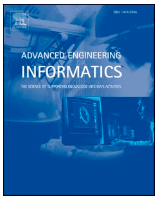
Contents lists available at ScienceDirect

Advanced Engineering Informatics

journal homepage: www.elsevier.com/locate/aei



Full length article

# Exact and metaheuristic algorithms for the vehicle routing problem with a factory-in-a-box in multi-objective settings

Junayed Pasha [a], Arriana L. Nwodu [b], Amir M. Fathollahi-Fard [c], Guangdong Tian [d], Zhiwu Li [e], Hui Wang [b], Maxim A. Dulebenets [f,*]

[a] Department of Civil & Environmental Engineering, Florida A&M University-Florida State University (FAMU-FSU) College of Engineering, 2035 E Paul Dirac Dr., Sliger Building, Suite 203, Tallahassee, FL 32310, USA
[b] Department of Industrial & Manufacturing Engineering, Florida A&M University-Florida State University (FAMU-FSU) College of Engineering, 2525 Pottsdamer Street, Building B, Suite 373D, Tallahassee, FL 32310-6046, USA
[c] Department of Electrical Engineering, École de Technologie Supérieure, University of Quebec, 1100 Notre-Dame St. W., Montreal, Quebec, Canada
[d] School of Mechanical-Electrical and Vehicle Engineering, Beijing University of Civil Engineering and Architecture, Beijing 100044, China
[e] School of Electro-Mechanical Engineering, Xidian University, Taibai South Road, Xi'an, China
[f] Department of Civil & Environmental Engineering, Florida A&M University-Florida State University (FAMU-FSU) College of Engineering, 2035 E Paul Dirac Dr., Sliger Building, Suite 275, Tallahassee, FL 32310, USA

ARTICLE INFO

*Keywords:*
Supply chains
Urgent demand
Factory-in-a-box
Vehicle routing problem
Metaheuristics
Hybrid algorithms

ABSTRACT

Emergencies, such as pandemics (e.g., COVID-19), warrant urgent production and distribution of goods under disrupted supply chain conditions. An innovative logistics solution to meet the urgent demand during emergencies could be the factory-in-a-box manufacturing concept. The factory-in-a-box manufacturing concept deploys vehicles to transport containers that are used to install production modules (i.e., factories). The vehicles travel to customer locations and perform on-site production. Factory-in-a-box supply chain optimization is associated with a wide array of decisions. This study focuses on selection of vehicles for factory-in-a-box manufacturing and decisions regarding the optimal routes within the supply chain consisting of a depot, suppliers, manufacturers, and customers. Moreover, in order to contrast the options of factory-in-a-box manufacturing with those of conventional manufacturing, the location of the final production is determined for each customer (i.e., factory-in-a-box manufacturing with production at the customer location or conventional manufacturing with production at the manufacturer locations). A novel multi-objective optimization model is presented for the vehicle routing problem with a factory-in-a-box that aims to minimize the total cost associated with traversing the edges of the network and the total cost associated with visiting the nodes of the network. A customized multi-objective hybrid metaheuristic solution algorithm that directly considers problem-specific properties is designed as a solution approach. A case study is performed for a vaccination project involving factory-in-a-box manufacturing along with conventional manufacturing. The case study reveals that the developed solution method outperforms the ε-constraint method, which is a classical exact optimization method for multi-objective optimization problems, and several well-known metaheuristics.

## 1. Introduction

In October 2020, the United Nations Educational, Scientific and Cultural Organization (UNESCO) reported that the frequency of pandemics is expected to increase under the existing projections. It is also anticipated that the future pandemics would be more infectious and deadlier [1]. Such predictions are in accordance with pandemic occurrences over the past decades. Indeed, the frequency of pandemics, epidemics, and outbreaks has increased over the years. The 21st century has already experienced some major outbreaks, such as SARS (2003), H1N1 (2009), MERS (2012), Ebola (2014), Zika (2015), and COVID-19 (2019) [2–5]. As of January 2022, the ongoing COVID-19 pandemic has taken

* Corresponding author.

*E-mail addresses:* jp17j@my.fsu.edu (J. Pasha), arriana1.nwodu@famu.edu (A.L. Nwodu), amirmohammad.fathollahifard.1@ens.etsmtl.ca (A.M. Fathollahi-Fard), tiangd2013@163.com (G. Tian), systemscontrol@gmail.com (Z. Li), hwang10@eng.famu.fsu.edu (H. Wang), mdulebenets@eng.famu.fsu.edu (M.A. Dulebenets).

https://doi.org/10.1016/j.aei.2022.101623
Received 16 January 2022; Received in revised form 3 April 2022; Accepted 25 April 2022
Available online 9 May 2022
1474-0346/© 2022 Elsevier Ltd. All rights reserved.

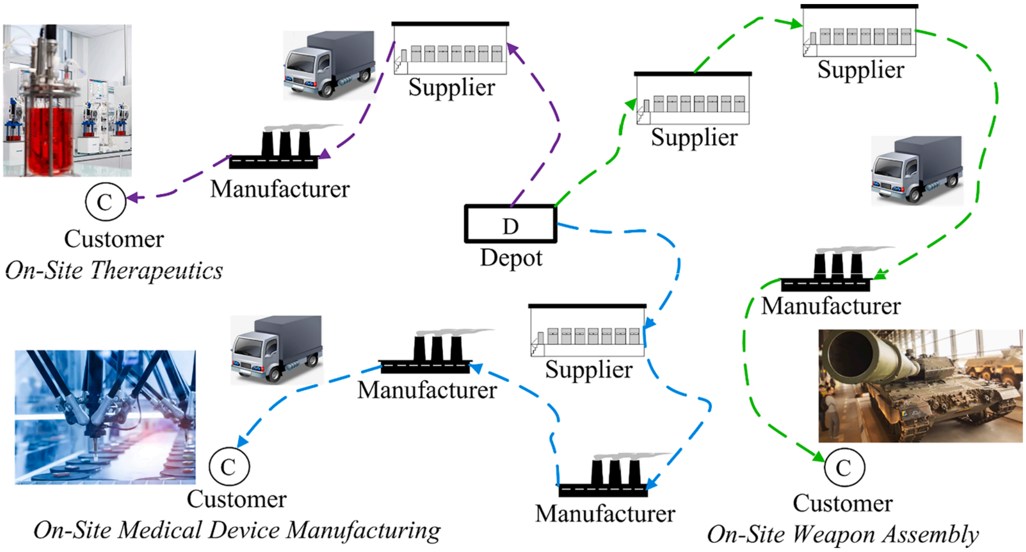


**Fig. 1.** Typical factory-in-a-box manufacturing routes.

more than 5.5 million lives, and more than 315 million cases have been reported [6]. Mitigating the effects of pandemics require extensive measures, such as production of medical supplies (e.g., personal protective equipment, testing kits, vaccines). An urgent demand for medical supplies occurs due to continuous losses of lives. At the same time, pandemics create emergencies and significant supply chain disruptions due to lockdowns, closure of distribution facilities, lack of personnel, and so on. Hence, meeting urgent demand during emergencies becomes challenging. Therefore, there is a need for effective logistics solutions to meet the customer demand during emergencies.

Creative logistics solutions can be effective in meeting urgent demand. One such solution could be the factory-in-a-box manufacturing concept, which deploys vehicles to transport containers that are used to install production modules (i.e., mobile factories). The vehicles travel to customer locations and perform on-site production. Typically, the vehicle routes involved with factory-in-a-box manufacturing start from a depot, traveling to supplier locations to pick up raw materials, moving to manufacturer locations to collect semi-finished products, and then stopping at customer locations for on-site production. Factory-in-a-box manufacturing could be helpful not only to address the challenges due to pandemics but also to meet the urgent demand during natural disasters or for military applications (e.g., production of military supplies during wars). Examples of typical factory-in-a-box manufacturing routes are depicted in Fig. 1. Factory-in-a-box supply chain optimization involves different decisions, which can be categorized into two groups. The first group of decisions is applicable to the pre-transport stage, which includes determination of raw materials for suppliers, decomposition of sub-assembly, assigning manufacturers to sub-assembly modules, and task-manufacturer assignment. The second group of decisions is made for the transport of sub-assembly modules. This study focuses on the latter group of decisions, which comprise selection of vehicles and decisions regarding the optimal routes within the supply chain consisting of a depot, suppliers, manufacturers, and customers.

The factory-in-a-box manufacturing concept has been applied in various industries. GE Healthcare has a system named KUBio that applies the factory-in-a-box concept for mass-production of therapeutics [7]. Furthermore, Nokia has employed the factory-in-a-box concept by packaging production modules in containers and shipping them to customer locations for production [8]. While factory-in-a-box manufacturing has a number of benefits, its necessity should be examined for specific cases. For instance, this manufacturing concept provides extended flexibility and mobility through on-site production at customer locations and by not spending a significant amount of time at manufacturer locations. On the other hand, in conventional manufacturing, the final products are produced by manufacturers, some of which may have an advantage of faster production than on-site production at customer locations (e.g., certain products may require a significant amount of time when manufacturing them at the customer locations using mobile factories as compared to traditional manufacturing at manufacturer locations that have the appropriate equipment and manufacturing resources). Hence, the options of factory-in-a-box manufacturing and conventional manufacturing for each customer need to be examined.

Furthermore, throughout factory-in-a-box supply chain planning, decision makers may have to compromise conflicting objectives. For example, selection of particular routes may minimize the total travel cost but, in the meantime, cause violation of the previously negotiated time windows at customer locations. Factory-in-a-box supply chain optimization has been assessed by only a few studies in the past [9,10]. However, no study has contrasted the options of factory-in-a-box manufacturing with those of conventional manufacturing in multi-objective settings. To fulfill this gap in the state-of-the-art, this study proposes a novel multi-objective optimization model for the vehicle routing problem with a factory-in-a-box, which captures the options of factory-in-a-box manufacturing and conventional manufacturing for each customer. A customized multi-objective hybrid metaheuristic solution algorithm is developed to solve the model. Numerical experiments are further performed to evaluate the proposed multi-objective hybrid metaheuristic solution algorithm and draw some managerial implications based on the solutions obtained. The remainder of this manuscript is organized as follows. The following section conducts a holistic review of the closely-related literature. The third section provides a detailed description of the problem studied herein, while the fourth section contains a mathematical formulation for the studied problem. The fifth section describes the primary solution approach employed, and the sixth section conducts some numerical experiments in order to analyze the proposed solution approach. The seventh section concludes this study.

## 2. Literature review

This section presents a review of the relevant literature, focusing on the following areas: (i) factory-in-a-box manufacturing; and (ii) recent studies on the vehicle routing problem.

### 2.1. Factory-in-a-box manufacturing

As indicated earlier, the factory-in-a-box manufacturing concept has been investigated by only a few studies. Bengtsson et al. [11] suggested that maintenance as well as monitoring could help achieving mobility, speed, and flexibility in case of factory-in-a-box manufacturing. Hedelind et al. [12] presented a project in Sweden that implemented factory-in-a-box manufacturing in order to perform on-demand mobile production. The project involved four other underlying projects and a total of five demonstrators, which were developed with industrial partners, for factory-in-a-box manufacturing. It was revealed that factory-in-a-box manufacturing was able not only to perform on-demand mobile production but also to implement a fast setup and increase production. The study underlined that the factory-in-a-box concept could assist with an effective reconfiguration of the existing manufacturing system and could enhance the production capacity when necessary. However, more comprehensive studies should be still administered to properly evaluate this innovative manufacturing concept. Jackson and Zaman [13] discussed that factory-in-a-box manufacturing could help fulfill uncertain demand through mobility and capacity improvement. Moreover, it was stated that the mobility and flexibility, obtained through factory-in-a-box manufacturing, could help companies reduce production cost.

Olsson et al. [14] analyzed three sub-systems that acted altogether as an enabler for factory-in-a-box manufacturing. The three sub-systems were: (1) cell configurator; (2) monitor agent; and (3) experience reuse server. Winroth and Jackson [15] highlighted three key features of factory-in-a-box manufacturing, which include: (1) mobility; (2) flexibility; and (3) speed. Based on a number of factory-in-a-box projects, Jackson et al. [16] examined the prospects of obtaining a product-service system. Various industrial advantages were found to be associated with factory-in-a-box manufacturing, such as improved service. It was also indicated that factory-in-a-box manufacturing could be an effective tool to reduce the global carbon dioxide emissions. Granlund et al. [17] examined the factory-in-a-box manufacturing concept for small and medium-sized enterprises. A case study was performed for a small company, which featured small volumes and craftsmanship. The study demonstrated factory-in-a-box manufacturing as an effective concept for small and medium-sized enterprises to realize product-service systems and achieve competitiveness.

Jiang et al. [9] implied that in case of vehicles making trips to various sites, factory-in-a-box manufacturing could pose some decision problems for supply chain network design. The major decisions were identified to be sub-assembly planning and supply chain reconfiguration. In order to address these decisions, the study presented a mathematical model, which had the objective of minimizing the sum of reconfiguration costs and production costs. Finally, some guidelines were provided for supply chain network design and reconfiguration under the scope of factory-in-a-box manufacturing. McHauser et al. [18] asserted that factory-in-a-box was an immersive manufacturing environment for industry personnel to develop new skills. Such skills spanned from digital technologies to lean manufacturing. Hence, the study analyzed factory-in-a-box manufacturing from a learning perspective. A simulation environment was created, so that a group of participants could learn different aspects of a factory-in-a-box manufacturing model. In spite of technical complexities, the model was proved to be a useful tool. Pasha et al. [10] presented a mathematical model and a set of optimization algorithms to optimize a factory-in-a-box supply chain. The model aimed to minimize the total supply chain cost. Numerical experiments exhibited the efficiency of the proposed solution algorithms and demonstrated some important managerial implications.

