notes: Maryam Mohammadi is currently a Post-doctoral Research Fellow in Process Control and Automation Research Group, School of Chemical Technology, Aalto University, Finland. She earned her Bachelor’s in Industrial Engineering (Systems Planning and Analysis) from the Islamic Azad University, South Tehran Branch, Iran, and Master’s in Industrial Engineering from the University Technology Malaysia, Malaysia. She received her PhD in Manufacturing Engineering from the University of Malaya, Malaysia. Her research interests include production planning and control systems, supply chain management, operations research and optimisation.

Siti Nurmaya Musa is a Senior Lecturer in Department of Mechanical Engineering, University of Malaya, Malaysia. She received her Bachelor’s in Industrial Engineering from the University of Wisconsin-Milwaukee, USA and Master’s in Manufacturing Engineering and Management from the University of Nottingham, UK. She obtained her PhD in Production Economics from the Linköping University, Sweden. Her areas of expertise are supply chain management, manufacturing management, production and logistics, production economics, operations research and management science.

Mohd Bin Omar is a Professor at the Institute of Mathematical Sciences, University of Malaya, Malaysia. He received his PhD in Operational Research from the University of Exeter, UK in 1999. His current research interests include production and inventory control and mathematical modelling. He has contributed papers to prestigious journals such as European Journal of Operational Research, Computers and Industrial Engineering, International Journal of Production Economics, International Journal of Production Research, International Journal of Systems Science, Applied Mathematical Modelling and Applied Soft Computing.

1 Introduction

Enterprises are facing highly competitive and fast-changing business environments. To lower the production cost and meet customer demands in a timely fashion, companies have used the strategy of outsourcing as a method to increase production capacity (De Kok, 2000; Tukel and Wasti, 2001; McCarthy and Anagnostou, 2004; Ruiz-Torres et al., 2006). In a global scale, companies cannot compete on their own in the market (Conklin and Perdue, 1994), thus requiring support from other partners by developing a multi-plant manufacturing supply chain (SC) to maximise competitive advantages of SC members (Chen, 2010). An integrated company may possess a hierarchy of production plants, in which the production and assembly processes for manufacturing a product can be distributed to different plants established in geographically scattered locations (Lin.
and Chen, 2007). Though, once a job is allocated to a plant, it is usually inefficient to transfer it to other factories (Chan et al., 2005), unexpected circumstances such as machine breakdowns and insufficient capacities and resources may justify the reallocation of jobs to other plants in real time (Alvarez, 2007).

For many organisations, the shift from the conventional single plant to multi-plant manufacturing environment may bring about difficulties in the production planning and distribution sectors. Thus, decision makers in multi-plant systems have to attempt towards integration of several manufacturing plants’ activities in such a way that they align their tasks in direction of improving the overall performance of the enterprise (Mussa, 2009; Olhager and Feldmann, 2017). The move towards incorporated multi-plant configurations would bring a wide range of opportunities in terms of cost reduction in manufacturing and logistics activities as well as competitive advantages in the global economic arena. In addition, it allows the company to establish reliable commitments with customers as efficiently as possible, and maximise the customer service level (Alvarez, 2007; Junqueira and Morabito, 2012; Darvish et al., 2016; Taxakis and Papadopoulos, 2016).

In a multi-plant scenario, a crucial managerial concern is the determination of production quantities (lot size) for each product in each plant and period, such that the demands of products are satisfied and total costs at all factories are minimised (Bhatnagar et al., 1993). Multi-plant capacitated lot-sizing problem (MPCLSP) with multiple time periods and products consists of several manufacturing centres that produce identical products and allows the inter-plant transfers. Lot-sizing problems are more complicated in multi-plant systems because of the interdependency between the plants.

In this paper, a multi-item, multi-period, capacitated lot-sizing problem with inter-plant interactions, multiple suppliers and distribution centres is addressed, which can cover a variety of problems arising in the literature and in practice. The fundamental concept is to simultaneously optimise decision variables of different functions in an integrated SC network, which have been conventionally optimised individually due to the complexity in their integration.

The subsequent sections of this paper are organised as follow: Section 2 is devoted to literature review. Section 3 presents the mathematical formulation of the proposed problem. The applied methods are explained in Section 4 and the algorithms’ parameters calibration procedure is presented in Section 5. A numerical example is described in Section 6, and the computational results are discussed in Section 7. Finally, the conclusions are shared in Section 8.

2 Literature review

The multi-plant production environment is extremely common in many manufacturing industries, such as electric power generating industry (Westfield, 1955), chemical process industry (Timpe and Kallrath, 2000), automotive manufacturing industry (Gnoni et al., 2003), steel corporation (Sambasivan and Yahya, 2005), production of thin film transistor-liquid crystal display (Lin and Chen, 2007), petrochemical (Alfares, 2009), papers processing (Mirzapour Al-E-Hashem et al., 2011), seed corn industry (Junqueira and Morabito, 2012), beverage industry (Guimarães et al., 2012), semiconductor foundry (Chen et al., 2013), ceramic industry (Alemany et al., 2013), cultivation system (Ke et al.,
Sambasivan and Schmidt (2002) proposed a heuristic approach based on transfers of production lots between the periods and the plants to solve the MPCLSP. Sambasivan and Yahya (2005) obtained better results for the problem studied in Sambasivan and Schmidt (2002) using the Lagrangian relaxation approach. Nascimento et al. (2010) embedded the setup carry-over to the proposed MPCLSP by Sambasivan and Schmidt (2002), and developed a greedy randomised adaptive search procedure, as well as a path relinking intensification procedure, which outperformed the Lagrangian relaxation approach presented by Sambasivan and Yahya (2005). However, in these studies, only the production and the inter-plant transportation of products were considered, and the coordination between production and distribution was not taken into account.

Integrating production and distribution functions may result in considerable amount of saving in operational and distribution costs, balancing production lots and vehicle loads, and reducing the inventory level and hence the inventory cost, as well as the stock-out risk (Fumero and Vercellis, 1999). Pirkul and Jayaraman (1998), Jolayemi and Olorunniwo (2004), Park (2005), Aghezzaf (2007), and Darvish et al. (2016) investigated the value of integrating production and distribution through computational studies in multi-plant manufacturing systems. Seyedhosseini and Ghoreyshi (2015) proposed a new formulation for integrating production and distribution planning of perishable products through lot-sizing and inventory routing problem. Due to the computational complexity of the problem, an efficient heuristic algorithm was developed to find good quality solutions in a reasonable time, and results were compared with the solutions provided by LINGO optimiser.

It is proven that the single plant multi-item capacitated lot-sizing problem is NP-Hard (Florian et al., 1980; Bitran and Yanasse, 1982; Mohammadi et al., 2014, 2015), so is the respective multi-plant version. Therefore, metaheuristic approaches can be used to efficiently tackle such complex problems and obtain good solutions within a reasonable computation time. Tseng et al. (2010a, 2010b) applied genetic algorithm (GA) and particle swarm optimisation (PSO) to solve an integrated assembly sequence planning and plant assignment problem, where products are assembled in a multi-plant system with the objective of minimising the assembly and multi-plant costs. Yang et al. (2010) proposed a quasi-transportation problem for multi-plant order allocation to minimise the total cost under the capacity load constraint, and solved it using the GA. Behnamian and Ghomi (2013) developed a heuristic algorithm and the GA approach for sequencing and scheduling of distributed multi-factory production network problem with parallel machines in order to minimise the maximum completion time of jobs.

Lin et al. (2015) considered a multi-plant remanufacturing system where decisions have to be made on the choice of plant to perform the remanufacturing and the remanufacturing options. The solution method was composed of the linear physical programming and the multi-level encoding GA. It was shown that the proposed integrated approach performs better compared with the regular GA in terms of makespan. Manimaran and Selladura (2014) developed a PSO algorithm for the multi-stage SC network problem, with the aim of selecting the optimum set of suppliers, plants and distribution centres, and determining the quantities to be supplied to satisfy the customer demand with a minimum total distribution cost. The performance of the PSO was
compared in terms of total distribution cost with spanning tree-based GA, improved Prüfer number encoding-based GA and LINGO. It was shown that the proposed PSO algorithm can produce better quality solutions.

Mohammadi and Musa (2016) developed a mathematical model to investigate the value of coordination of production and distribution planning in a three-layer SC consisting of multiple suppliers, manufacturers, and distributors. The proposed NP-Hard model was solved by simulated annealing and the GA. The results showed that the presented algorithms are computationally effective and beneficial for obtaining the optimal solution for the proposed lot-sizing problem. Atabaki and Mohammadi (2017) presented a nonlinear programming of the multiple suppliers and multiple period’s inventory lot-sizing problem with supplier and storage capacity constraints. A calibrated GA was used to solve the problem and results were compared with GAMS software. It was shown that the proposed GA is a more suitable solution methodology for large size problems.

Numerous studies investigating the lot-sizing problems have concentrated on how to effectively schedule production operations within the confines of a single production facility. However, from the perspective of minimising the total cost in the SC network, companies usually acknowledge that the cost of a product is not only determined with the amount of factory resources used to convert the raw material into a finished product, but also with the amount of resources used to deliver the product to the customer. Hence, concentrating only on lot-sizing and scheduling of production operations within plants may not be sufficient to obtain the desired low levels in the production and logistics costs of the SC.

Based on the existing literature, little attention has been paid to the optimisation of MPCLSP in an integrated SC network. Generally, the combinations of essential functions associated to the integrated SC such as purchasing, production, storage, backordering, and transportation have not been discussed entirely and comprehensively in the previous studies. Thus, there is a need for developing an integrated mathematical model and effective solution approaches to address these shortcomings. Three well-known metaheuristic algorithms namely the GA, PSO, and imperialist competitive algorithm (ICA) are selected to solve the proposed model.