### 2.2. Recent studies on the vehicle routing problem

The vehicle routing problem (VRP) is one of the well-studied decision problems in operations research. This section provides a concise review of some of the recent and relevant efforts on the VRP. For more comprehensive state-of-the-art reviews regarding the VRP, interested readers can refer to Braekers et al. [19], Elshaer and Awad [20], and Mor and Speranza [21]. There are many different VRP variations. For example, under the open VRP, vehicles do not go back to the depot after serving customers [22–24]. The open VRP would be the closest one to the VRP with a factory-in-a-box, as vehicles transporting production modules in containers do not necessarily have to travel back to the depot after they visit the last customer assigned. Brandão [22] presented an iterated local search algorithm for the multi-depot open VRP. In order to define perturbation procedures, the algorithm utilized the search memory. The latter strategy assisted with improving the local search procedure through counting the number of moves for each customer. Numerical experiments indicated that the algorithm could examine potentially better regions of the search space and avoid cycling. Sánchez-Oro et al. [23] assessed a multi-objective open VRP. A total of three objectives were considered, including minimization of the total cost, makespan, and number of vehicles. Variable Neighborhood Search was used to tackle the problem, whose performance was compared with that of Non-Dominated Sorting Genetic Algorithm II. Numerical experiments revealed that the developed Variable Neighborhood Search was the superior of the two tested algorithms. Lalla-Ruiz and Mes [24] presented a two-index-based mathematical formulation for the multi-depot open VRP. The mathematical formulation attempted to enhance sub-tour elimination constraints with the objective of minimizing the total travel cost for delivery of goods. The proposed model was solved with CPLEX, while the maximum CPU time was set to 2 h. It was stated that the proposed methodology could reduce the associated computational complexity and provide good-quality solutions.

A number of studies have addressed the VRP with soft time windows [25,26] and strict time windows [27,28]. Li and Li [25] implied that travel times and service times in the real world could present a random state due to being impacted by various factors, such as inclement weather, congestion, traffic accidents, and so on. Therefore, the study considered stochastic travel times and service times for the VRP with soft time windows. A stochastic programming model was presented to minimize the total distribution cost. A greedy algorithm that was based on Tabu Search was developed to solve the model. The effectiveness of the proposed algorithm was verified via computational experiments. To obtain vehicle routes considering soft time windows, Zhang et al. [26] utilized a reinforcement learning algorithm. The developed problem tackled the VRP with soft time windows as a vehicle tour generation process. In order to generate tours, an encoder-decoder framework was proposed that featured attention layers. It was demonstrated that the algorithm performed better than Google OR-Tools. Keskin et al. [27] assessed the electric VRP with strict time windows. Queuing times at charging stations were modeled as stochastic. A bi-stage linear programming model was presented to generate vehicle routes. A heuristic algorithm was employed for solution, which involved simulation to model stochasticity. Pan et al. [28] considered strict time windows and time-dependent speed for the VRP. Multiple trips were allowed, but a maximum trip duration was imposed. The proposed model had the objective of minimizing the total travel distance. It was indicated that the proposed model was solvable with exact optimization solvers (e.g., CPLEX).

A number of studies have concluded that the VRP along with its variants are classified as NP-hard problems. Therefore, due to computational complexity of the VRP mathematical formulations, heuristics [27,29,30], metaheuristics [20,23,31], and hybrid algorithms [32–35] have been applied by most of the studies to solve the problem. Especially, hybrid algorithms considering problem-specific features could be efficient in solving large-size instances of the VRP.

### 2.3. Literature summary and contributions

A review of the relevant literature implies that very few studies have addressed the factory-in-a-box manufacturing concept, while only one

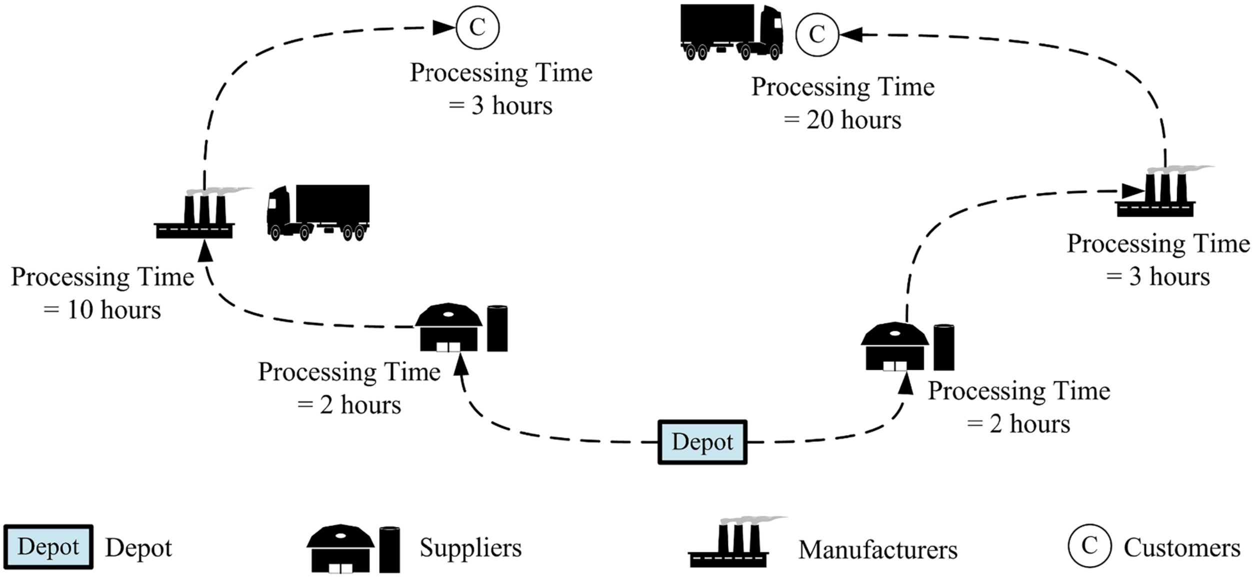


**Fig. 2.** Conventional manufacturing and typical factory-in-a-box manufacturing.

study has provided a supporting model for vehicle routing with a factory-in-a-box [10], which the relevant supply chain stakeholders could employ. Moreover, the benefits of vehicle routing with factory-in-a-box manufacturing have not been contrasted with those of conventional manufacturing in multi-objective settings. Hence, this study aims to make the following contributions to the state-of-the-art:

- A novel mathematical model is formulated for the VRP, which directly captures the options of factory-in-a-box manufacturing and conventional manufacturing for each customer.
- A multi-objective framework is presented to analyze the trade-offs between minimizing the total cost associated with traversing the edges of the network and minimizing the total cost associated with visiting the nodes of the network.
- Complex supplier-to-customer along with manufacturer-to-customer relationships are captured.
- A novel customized nature-inspired Hybrid Multi-Objective Evolutionary Algorithm is developed to solve the problem. The proposed algorithm relies on evolutionary operators for route generation and an exact optimization approach to optimize the locations of the final production.
- Numerical experiments are conducted to examine the computational performance of the developed hybrid multi-objective algorithm in comparison with an exact optimization method along with alternative metaheuristics.
- A detailed analysis is performed for the solutions provided by the developed hybrid multi-objective algorithm for the proposed mathematical model.

## 3. Problem description

The generic VRP involves two types of nodes, including a depot node and customer nodes. Since this research studies a special variant of the VRP, which is associated with factory-in-a-box manufacturing, it involves four basic categories of nodes. These categories of nodes include: (1) the depot (which is further divided into the depot and the dummy depot for mathematical convenience); (2) the suppliers; (3) the manufacturers; and (4) the customers. Similar to the generic VRP, the vehicles will be deployed from the depot node (denoted as “0”). Once deployed, they will visit the suppliers and collect raw materials and then travel to the manufacturers to load semi-processed goods. Unlike typical factory-in-a-box manufacturing, this study weighs the options of factory-in-a-box manufacturing and conventional manufacturing for each customer. In other words, the final products can be manufactured at the manufacturer locations, picked up by the designated vehicle, and delivered to the assigned customer (i.e., conventional manufacturing). Alternatively, semi-finished goods could be collected from the manufacturers by one of the vehicles available, which will travel to the customers afterwards, where the factory will be assembled (i.e., factory-in-a-box manufacturing). After serving the last customer, the vehicles will go to the dummy depot. Note that all time and cost components, associated with the dummy depot, are zero (which resembles open vehicle routing).

According to the graph theory, the studied problem may be modeled using a directed graph $G = (N, E)$, where $N$ stands for the set of all nodes, and $E = \{(i,j), i \in N, j \in N\}$ denotes the set of all edges. Each edge $(i,j) \in E$ involves a travel time $t_{ij}, i \in N, j \in N$ (hours). Every node in the considered graph is associated with a demand $q_i, i \in N$ (lbs). Positive demand of a node indicates linehaul (i.e., pick-up demand), whereas negative demand implies backhaul (i.e., delivery demand). The set of all nodes can be construed as follows: $N = N^{'} \cup \{0\} \cup \{m\}$, where $N^{'} = N^s \cup N^m \cup N^c$. Note that $N^s = \{1, 2, 3, \cdots, m^1\}$, $N^m = \{1, 2, 3, \cdots, m^2\}$, and $N^c = \{1, 2, 3, \cdots, m^3\}$ denote the sets of supplier, manufacturer, and customer nodes, respectively. Node $\{0\}$ denotes the depot node, and $\{m\}$ stands for the dummy depot node. The fleet of vehicles, all of which carry a factory-in-a-box, is denoted by $K = \{1, 2, 3, \cdots, m^4\}$. The vehicles are heterogeneous and have a load carrying limit $Q_k, k \in K$ (lbs), along with a unit travel cost $c_k^v, k \in K$ (USD/hour).

A real-world organization may manufacture various products. However, a unique product is manufactured from specific raw materials and semi-finished goods from specific suppliers and manufacturers, respectively. Hence, this study dictates that if a vehicle is assigned to serve a given customer (that can even be involved in manufacturing the product demanded by the customer in case of factory-in-a-box manufacturing), it first visits the associated suppliers and manufacturers, respectively. In order to ensure this relationship, two binary parameters $b_{ij}^{sc}, i \in N^s, j \in N^c$ and $b_{ij}^{mc}, i \in N^m, j \in N^c$ are employed. If supplier $i$ must be visited before serving customer $j$, then, the value of $b_{ij}^{sc}$ equals to 1 (=0 otherwise). Similarly, if manufacturer $i$ must be visited before serving customer $j$, then, the value of $b_{ij}^{mc}$ equals to 1 (=0 otherwise). Furthermore, in an attempt to ensure a specific order of visits, a precedence level $pl_i, i \in N$ is enforced for each node. Multiple nodes may have the same precedence level. Nonetheless, nodes with decreasing precedence levels cannot be visited.

This study permits vehicle arrivals outside a (soft) time window $[a_i, b_i], i \in N$ (hours) at a node, in exchange of extra costs. Therefore, an early arrival cost ($c_i^e, i \in N$ – USD/hour) is incurred for vehicle arrivals before the time window begins at a node, and a late arrival cost ($c_i^l, i \in N$ – USD/hour) must be paid in case of vehicle arrivals after the time window ends at a node. Furthermore, each node is associated with a processing time ($\boldsymbol{tp}_i, i \in N$ – hours), which further involves a loading/unloading time ($tl_i, i \in N$ – hours) and a manufacturing time ($tm_i, i \in N$ – hours). In case of the supplier nodes, processing times will be equal to loading times. However, the location where the final products will be manufactured is unknown. If the final products are manufactured at a manufacturer location, then, the processing time at the manufacturer location will be the sum of the loading time and the manufacturing time at that location, and the processing time at the associated customer location will be the unloading time at that location. This option is similar to conventional manufacturing, as indicated earlier. On the other hand, if the final products are manufactured at a customer location, then, the processing time at the manufacturer location will be the loading time at that location, and the processing time at the associated customer location will be the sum of the unloading time and the manufacturing time at that location. This option is a typical factory-in-a-box manufacturing process.

Note that the owner of the location where the final products are manufactured, whether they are a manufacturer or a customer, will be paid a compensation cost ($c_i^c, i \in N$ – USD/hour) for manufacturing the final products to compensate for the use of required resources. Fig. 2 illustrates two vehicle routes, where the route on the left side of the depot resembles conventional manufacturing, and the route on the right side of the depot signifies typical factory-in-a-box manufacturing. In the conventional manufacturing route (i.e., the route on the left), the vehicle spends a longer time at the manufacturer location, as the final products are manufactured there. On the other hand, in the typical factory-in-a-box manufacturing route (i.e., the route on the right), the vehicle spends a shorter time at the manufacturer location, which is required only for loading semi-finished goods into the vehicle. In the meantime, the vehicle will have to spend a longer time at the customer location, as the final products will be manufactured there. The decision regarding the type of manufacturing (i.e., conventional manufacturing vs. factory-in-a-box manufacturing) should be made considering different factors, including the following: the time required to manufacture the products at the manufacturer locations, the time required to manufacture the products at the customer locations, the associated compensation costs, the early and late arrival costs at nodes, as well as the vehicle routing choices. For instance, it could be faster to manufacture the products at the manufacturer location but will incur a higher compensation cost as compared to manufacturing the products at the customer location. On the other hand, manufacturing the products at the customer location can be cheaper than manufacturing the products at the manufacturer location but may take more time, which may not be viewed as desirable from the customer perspective. Furthermore, time savings at the manufacturer nodes in case of factory-in-a-box manufacturing can be used to reduce or even prevent time window violations at subsequent nodes during the journey of a given vehicle.