### 3 Problem description and mathematical formulations

The considered SC problem consists of $M$ suppliers, $J$ plants, and $W$ distribution centres, as shown in Figure 1. It is assumed that production takes place in a multi-plant manufacturing company, where the plants are located in different areas of a country. Each product is made of $K$ raw materials, which are provided by the suppliers. It is designated that for each raw material type there is only one particular supplier. Each plant is characterised by its own inventory and production capacities. It is possible to store excess production at the plant storage that has capacity limit, but no storage is possible for end products at distribution centres.

Any of the products produced in each plant can be transported to any of the distribution centres that are located in different areas. Obviously, the demand in a distribution centre is initially served by the closest plant. Transfer decisions between plants are made when demand observed at a plant can be satisfied by other production
sites to cope with the under-capacity and/or stock-out problem of that particular plant. It should be noted that the customer would pay only for the transportation from the nearest plant. The transportation cost from other plants to the plant where demand has been placed, has to be borne by the company.

Since all factories, suppliers, and distribution centres are spread out geographically, the transportation cost can vary. Homogenous vehicles of given capacity are stationed at each supplier and plant to deliver raw materials from suppliers to plants, transfer the products between the production plants and from plants to distribution centres. The model is developed with the assumption that sales are made at the distribution centres. In addition, backordering is allowed when demand at a distribution centre cannot be entirely satisfied.

**Figure 1** A schematic representing the proposed multi-plant problem (see online version for colours)

![Schematic diagram](image)

### 3.1 Notation

**Indices**

- \( i \) product, \( i \in \{1, 2, \ldots, N\} \)
- \( k \) raw material, \( k \in \{1, 2, \ldots, K\} \)
- \( v \) resource, \( v \in \{1, 2, \ldots, V\} \)
- \( m \) supplier, \( m \in \{1, 2, \ldots, M\} \)
- \( j, j' \) plant, \( j, j' \in \{1, 2, \ldots, J\} \)
- \( w \) distribution centre, \( w \in \{1, 2, \ldots, W\} \)
- \( t \) time period, \( t \in \{1, 2, \ldots, T\} \).
Optimisation of multi-plant capacitated lot-sizing problems

Parameters

\( d_{iwt} \) demand of product \( i \) in distribution centre \( w \) in period \( t \)
\( U_{iwt} \) selling price of product \( i \) in distribution centre \( w \) in period \( t \)
\( A_{ijt} \) setup time of product \( i \) in plant \( j \) in period \( t \)
\( P_{ijt} \) production time of product \( i \) in plant \( j \) in period \( t \)
\( F_{jt} \) total available production time in plant \( j \) in period \( t \)
\( x_{ki} \) amount of raw material \( k \) required to produce a unit of product \( i \)
\( E_{kmt} \) number of raw material \( k \) that can be provided by supplier \( m \) in period \( t \)
\( \lambda_{kmjt} \) percentage of rejected raw material \( k \) delivered by supplier \( m \) to plant \( j \) in period \( t \)
\( R_{vij} \) amount of resource \( v \) required to produce a unit of product \( i \) in plant \( j \)
\( N_{vjt} \) total amount of resource \( v \) available in plant \( j \) in period \( t \)
\( \pi_{kmt} \) ordering cost of raw material \( k \) at supplier \( m \) in period \( t \)
\( \tau_{kmt} \) purchasing cost of raw material \( k \) at supplier \( m \) in period \( t \)
\( S_{ijt} \) setup cost for product \( i \) in plant \( j \) in period \( t \)
\( O_{ijt} \) production cost of product \( i \) in plant \( j \) in period \( t \)
\( H_{kjt} \) holding cost of raw material \( k \) in plant \( j \) in period \( t \)
\( H_{ijt}^* \) holding cost of product \( i \) in plant \( j \) in period \( t \)
\( B_{iwt} \) backordering cost of product \( i \) in distribution centre \( w \) in period \( t \)
\( \sigma_{pj} \) distance between supplier \( m \) and plant \( j \)
\( \mu_{jl} \) distance between plant \( j \) and plant \( l \)
\( \zeta_{wj} \) distance between plant \( j \) and distribution centre \( w \)
\( \sigma_{kj} \) storage capacity for raw material \( k \) in plant \( j \)
\( \sigma_{ij}^* \) storage capacity for item \( i \) in plant \( j \)
\( \varsigma_k \) vehicle available capacity respect to raw material \( k \)
\( \varsigma_i^* \) vehicle available capacity respect to product \( i \)
\( \eta^f \) fixed transportation cost of vehicle
\( \eta^v \) variable transportation cost of vehicle per trip
\( \rho_k \) safety stock coefficient respect to raw material \( k \)
\( \zeta_j \) performance percentage of available time in plant \( j \)
\( \zeta_{vij} \) productivity percentage of resource \( v \) in plant \( j \)
\( \delta \) a very large number
M. Mohammadi et al.

\( \delta' \) a very large number.

**Decision variables**

- \( Q_{ijt} \) quantity of product \( i \) produced in plant \( j \) in period \( t \)
- \( \alpha_{kmjt} \) purchase amount of raw material \( k \) shipped from supplier \( m \) to plant \( j \) in period \( t \)
- \( I_{ijt} \) inventory level of raw material \( k \) stored in plant \( j \) at the end of period \( t \)
- \( I'_{ijt} \) inventory level of product \( i \) stored in plant \( j \) at the end of period \( t \)
- \( C_{ijwt} \) quantity of product \( i \) that is available to be shipped from plant \( j \) to distribution centre \( w \) in period \( t \)
- \( Z_{ijlt} \) quantity of product \( i \) transferred from plant \( j \) to plant \( l \) in period \( t \)
- \( Y_{ijwt} \) total number of product \( i \) shipped from plant \( j \) to distribution centre \( w \) in period \( t \)
- \( b_{ijw} \) shortage amount of product \( i \) in distribution centre \( w \) in period \( t \)
- \( \phi_{kmjt} \) number of vehicles required to ship products from supplier \( m \) to plant \( j \) in period \( t \)
- \( \nu_{ijlt} \) number of vehicles required to transfer products from plant \( j \) to plant \( l \) in period \( t \)
- \( \Omega_{jwt} \) number of vehicles required to ship products from plant \( j \) to distribution centre \( w \) in period \( t \)
- \( \chi_{ijt} \) binary variable, \( \chi_{ijt} = 1 \) if there is a setup for product \( i \) in plant \( j \) in period \( t \), else \( \chi_{ijt} = 0 \)
- \( \phi_{kmjt} \) binary variable, \( \phi_{kmjt} = 1 \) if an order of raw material \( k \) is allocated to supplier \( m \) by plant \( j \) in period \( t \), else \( \phi_{kmjt} = 0 \).

### 3.2 Constraints

#### 3.2.1 Raw material purchasing constraint

Equation (1) shows the required amount of raw material \( k \) that plant \( j \) must purchase from supplier \( m \) in period \( t \). If total amount of raw material \( k \) used in production of all items multiplied by a safety stock coefficient \( (\rho_k) \) is less than the existing inventory of raw material \( k \), then the factory does not need to order any raw material. The safety stock coefficient is considered to protect the firm in uncertain conditions, i.e., if a supplier fails to deliver the raw material at the required time, or the supplier’s quality is found to be substandard upon inspection, which would leave the plant without the sufficient raw materials.

\[
\alpha_{kmjt} = \max \left\{ 0, \sum_i \rho_k (x_{ki}Q_{ijt}) - I_{ij(t-1)} \right\} \quad \forall k, m, j, t
\]

### 3.2.2 Inventory constraints for raw materials

Initial inventory level of raw materials is considered to be zero as shown in equation (2).

\[
I_{ij(t-1)} = 0 \quad \forall k, j, t = 1
\]
Equation (3) represents the balance equation for the inventory of raw materials in plants at the end of period $t$. It must be noted that there is only one supplier for each type of raw material. Furthermore, plants do not pay for the rejected raw materials, and their associated cost is paid by the respective supplier.

$$I_{ijt} = I_{ij(t-1)} + \alpha_{tij} - \lambda_{tij} - \sum_{l} s_{ltij} Q_{ijt} \quad \forall k, m, j, t$$  \hspace{1cm} (3)

### 3.2.3 Charging ordering cost constraint

Equation (4) describes that a plant cannot place a procurement order without charging an ordering cost. $\phi_{tij}$ is a binary variable with value of 1 if an order is allocated to supplier $m$ during time period $t$, otherwise, it is 0. The symbol $\delta$ is defined as a sufficiently large number to ensure that it is greater than each $\alpha_{tij}$.

$$\alpha_{tij} \leq \delta \phi_{tij} \quad \forall k, m, j, t$$  \hspace{1cm} (4)

### 3.2.4 Supplier capacity constraint

Equation (5) ensures that the ordering size of raw materials is limited by the supplier’s capacity.

$$\sum_{j} \alpha_{tij} \leq E_{km} \quad \forall k, m, t$$  \hspace{1cm} (5)

### 3.2.5 Inventory constraints for finished items at plants

Equation (6) shows the initial inventory level of products at the beginning of planning horizon.

$$I_{i’j’t(-1)} = 0 \quad \forall i, j, t = 1$$  \hspace{1cm} (6)

Equation (7) is the inventory balance equation for products in plants at the end of each period.

$$I_{ijt} = I_{ij(t-1)} + Q_{ijt} + \sum_{l \neq j} Z_{ijlt} - Y_{ijwt} - \sum_{i \neq j} Z_{ijlt} \quad \forall i, j, w, t$$  \hspace{1cm} (7)

It is supposed that if during period $t$ there is a transfer into plant $j$, there cannot be any transfer out from plant $j$ to other plants during that period. Hence:

$$Z_{ijlt} \times Z_{ijjt} = 0 \quad \forall i, j, l \text{ and } l' \neq j, t$$  \hspace{1cm} (8)

### 3.2.6 Setup forcing constraint

Equation (9) forces variable $\chi_{ijt}$ to be nonzero if $Q_{ijt}$ is nonzero; therefore, if there is production of product $i$ in period $t$, i.e., $Q_{ijt} > 0$, then $\chi_{ijt} = 1$ and the fixed cost of $S_{ijt}$ is charged. The symbol $\delta'$ is defined as a sufficiently large number to ensure that it is greater than each $Q_{ijt}$.

$$Q_{ijt} \leq \delta' \chi_{ijt} \quad \forall i, j, t$$  \hspace{1cm} (9)
3.2.7 Total available time constraint

Equation (10) limits the production time available in a plant during period \( t \). The overall time consumptions for production and setup in each plant for all products must be lower than or equal to the available time capacity. The efficiency of the available time is also considered.

\[
\sum_i (P_{q_{ij}}Q_{q_{ij}} + A_{q_{ij}}X_{q_{ij}}) \leq F_{q_{ij}}{\varepsilon_{ij}} \quad \forall j, t
\]  

(10)

3.2.8 Resource constraint

Equation (11) ensures that a manufacturer does not plan beyond the available resources (machine or human) in the plant in each period. The resources’ productivity is also taken into account.

\[
\sum_i R_{n_{ij}}Q_{n_{ij}} \leq N_{n_{ij}}{\varepsilon_{ij}} \quad \forall v, j, t
\]

(11)

3.2.9 Transportation limitation constraint

Equation (12) shows that the number of products available to be transferred from plant \( j \) to distribution centre \( w \) and other plants in period \( t \) should not exceed the previous period inventory plus the current production quantity in plant \( j \) and transferred products to plant \( j \) in period \( t \).

\[
C_{wjt} + \sum_{i \in j} Z_{ijt} \leq I_{ijt-1} + Q_{ijt} + \sum_{l \neq j} Z_{ijl} \quad \forall i, j, w, t
\]

(12)

Equation (13) restricts receiving the products in plant \( j \) from other plants during period \( t \). It implies that if total amount of item \( i \) available in plant \( j \) to be transferred to distribution centre \( w \) in period \( t \) is greater than the backorder amount from previous period plus the demand in distribution \( w \) in period \( t \), then plant \( j \) does not need outsourcing. In this condition \( C_{wjt} \) is equal to \( Y_{wjt} \). Otherwise, plant \( j \) needs to request the shortage amount of item \( i \) from other plants.

\[
\sum_{l \neq j} Z_{ijl} \leq \max \{ (b_{jti(t-1)} + d_{jti}) - C_{wjt}, 0 \} \quad \forall i, j, w, t
\]

(13)

3.2.10 Vehicles constraints

Equations (14) to (16) calculate the number of vehicles required for transportation of raw materials from suppliers to plants, and the products from a plant to other plants and distribution centres respectively.

\[
\sum_k \alpha_{kimj} \leq \phi_{kimj} \quad \forall m, j, t
\]

(14)

\[
\sum_i Z_{ijl} \leq v_{jlt} \quad \forall j, l \neq j, t
\]

(15)
Optimisation of multi-plant capacitated lot-sizing problems

\[ \sum_{i} Y_{jw}^{i} \leq \Omega_{jw} \quad \forall j, w, t \]  

(16)

3.2.11 Backordering constraint in distribution centres

Equation (17) limits the backorder quantity in period \( t \) by the current demand plus the backorder amount from the previous period. The shortage amount in period \( t \) is zero if the amount of demand for item \( i \) in distribution centre \( w \) in period \( t \) plus the backorder quantity from previous period is equal to or smaller than the total quantity of item \( i \) transferred to distribution centre \( w \) in period \( t \).

\[ b_{jw} = \max \left\{ (b_{jw(t-1)} + d_{jw}^t) - Y_{jw}^i, 0 \right\} \quad \forall i, j, w, t \]  

(17)

3.2.12 Storage capacity constraints

Equations (18) and (19) determine the upper limit of inventory level for each type of raw material and product in plants respectively.

\[ I_{k}^{ij} \leq \sigma_{k} \quad \forall k, j, t \]  

(18)

\[ I_{i}^{ij} \leq \sigma_{i} \quad \forall i, j, t \]  

(19)

3.2.13 Non-negativity and binary constraints

The restrictions of non-negativity and binary nature on the decision variables are shown below.

\[ Q_{ij}, \alpha_{kij}, I_{k}^{ij}, I_{i}^{ij}, b_{jw}^t, Z_{ij}^{kw}, C_{ijw}, Y_{ijw} \in \mathbb{R}^{+} \quad \forall k, i, j, j \neq j, w, t \]

\[ \phi_{ij}, \psi_{ij}, \Omega_{jw} \in \mathbb{Z}^{+} \quad \forall m, j, i \neq j, w, t \]  

(20)

3.3 Objective function

3.3.1 Procurement cost

Equation (21) shows the total procurement cost. It consists of the ordering cost that depends on whether procurement has taken place or not, and the purchasing cost of raw materials over the planning horizon.

\[ \sum_{k} \sum_{m} \sum_{j} \sum_{t} \phi_{kij} R_{km}^{i} + \alpha_{kij} C_{km} \]  

(21)

3.3.2 Production cost

Equation (22) expresses the total production cost. The first term represents the setup cost and the second term shows the production cost. It must be noted that the setup cost depends on whether production takes place or not; therefore, binary variable \( \chi_{ij} \) is used in expression of setup cost.
\[
\sum_{i} \sum_{j} \sum_{t} x_{ijt} s_{ijt} + Q_{ijt} o_{ijt} \tag{22}
\]

### 3.3.3 Inventory cost

Equation (23) shows the inventory costs of raw materials and finished items in plants.

\[
\sum_{k} \sum_{j} \sum_{t} I_{kjt} H_{kjt} + \sum_{i} \sum_{j} \sum_{t} I'_{ijt} H'_{ijt} \tag{23}
\]

### 3.3.4 Transportation cost

Equation (24) shows the transportation cost from the suppliers and plants, between the plants, and from plants to distribution centres. This cost depends on the fixed and variables costs of vehicles. The movement of a vehicle incurs a fixed cost associated to vehicle’s depreciation and insurance, cost of capital, and driver wages, and the variable transportation cost relates to the transported item, its quantity, and the path taken for each route travelled. Without loss of generality, it is assumed that transfer cost from plant \( j \) to plant \( l \) is smaller than transfer cost from plant \( j \) to plant \( l' \) plus transfer cost from plant \( l' \) to plant \( l \).

\[
\eta' \sum_{m} \sum_{j} \sum_{t} \sum_{i} \phi_{mij} + \eta' \sum_{j} \sum_{i} \sum_{t} \psi_{jit} + \eta' \sum_{j} \sum_{w} \sum_{t} \Omega_{jtw} + \eta' \sum_{m} \sum_{j} \sum_{i} \sum_{t} \sum_{w} \sum_{t} \Omega_{jtw} \tag{24}
\]

### 3.3.5 Shortage cost

Equation (25) shows the shortage cost in distribution centre \( w \). Demand in a distribution centre during any period can be satisfied by direct transfer of produces from the nearest plant. If that plant cannot fully satisfy the orders, it can be met by requesting the unfulfilled amount of demand from other production plants. In the case when the demand cannot be fulfilled by any other plants, then a shortage will occur, and the backordered amount at the distribution centre must be satisfied in the next time period.

\[
\sum_{i} \sum_{w} \sum_{t} b_{iwt} b_{iwt} \tag{25}
\]

### 3.3.6 Sales income

Equation (26) expresses the total sales income over the planning horizon, which is obtained from the sales of products shipped to the distribution centres.

\[
\sum_{i} \sum_{j} \sum_{w} \sum_{t} \gamma_{iwt} x_{iwt} \tag{26}
\]

Finally, equation (27) presents the objective function of the proposed model, where the sum of procurement, production, inventory, transportation, and shortage costs over the planning horizon should be minimised from which the total sales is deducted.
Optimisation of multi-plant capacitated lot-sizing problems

\[
\begin{align*}
\min f &= \left( \sum_{k} \sum_{w} \sum_{j} \sum_{t} \phi_{kjt} \tau_{kwt} + \alpha_{kjt} \tau_{kwt} + \sum_{j} \sum_{t} \sum_{f} \chi_{jit} S_{jlt} + Q_{jlt} O_{jlt} \\
&+ \sum_{k} \sum_{j} \sum_{t} I_{kjt} H_{kjt} + \sum_{i} \sum_{j} \sum_{t} I_{jlt} H_{jlt} + \eta^F \sum_{m} \sum_{j} \sum_{t} \phi_{ijt} \\
&+ \eta^F \sum_{l} \sum_{j} \sum_{t} \psi_{jlt} + \eta^F \sum_{j} \sum_{s} \sum_{t} \Omega_{jst} + \eta^F \sum_{m} \sum_{j} \sum_{t} \sigma_{mj} \phi_{ijt} \right) \\
&+ \eta^F \sum_{j} \sum_{l} \sum_{t} \mu_{jlt} \psi_{jlt} + \eta^F \sum_{j} \sum_{w} \sum_{t} \zeta_{jw} \Omega_{jwt} + \sum_{i} \sum_{w} \sum_{t} b_{iwt} B_{iwt} \right) \\
&- \sum_{j} \sum_{w} \sum_{t} Y_{jtw} U_{jtw} 
\end{align*}
\]

(27)

4 Solution algorithms

It is known that the decision making associated with the lot-sizing and scheduling problem belongs to the category of combinatorial optimisation problems. The difficulty to find a general approach for the lot-sizing and scheduling problem is considered in complexity theory as a NP-hard problem (França et al., 1997). Metaheuristic algorithms have become a great choice for solving NP-hard problems because of their multi-solution and strong neighbourhood search capabilities in a reasonable computational time. Therefore, metaheuristic approaches namely the GA, PSO, and ICA are used to deal with the presented model’s intricacy and to obtain optimal or near-optimal solution. These applied methods are described below.