The goal of this study involves minimizing four cost components, which include: (1) the total travel cost; (2) the total early arrival cost; (3) the total late arrival cost; and (4) the total compensation cost. If routes are planned to minimize the total travel time, then, they might incur significant violations of the previously negotiated time windows at the nodes as well as high compensation cost. On the other hand, a route could incur insignificant time window violations at the nodes as well as low compensation cost; however, the associated travel time might be substantial. Therefore, this study groups the cost components into two potentially conflicting objective functions. One objective function is associated with traversing the edges of transportation network by the available vehicles and aims to minimize the total travel cost. On the other hand, the second objective function is associated with operations at the nodes of transportation network and aims to minimize the sum of the total early arrival cost, the total late arrival cost, and the total compensation cost.

## 4. Mathematical model

This section of the manuscript presents a mathematical formulation for the Multi-Objective Factory-in-a-Box Routing (**MOFIBR**) model along with the adopted nomenclature.

### 4.1. Nomenclature

| **Sets** | |
|---|---|
| $N^s = \{1,2,3,\cdots,m^1\}$ | set of supplier nodes (nodes) |
| $N^m = \{1,2,3,\cdots,m^2\}$ | set of manufacturer nodes (nodes) |
| $N^c = \{1,2,3,\cdots,m^3\}$ | set of customer nodes (nodes) |
| $N^{'} = N^s \cup N^m \cup N^c$ | set of supplier, manufacturer, and customer nodes (nodes) |
| $N = N^{'} \cup \{0\} \cup \{m\}$ | set of all nodes (nodes) |
| $E = \{(i,j), i \in N, j \in N\}$ | set of edges (edges) |
| $K = \{1,2,3,\cdots,m^4\}$ | set of vehicles (vehicles) |
| **Decision Variable** | |
| $\boldsymbol{x}_{ijk} \in \mathbb{B} \forall i \in N, j \in N, k \in K$ | =1 if vehicle $k$ traverses edge $(i,j)$ (=0 otherwise) |
| **Auxiliary Variables** | |
| $\boldsymbol{z}_{ik} \in \mathbb{B} \forall i \in N^{'}, k \in K$ | =1 if vehicle $k$ visits node $i$ (=0 otherwise) |
| $\boldsymbol{y}_{ik} \in \mathbb{R}^+ \forall i \in N, k \in K$ | current load on vehicle $k$ upon arrival at node $i$ (lbs) |
| $\boldsymbol{s}_{ik} \in \mathbb{R}^+ \forall i \in N, k \in K$ | service start time at node $i$ by vehicle $k$ (hours) |
| $\boldsymbol{e}_{ik} \in \mathbb{R}^+ \forall i \in N, k \in K$ | early arrival time at node $i$ by vehicle $k$ (hours) |
| $\boldsymbol{l}_{ik} \in \mathbb{R}^+ \forall i \in N, k \in K$ | late arrival time at node $i$ by vehicle $k$ (hours) |
| $\boldsymbol{tp}_i \in \mathbb{R}^+ \forall i \in N$ | processing time at node $i$ (hours) |
| $\boldsymbol{mF}_i \in \mathbb{B} \forall i \in N$ | =1 if the final production is done at manufacturer node $i$ (=0 otherwise) |
| $\boldsymbol{cF}_i \in \mathbb{B} \forall i \in N$ | =1 if the final production is done at customer node $i$ (=0 otherwise) |
| $\boldsymbol{TTC} \in \mathbb{R}^+$ | total travel cost (USD) |
| $\boldsymbol{TEAC} \in \mathbb{R}^+$ | total early arrival cost (USD) |
| $\boldsymbol{TLAC} \in \mathbb{R}^+$ | total late arrival cost (USD) |
| $\boldsymbol{TCC} \in \mathbb{R}^+$ | total compensation cost (USD) |
| $\boldsymbol{F}^1 \in \mathbb{R}^+$ | total cost associated with traversing the edges of the network (USD) |
| $\boldsymbol{F}^2 \in \mathbb{R}^+$ | total cost associated with visiting the nodes of the network (USD) |
| **Parameters** | |
| $m \in \mathbb{N}$ | total number of nodes (nodes) |
| $m^1 \in \mathbb{N}$ | number of supplier nodes (nodes) |
| $m^2 \in \mathbb{N}$ | number of manufacturer nodes (nodes) |
| $m^3 \in \mathbb{N}$ | number of customer nodes (nodes) |
| $m^4 \in \mathbb{N}$ | maximum number of vehicles (vehicles) |
| $Q_k \in \mathbb{R}^+ \forall k \in K$ | load carrying limit of vehicle $k$ (lbs) |
| $q_i \in \mathbb{R} \forall i \in N$ | demand at node $i$ (lbs) |
| $a_i \in \mathbb{R}^+ \forall i \in N$ | time window start at node $i$ (hours) |
| $b_i \in \mathbb{R}^+ \forall i \in N$ | time window end at node $i$ (hours) |
| $t_{ij} \in \mathbb{R}^+ \forall i \in N, j \in N$ | time to travel from node $i$ to node $j$ (hours) |
| $tl_i \in \mathbb{R}^+ \forall i \in N$ | loading/unloading time at node $i$ (hours) |
| $tm_i \in \mathbb{R}^+ \forall i \in N$ | manufacturing time at node $i$ (hours) |
| $pl_i \in \mathbb{N} \forall i \in N$ | precedence level of node $i$ (precedence level) |
| $b_{ij}^{sc} \in \mathbb{B} \forall i \in N^s, j \in N^c$ | binary relationship parameter denoting if supplier $i$ must be visited before visiting customer $j$ |
| $b_{ij}^{mc} \in \mathbb{B} \forall i \in N^m, j \in N^c$ | binary relationship parameter denoting if manufacturer $i$ must be visited before visiting customer $j$ |
| $c_k^v \in \mathbb{R}^+ \forall k \in K$ | unit travel cost of vehicle $k$ (USD/hour) |
| $c_i^e \in \mathbb{R}^+ \forall i \in N$ | unit early arrival cost at node $i$ (USD/hour) |
| $c_i^l \in \mathbb{R}^+ \forall i \in N$ | unit late arrival cost at node $i$ (USD/hour) |
| $c_i^c \in \mathbb{R}^+ \forall i \in N$ | unit compensation cost at node $i$ (USD/hour) |
| $\boldsymbol{M} \in \mathbb{R}^+$ | large positive number |

### 4.2. Model formulation

The mathematical formulation for the **MOFIBR** optimization model with two conflicting objective functions can be presented as follows.

$$min\, \boldsymbol{F}^1 = \boldsymbol{TTC} \quad (1)$$

$$min\, \boldsymbol{F}^2 = \boldsymbol{TEAC} + \boldsymbol{TLAC} + \boldsymbol{TCC} \quad (2)$$

$$\boldsymbol{TTC} = \sum_{i\in N}\sum_{j\in N}\sum_{k\in K} c_k^v t_{ij} \boldsymbol{x}_{ijk} \quad (3)$$

$$\boldsymbol{TEAC} = \sum_{i\in N}\sum_{k\in K} c_i^e \boldsymbol{e}_{ik} \quad (4)$$

$$\boldsymbol{TLAC} = \sum_{i\in N}\sum_{k\in K} c_i^l \boldsymbol{l}_{ik} \quad (5)$$

$$\boldsymbol{TCC} = \sum_{i\in N} c_i^c tm_i (\boldsymbol{mF}_i + \boldsymbol{cF}_i) \quad (6)$$

The objective function (1) aims to minimize the total travel cost, while the objective function (2) aims to minimize the sum of the total early arrival cost, the total late arrival cost, and the total compensation cost. Constraints (3) to (6) quantify the total travel cost, the total early arrival cost, the total late arrival cost, and the total compensation cost, respectively. Note that the unit cost components $c_k^v$, $c_i^e$, $c_i^l$, and $c_i^c$ play the role of normalizing coefficients within the **MOFIBR** mathematical model. Relevant stakeholders may differently perceive the total travel time, the total early arrival time, the total late arrival time, and the total manufacturing time (e.g., stakeholders can be more sensitive to the total manufacturing time rather than the total early arrival time from the operational perspective). Therefore, it may not be appropriate to sum the total early arrival time, the total late arrival time, and the total manufacturing time without application of normalizing coefficients.

The **MOFIBR** mathematical model involves a number of operational constraints to incorporate various features of factory-in-a-box manufacturing within the framework of the VRP. In particular, a total of four groups of constraints are included in this model. The first group of constraints [constraints (7) to (12)] applies some basic routing rules of the **MOFIBR** mathematical model. Constraints (7) ensure that routes are not generated between one single node. Constraints (8) indicate that each node, except the depot and the dummy depot, is served by a single vehicle and only once. Constraints (9) imply that each node, except the depot and the dummy depot, is visited to and from by the same vehicle. Constraints (10) guarantee that each vehicle starts its journey from the depot. Constraints (11) and (12) ensure that a vehicle cannot go to the dummy depot directly after serving a supplier or a manufacturer, respectively.

$$\boldsymbol{x}_{iik} = 0\, \forall i \in N, k \in K \quad (7)$$

$$\sum_{i\in N}\sum_{k\in K} \boldsymbol{x}_{ijk} = 1\, \forall j \in N' \quad (8)$$

$$\sum_{i\in N} \boldsymbol{x}_{ijk} = \sum_{i\in N} \boldsymbol{x}_{jik}\, \forall j \in N', k \in K \quad (9)$$

$$\sum_{j\in N} \boldsymbol{x}_{ijk} = 1\, \forall i = 0, k \in K \quad (10)$$

$$\sum_{i\in N^s} \boldsymbol{x}_{ijk} = 0\, \forall j = m, k \in K \quad (11)$$

$$\sum_{i\in N^m} \boldsymbol{x}_{ijk} = 0\, \forall j = m, k \in K \quad (12)$$

The second group of constraints [constraints (13) and (14)] regulates the current loads on the vehicles. Constraints (13) guarantee that a vehicle's load carrying limit is never exceeded throughout a journey of that vehicle. Constraints (14) ensure that when a vehicle traverses edge $(i,j)$, the sum of the current load on the vehicle upon arrival at node $i$ and the demand at node $i$ should not exceed the current load on the vehicle upon arrival at node $j$.

$$\boldsymbol{y}_{ik} \le Q_k\, \forall i \in N, k \in K \quad (13)$$

$$\boldsymbol{y}_{ik} + q_i - M(1 - \boldsymbol{x}_{ijk}) \le \boldsymbol{y}_{jk}\, \forall i \in N, j \in N, k \in K \quad (14)$$

The third group of constraints [constraints (15) to (22)] deals with the time components of the **MOFIBR** mathematical model. Constraints (15) indicate that when a vehicle traverses edge $(i,j)$, the sum of the service start time at node $i$, the processing time at node $i$, and the time to travel from node $i$ to node $j$ should not exceed the service start time at node $j$. Constraints (16) and (17) compute the early and late arrival times, respectively. Constraints (18) to (22) estimate the processing time at each node.

$$\boldsymbol{s}_{ik} + \boldsymbol{tp}_i + t_{ij} - M(1 - \boldsymbol{x}_{ijk}) \le \boldsymbol{s}_{jk}\, \forall i \in N, j \in N, k \in K \quad (15)$$

$$\boldsymbol{e}_{ik} \ge a_i - \boldsymbol{s}_{ik} - M(1 - \boldsymbol{z}_{ik})\, \forall i \in N, k \in K \quad (16)$$

$$\boldsymbol{l}_{ik} \ge \boldsymbol{s}_{ik} - b_i - M(1 - \boldsymbol{z}_{ik})\, \forall i \in N, k \in K \quad (17)$$

$$\boldsymbol{tp}_i = 0\, \forall i = 0 \quad (18)$$

$$\boldsymbol{tp}_i = 0\, \forall i = m \quad (19)$$

$$\boldsymbol{tp}_i = tl_i\, \forall i \in N^s \quad (20)$$

$$\boldsymbol{tp}_i = tl_i + tm_i \boldsymbol{mF}_i\, \forall i \in N^m \quad (21)$$

$$\boldsymbol{tp}_i = tl_i + tm_i \boldsymbol{cF}_i\, \forall i \in N^c \quad (22)$$

The fourth group of constraints [constraints (23) to (30)] satisfies various production requirements. Constraints (23) to (26) indicate that the final production can be done either at a manufacturer node or the associated customer node. Constraints (27) check if a vehicle has served a supplier, manufacturer, or customer node. Constraints (28) and (29) indicate that if a vehicle is assigned to serve a given customer, it first visits the associated suppliers and manufacturers of that particular customer, respectively. Constraints (30) indicate that the nodes with decreasing precedence levels cannot be visited by a vehicle.