4.1 Genetic algorithm

The fundamental of the GA was primarily established by Holland (1975). The GA is based on the concept of natural evolution, derived from the Darwinian principles of survival of the fittest and natural selection. The required steps to solve the proposed model by the GA are explained in the following subsections.

4.1.1 Parameters

The initial information required to begin a GA includes the number of chromosomes kept in each generation called population size, ‘\(N_{pop}\)’, the probability of operating crossover, ‘\(P_c\)’, the probability of operating mutation, ‘\(P_m\)’, and maximum number of generations, ‘\(\text{max generation}\)’.

4.1.2 Chromosome representation

A GA starts with encoding the variables of the problem as finite-length strings or chromosomes. In this study, the chromosomes are considered as strings of the quantities of the produced items (lot size \(Q\)) with \(N \times J \times T\) dimensions where \(N\) shows total number of products, \(J\) indicates total number of plants, and \(T\) denotes total number of periods. The representation of a chromosome is illustrated in Figure 2.
4.1.3 Initial population

The GA generates a randomly initial population of \( g \) chromosomes within the boundary of the component. Let \( Q_g \) represent the \( g \)th chromosome in the population. Then, each chromosome is generated by:

\[
Q_g = \text{round}[\text{lb}(i, j, t), \text{ub}(i, j, t)]
\]  

(28)

where \( g \) denotes the size of population (\( g = 1, 2, \ldots, N_{\text{pop}} \)), and \( \text{lb}Q \) and \( \text{ub}Q \) are the lower and upper bounds for variable \( Q \) respectively. Therefore, equation (28) produces integer random numbers for variable \( Q \) within the predetermined limits.

The population size depends only on the nature of problems and must be balanced between time complexity and search space measure. Larger population size may increase the probability of finding optimal solution, but would correspondingly increase the computational time, in addition to an increase in the number of function evaluations (NFEs). The NFEs determines the speed (computational effort) and the robustness of the algorithm. Smaller NFEs would result in a shorter time to reach the global optimum (Almasi et al., 2014).

4.1.4 Selection

In each generation, a collection of offspring chromosomes is generated through a recombination process of parents using the roulette wheel procedure. The selection process is based on spinning the roulette wheel \( N_{\text{pop}} \) times. The following process is used to choose two parents:

1. The fitness value of the population is obtained.
2. A particular population member to be a parent with a probability is selected. The selection probability, \( a_g \), for individual \( g \) with objective function value \( f_g \), is calculated by equation (29):

\[
a_g = \frac{f_g}{\sum_{g=1}^{N_{\text{pop}}} f_g}
\]

(29)

3. Cumulative normalised fitness value for each chromosome is then calculated.
4. A real random number \( r \) in range [0, 1] is generated.
5. Two chromosomes are selected whose cumulative probabilities are greater than \( r \).

Although all individuals in the population have a chance of being selected to reproduce the next generation, those with higher fitness value are more likely to be selected for the mating pool.
4.1.5 Crossover

In a crossover process, it is essential to mate pairs of chromosomes to produce offspring. This is carried out by a random selection of a pair of chromosomes from the generation with probability \( P_c \). The number of chromosomes for carrying out the crossover operator is obtained by equation (30):

\[
N_{\text{crossover}} = N_{\text{pop}} \times P_c
\]  

(30)

In this study, the arithmetic crossover operator that linearly combines the parent chromosome vector is used to produce offspring based on equations (31) and (32).

\[
\text{offspring}_{(1)} = y \times \text{parent}_{(1)} + (1 - y) \times \text{parent}_{(2)}
\]  

(31)

\[
\text{offspring}_{(2)} = y \times \text{parent}_{(2)} + (1 - y) \times \text{parent}_{(1)}
\]  

(32)

where \( y \) is a random vector in range \([0, 1]\), and has a dimension equal to the size of the selected part (say the first part) of the chosen parent. Because variable \( Q \) is integer, the amounts of produced offsprings are rounded. The crossover process will be repeated \( N_{\text{crossover}}/2 \) times.

4.1.6 Mutation

The mutation operator injects diversity in the population of solutions with probability \( P_m \). It is considered as a background operator that keeps genetic diversity within the population. Mutation process prevents the algorithm to get stuck at a local minimum as well as reaching an untimely convergence. The solution spaces that are not discovered by the crossover operator are found using the mutation operator. The expected number of chromosomes that go through the mutation operation is obtained using equation (33):

\[
N_{\text{mutation}} = N_{\text{pop}} \times P_m
\]  

(33)

The steps involved in the mutation operation are as follow:

1. An integer random number in range \([1, N_{\text{pop}}]\) is generated in order to select a chromosome \( Q_g \). The total number of elements selected for mutation are \( nQ = N \times J \times T \).

2. Two integer random numbers \( r_1 \) and \( r_2 \) are produced in order to select the elements of the chromosome for mutation. The considered range for \( r_1 \) is \([1, nQ-1]\) and \( r_2 \) is \([r_1 + 1, nQ]\).

3. The value of selected elements of chromosome \( Q_g \) is changed using equation (34):

\[
Q'_g (\eta : r_2) = Q_g (\eta : r_2) + \text{distance}Q_g (\eta : r_2)
\]  

(34)

The result obtained by equation (34) is rounded to obtain an integer value. The distance is the amount that an element can be changed, which is obtained by equation (35):

\[
\text{distance}Q_g = 0.1 \times r \times (\text{ub}Q_g - \text{lb}Q_g)
\]  

(35)

where \( r \) is a random number generated from the continuous uniform distribution within the range of \([-1, 1]\). The size of generated random numbers is equal to the upper bound
Equation (35) causes the values of chosen elements exchange with uniform values randomly selected between the upper and lower ranges. However, there is a possibility that the calculated value by equation (34) exceeds the upper and lower bounds of $Q_g$. Therefore, it can be controlled by:

$$
\begin{align*}
    \text{if } Q'_g > u_b Q_g & \text{ then } Q'_g = u_b Q_g \\
    \text{if } Q'_g < l_b Q_g & \text{ then } Q'_g = l_b Q_g
\end{align*}
$$

In this study, various combinations of the crossover and mutation rates in the range $[0.1, 1]$ are examined. The computational results showed that as $P_c$ and $P_m$ increase, the total cost decreases.

### 4.1.7 Evaluation and constraint handling

As the chromosomes are produced, a fitness value is assigned to chromosomes of each generation for their evaluation. This evaluation is achieved by the objective function given in equation (27), which measures the fitness of each individual in the population. As shown in Subsection 3.2, the proposed model contains various constraints, which may lead to the production of infeasible chromosomes. In order to deal with infeasibility, a penalty value is assigned to chromosomes that are not feasible. It can be obtained by adding a specific amount to the objective function value according to the amount of constraints' violations gained in a solution.

When a chromosome is feasible, its penalty is set to zero, whereas in case of infeasibility, the coefficient is selected sufficiently large. Therefore, the fitness function for a chromosome will be equal to the sum of the objective function value and penalties as shown in equation (37), where $s$ represents a solution, and $f(s)$ is the objective function value for solution $s$. The penalty policy is employed for all metaheuristic algorithms utilised in this paper.

$$
\text{fitness}(s) = \begin{cases} 
  f(s); & \text{if } s \text{ is feasible} \\
  f(s) + \text{Penalty}(s); & \text{otherwise}
\end{cases}
$$

### 4.1.8 New population

The fitness function values of all members, including parents and offspring are assessed in this stage. The chromosomes with higher fitness scores are then chosen to create a new population. To attain a better solution, the fittest chromosomes would be maintained at the end of this stage. The number of chosen chromosomes must be equal to $N_{pop}$.

### 4.1.9 Termination

The selection and reproduction of parents will be continued until the algorithm reaches a stopping criterion. The procedure can be ended after a predetermined number of generations, or when no substantial improvement over a successive generation is achieved.
4.2 Particle swarm optimisation

The PSO, initially introduced by Kennedy and Eberhart (1995), is an algorithm for finding optimal regions of complex search spaces through the interaction of individuals in a population of particles. The procedure was inspired by the social behaviour of fish schooling or bird flocking choreography. In any iteration of PSO, the velocity and position of particles are updated according to equations (38) and (39):

\[
V_{g}^{(e+1)} = \omega V_{g}^{e} + c_1 y_1 (best_{g} - \psi_{g}^{e}) + c_2 y_2 (global_{g} - \psi_{g}^{e}) \tag{38}
\]

\[
\psi_{g}^{(e+1)} = \psi_{g}^{e} + V_{g}^{(e+1)} \tag{39}
\]

where \(g = 1, 2, \ldots, N_{\text{swarm}}\) and \(e\) denotes the iteration \((e = 1, 2, \ldots, \text{max iteration})\). \(V_{g}\) is the velocity of \(g^{th}\) particle. \(\omega\) is the inertia weight controlling the impact of the previous velocity of the particle on its current velocity, which plays an important role in balancing global and local search ability of the PSO. Applying a large inertia weight at the start of the algorithm and making it decay to a small value through the PSO execution makes the algorithm search globally at the beginning of the search, and search locally at the end of the execution (Coelho and Sierakowski, 2008). The inertia weight is updated in each iteration by \(\omega^{(e+1)} = \omega^{e} \times (1 - \beta)\) where in this study \(\beta\) is considered to be 0.01. \(c_1\) is the cognitive parameter, and \(c_2\) is the social parameter; \(y_1\) and \(y_2\) are random numbers within the range \([0, 1]\), which should be the same size as \(V_{g}\); \(best_{g}\) is the best searching experience by particle \(g\); \(\psi_{g}\) is the current position of particle \(g\); and \(global_{g}\) is the position in parameter space of the best fitness returned for the entire swarm. The search procedure of the PSO is summarised as follows:

1. Generate an initial population of \(g\) particles (solutions) with random positions and velocities within the boundary of the component according to equation (28), where \(g\) denotes the size of swarm.

2. Evaluate the fitness value of each particle in the swarm.

3. Compare each particle’s fitness with the current particle’s own best. If current value is better than own best, own best value will be set to the current value, and the own best location to the current location.

4. Compare the fitness value with the population’s overall previous best. If the current value is better than global best, then global best will be set to the current particle’s array index and value.

5. Update the velocity of each particle \(g\) using equation (38), and the position of particle \(g\) using equation (39). The values obtained by equation (38) for the velocity is rounded to the nearest integer number. If the value obtained by equation (39) exceeds upper and lower bounds of the particle, then equation (36) is used to set the value to its boundaries.

6. Terminate the procedure if the termination criterion is satisfied, otherwise go to step 2.
4.3 Imperialist competitive algorithm

The ICA approach is a population based evolutionary algorithm proposed by Atashpaz-Gargari and Lucas (2007), which has been used extensively to solve various kinds of combinatorial optimisation problems. This method is based on socio-political process of imperialistic competition. The main steps of the ICA are described in the following subsections.

4.3.1 The initialisation mechanism

Similar to other optimisation methods, the ICA first creates the initial population. Each individual of the population is named a ‘country’. The word ‘country’ corresponds to the ‘chromosome’ in the GA terminology. This array is shown in equation (40).

\[
\text{country} = [L_1, L_2, \ldots, L_N]
\]  

(40)

where \(L_s\) are the variables to be optimised. The power of a country is inversely proportional to its fitness function value which is obtained by evaluation of cost function \(f\) at variables as shown in equation (41).

\[
\text{cost} = f(\text{country}) = f([L_1, L_2, \ldots, L_N])
\]  

(41)

Equation (28) is used to create the initial population of size \(N_{\text{country}}\) and equation (27) is used to evaluate the fitness of each individual. Based on the cost values, a certain number of countries that have the lowest costs are selected as imperialist (\(N_{\text{imp}}\)), and the rest known as colonies (\(N_{\text{colony}} = N_{\text{country}} - N_{\text{imp}}\)), are divided among these imperialists. Each imperialist and its allocated colonies form an empire.

4.3.2 Assimilation of colonies

The imperialist countries try to absorb their colonies toward themselves. For this purpose, the assimilation policy is considered in the ICA. Based on this concept, each colony moves toward its imperialist by \(X\) units as shown in equation (42).

\[
Q_{g+1}^{\text{col}} = Q_g^{\text{col}} + X
\]

(42)

where \(Q_g^{\text{col}}\) and \(Q_{g+1}^{\text{col}}\) are the current and new position of \(g^{\text{th}}\) colony of each empire respectively, \(e\) is the number of iteration (knows as decade in the ICA), and \(X\) is:

\[
X = \beta \times y \times D
\]

(43)

where \(D\) is the distance between the initial position of colony and its imperialist (\(D = \text{imp. } Q_g^{\text{col}} - \text{colony. } Q_g^{\text{col}}\)). The position of the colony after movement is defined by the random parameter \(y\) in range [0, 1], which must have the dimension equal to the size of \(D\). Parameter \(\beta > 1\) causes the colony to get closer to its imperialist from different direction. \(\beta = 2\) results in good convergence of countries to the global minimum in most of the implementations. If the value of \(Q_{g+1}^{\text{col}}\) exceeds its predetermined limit, it must be set to its upper and lower bounds as shown in equation (36). Then, fitness of \(Q_{g+1}^{\text{col}}\) will be evaluated.
4.3.3 Revolution

Revolution is a fundamental change in power or organisational structures that takes place in a relatively short period of time. The revolution increases the exploration of the algorithm and avoids the early convergence of countries to local minimum with a revolution rate. A very high value of revolution rate reduces the exploitation power of algorithm as well as the convergence rate (Nazari-Shirkouhi et al., 2010). This mechanism is similar to mutation process in the GA for creating diversification in solutions, increasing the variety in the population.

For creating a revolution, random number $r$ in range $[0, 1]$ is generated. If $r$ is smaller than revolution rate ($\theta$), a change will be applied on the selected colony of the particular imperialist. Steps of creating a change are as follow:

1. Find the total number of elements of the colony that can be chosen for change and store it in $nQ$ where $nQ = N \times J \times T$.
2. Generate an integer random number $n$ in range $[0, \text{round}(0.01 \times nQ + 1)]$.
3. Generate two integer random numbers $r_1$ and $r_2$. The considered range for $r_1$ is $[1, nQ-n]$ and $r_2$ is $r_1 + n$.
4. Create the change using equation (44):

$$Q_{eg}^{r1} (n : r_2) = \text{lb} Q_{eg} (n : r_2) + y \times \left[ \text{ub} Q_{eg} (n : r_2) - \text{lb} Q_{eg} (n : r_2) \right]$$

where $y$ is a matrix with size $1 \times n + 1$ containing real random numbers in the range $[0, 1]$, and $\text{lb} Q_{eg}$ and $\text{ub} Q_{eg}$ are the lower and upper bounds of $g^{th}$ colony in decade $e$. Values obtained by equation (44) are rounded to the nearest integers. Then, fitness of new colony is evaluated.

4.3.4 Exchange the colony with imperialist

After moving toward the imperialist, a colony may reach a position with lower cost than it’s imperialist. In this case, the colony will become the imperialist in the current empire and vice versa. In the next decades, colonies in the empire will move towards the new imperialist.

4.3.5 Total power of an empire

Based on total power of empires, an imperialistic competition takes place between empires. Total power of an empire is mainly affected by the power of its imperialist and slightly by its colonies. Hence, total power of an empire is calculated by equation (45):

$$T_{eg}^{f} = f \left( \text{imp}_{g} \right) + \gamma \times \text{mean} \left[ f \left( \text{colony of empire}_{g} \right) \right]$$

where $T_{eg}^{f}$ is the total cost of the $g^{th}$ empire and $\gamma$ is a positive number in the range of $[0, 1]$. The small value of $\gamma$ makes the total power of the empire to be determined by its imperialist, and increasing it will enhance the role of the colonies in determining the total power of an empire.
4.3.6 Imperialistic competition

Every empire tries to take over the colonies of other empires. The imperialistic competition gradually causes reduction in power of weaker empires and growth in power of stronger ones. The imperialistic competition is modelled by selecting the weakest colonies of the weakest empire in every iteration and making a competition among all empires to take control over this colony. The likelihood of possession of the colony for each empire is proportionate to its total power. The normalised total cost of an empire can be obtained by equation (46):

$$ NT_g = \max (T_f g', T_f g) + \varepsilon $$

(46)

where $\varepsilon$ is a small number to avoid the value of $NT_g$ to become exactly zero. $NT_f g$ and $T_f g$ are the normalised total cost and total cost of $g^{th}$ empire respectively. The possession probability of $g^{th}$ empire is obtained by equation (47):

$$ a_g = \frac{NT_f g}{\sum_{g'=1}^{N} NT_f g'} $$

(47)

Powerful empires have higher chance of possessing the colony. To distribute the colony among empires, vector $c$ is formed as give in equation (48):

$$ c = [c_1, c_2, \ldots, c_{N_{emp}}] $$

(48)

Then, vector $h$ with uniformly distributed random numbers in range $[0, 1]$ is created as shown in equation (49), which has the same size as $c$:

$$ h = [h_1, h_2, \ldots, h_{N_{emp}}] $$

(49)

Vector $z$ is formed by subtracting $h$ from $c$ as shown in equation (50):

$$ z = c - h = [z_1, z_2, \ldots, z_{N_{emp}}] $$

(50)

Referring to vector $z$, the colony is handed to an empire, whose its corresponding index in $z$ is maximum.

4.3.7 Elimination of powerless empires

Powerless empires will collapse in the imperialistic competition and their colonies will be divided among other empires. It is assumed that an empire would collapse and be removed when it loses all of its colonies. This process will be continued and causes the countries to converge to the global minimum of the cost function. At the end, all the empires will collapse except the most powerful one, and all the colonies would be under this unique empire. At this stage, the imperialist and colonies would have the same position and power.
4.4 Assignment of demand to plants

The following procedure is employed to assign demand of a product received in a distribution centre to the plants.

1. Supply the demand of product \( i \) in distribution centre \( w \) from the nearest plant \( j \) as long as there is inventory from previous period and enough capacity for production in plant \( j \) in period \( t \). This model allows outsourcing from other plants only when the demand cannot be met thoroughly at the current plant. After the assignment, inventory at plant \( j \) is updated.

2. If demand of product \( i \) in period \( t \) is not fully satisfied by plant \( j \), the remaining demand will be supplied from the second nearest plant subject to the available capacity and inventory at that plant.

3. Steps 1 and 2 are repeated until all distribution centres have satisfied their demands for all the products.

4. If the demand in period \( t \) cannot be fully satisfied by the inventory, production, and inter-plant transfers, it will be backordered, but the backorder demand must be fulfilled in the next period.

5 Algorithms’ parameters calibration using the Taguchi method

Taguchi method is utilised to calibrate the parameters of the applied algorithms, since the quality of the solutions obtained by metaheuristic algorithms depends on the values of their parameters. Taguchi divides the factors affecting the performance (response) of a process into two groups: noise factors that cannot be controlled and controllable factors such as the parameters of a metaheuristic algorithm that can be controlled by designers (Sadeghi et al., 2013). In Taguchi’s parameter design phase, an experimental design is used to arrange the control and noise factors in the inner and outer orthogonal arrays respectively. Then, the signal-to-noise (S/N) ratio is calculated for each experimental combination. The term ‘signal’ indicates the desirable value (response variable) and ‘noise’ denotes the undesirable value (standard deviation) (SD). Taguchi method categorises objective functions into three groups: larger is better, nominal is best, and smaller is better. Since the objective function of the proposed model is the minimisation, ‘smaller is better’ type of response has been utilised, where \( S/N \) is obtained by equation (51) (Phadke, 1989):

\[
S / N = -10 \log_{10} \left( \frac{1}{n} \sum_{i=1}^{n} f_i^2 \right) \tag{51}
\]
where $f_e$ is the objective function value of a given experiment $e$, and $n$ is the number of times the experiment is performed. In the Taguchi method, the parameters that most likely to have considerable effects on the process output are initially chosen for tuning. The parameters that require calibration are $N_{pop}$, $P_c$, and $P_m$ in the GA, $N_{swarm}$, $c_1$, $c_2$, and $\omega$ in the PSO, and $N_{countries}$, $N_{imp}$, $\theta$, and $\gamma$ in the ICA. The range for the parameters that produces satisfactory fitness function values are chosen by trial and error methods.