$$\boldsymbol{mF}_i + \sum_{j\in N^c} b_{ij}^{mc} \boldsymbol{cF}_j \le \sum_{j\in N^c} b_{ij}^{mc}\, \forall i \in N^m \quad (23)$$

$$\boldsymbol{mF}_i + \sum_{j\in N^c} b_{ij}^{mc} \boldsymbol{cF}_j \ge 1\, \forall i \in N^m \quad (24)$$

$$\sum_{i\in N^m} b_{ij}^{mc} \boldsymbol{mF}_i + \boldsymbol{cF}_j \le \sum_{i\in N^m} b_{ij}^{mc}\, \forall j \in N^c \quad (25)$$

$$\sum_{i\in N^m} b_{ij}^{mc} \boldsymbol{mF}_i + \boldsymbol{cF}_j \ge 1\, \forall j \in N^c \quad (26)$$

$$\boldsymbol{z}_{ik} = \sum_{j\in N} \boldsymbol{x}_{ijk}\, \forall i \in N', k \in K \quad (27)$$

$$\sum_{i\in N} \boldsymbol{x}_{ijk} = \frac{\sum_{i\in N^s} b_{ij}^{sc} \boldsymbol{z}_{ik}}{\sum_{i\in N^s} b_{ij}^{sc}}\, \forall j \in N^c, k \in K \quad (28)$$

$$\sum_{i\in N} \boldsymbol{x}_{ijk} = \frac{\sum_{i\in N^m} b_{ij}^{mc} \boldsymbol{z}_{ik}}{\sum_{i\in N^m} b_{ij}^{mc}}\, \forall j \in N^c, k \in K \quad (29)$$

$$pl_i - M(1 - \boldsymbol{x}_{ijk}) \le pl_j\, \forall i \in N, j \in N, k \in K \quad (30)$$

## 5. Solution approach

The solution approach employed by this study is presented in this section of the manuscript. Since **MOFIBR** has multiple objective functions, it is associated with a set of non-dominated solutions (i.e., Pareto Front – PF), instead of a single solution. In order to effectively solve

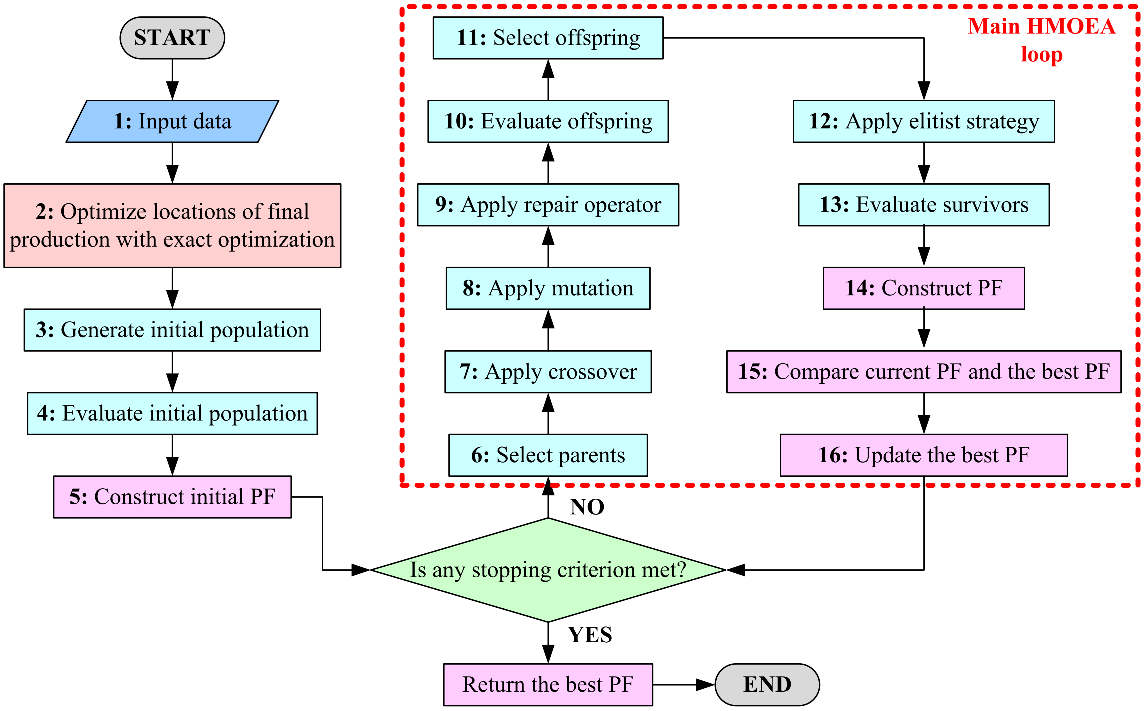


**Fig. 3.** Main steps of the developed HMOEA.

large-size problem instances of **MOFIBR**, a customized Hybrid Multi-Objective Evolutionary Algorithm (HMOEA) was developed that relies on evolutionary operators for route generation and an exact optimization approach to optimize the locations of the final production. The main steps of HMOEA are outlined in Fig. 3. At the start, the algorithmic parameters along with the input data for the **MOFIBR** mathematical model are provided to HMOEA. Then, the locations of the final production are optimized with an exact optimization approach. From this point onward, the evolutionary algorithmic steps are executed. At first, HMOEA generates the initial population, and all the chromosomes of the initial population are evaluated. Then, the PF for the initial population is determined. Afterwards, HMOEA enters a loop, which continues until any of the stopping criteria is met. When a stopping criterion is met, the best PF is returned. The stopping criteria in this study were defined as: (1) a pre-defined number of generations; and (2) a specified consecutive number of generations during which the best PF does not change [36].

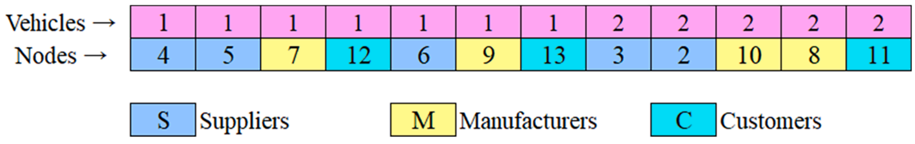
As for the next step inside the loop, parent chromosomes are selected using the Boltzmann selection. Then, the cycle crossover and a custom mutation operator are performed to produce and mutate offspring chromosomes. A repair operator is applied to infeasible offspring chromosomes after crossover and mutation. Afterwards, all the offspring chromosomes are evaluated, and survivor selection is performed. In order to select survivors, the ranking selection is applied. Furthermore, as a part of the survivor selection, the population is injected with the solutions belonging to the best PF discovered along with the two solutions with the best fitness functions (as **MOFIBR** has two objective functions) found from all of the performed generations (i.e., the elitist strategy). Then, the surviving population is evaluated. In the next step, the PF for the current population is determined and compared with the best PF discovered. The best PF is updated as the superior of the two compared PFs. The loop is continued until any stopping criterion is met.

### 5.1. *Chromosome representation*

A 2-dimensional integer chromosome representation has been selected for HMOEA, where the first row is used to denote vehicles and the second row represents the order of nodes to be visited. A chromosome representation with 2 vehicles and 14 nodes is shown in Fig. 4. Based on Fig. 4, the order of visits for vehicle “1” is nodes “4”, “5”, “7”, “12”, “6”, “9”, and “13”. Moreover, the order of visits for vehicle “2” is nodes “3”, “2”, “10”, “8”, and “11”. Nodes “2”, “3”, “4”, “5”, and “6” are the supplier nodes, whereas nodes “7”, “8”, “9”, and “10” are the manufacturer nodes. Furthermore, nodes “11”, “12”, and “13” are the customer nodes. Note that nodes “1” and “14” represent the depot and the dummy depot, respectively, and do not appear in the chromosome (even though each vehicle is deployed from the depot). The proposed solution algorithm was specifically customized for the **MOFIBR** optimization model. In particular, the adopted solution representation allows generating feasible solutions for the **MOFIBR** optimization model that clearly show the assignment of the vehicles to the nodes of the transportation network and the order in which the nodes should be visited by the assigned vehicles.

### 5.2. *Initial population generation*

The initial population generation comprises two steps. At first, a hybridization procedure within the developed algorithm (i.e., optimization of the locations of the final production with an exact optimization



| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Vehicles → | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 2 |
| Nodes → | 4 | 5 | 7 | 12 | 6 | 9 | 13 | 3 | 2 | 10 | 8 | 11 |

S Suppliers M Manufacturers C Customers

**Fig. 4.** The chromosome representation.

| | Customer 1 (*Node 11*) | Customer 2 (*Node 12*) | Customer 3 (*Node 13*) |
|---|---|---|---|
| Supplier 1 (*Node 2*) | 1 | 0 | 0 |
| Supplier 2 (*Node 3*) | 1 | 0 | 0 |
| Supplier 3 (*Node 4*) | 0 | 1 | 0 |
| Supplier 4 (*Node 5*) | 0 | 1 | 0 |
| Supplier 5 (*Node 6*) | 0 | 0 | 1 |

| | Customer 1 (*Node 11*) | Customer 2 (*Node 12*) | Customer 3 (*Node 13*) |
|---|---|---|---|
| Manufacturer 1 (*Node 7*) | 0 | 1 | 0 |
| Manufacturer 2 (*Node 8*) | 1 | 0 | 0 |
| Manufacturer 3 (*Node 9*) | 0 | 0 | 1 |
| Manufacturer 4 (*Node 10*) | 1 | 0 | 0 |

**Fig. 5.** Sample binary data structures denoting the pre-specified suppliers and manufacturers for each customer.

approach) is performed. Then, a stochastic operator is deployed for the chromosome generation (i.e., generation of routes for vehicles).

### 5.2.1. *Optimizing locations of final production with exact optimization*

In order to obtain the locations of the final production, the Final Production Location Identification Problem (**FPLIP**) should be solved. A mathematical formulation of the **FPLIP** decision problem, which is essentially a relaxation of the original **MOFIBR** model, can be formulated as follows:

**Final Production Location Identification Problem (FPLIP):**

$$min\,(\boldsymbol{TCC}) \tag{31}$$

**Subject to:**

$$\boldsymbol{mF}_i + \sum_{j\in N^c} b_{ij}^{mc}\boldsymbol{cF}_j \leq \sum_{j\in N^c} b_{ij}^{mc}\;\forall i \in N^m \tag{32}$$

$$\boldsymbol{mF}_i + \sum_{j\in N^c} b_{ij}^{mc}\boldsymbol{cF}_j \geq 1\;\forall i \in N^m \tag{33}$$

$$\sum_{i\in N^m} b_{ij}^{mc}\boldsymbol{mF}_i + \boldsymbol{cF}_j \leq \sum_{i\in N^m} b_{ij}^{mc}\;\forall j \in N^c \tag{34}$$

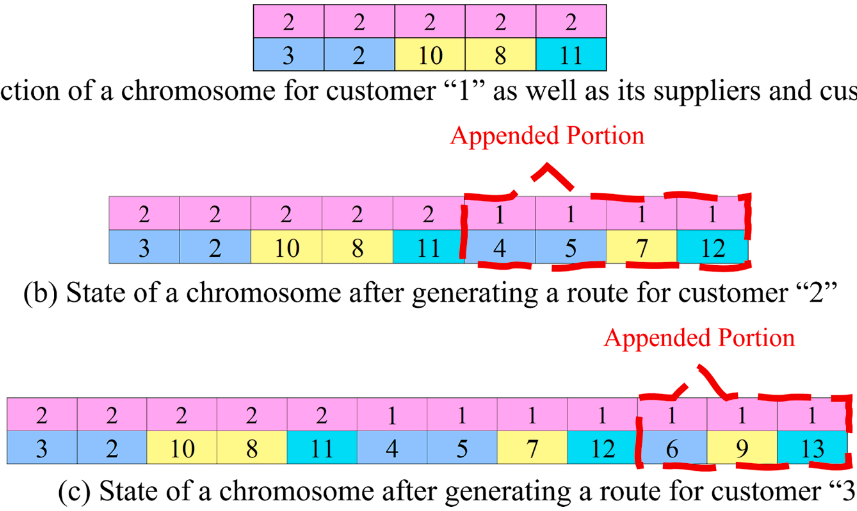
$$\sum_{i\in N^m} b_{ij}^{mc}\boldsymbol{mF}_i + \boldsymbol{cF}_j \geq 1\;\forall j \in N^c \tag{35}$$

$$\boldsymbol{TCC} = \sum_{i\in N} c_i^c tm_i(\boldsymbol{mF}_i + \boldsymbol{cF}_i) \tag{36}$$

The objective function (31) of **FPLIP** is to minimize the total compensation cost. Constraints (32) to (35) indicate that the final production can be done either at a manufacturer node or the associated customer node, while constraint (36) quantifies the total compensation cost. Due to fairly low computational complexity, the **FPLIP** decision problem can be optimally solved in a reasonable computational time using exact mixed-integer programming methods (e.g., CPLEX). The total compensation cost and the locations of the final production are obtained from the solution of this model. Then, the processing time at each node ($\boldsymbol{tp}_i, i \in N$ – hours) can be estimated from the following equation:

$$\boldsymbol{tp}_i = tl_i + tm_i(\boldsymbol{mF}_i + \boldsymbol{cF}_i)\;\forall i \in N \tag{37}$$

Note that the proposed hybridization procedure optimizes the cost that is paid for manufacturing the final product either at a manufacturer location or at a customer location (i.e., the total compensation cost – $\boldsymbol{TCC}$), which serves as a component of the objective function $\boldsymbol{F}^2$. Therefore, the deployment of the proposed hybridization procedure will not affect the objective function $\boldsymbol{F}^1$. In other words, optimization of the location selection for the final production will not affect the total travel time of vehicles traversing the edges of the transportation network and



| | | | | |
|---|---|---|---|---|
| 2 | 2 | 2 | 2 | 2 |
| 3 | 2 | 10 | 8 | 11 |

(a) Fraction of a chromosome for customer “1” as well as its suppliers and customers

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 2 | 2 | 2 | 2 | 2 | 1 | 1 | 1 | 1 |
| 3 | 2 | 10 | 8 | 11 | 4 | 5 | 7 | 12 |

(b) State of a chromosome after generating a route for customer “2”

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 2 | 2 | 2 | 2 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 3 | 2 | 10 | 8 | 11 | 4 | 5 | 7 | 12 | 6 | 9 | 13 |

(c) State of a chromosome after generating a route for customer “3”

**Fig. 6.** Chromosome generation.

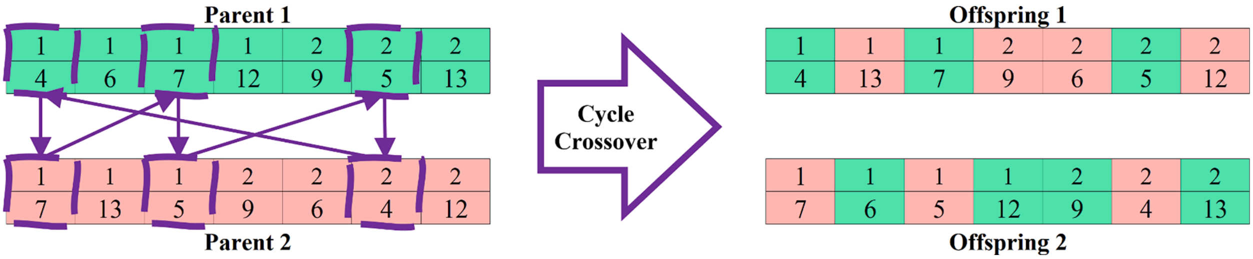


**Fig. 7.** Cycle crossover.

the total number of vehicles deployed.

### 5.2.2. Chromosome generation

Fig. 5 shows an illustrative example of the binary data structures that can be potentially used for the supplier-to-customer relationship ($b_{ij}^{sc}, i \in N^s, j \in N^c$) and the manufacturer-to-customer relationship ($b_{ij}^{mc}, i \in N^m$, $j \in N^c$), where the products that were previously requested by customer “1” (i.e., node “11”) require the raw materials from suppliers “1” and “2” (i.e., nodes “2” and “3”, respectively) as well as the semi-finished products from manufacturers “2” and “4” (i.e., nodes “8” and “10”, respectively). Based on the number of vehicles, a vehicle is randomly assigned to a customer, and the same vehicle is assigned to the customer’s suppliers and manufacturers. In case of customer “1” (i.e., node “11”), vehicle “2” is assigned randomly. After determining the suppliers for a customer, the order of those suppliers is randomly permuted to increase the population diversity. Thus, the order of visits for customer “1” suppliers is randomly permuted to nodes “3” and then “2”. The order of visits for manufacturers is generated in a similar fashion (i.e., nodes “10” and then “8”). Therefore, the order of visits of nodes for customer “1” (i.e., node “11”) is “3”, “2”, “10”, “8”, and “11”. The construction of the chromosome is started after this step. The top row indicates the index of the vehicle serving the group of nodes (e.g., vehicle “2” for the group of nodes corresponding to customer “1”). The bottom row is filled with the order of node visits for that particular group. The fraction of the chromosome for customer “1” along with its suppliers and manufacturers is depicted by Fig. 6(a).

The same steps are repeated for the rest of customers. However, the respective fractions of the chromosome are appended to the right side of the previously constructed chromosome. Fig. 6(b) illustrates the state of the chromosome after generating a route for customer “2” (i.e., node “12”), which is randomly assigned to vehicle “1”. Similarly, Fig. 6(c) illustrates the state of the chromosome after generating a route for customer “3” (i.e., node “13”). Finally, the nodes are sorted based on the assigned vehicles (see Fig. 4). All the other chromosomes in the population are also generated following the aforementioned steps.

## 5.3. Fitness function

The **MOFIBR** model has two objective functions: $\boldsymbol{F}^1$ and $\boldsymbol{F}^2$, which are reflected by two fitness functions. The fitness values ($Fit_{cg}^1$ and $Fit_{cg}^2$) of chromosome $c$, belonging to the set of chromosomes $Chrm = \{1, \cdots, PopSize\}$, in generation $g$, which belongs to the set of generations $Gen = \{1, \cdots, gens\}$, are computed based on the following equations:

$$Fit_{cg}^1 = \boldsymbol{F}_{cg}^1 + \alpha\vartheta_{cg} \, \forall c \in Chrm, g \in Gen \tag{38}$$

$$Fit_{cg}^2 = \boldsymbol{F}_{cg}^2 + \alpha\vartheta_{cg} \, \forall c \in Chrm, g \in Gen \tag{39}$$

Here, $\alpha$ is the penalty coefficient, and $\vartheta_{cg}$ is the cumulative violation of the vehicular load carrying limits for chromosome $c$ in generation $g$, which can be estimated from the load carrying limit ($Q_k$) and the current load ($y_k$) of vehicle $k$ as follows:

$$\vartheta_{cg} = \sum_{i \in N}\sum_{j \in N}\sum_{k \in K} max\{0; (y_k - Q_k)\boldsymbol{x}_{ijk}\} \forall c \in Chrm, g \in Gen \tag{40}$$

Note that for each chromosome, the selection operators of HMOEA use the sum of normalized fitness values instead of two fitness values.

**Algorithm 1** Boltzmann Selection

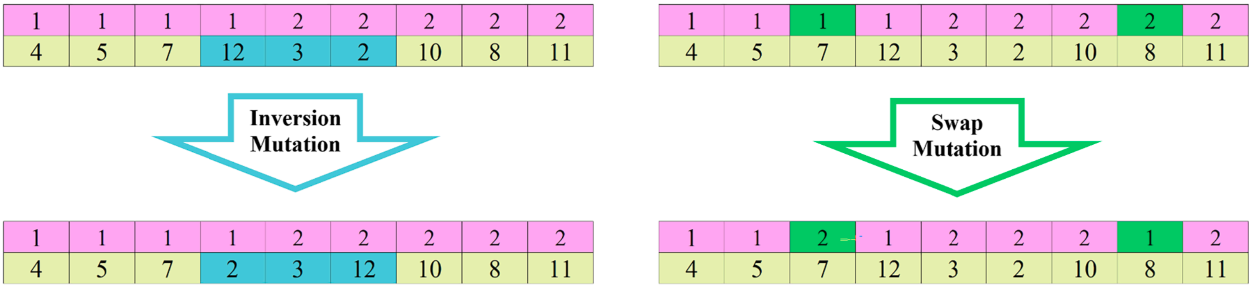
```
In: $Pop_g$ – population in generation $g$; $Fit_g$ – sum of normalized fitness values of
    chromosomes in generation $g$; $T^0$ – initial temperature; $dT$ – temperature interval;$NC$
    – normalizing coefficient
    Out: $Parents_g$ – parent chromosomes in generation $g$
    1: $Parents_g = \oslash$ ◃ Initialization
    2: $T = \max\{1; T^0 - dT \bullet g\}$ ◃ Determine the temperature
    3: $k = 1$ ◃ Choose the first chromosome from the population
    4: while $|Parents_g| \langle |Pop_g|$ do
    5:   $P_k = \frac{exp\left(\frac{-Fit_k}{T \bullet NC \bullet mean\{Fit_g\}}\right)}{\sum_{a=1}^{|Pop_g|} exp\left(\frac{-Fit_a}{T \bullet NC \bullet mean\{Fit_g\}}\right)}$ ◃ Estimate selection probability of the
    chosen chromosome
    6:   if $P_k > rand\{0; 1\}$ then
    7:       $Parents_g = Parents_g \cup Pop_{gk}$ ◃ Assign the chosen chromosome as a parent
    8:   end if
    9:   $k = k \bullet min\left\{1; abs\left(k - |Pop_g|\right)\right\} + 1$ ◃ Choose the next chromosome from the
    population
    10: end while
    11: return $Parents_g$
```

## 5.4. Parent selection

The Boltzmann selection was applied in this study to select the parent chromosomes that are used to produce the offspring chromosomes. The Boltzmann selection was chosen due to its capability to alter the selection pressure throughout different generations. At earlier generations, low-quality chromosomes could survive since the selection pressure is reduced by using a high temperature. On the other hand, at later generations, only high-quality chromosomes could survive since the selection pressure is increased by using a low temperature. The steps of the Boltzmann selection are outlined in **Algorithm 1** [36]. In step 2, the minimum value of the temperature ($T$) is assumed to be 1, as the **MOFIBR** mathematical model has minimization objective functions. For the same reason, a (-)ve sign is used to estimate the selection probability ($P_k$) of chromosome $k$ in step 5. The Boltzmann selection iteratively selects chromosomes depending on the assigned temperature in steps 4–10 until the required number of parent chromosomes have been selected.

## 5.5. Crossover and mutation operators

Crossover and mutation operators are generally used within Evolutionary Algorithms for exploration and exploitation of the search space, respectively (i.e., during diversification and intensification phases) [36]. The cycle crossover operator was adopted in this study to produce offspring chromosomes in order to explore the search space. An illustrative example of the cycle crossover is shown in Fig. 7. Based on the



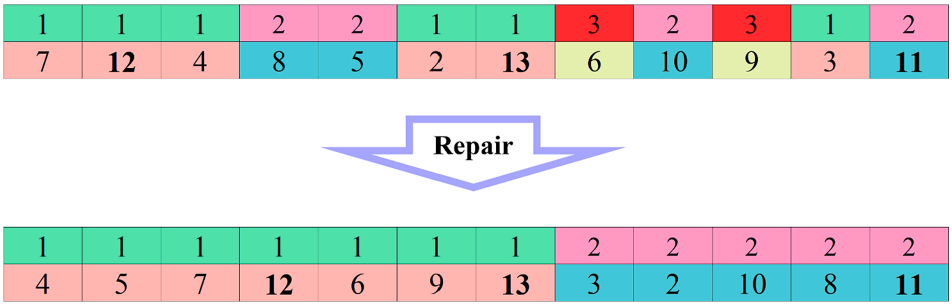
| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 2 |
| 4 | 5 | 7 | 12 | 3 | 2 | 10 | 8 | 11 |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 2 |
| 4 | 5 | 7 | 2 | 3 | 12 | 10 | 8 | 11 |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 2 |
| 4 | 5 | 7 | 12 | 3 | 2 | 10 | 8 | 11 |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 2 | 1 | 2 | 2 | 2 | 1 | 2 |
| 4 | 5 | 7 | 12 | 3 | 2 | 10 | 8 | 11 |

**Fig. 8.** Custom mutation operator.



| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 2 | 2 | 1 | 1 | 3 | 2 | 3 | 1 | 2 |
| 7 | **12** | 4 | 8 | 5 | 2 | **13** | 6 | 10 | 9 | 3 | **11** |

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 2 |
| 4 | 5 | 7 | **12** | 6 | 9 | **13** | 3 | 2 | 10 | 8 | **11** |

**Fig. 9.** Repair operator.

crossover probability, a pair of randomly selected parent chromosomes is used for crossover. The first allele (i.e., value) of the first parent chromosome (i.e., node "4") is appended to the cycle, and the first allele of the second parent chromosome (i.e., node "7") is identified. Then, the locus (i.e., location) of the gene of the first parent chromosome, which contains node "7" (i.e., locus "3") is determined, and node "7" is appended to the cycle. Afterwards, the allele of locus "3" of the second parent chromosome is identified (i.e., node "5"). Then, the locus of the gene of the first parent chromosome, which contains node "5" (i.e., locus "6") is determined, and node "5" is appended to the cycle. Afterwards, the allele of locus "6" of the second parent chromosome is identified (i.e., node "4"). However, node "4" is already included in the cycle, and so, the cycle determination process is terminated. Therefore, the cycle comprises nodes "4", "7", and "5". The genes of the alleles that match the cycle are copied to the first offspring from the first parent, while the rest of the genes are adopted from the second parent. Similarly, the genes of the alleles that match the cycle are copied to the second offspring from the second parent, while the rest of the genes are adopted from the first parent. The cycle crossover would be effective in changing the order of visited nodes for a vehicle as well as altering the assignment of vehicles to nodes.

In order to efficiently exploit the search space, a custom mutation operator was employed by this study to mutate offspring chromosomes. Under this custom mutation operator, a row of a given offspring chromosome is randomly selected (from the top and bottom rows), which will undergo mutation. Then, either inversion mutation or swap mutation is randomly chosen, and the chosen mutation is applied to the selected row (see Fig. 8, where inversion mutation was applied for nodes "12", "3", and "2", whereas swap mutation was applied for vehicles "1" and "2"). The same approach is applied for each chromosome of the population.

### 5.6. Repair operator

After crossover and mutation, a repair operator is applied to offspring chromosomes, in case of infeasibility. Fig. 9 presents an example of the application of the repair operator. Here, vehicle "1" serves customer nodes "12" and then "13", while vehicle "2" serves customer node "11". Vehicle "3" serves two nodes; however, they are not customer nodes. The route of vehicle "1" is infeasible, based on the supplier-to-customer relationships and as well as the manufacturer-to-customer relationships shown in Fig. 5. Hence, the route of vehicle "1" is reconstructed by including (permutations of) the supplier and manufacturer nodes of customer node "12" and then including (permutations of) the supplier and manufacturer nodes of customer node "13". The route of vehicle "2" is also infeasible, and so, it is reconstructed in the same manner. Since vehicle "3" does not serve any customer nodes, its route is removed from the chromosome. Finally, the nodes are sorted based on the assigned vehicles (i.e., vehicle "1" appears before vehicle "2"). The developed repair operator is applied to every infeasible chromosome in the HMOEA population.