Table 1: The GA, PSO, and ICA parameters’ levels

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Parameters</th>
<th>Levels</th>
</tr>
</thead>
<tbody>
<tr>
<td>GA</td>
<td>$N_{pop}$</td>
<td>50 100 150</td>
</tr>
<tr>
<td></td>
<td>$P_c$</td>
<td>0.85 0.9 0.95</td>
</tr>
<tr>
<td></td>
<td>$P_m$</td>
<td>0.70 0.80 0.90</td>
</tr>
<tr>
<td>PSO</td>
<td>$N_{swarm}$</td>
<td>50 100 150</td>
</tr>
<tr>
<td></td>
<td>$c_1$</td>
<td>1 1.5 2</td>
</tr>
<tr>
<td></td>
<td>$c_2$</td>
<td>1 1.5 2</td>
</tr>
<tr>
<td></td>
<td>$\omega$</td>
<td>0.80 1 1.2</td>
</tr>
<tr>
<td>ICA</td>
<td>$N_{country}$</td>
<td>50 100 150</td>
</tr>
<tr>
<td></td>
<td>$N_{imp}$</td>
<td>5 10 15</td>
</tr>
<tr>
<td></td>
<td>$\theta$</td>
<td>0.10 0.20 0.30</td>
</tr>
<tr>
<td></td>
<td>$\gamma$</td>
<td>0.15 0.25 0.35</td>
</tr>
</tbody>
</table>

Table 2: Optimal values of the algorithms’ parameters

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Parameters</th>
<th>Optimal values</th>
</tr>
</thead>
<tbody>
<tr>
<td>GA</td>
<td>$N_{pop}$</td>
<td>150</td>
</tr>
<tr>
<td></td>
<td>$P_c$</td>
<td>0.95</td>
</tr>
<tr>
<td></td>
<td>$P_m$</td>
<td>0.80</td>
</tr>
<tr>
<td></td>
<td>Max generation</td>
<td>200</td>
</tr>
<tr>
<td>PSO</td>
<td>$N_{swarm}$</td>
<td>150</td>
</tr>
<tr>
<td></td>
<td>$c_1$</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>$c_2$</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>$\omega$</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Max iteration</td>
<td>200</td>
</tr>
<tr>
<td>ICA</td>
<td>$N_{country}$</td>
<td>150</td>
</tr>
<tr>
<td></td>
<td>$N_{imp}$</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>$\theta$</td>
<td>0.30</td>
</tr>
<tr>
<td></td>
<td>$\gamma$</td>
<td>0.25</td>
</tr>
<tr>
<td></td>
<td>max decade</td>
<td>200</td>
</tr>
</tbody>
</table>
Table 1 shows the algorithms’ parameters, each at three levels with nine observations. Figure 3 shows the mean S/N ratio plot of the applied algorithms (ω shown in Figure 3 stands for $\omega$, $\theta$ stands for $\theta$, and $\gamma$ stands for $\gamma$ as indicated in Table 1). The best parameters’ levels are the highest mean of S/N values. Table 2 shows the optimal levels of the parameters for all algorithms.

Figure 3 The mean S/N ratio plot for each level of the factors of the GA, PSO, and ICA algorithms (see online version for colours)
6 A numerical example

A company owns three plants, which are spread geographically around a country, and two distribution centres located in two different cities. The planning horizon is assumed to be six periods. The number of products is considered to be three. There are three types of raw materials, each supplied by a different supplier. The total available time for each plant is assumed to be 10,560 min per period (22 days × 8 hours × 60 minutes). There are 47, 44 and 30 workers at plant 1, 2 and 3 respectively, with no hiring and firing of workers during the considered planning horizon. Safety stock coefficient for each raw material type is considered 1.5 per period. It is supposed that the chance of rejecting each type of raw material by plant 1, 2 and 3 is 0.02, 0.01, and 0.02 respectively in each period.

Table 3  Demand data

<table>
<thead>
<tr>
<th>Product i</th>
<th>Distribution centre 1</th>
<th>Distribution centre 2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Period t</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 2 3 4 5 6</td>
<td>1 2 3 4 5 6</td>
</tr>
<tr>
<td>1</td>
<td>680 600 550 520 650 630</td>
<td>630 550 530 500 550 675</td>
</tr>
<tr>
<td>2</td>
<td>550 530 600 500 600 550</td>
<td>700 680 570 680 530 530</td>
</tr>
<tr>
<td>3</td>
<td>500 700 600 550 590 630</td>
<td>590 680 530 660 540 500</td>
</tr>
</tbody>
</table>

Table 4  Selling price ($/unit period)

<table>
<thead>
<tr>
<th>Product i</th>
<th>Distribution centre 1</th>
<th>Distribution centre 2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Period t</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 2 3 4 5 6</td>
<td>1 2 3 4 5 6</td>
</tr>
<tr>
<td>1</td>
<td>53 53 53 53.5 53.5 53.5</td>
<td>51 51 51 51.5 51.5 51.5</td>
</tr>
<tr>
<td>2</td>
<td>55 55 55 55.5 55.5 55.5</td>
<td>52.5 52.5 52.5 53 53 53</td>
</tr>
<tr>
<td>3</td>
<td>57 57 57 57.5 57.5 57.5</td>
<td>53 53 53 53.5 53.5 53.5</td>
</tr>
</tbody>
</table>

The time and workers’ performance factors at each plant are assumed to be 85 and 80 percent respectively. The fixed transportation cost of a vehicle is considered to be $1,000 and the variable cost is $2 per trip. The capacity of each vehicle is set to be 100 units for each product type and 250 units for raw materials. Storage capacity for raw material 1, 2 and 3 in each plant is considered to be 4,000, 3,500, and 3,000 units, and for product types 1, 2 and 3 is set to be 4,500, 4,000, and 3,500 units respectively. The suppliers’ capacity is assumed to be 15,000 units for each type of raw material per period. The values for the remaining data are presented in Tables 3–11. It is assumed that setup time and production time (see Table 5), ordering and purchasing costs (see Table 7), setup and production costs (see Table 8), inventory cost (see Table 9), and backordering cost (see Table 10) remain fixed during the planning horizon.
Table 5  Setup time and production time (min)

<table>
<thead>
<tr>
<th>Plant j</th>
<th>Aij</th>
<th>Pij</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>6.06</td>
<td>4.90</td>
</tr>
<tr>
<td>2</td>
<td>7.18</td>
<td>5.20</td>
</tr>
<tr>
<td>3</td>
<td>7.73</td>
<td>5.27</td>
</tr>
</tbody>
</table>

Table 6  Required resource for each product (manpower-min)

<table>
<thead>
<tr>
<th>Resource v</th>
<th>Plant 1</th>
<th>Plant 2</th>
<th>Plant 3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Product i</td>
<td>Product i</td>
<td>Product i</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>1</td>
<td>13.8</td>
<td>12</td>
<td>9.6</td>
</tr>
</tbody>
</table>

Table 7  Ordering cost and purchasing cost of raw material ($/unit period)

<table>
<thead>
<tr>
<th>Supplier m</th>
<th>πkmt</th>
<th>τkmt</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Raw material k</td>
<td>Raw material k</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>0.64</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0.37</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 8  Setup cost and production cost ($/unit period)

<table>
<thead>
<tr>
<th>Plant j</th>
<th>Sijt</th>
<th>Oijt</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Product i</td>
<td>Product i</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>2.55</td>
<td>1.58</td>
</tr>
<tr>
<td>2</td>
<td>2.50</td>
<td>1.50</td>
</tr>
<tr>
<td>3</td>
<td>2.80</td>
<td>1.70</td>
</tr>
</tbody>
</table>

Table 9  Raw material and end product inventory costs in plants ($/unit period)

<table>
<thead>
<tr>
<th>Plant j</th>
<th>Hkjt</th>
<th>H′ijt</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Raw material k</td>
<td>Product i</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>4</td>
</tr>
</tbody>
</table>
Table 10  Raw material consumption rate and backordering cost in distribution centres ($/unit period)

<table>
<thead>
<tr>
<th>Product i</th>
<th>Raw material k</th>
<th>Raw material k</th>
<th>Distribution centre w</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>3</td>
</tr>
</tbody>
</table>

Table 11  Distances between SC entities (km)

<table>
<thead>
<tr>
<th>Plant j</th>
<th>Supplier m</th>
<th>Plant l</th>
<th>Distribution centre w</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>1</td>
<td>150</td>
<td>760</td>
<td>670</td>
</tr>
<tr>
<td>2</td>
<td>800</td>
<td>850</td>
<td>155</td>
</tr>
<tr>
<td>3</td>
<td>415</td>
<td>750</td>
<td>1,000</td>
</tr>
</tbody>
</table>

7  Results and discussions

The applied optimisers were written and coded in MATLAB software version R2012a on a laptop with 2.5-GHz AMD and 4GB RAM. In order to compare the performances of three algorithms and have reliable results, 20 independent runs were carried out using the parameters settings given in Table 2. The objective function values for all runs are given in Table 12. It can be observed that all methods produced similar results. It approves that if the proposed MPCLSP model is solved by any metaheuristic method, similar results will be obtained, which it verifies the correctness of the developed model. The best cost solution is $848,067.30, which is resulted from total cost ($1,989,429.62) minus the total revenue obtained from selling the products ($1,141,362.32). Figure 4 provides an insight into each component of the total cost of the SC, which mainly is caused by transportation cost and procurement cost. The product inventory cost constructs a small portion of the total cost.

Figure 5 shows the performance of three metaheuristic approaches in terms of objective function values. It illustrates that for all runs, the GA performed better for the proposed problem.

Furthermore, using the Minitab software a one-way analysis of variance (ANOVA) is generated as shown in Table 13, to compare the performance of the algorithms based on the results obtained for 20 runs. The p-value of the test-statistics on the equality of the mean of the objective function values is 0.002. This means that the null hypothesis of the test is rejected at 95% confidence level, i.e., there is difference between the mean of objective function values obtained by three algorithms. Figure 6 supports this conclusion as well.
Table 12  Objective function values obtained by applied metaheuristic algorithms

<table>
<thead>
<tr>
<th>Run number</th>
<th>GA</th>
<th>PSO</th>
<th>ICA</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>848,067.30</td>
<td>849,176.44</td>
<td>855,034.49</td>
</tr>
<tr>
<td>2</td>
<td>850,082.43</td>
<td>854,445.35</td>
<td>857,533.07</td>
</tr>
<tr>
<td>3</td>
<td>852,541.96</td>
<td>855,613.04</td>
<td>858,755.59</td>
</tr>
<tr>
<td>4</td>
<td>853,761.39</td>
<td>865,922.48</td>
<td>860,029.49</td>
</tr>
<tr>
<td>5</td>
<td>854,389.84</td>
<td>867,406.98</td>
<td>861,616.09</td>
</tr>
<tr>
<td>6</td>
<td>855,894.30</td>
<td>869,235.74</td>
<td>862,392.66</td>
</tr>
<tr>
<td>7</td>
<td>857,554.22</td>
<td>870,423.49</td>
<td>863,818.04</td>
</tr>
<tr>
<td>8</td>
<td>858,196.94</td>
<td>871,761.29</td>
<td>869,408.87</td>
</tr>
<tr>
<td>9</td>
<td>859,963.66</td>
<td>872,441.48</td>
<td>870,034.35</td>
</tr>
<tr>
<td>10</td>
<td>863,933.85</td>
<td>876,301.24</td>
<td>871,176.70</td>
</tr>
<tr>
<td>11</td>
<td>864,624.39</td>
<td>877,615.81</td>
<td>872,509.70</td>
</tr>
<tr>
<td>12</td>
<td>865,428.18</td>
<td>878,972.55</td>
<td>873,723.78</td>
</tr>
<tr>
<td>13</td>
<td>868,335.94</td>
<td>881,333.33</td>
<td>874,842.02</td>
</tr>
<tr>
<td>14</td>
<td>868,412.71</td>
<td>883,587.86</td>
<td>875,282.19</td>
</tr>
<tr>
<td>15</td>
<td>871,364.99</td>
<td>883,804.79</td>
<td>876,521.90</td>
</tr>
<tr>
<td>16</td>
<td>872,345.41</td>
<td>884,404.79</td>
<td>877,019.07</td>
</tr>
<tr>
<td>17</td>
<td>873,241.66</td>
<td>885,163.58</td>
<td>878,407.67</td>
</tr>
<tr>
<td>18</td>
<td>874,193.48</td>
<td>887,855.43</td>
<td>880,179.76</td>
</tr>
<tr>
<td>19</td>
<td>874,381.70</td>
<td>888,041.52</td>
<td>883,934.97</td>
</tr>
<tr>
<td>20</td>
<td>874,932.03</td>
<td>896,651.68</td>
<td>886,178.31</td>
</tr>
</tbody>
</table>

Table 13  One-way ANOVA results for objective function values

<table>
<thead>
<tr>
<th>Source</th>
<th>Degree of freedom (DF)</th>
<th>SS</th>
<th>MS</th>
<th>F-test</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optimisation engines</td>
<td>2</td>
<td>1,447,406,372</td>
<td>723,703,186</td>
<td>6.93</td>
<td>0.002</td>
</tr>
<tr>
<td>Error</td>
<td>57</td>
<td>5,949,027,657</td>
<td>104,368,906</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>59</td>
<td>7,396,434,029</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The statistical optimisation results obtained from three algorithms including the worst mean, best solutions, SD, and NFEs are listed in Table 14. Based on Table 14, the GA is found to be superior to other methods and surpassed the PSO and ICA in terms of function value (accuracy). The GA method has found the best solution (848,067.30) in 37,616 function evaluations (143 generations). Although the PSO offered modest solution quality with a smaller NFEs (20,550), its average cost and SD in 20 experiments are inferior to the other two methods. The ICA surpassed the PSO for offering the best solution in terms of SD and mean solution.
Figure 4  Percentage of each component cost in the total cost obtained from the best run of the GA (see online version for colours)

Figure 5  Graphical comparison of applied methods in terms of objective function value (see online version for colours)

Figure 6  Box-plot of objective function values (see online version for colours)
Figure 7  The convergence path of fitness function for the best run of the GA, PSO, and ICA algorithms (see online version for colours)
Table 14  Statistical results obtained from applied metaheuristic algorithms

<table>
<thead>
<tr>
<th>Methods</th>
<th>Worst solution</th>
<th>Mean solution</th>
<th>Best solution</th>
<th>SD</th>
<th>NFEs</th>
</tr>
</thead>
<tbody>
<tr>
<td>GA</td>
<td>874,932.03</td>
<td>863,082.32</td>
<td>848,067.30</td>
<td>8,878.11</td>
<td>37,616</td>
</tr>
<tr>
<td>PSO</td>
<td>896,651.68</td>
<td>875,007.94</td>
<td>849,176.44</td>
<td>12,328.95</td>
<td>20,550</td>
</tr>
<tr>
<td>ICA</td>
<td>886,178.31</td>
<td>870,419.94</td>
<td>855,034.49</td>
<td>9,071.00</td>
<td>36,643</td>
</tr>
</tbody>
</table>

Table 15  Purchase amount of raw materials

<table>
<thead>
<tr>
<th>Raw material k</th>
<th>Supplier m</th>
<th>Plant j</th>
<th>Period t</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1*</td>
<td>1</td>
<td>α</td>
<td>α</td>
<td>α</td>
<td>α</td>
<td>α</td>
<td>α</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: *there is only one supplier for each type of raw material, i.e., raw materials 1, 2, and 3 are supplied by suppliers 1, 2, and 3 respectively.

Table 16  Inventory level of raw materials

<table>
<thead>
<tr>
<th>Raw material k</th>
<th>Plant j</th>
<th>Period t</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1,612</td>
<td>1,611</td>
<td>1,626</td>
<td>1,279</td>
<td>1,421</td>
<td>1,149</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1,083</td>
<td>1,127</td>
<td>1,099</td>
<td>881</td>
<td>924</td>
<td>781</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1,513</td>
<td>1,719</td>
<td>1,548</td>
<td>1,404</td>
<td>1,242</td>
<td>1,111</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>2,177</td>
<td>1,671</td>
<td>1,634</td>
<td>1,780</td>
<td>1,608</td>
<td>1,458</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1,222</td>
<td>1,152</td>
<td>1,135</td>
<td>1,227</td>
<td>1,087</td>
<td>991</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1,745</td>
<td>1,777</td>
<td>1,731</td>
<td>1,798</td>
<td>1,573</td>
<td>1,449</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

To have fair comparisons, the same number of iteration was used for all applied optimisers, which was set to 200. The convergence paths of applied methods in terms of best and mean costs for the best run are plotted in Figure 7. As it can be seen, there was no significant improvement in the fitness function values attained for higher number of successive iterations. However, as the problem is cost minimisation, it is worth to
continue the iterations until the search reaches the lowest possible cost. Using the parallel computers can reduce the computational time dramatically.

Table 17  Production quantity and inventory level of products

<table>
<thead>
<tr>
<th>Product i</th>
<th>Plant j</th>
<th>Period t</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Q</td>
<td>I'</td>
<td>Q</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>651</td>
<td>0</td>
<td>460</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>545</td>
<td>0</td>
<td>598</td>
<td>0</td>
<td>601</td>
</tr>
<tr>
<td>3</td>
<td>493</td>
<td>0</td>
<td>645</td>
<td>0</td>
<td>489</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>583</td>
<td>0</td>
<td>591</td>
<td>41</td>
</tr>
<tr>
<td>2</td>
<td>670</td>
<td>0</td>
<td>540</td>
<td>0</td>
<td>579</td>
</tr>
<tr>
<td>3</td>
<td>596</td>
<td>6</td>
<td>678</td>
<td>4</td>
<td>646</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>123</td>
<td>47</td>
<td>112</td>
<td>19</td>
</tr>
<tr>
<td>2</td>
<td>99</td>
<td>64</td>
<td>97</td>
<td>89</td>
<td>95</td>
</tr>
<tr>
<td>3</td>
<td>110</td>
<td>103</td>
<td>71</td>
<td>119</td>
<td>129</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Product i</th>
<th>Plant j</th>
<th>Period t</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Q</td>
<td>I'</td>
<td>Q</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>487</td>
<td>0</td>
<td>619</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>371</td>
<td>0</td>
<td>461</td>
<td>0</td>
<td>440</td>
</tr>
<tr>
<td>3</td>
<td>564</td>
<td>14</td>
<td>347</td>
<td>0</td>
<td>360</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>541</td>
<td>63</td>
<td>569</td>
<td>82</td>
</tr>
<tr>
<td>2</td>
<td>671</td>
<td>0</td>
<td>560</td>
<td>30</td>
<td>509</td>
</tr>
<tr>
<td>3</td>
<td>623</td>
<td>83</td>
<td>527</td>
<td>70</td>
<td>498</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>112</td>
<td>222</td>
<td>87</td>
<td>278</td>
</tr>
<tr>
<td>2</td>
<td>111</td>
<td>117</td>
<td>61</td>
<td>39</td>
<td>65</td>
</tr>
<tr>
<td>3</td>
<td>125</td>
<td>262</td>
<td>105</td>
<td>138</td>
<td>64</td>
</tr>
</tbody>
</table>