**Algorithm 2 Ranking Selection**

```
In: $Off_g$ – offspring chromosomes in generation g; $Fit_g$ – sum of normalized fitness
  values of offspring chromosomes in generation g; $PF^{Best}$ – chromosomes belonging to
  the best PF discovered from all of the performed generations; $c^{Fit^1}$ – chromosome
  with the best value of the first fitness function found from all of the performed
  generations; $c^{Fit^2}$ – chromosome with the best value of the second fitness function
  found from all of the performed generations
Out: $Survivors_g$ – survivor chromosomes in generation g
$Survivors_g = PF^{Best} \cup c^{Fit^1} \cup c^{Fit^2}$ ◁ Initialization
$Off_g^{cand} = Off_g \cup Off_g$ ◁ Create the list of candidate chromosomes
$Fit_g^{cand} = Fit_g \cup Fit_g$ ◁ For each candidate chromosome, assign the sum of
  normalized fitness values
while $|Survivors_g| \langle |Off_g|$ do
    $c^* = argmin\{Fit_g^{cand}\}$ ◁ Locate the fittest chromosome
    $Survivors_g = Survivors_g \cup \{Off_{c^*g}^{cand}\}$ ◁ Assign the fittest chromosome as a survivor
    $Off_g^{cand} = Off_g^{cand} - \{Off_{c^*g}^{cand}\}$ ◁ Update the list of candidate chromosomes
end while
return $Survivors_g$
```

### 5.7. Survivor selection

This study applied the ranking selection to choose offspring chromosomes that will survive in the current generation and be transferred to the following generation, where they could become parent

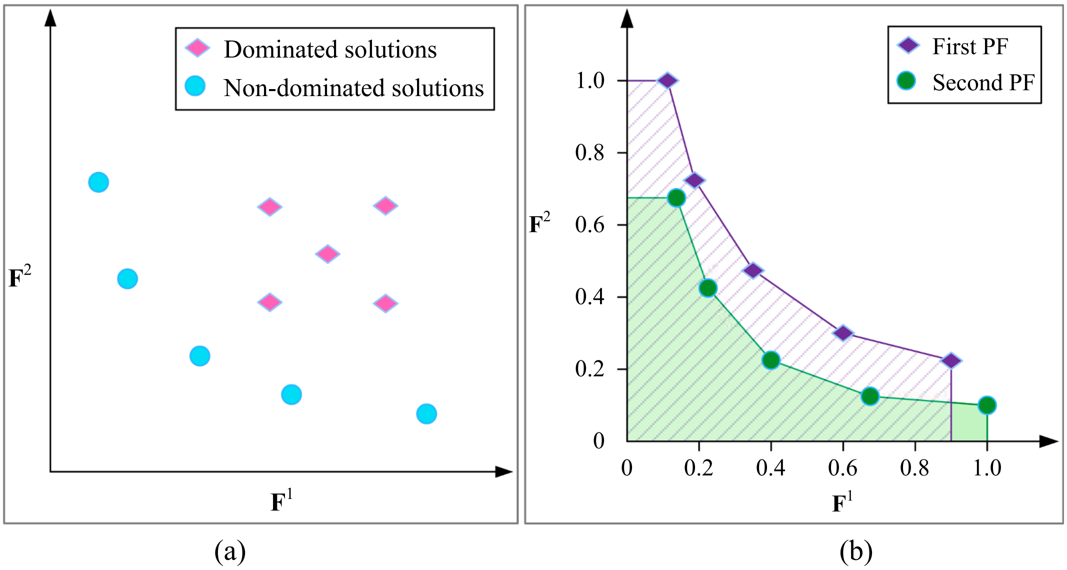


**Fig. 10.** PF construction (a) and evaluation of PFs (b).

chromosomes. The steps of the ranking selection are outlined in **Algorithm 2** [36,37]. In step 1, the ranking selection procedure ensures that the chromosomes belonging to the best PF discovered along with the two chromosomes with the best fitness functions (as **MOFIBR** has two objective functions) found from all of the previous generations will be selected as survivors (i.e., the elitist strategy). Note that in step 2, two copies of the offspring chromosomes are used to develop a pool of the candidate chromosomes for the survivor selection. More copies could be used; however, that would lead to a reduction in the selection pressure, which may not be desirable (especially, at the beginning of the search process). In step 3, the sum of normalized fitness values is assigned to each candidate chromosome. Then, the required number of fittest chromosomes are selected from the pool of candidate chromosomes in steps 4–8 that will be further used as surviving chromosomes for the following generation.

### 5.8. PF construction and evaluation

In order to construct the PF for the population in a given generation, the non-dominated solutions (i.e., chromosomes) are separated from the dominated solutions. Here, the dominated solutions indicate the ones, whose both fitness values (i.e., $Fit_{cg}^{1}$ and $Fit_{cg}^{2}$) are worse than at least one of the considered solutions – see Fig. 10(a). In each generation, two PFs are compared, and the superior of the compared PFs is considered as the best PF onwards. For the comparison of two PFs, this study estimates the surface area beneath normalized objective/fitness functions [38,39]. At first, the fitness functions of the PFs are normalized, such that they range from 0 to 1. Then, the points of each PF are joined. Afterwards, the surface area under each PF is computed. The PF with the smaller surface area is considered as the superior one. An illustrative example of the comparison between two PFs is shown in Fig. 10(b). Since the surface area under the second PF is smaller, the second PF is considered superior to the first PF.

## 6. Numerical experiments

A set of numerical experiments are presented in this section to demonstrate the applicability and performance of the **MOFIBR** mathematical model and the developed HMOEA algorithm. Several other metaheuristic algorithms were employed for comparison with HMOEA. In particular, Non-Dominated Sorting Genetic Algorithm II [40], which is a well-known multi-objective optimization algorithm, and multi-objective versions of Simulated Annealing [41], Tabu Search [42], and Variable Neighborhood Search [43] were employed to evaluate the HMOEA performance. Each of these algorithms was hybridized with an exact optimization approach to optimize the locations of the final production (see section 5.2.1 for details). Hence, the hybridized versions of the algorithms were named as Hybrid Non-Dominated Sorting Genetic Algorithm II (HNSGA-II), Hybrid Multi-Objective Simulated Annealing (HMOSA), Hybrid Multi-Objective Tabu Search (HMOTS), and Hybrid Multi-Objective Variable Neighborhood Search (HMOVNS), respectively. However, as a part of the numerical experiments, all the hybridized algorithms will be compared against their non-hybridized versions (as will be discussed more in detail in section 6.3.1 of the manuscript).

Small-size problem instances were generated to contrast the performances of HMOEA, HMOSA, HMOTS, HMOVNS, and HNSGA-II with that of the ε-constraint method (ECON), which is a well-known exact optimization approach for multi-objective optimization problems. A thorough description of ECON can be found in Mavrotas [44] and Dulebenets [45]. The desired number of PF points was set to 5 for ECON, while the maximum CPU time was set to 24 min for generating each PF point. Thus, the overall CPU time limit was 2 h for ECON. Furthermore, large-size problem instances were generated to select the superior algorithm among HMOEA, HMOSA, HMOTS, HMOVNS, and HNSGA-II. The rationale behind conducting the experiments for two sets of instances consists in the fact that the exact optimization method (i.e., ECON) might be able to produce optimal PFs only for the small-size instances due to the computational complexity of the **MOFIBR** optimization model. However, for the large-size problem instances where the ECON method is not able to obtain the optimal PFs, the developed HMOEA algorithm was evaluated based on the comparative analysis against the alternative metaheuristic algorithms.

In this study, ECON was executed with CPLEX (the target optimality gap was set to 1% at each iteration of ECON), and HMOEA, HMOSA, HMOTS, HMOVNS, and HNSGA-II were encoded in MATLAB (version 2016a). The same CPLEX settings were used to solve the **FPLIP** mathematical model at the initial population generation stage (see section 5.2.1 for details). The numerical experiments were conducted on an Intel (R) Core™ i7-7700 K processor with a 32 GB RAM. The following sections of the manuscript elaborate on the case study that was considered during the experiments, tuning of the parameters for the considered algorithms, evaluation of the considered algorithms in terms of different performance indicators, and managerial insights from the solutions

**Table 1**
The parameter values adopted for the case study.

| Parameter | Value |
|---|---|
| Number of supplier nodes (nodes): $m^1 \in \mathbb{N}$ | $[3;80]$ |
| Number of manufacturer nodes (nodes): $m^2 \in \mathbb{N}$ | $[3;80]$ |
| Number of customer nodes (nodes): $m^3 \in \mathbb{N}$ | $[3;80]$ |
| Maximum number of vehicles (vehicles): $m^4 \in \mathbb{N}$ | 20 |
| Load carrying limit of vehicle $k$ (lbs): $Q_k \in \mathbb{R}^+ \forall k \in K$ | $U[6,000;6,300]$ |
| Demand at supplier node $i$ (lbs): $q_i \in \mathbb{R}^+ \forall i \in N^s$ | $U[280;420]$ |
| Demand at manufacturer node $i$ (lbs): $q_i \in \mathbb{R}^+ \forall i \in N^m$ | $U[1,320;1,980]$ |
| Demand at customer node $i$ (lbs): $q_i \in \mathbb{R}^- \forall i \in N^c$ | $-U[1,600;2,400]$ |
| Duration of time window at node $i$ (hours): $[b_i - a_i] \in \mathbb{R}^+ \forall i \in N$ | 10 |
| Time to travel from node $i$ to node $j$ (hours): $t_{ij} \in \mathbb{R}^+ \forall i \in N, j \in N'$ | $U[5;10]$ |
| Loading/unloading time at node $i$ (hours): $tl_i \in \mathbb{R}^+ \forall i \in N'$ | $U[1.5;2.5]$ |
| Manufacturing time at manufacturer/customer node $i$ (hours): $tm_i \in \mathbb{R}^+ \forall i \in N^m \cup N^c$ | $U[4;5]$ |
| Precedence level of node $i$ (precedence level): $pl_i \in \mathbb{N} \forall i \in N$ | $[0;4]$ |
| Binary relationship parameter denoting if supplier $i$ must be visited before visiting customer $j$: $b_{ij}^{sc} \in \mathbb{B} \forall i \in N^s, j \in N^c$ | $U[0;1]$ |
| Binary relationship parameter denoting if manufacturer $i$ must be visited before visiting customer $j$: $b_{ij}^{mc} \in \mathbb{B} \forall i \in N^m, j \in N^c$ | $U[0;1]$ |
| Unit travel cost of vehicle $k$ (USD/hour): $c_k^v \in \mathbb{R}^+ \forall k \in K$ | $U[900;950]$ |
| Unit early arrival cost at node $i$ (USD/hour): $c_i^e \in \mathbb{R}^+ \forall i \in N'$ | $U[0.1;100]$ |
| Unit late arrival cost at node $i$ (USD/hour): $c_i^l \in \mathbb{R}^+ \forall i \in N'$ | $U[0.1;100]$ |
| Unit compensation cost at manufacturer/customer node $i$ (USD/hour): $c_i^c \in \mathbb{R}^+ \forall i \in N^m \cup N^c$ | $U[5;10]$ |
| Large positive number: $M \in \mathbb{R}^+$ | $1,000,000$ |

returned by the most promising metaheuristic algorithm.

### 6.1. Case study

A case study was performed for a vaccination project involving factory-in-a-box manufacturing along with conventional manufacturing, where raw products (i.e., seeding liquids) were picked up from suppliers. Semi-finished products (i.e., vials) could be picked up from manufacturers for factory-in-a-box manufacturing, or the final products (i.e., vaccines) could be directly manufactured at manufacturer locations (i.e., conventional manufacturing). The vaccination projects, as the one considered in the present study, play a critical role for many geographical locations around the globe due to devastating impacts of the COVID-19 pandemic [6]. The parameter values adopted for the case study are shown in Table 1 [10]. The maximum number of supplier/manufacturer/customer nodes was set to 17 for small-size problem instances and 80 for large-size problem instances. Throughout the numerical experiments, a maximum of 20 vehicles were allowed for utilization, even though all of them might not be used.

### 6.2. Parameter tuning

Parameter tuning is essential to ensure adequate performance of metaheuristics [46–48]. A parameter tuning analysis was performed to select the appropriate parameter values for HMOEA, HMOSA, HMOTS, HMOVNS, and HNSGA-II using a total of 5 problem instances generated based on the data reported in Table 1. The $3^k$ factorial design method was employed, where an algorithm had $k$ parameters, and each parameter was tested with 3 candidate values. Table 2 underlines the parameter tuning analysis results for the considered hybrid multi-objective algorithms. HMOEA has a number of parameters, including the population size (*PopSize*), initial temperature ($T^0$), temperature interval (*dT*), and normalizing coefficient (*NC*) for Boltzmann selection, crossover probability ($\sigma^c$), mutation probability ($\sigma^m$), maximum number of generations the algorithm is allowed to run for (*MaxGens*), and maximum consecutive number of generations during which the best PF does not change (*ConGens*). The best values of these parameters were found to be 200, 1,000, 1.00, 0.10, 0.25, 0.10, 500, and 100, respectively. Unlike HMOEA, HMOSA, HMOTS, HMOVNS, and HNSGA-II have fewer parameters, and their best values are reported in Table 2.