Table 18  Shortage amount before inter-plant transactions

<table>
<thead>
<tr>
<th>Product i</th>
<th>Plant j</th>
<th>Period t</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Q</td>
<td>I'</td>
<td>Q</td>
<td>I'</td>
<td>Q</td>
<td>I'</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>29</td>
<td>140</td>
<td>0</td>
<td>33</td>
<td>31</td>
<td>278</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>0</td>
<td>49</td>
<td>129</td>
<td>139</td>
<td>110</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>7</td>
<td>55</td>
<td>111</td>
<td>0</td>
<td>229</td>
<td>270</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>47</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>89</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>30</td>
<td>140</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>
Table 19    Quantity of transported products between plants

<table>
<thead>
<tr>
<th>Product i</th>
<th>Plant j</th>
<th>Plant l</th>
<th>Period t</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>Z</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>1</td>
<td>29</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>0</td>
<td>49</td>
</tr>
<tr>
<td>3</td>
<td>7</td>
<td>55</td>
<td>111</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>2</td>
<td>47</td>
</tr>
<tr>
<td>2</td>
<td>30</td>
<td>72</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Figure 8    Function values versus number of iterations for all applied algorithms (see online version for colours)

Figure 8 compares the function values for the best run of all methods versus the number of iterations. It shows that the convergence rate of the GA and PSO is almost similar, both converged to near optimum point in the early iterations.

Based on the results, the GA is subsequently used to find the optimal solution of decision variables. Table 15 shows the exact amount of each type of raw material that a plant should order from the corresponding suppliers during each period. The inventory level of raw materials stored in plants at the end of each period is presented in Table 16.
### Table 20
Quantity of available products to be shipped from plants to distribution centres and total number of products transferred from plants to distribution centres

<table>
<thead>
<tr>
<th>Period t</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product i, Plant j, Distribution center w</td>
<td>C Y</td>
<td>C Y</td>
<td>C Y</td>
<td>C Y</td>
<td>C Y</td>
<td>C Y</td>
</tr>
<tr>
<td>1, 2, 1</td>
<td>631 680</td>
<td>460 660</td>
<td>650 550</td>
<td>497 520</td>
<td>619 650</td>
<td>352 630</td>
</tr>
<tr>
<td>1, 2, 5</td>
<td>545 550</td>
<td>530 530</td>
<td>601 650</td>
<td>277 500</td>
<td>461 600</td>
<td>440 530</td>
</tr>
<tr>
<td>1, 2, 3</td>
<td>493 500</td>
<td>645 700</td>
<td>489 600</td>
<td>550 550</td>
<td>361 590</td>
<td>360 630</td>
</tr>
<tr>
<td>1, 2, 5</td>
<td>670 700</td>
<td>543 680</td>
<td>640 530</td>
<td>550 550</td>
<td>530 530</td>
<td>530 530</td>
</tr>
<tr>
<td>1, 2, 3</td>
<td>590 590</td>
<td>680 590</td>
<td>680 590</td>
<td>550 550</td>
<td>530 530</td>
<td>530 530</td>
</tr>
<tr>
<td>1, 2, 5</td>
<td>700 680</td>
<td>590 530</td>
<td>680 530</td>
<td>530 530</td>
<td>530 530</td>
<td>530 530</td>
</tr>
</tbody>
</table>
The production quantity and inventory level of each type of product in each plant during the considered planning horizon are indicated in Table 17. It can be seen that the inventory levels of all products in plant 1 during the entire considered planning horizon (except for product 3 in period 4) and for plant 2 during some periods are zero, and for plant 3 is non-zero in every period except the last. It means that plants 1 and 2 may encounter the problem of shortage of products if the demand of products is higher than the production quantity in the current period. The shortage amount of each product during all periods is presented in Table 18.

Table 21  Number of vehicles required to ship products from suppliers to plants

<table>
<thead>
<tr>
<th>Supplier m</th>
<th>Plant j</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>21</td>
<td>14</td>
<td>14</td>
<td>10</td>
<td>13</td>
<td>9</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>14</td>
<td>10</td>
<td>9</td>
<td>7</td>
<td>8</td>
<td>6</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>20</td>
<td>16</td>
<td>13</td>
<td>12</td>
<td>10</td>
<td>9</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>23</td>
<td>14</td>
<td>14</td>
<td>16</td>
<td>13</td>
<td>12</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>16</td>
<td>10</td>
<td>10</td>
<td>11</td>
<td>9</td>
<td>8</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>23</td>
<td>16</td>
<td>15</td>
<td>16</td>
<td>13</td>
<td>12</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

Table 22  Number of vehicles required for inter-plant transportation

<table>
<thead>
<tr>
<th>Plant j</th>
<th>Plant l</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

The transportation amount of each product from a plant to another is shown in Table 19. Since distribution centre 1 is close to plant 1 and distribution centre 2 is close to plant 2, demand in distribution centres 1 and 2 are served by plants 1 and 2 respectively. It was assumed that if the demand of an item cannot be met completely by the given plant, the rest of demand can be supplied by another nearby plant. The results show that when plant 1 experienced under-capacity and insufficient inventory problems, it requested plant 3 to transfer the rest of unmet demand. In case there was not enough end products available in plant 3, then plant 1 requested plant 2 to satisfy the rest of required products. It should be mentioned that the number of products that plant 1 requested from plant 2, is the shortage quantity minus those provided by plant 3. Similarly, plant 2 initially requested plant 1 to supply the unmet demands and in the case of inadequate availability of products in plant 1, the request was sent to plant 3 to provide
the rest of unfulfilled demand. It is indicated that plant 3 is mostly capable of satisfying the shortage amounts. In case, none of the plants are able to meet the demand of products in distribution centres, the unmet demand is backordered and can be satisfied in the next periods. However, it incurs a penalty cost for the distribution centre due to lost units of demand.

Based on Table 18, the total shortage cost when distribution centres cannot meet the demand of products will be $53,507, where the transportation cost caused by outsourcing will be $51,430. Since in this case, the shortage cost is higher than the transportation and inventory costs, bearing some inventories and outsourcing from other plants is preferable than having backorders. The reason lies in the fact that the cost of having backorder is far higher than keeping inventory. In addition, meeting demand in a timely fashion increases customer satisfaction.

Table 20 indicates the number of products available in plants before the inter-plant transfers (produced products plus the pervious inventory) during each period, and total number of products that are finally delivered to distribution centres. The results showed that after all inter-plant transfers, demands of produces in both distribution centres were entirely fulfilled during all periods.

The number of vehicles required for shipment of raw materials from suppliers to plant, inter-plant transfers and delivering the products from plants to distribution centres are shown in Tables 21–23.

The results also showed that the binary variables $\chi_{ijt}$ and $\phi_{kmjt}$ have value of one during all periods, indicating that all three products were produced in all three plants, and all plants have ordered all types of raw materials throughout the six-period planning horizon.

Table 23

<table>
<thead>
<tr>
<th>Plant j</th>
<th>Distribution centre w</th>
<th>Period t</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Ω</td>
<td>Ω</td>
<td>Ω</td>
<td>Ω</td>
<td>Ω</td>
<td>Ω</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td>18</td>
<td>19</td>
<td>18</td>
<td>16</td>
<td>19</td>
<td>19</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td>20</td>
<td>20</td>
<td>17</td>
<td>19</td>
<td>17</td>
<td>18</td>
</tr>
</tbody>
</table>

8 Conclusions

A multi-plant SC is one in which a core exists simultaneously within several manufacturing plants. Such SC network coordinates the planning and scheduling tasks and shares the flow of information among plants, upstream suppliers and downstream distributors, in order to enhance the whole chain performance. In this paper, a mathematical model was presented for the optimisation of the multi-item multi-period capacitated lot-sizing problem in a multi-supplier multi-plant multi-distribution centre logistic environment. A number of operational constraints such as resource utilisation, demand fulfilment, production capacity, inventory storage capacity, supplier capacity, and vehicle utilisation capacity were considered. The objective was to minimise the total cost of the SC including procurement, production, inventory, backordering, and
transportation, such that demands of products are satisfied and the capacity restrictions are not violated.

Besides, the efficiencies of a number of metaheuristic approaches for solving the MPCLSP in an integrated SC were evaluated. Since the proposed model is NP-Hard, three metaheuristic algorithms namely the GA, PSO, and ICA were used to find the optimal solutions within a moderate computational time. The GA, PSO, and ICA methods have shown great potentials for solving optimisation problems as they conduct global stochastic search. In addition, Taguchi method was used to calibrate the parameters of the metaheuristic methods and the ANOVA was performed to compare the performance of the proposed algorithms.

The statistical optimisation results showed the efficiency, effectiveness and robustness of the applied methods in solving the proposed multi-plant coordination problem. However, the GA resulted in the best-known solutions and generated lower costs, and in general was found to be superior to other two optimisation approaches. In terms of NFEs (computational cost), the PSO was superior to the other methods. The results indicated that the proposed model can provide a promising approach to fulfil an efficient production and distribution planning in such integrated SC situation.

Future research may consider solving the proposed problem in stochastic manufacturing environments, in which the uncertainty and dynamic nature of parameters are taken into account.

Acknowledgements

The authors would like to acknowledge University of Malaya, Kuala Lumpur, Malaysia for providing the necessary facilities and resources for this research. This research was fully funded by University of Malaya Research Grant with grant number of UMRG: RP035C-15AET.

References


Optimisation of multi-plant capacitated lot-sizing problems