**Table 2**
Parameter tuning analysis for the hybrid multi-objective algorithms.

| Algorithm | Parameter | Candidate Values | Best Value |
|---|---|---|---|
| HMOEA | Population size (*PopSize*) | [50; 100; 200] | 200 |
| | Initial temperature ($T^0$) | [500; 1,000; 2,000] | 1,000 |
| | Temperature interval (*dT*) | [0.25; 0.50; 1.00] | 1.00 |
| | Normalizing coefficient (*NC*) | [0.10; 0.15; 0.20] | 0.10 |
| | Crossover probability ($\sigma^c$) | [0.20; 0.25; 0.30] | 0.25 |
| | Mutation probability ($\sigma^m$) | [0.02; 0.05; 0.10] | 0.10 |
| | Maximum number of generations (*MaxGens*) | [250; 500; 1,000] | 500 |
| | Maximum consecutive number of generations during which the best PF does not change (*ConGens*) | [100; 250; 500] | 100 |
| HMOSA | Population size (*PopSize*) | [50; 100; 200] | 100 |
| | Initial temperature ($T^0$) | [500; 1,000; 2,000] | 1,000 |
| | Temperature interval (*dT*) | [0.25; 0.50; 1.00] | 1.00 |
| | Normalizing coefficient (*NC*) | [0.10; 0.15; 0.20] | 0.10 |
| | Maximum number of iterations (*MaxIters*) | [500; 1,000; 2,000] | 1,000 |
| | Maximum consecutive number of iterations during which the best PF does not change (*ConIters*) | [100; 250; 500] | 100 |
| HMOTS | Population size (*PopSize*) | [50; 100; 200] | 100 |
| | Maximum number of iterations (*MaxIters*) | [500; 1,000; 2,000] | 1,000 |
| | Maximum consecutive number of iterations during which the best PF does not change (*ConIters*) | [100; 250; 500] | 100 |
| HMOVNS | Population size (*PopSize*) | [50; 100; 200] | 100 |
| | Maximum number of iterations (*MaxIters*) | [500; 1,000; 2,000] | 1,000 |
| | Maximum consecutive number of iterations during which the best PF does not change (*ConIters*) | [100; 250; 500] | 100 |
| HNSGA-II | Population size (*PopSize*) | [50; 100; 200] | 200 |
| | Crossover probability ($\sigma^c$) | [0.20; 0.25; 0.30] | 0.25 |
| | Mutation probability ($\sigma^m$) | [0.02; 0.05; 0.10] | 0.10 |
| | Maximum number of generations (*MaxGens*) | [250; 500; 1,000] | 500 |
| | Maximum consecutive number of generations during which the best PF does not change (*ConGens*) | [100; 250; 500] | 100 |

### 6.3. Algorithmic performance evaluation

A total of 15 small-size problem instances as well as 15 large-size problem instances were developed to assess the performance of the solution algorithms. For small-size problem instances, the number of supplier/manufacturer/customer nodes was increased from 3 in instance S-1 to 17 in instance S-15, with an increment of 1 node per small-size problem instance. On the other hand, for large-size problem instances, the number of supplier/manufacturer/customer nodes was increased from 52 in instance L-1 to 80 in instance L-15, with an increment of 2 nodes per large-size problem instance. A detailed evaluation of algorithmic performances for the aforementioned problem instances is exhibited in this section of the manuscript. Since **MOFIBR** is a multi-objective optimization model, several multi-objective optimization performance indicators were examined for the candidate solution algorithms, including the following:

***Quality Metric (QM)***: This metric combines the PF points obtained

**Table 3**
Mean of $\boldsymbol{F}^2$ for the PFs returned by the hybridized and non-hybridized versions of the metaheuristic algorithms.

| Problem Instance | HMOEA (USD) | MOEA (USD) | HMOSA (USD) | MOSA (USD) | HMOTS (USD) | MOTS (USD) | HMOVNS (USD) | MOVNS (USD) | HNSGA-II (USD) | NSGA-II (USD) |
|---|---|---|---|---|---|---|---|---|---|---|
| S-1 | 68.39 | 71.08 | 68.66 | 71.35 | 68.44 | 71.13 | 68.64 | 71.33 | 68.50 | 71.19 |
| S-2 | 91.48 | 95.61 | 91.48 | 95.62 | 91.50 | 95.63 | 91.52 | 95.65 | 91.48 | 95.61 |
| S-3 | 115.25 | 119.38 | 115.54 | 119.67 | 115.83 | 119.97 | 115.37 | 119.50 | 115.22 | 119.35 |
| S-4 | 138.13 | 143.74 | 137.90 | 143.51 | 138.03 | 143.63 | 138.21 | 143.81 | 137.84 | 143.44 |
| S-5 | 163.00 | 168.68 | 163.07 | 168.74 | 162.70 | 168.38 | 163.05 | 168.72 | 163.18 | 168.86 |
| S-6 | 184.77 | 191.93 | 186.20 | 193.35 | 185.45 | 192.53 | 186.51 | 193.66 | 186.67 | 193.82 |
| S-7 | 209.01 | 216.63 | 209.64 | 217.26 | 208.72 | 216.34 | 208.54 | 215.95 | 209.04 | 216.52 |
| S-8 | 231.00 | 241.13 | 233.02 | 243.48 | 234.26 | 244.73 | 232.79 | 243.32 | 234.46 | 244.99 |
| S-9 | 255.08 | 266.20 | 255.94 | 266.67 | 256.62 | 267.01 | 255.51 | 266.70 | 254.99 | 266.18 |
| S-10 | 277.14 | 288.30 | 281.18 | 292.30 | 279.92 | 290.90 | 280.84 | 292.00 | 279.34 | 290.57 |
| S-11 | 300.94 | 312.33 | 307.14 | 319.13 | 303.46 | 314.11 | 306.00 | 317.10 | 308.17 | 319.51 |
| S-12 | 325.91 | 338.87 | 327.48 | 339.67 | 335.16 | 346.58 | 331.27 | 343.27 | 333.89 | 345.95 |
| S-13 | 351.09 | 363.12 | 357.12 | 370.79 | 357.89 | 370.55 | 360.80 | 373.44 | 358.64 | 370.30 |
| S-14 | 375.80 | 389.33 | 376.73 | 389.33 | 378.10 | 391.20 | 379.60 | 393.52 | 384.27 | 397.52 |
| S-15 | 398.16 | 411.63 | 400.64 | 414.75 | 406.29 | 419.71 | 403.89 | 418.86 | 398.94 | 412.51 |
| L-1 | 1,327.76 | 1,361.29 | 1,427.50 | 1,460.56 | 1,405.80 | 1,439.32 | 1,385.96 | 1,418.32 | 1,362.30 | 1,396.08 |
| L-2 | 1,367.89 | 1,400.95 | 1,422.21 | 1,455.77 | 1,471.07 | 1,505.30 | 1,446.15 | 1,481.61 | 1,395.84 | 1,428.84 |
| L-3 | 1,445.88 | 1,480.39 | 1,495.80 | 1,531.01 | 1,522.55 | 1,557.99 | 1,522.81 | 1,556.00 | 1,479.32 | 1,514.03 |
| L-4 | 1,488.91 | 1,524.38 | 1,590.96 | 1,626.95 | 1,530.64 | 1,566.16 | 1,607.51 | 1,642.50 | 1,525.82 | 1,561.64 |
| L-5 | 1,558.56 | 1,595.07 | 1,669.92 | 1,706.80 | 1,650.19 | 1,687.96 | 1,658.62 | 1,695.52 | 1,564.09 | 1,601.89 |
| L-6 | 1,606.15 | 1,646.40 | 1,753.14 | 1,790.99 | 1,748.40 | 1,786.05 | 1,697.07 | 1,733.87 | 1,630.81 | 1,668.66 |
| L-7 | 1,655.39 | 1,694.36 | 1,763.34 | 1,802.35 | 1,799.16 | 1,837.67 | 1,793.62 | 1,830.76 | 1,677.62 | 1,715.70 |
| L-8 | 1,756.21 | 1,795.97 | 1,832.86 | 1,872.24 | 1,876.97 | 1,917.55 | 1,902.72 | 1,942.53 | 1,802.07 | 1,843.43 |
| L-9 | 1,773.41 | 1,814.52 | 1,916.75 | 1,958.67 | 1,899.27 | 1,940.94 | 1,973.69 | 2,013.78 | 1,789.92 | 1,830.54 |
| L-10 | 1,824.92 | 1,867.41 | 2,033.22 | 2,074.79 | 2,003.35 | 2,045.52 | 2,016.54 | 2,058.19 | 1,946.55 | 1,988.29 |
| L-11 | 1,906.06 | 1,948.85 | 1,995.39 | 2,037.49 | 2,161.89 | 2,205.78 | 2,086.48 | 2,128.57 | 1,963.62 | 2,006.81 |
| L-12 | 1,984.36 | 2,027.32 | 2,110.59 | 2,156.75 | 2,117.63 | 2,161.81 | 2,162.33 | 2,205.60 | 2,025.26 | 2,068.65 |
| L-13 | 2,017.78 | 2,061.75 | 2,141.42 | 2,186.29 | 2,163.48 | 2,207.48 | 2,107.31 | 2,149.85 | 2,053.39 | 2,099.18 |
| L-14 | 2,140.07 | 2,185.57 | 2,250.29 | 2,296.28 | 2,286.34 | 2,332.51 | 2,209.22 | 2,254.16 | 2,139.88 | 2,184.66 |
| L-15 | 2,171.12 | 2,219.28 | 2,307.40 | 2,353.78 | 2,288.21 | 2,334.88 | 2,376.40 | 2,424.33 | 2,207.31 | 2,255.75 |

from all the candidate algorithms and constructs a Pareto Set from those points. Then, the number of points from each candidate algorithm belonging to that Pareto Set is estimated [49,50]. A high *QM* indicates better performance.

***Spacing Metric (SM)***: This metric determines the uniformity in distribution of PF points. A low *SM* indicates better performance. *SM* can be estimated from the Euclidean distance between two consecutive PF points ($d_i$), the mean of such Euclidean distances ($\overline{d}$), and the number of PF points (*N*) as follows [49,50]:

$$SM = \frac{\sum_{i=1}^{N-1} |\overline{d} - d_i|}{(N-1)\overline{d}} \tag{41}$$

***Mean Ideal Distance (MID)***: This metric signifies the closeness between the PF points of a candidate algorithm and the ideal points (this study assumes that the ideal points are the ones that belong to the optimal PF generated by the ECON method for a given problem instance). A low *MID* indicates better performance. *MID* can be obtained from the values of the objective functions for a given PF point of a candidate algorithm ($F_i^1$, $F_i^2$), the ideal points ($F^{1^{ideal}}$, $F^{2^{ideal}}$), and the maximum and minimum values of each objective function obtained from combining the PF points of all the candidate algorithms ($\max F^{1^{all}}$, $\min F^{1^{all}}$, $\max F^{2^{all}}$, $\min F^{2^{all}}$) as follows [50,51]:

$$MID = \frac{\sum_{i=1}^{N} \sqrt{\left(\frac{F_i^1 - F^{1^{ideal}}}{\max F^{1^{all}} - \min F^{1^{all}}}\right)^2 + \left(\frac{F_i^2 - F^{2^{ideal}}}{\max F^{2^{all}} - \min F^{2^{all}}}\right)^2}}{N} \tag{42}$$

***Diversification Metric (DM)***: This metric determines the diversity of PF points distributed in the search space. A high *DM* indicates better performance. *DM* can be estimated from the maximum and minimum values of the objective functions returned by a candidate algorithm ($\max F^1$, $\min F^1$, $\max F^2$, $\min F^2$) as follows [49,50]:

$$DM = \sqrt{\left(\frac{\max F^1 - \min F^1}{\max F^{1^{all}} - \min F^{1^{all}}}\right)^2 + \left(\frac{\max F^2 - \min F^2}{\max F^{2^{all}} - \min F^{2^{all}}}\right)^2} \tag{43}$$

The following sections of the manuscript elaborate more on the results from numerical experiments and focus on the assessment of the effects of hybridization for the considered multi-objective metaheuristic algorithms, analysis of the algorithmic performance for small-size problem instances, and analysis of the algorithmic performance for large-size problem instances.

### 6.3.1. *Assessment of the effects of hybridization*

Each of the metaheuristic algorithms was hybridized with an exact optimization approach to optimize the locations of the final production. Since the locations of the final production are associated with the second objective function of the **MOFIBR** mathematical model ($\boldsymbol{F}^2$), an improvement in $\boldsymbol{F}^2$ was noted due to hybridization. The effects of hybridization can be captured by comparing the mean of $\boldsymbol{F}^2$ for the PFs returned by HMOEA, HMOSA, HMOTS, HMOVNS, and HNSGA-II with that of their non-hybridized versions, including the following: (1) Multi-Objective Evolutionary Algorithm (MOEA); (2) Multi-Objective Simulated Annealing (MOSA); (3) Multi-Objective Tabu Search (MOTS); (4) Multi-Objective Variable Neighborhood Search (MOVNS); and (5) Non-Dominated Sorting Genetic Algorithm II (NSGA-II). All the considered hybridized algorithms and their non-hybridized versions were executed 10 times for the developed small-size and large-size problem instances to estimate the mean values of $\boldsymbol{F}^2$, and the results are presented in Table 3. Based on the conducted analysis, it can be observed that hybridized metaheuristic algorithms HMOEA, HMOSA, HMOTS, HMOVNS, and HNSGA-II consistently outperformed their non-hybridized versions and returned lower $\boldsymbol{F}^2$ values for all the developed problem instances. The objective improvements of up to ≈5% were recorded during the computational experiments. Hence, the hybridization technique proposed in this study could be considered as effective in enhancing the quality of produced PFs.

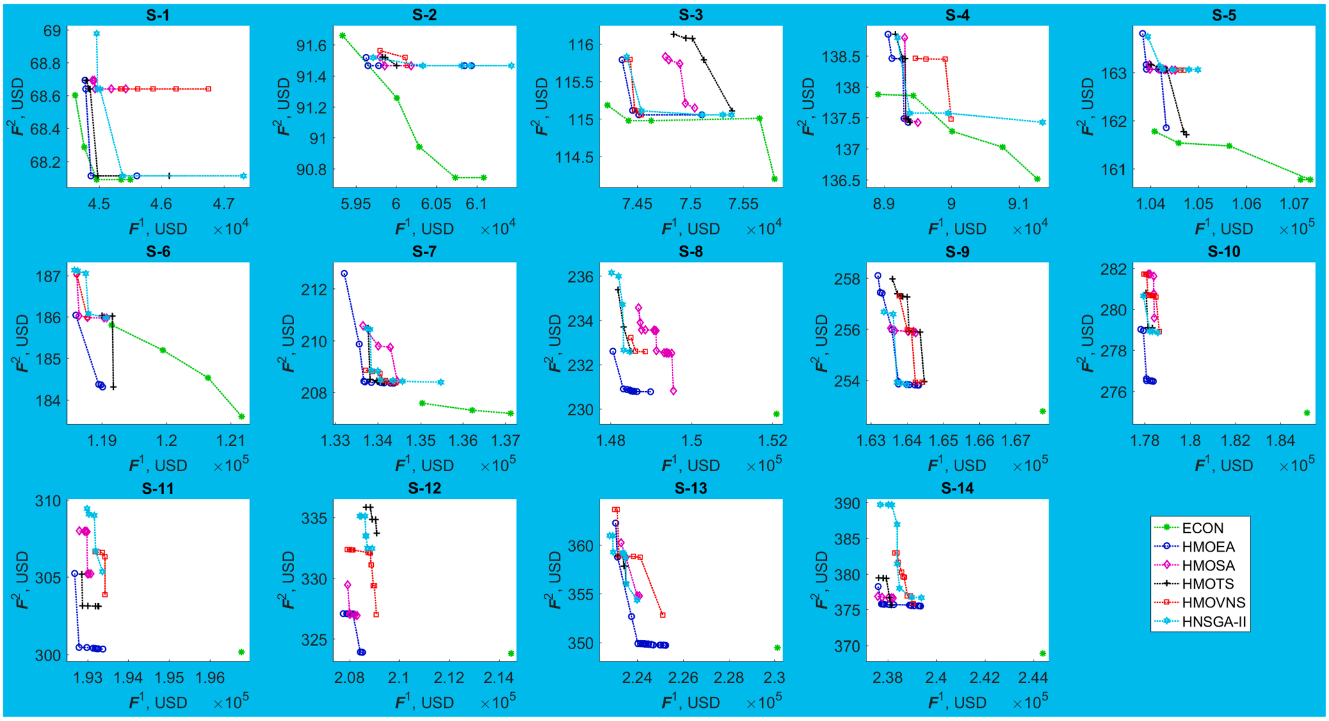


**Fig. 11.** PFs generated for the small-size problem instances.

**Table 4**
The *QM* values for the candidate algorithms and small-size problem instances.

| Problem Instance | Total Points | ECON | % | HMOEA | % | HMOSA | % | HMOTS | % | HMOVNS | % | HNSGA-II | % |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| S-1 | 6 | **5** | **83.33** | 1 | 16.67 | 0 | 0.00 | 0 | 0.00 | 0 | 0.00 | 0 | 0.00 |
| S-2 | 9 | **5** | **55.56** | 3 | 33.33 | 1 | 11.11 | 0 | 0.00 | 0 | 0.00 | 0 | 0.00 |
| S-3 | 5 | **5** | **100.00** | 0 | 0.00 | 0 | 0.00 | 0 | 0.00 | 0 | 0.00 | 0 | 0.00 |
| S-4 | 8 | **4** | **50.00** | 2 | 25.00 | 1 | 12.50 | 2 | 25.00 | 0 | 0.00 | 0 | 0.00 |
| S-5 | 9 | **5** | **55.56** | 3 | 33.33 | 1 | 11.11 | 0 | 0.00 | 0 | 0.00 | 0 | 0.00 |
| S-6 | 9 | 1 | 11.11 | **4** | **44.44** | 2 | 22.22 | 1 | 11.11 | 1 | 11.11 | 1 | 11.11 |
| S-7 | 12 | 3 | 25.00 | **9** | **75.00** | 0 | 0.00 | 0 | 0.00 | 0 | 0.00 | 0 | 0.00 |
| S-8 | 12 | 1 | 8.33 | **10** | **83.33** | 0 | 0.00 | 0 | 0.00 | 0 | 0.00 | 1 | 8.33 |
| S-9 | 16 | 1 | 6.25 | **10** | **62.50** | 2 | 12.50 | 0 | 0.00 | 0 | 0.00 | 3 | 18.75 |
| S-10 | 9 | 1 | 11.11 | **8** | **88.89** | 0 | 0.00 | 0 | 0.00 | 0 | 0.00 | 0 | 0.00 |
| S-11 | 10 | 1 | 10.00 | **9** | **90.00** | 0 | 0.00 | 0 | 0.00 | 0 | 0.00 | 0 | 0.00 |
| S-12 | 11 | 1 | 9.09 | **7** | **63.64** | 3 | 27.27 | 0 | 0.00 | 0 | 0.00 | 0 | 0.00 |
| S-13 | 27 | 1 | 3.70 | **18** | **66.67** | 0 | 0.00 | 3 | 11.11 | 0 | 0.00 | 5 | 18.52 |
| S-14 | 20 | 1 | 5.00 | **16** | **80.00** | 2 | 10.00 | 1 | 5.00 | 0 | 0.00 | 0 | 0.00 |
| **Mean:** | | 2.43 | 31.00 | **7.14** | **54.49** | 0.86 | 7.62 | 0.50 | 3.73 | 0.07 | 0.79 | 0.71 | 4.05 |

**Table 5**
The *SM* values for the candidate algorithms and small-size problem instances.

| Problem Instance | ECON | HMOEA | HMOSA | HMOTS | HMOVNS | HNSGA-II |
|---|---|---|---|---|---|---|
| S-1 | **0.386** | 1.103 | 0.843 | 1.224 | 0.467 | 1.148 |
| S-2 | **0.282** | 1.131 | 0.696 | 0.610 | 0.837 | 0.381 |
| S-3 | 0.797 | 0.912 | 0.399 | 0.466 | **0.159** | 0.659 |
| S-4 | **0.135** | 0.577 | 0.846 | 0.607 | 0.381 | 0.634 |
| S-5 | 0.591 | 0.677 | 0.577 | 0.704 | **0.212** | 0.349 |
| S-6 | 0.149 | 1.046 | 0.677 | 0.816 | **0.000** | 0.629 |
| S-7 | **0.132** | 0.526 | 0.274 | 0.648 | 0.508 | 0.805 |
| S-8 | – | 0.832 | 0.711 | 0.667 | **0.338** | 0.682 |
| S-9 | – | 0.674 | 0.559 | 0.413 | 0.497 | **0.364** |
| S-10 | – | 0.581 | 0.972 | 0.725 | 0.581 | **0.335** |
| S-11 | – | 0.639 | 0.890 | 0.745 | **0.473** | 0.795 |
| S-12 | – | 0.839 | **0.483** | 0.639 | 0.922 | 0.694 |
| S-13 | – | 0.751 | **0.572** | 0.777 | 1.072 | 1.015 |
| S-14 | – | 0.925 | 0.613 | **0.231** | 0.345 | 0.718 |
| **Mean:** | – | 0.801 | 0.651 | 0.662 | **0.485** | 0.658 |

### *6.3.2. Analysis of small-size problem instances*

As a part of the numerical experiments, the developed HMOEA, HMOSA, HMOTS, and HMOVNS metaheuristic algorithms were evaluated against the ECON algorithm, which is a well-known exact multi-objective optimization algorithm, for all the considered small-size problem instances, where the number of supplier/manufacturer/customer nodes was increased from 3 in instance S-1 to 17 in instance S-15. The PFs generated by all the solution algorithms for the small-size problem instances are plotted in Fig. 11. Fig. 11 demonstrates that ECON generally performed better than the metaheuristic algorithms in instances S-1 through S-5, even though the objective functions of the metaheuristic algorithms were close to those of ECON (i.e., they provided good-quality solutions). The ECON performance started to decline from instance S-6. In fact, ECON could not provide solutions for the required 5 PF points for instance S-6 within the CPU time limit of 24 min per PF point. Moreover, ECON was not able to provide even one PF point for instance S-15 or larger within the CPU time limit. Hence, the results for instance S-15 were not plotted. Thus, the ECON algorithm, which is a classical exact optimization method for multi-objective optimization

**Table 6**
The *MID* values for the candidate algorithms.

| Problem Instance | HMOEA | HMOSA | HMOTS | HMOVNS | HNSGA-II |
|---|---|---|---|---|---|
| S-1 | **0.443** | 0.671 | 0.526 | 0.787 | 0.713 |
| S-2 | 0.928 | **0.864** | 0.872 | 0.907 | 1.041 |
| S-3 | **0.631** | 0.827 | 1.028 | 0.632 | 0.754 |
| S-4 | 0.712 | **0.633** | 0.673 | 0.818 | 0.793 |
| S-5 | 0.736 | 0.760 | **0.651** | 0.766 | 0.808 |
| **Mean:** | **0.690** | 0.751 | 0.750 | 0.782 | 0.822 |

**Table 7**
The *DM* values for the candidate algorithms and small-size problem instances.

| Problem Instance | ECON | HMOEA | HMOSA | HMOTS | HMOVNS | HNSGA-II |
|---|---|---|---|---|---|---|
| S-1 | 0.664 | 0.724 | 0.207 | 0.820 | 0.521 | **1.308** |
| S-2 | **1.303** | 0.624 | 0.193 | 0.104 | 0.194 | 0.820 |
| S-3 | **1.123** | 0.610 | 0.397 | 0.634 | 0.391 | 0.745 |
| S-4 | **1.131** | 0.621 | 0.590 | 0.615 | 0.472 | 1.058 |
| S-5 | **0.986** | 0.660 | 0.156 | 0.524 | 0.052 | 0.376 |
| S-6 | **0.998** | 0.516 | 0.350 | 0.492 | 0.303 | 0.378 |
| S-7 | 0.538 | **0.839** | 0.444 | 0.399 | 0.202 | 0.587 |
| S-8 | – | 0.367 | **0.624** | 0.442 | 0.133 | 0.570 |
| S-9 | – | **0.845** | 0.156 | 0.781 | 0.654 | 0.554 |
| S-10 | – | 0.382 | 0.318 | 0.256 | **0.424** | 0.278 |
| S-11 | – | **0.553** | 0.309 | 0.247 | 0.307 | 0.450 |
| S-12 | – | 0.286 | 0.217 | 0.189 | **0.478** | 0.234 |
| S-13 | – | **0.935** | 0.403 | 0.199 | 0.818 | 0.496 |
| S-14 | – | 0.289 | 0.095 | 0.199 | 0.361 | **0.675** |
| **Mean:** | – | 0.589 | 0.319 | 0.421 | 0.379 | **0.609** |

problems, would not be able to handle large-size problem instances because of the computational complexity of the **MOFIBR** mathematical model. On the other hand, the performance of developed metaheuristic algorithms remained consistent.

As for multi-objective optimization performance indicators, HMOEA exhibited the best values of *QM* when comparing to the ECON, HMOSA, HMOTS, HMOVNS, and HNSGA-II algorithms (see Table 4). HMOEA on average contributed more than 50% of the total PF points for the considered small-size problem instances. Furthermore, the HMOEA PFs were found to be the closest ones to the optimal PFs obtained by the ECON method based on the estimated *MID* values (see Table 6). Note that the *MID* values were only computed for the small-size problem instances S-1 through S-5, as ECON was not able to generate a full PF with 5 points for the rest of small-size problem instances within the CPU time limit imposed due to the computational complexity of the **MOFIBR** mathematical model. Throughout the numerical experiments, it was noticed that HMOEA did not demonstrate the superiority in terms of the best *SM* and *DM* performance indicators (see Tables 5 and 7), as some of the PF points generated by HMOEA were located close to each other. However, the *SM* and *DM* values recorded for HMOEA still can be viewed as acceptable, when comparing to other solution algorithms.

As for the computational time, on average over 10 replications, ECON required 2,402.75 s per small-size problem instance, whereas HMOEA, HMOSA, HMOTS, HMOVNS, and HNSGA-II required only 34.92 s, 28.00 s, 19.26 s, 23.88 s, and 37.50 s, respectively. Therefore, based on the numerical experiments conducted for the developed small-size problem instances, HMOEA was found to be superior to the ECON method and metaheuristic algorithms considered, taking into account the shapes of PFs obtained, recorded values of multi-objective optimization performance indicators, and average computational time incurred.

### 6.3.3. *Analysis of large-size problem instances*

As a part of the numerical experiments, the developed HMOEA, HMOSA, HMOTS, and HMOVNS metaheuristic algorithms were evaluated against the HNSGA-II algorithm, which is a well-known multi-objective optimization algorithm with advanced operators (e.g., non-dominated sorting, crowding distance sorting, genetic operators), for all the considered large-size problem instances, where the number of supplier/manufacturer/customer nodes was increased from 52 in instance L-1 to 80 in instance L-15. The PFs generated by all the metaheuristic algorithms for the large-size problem instances are plotted in Fig. 12. As indicated earlier, the ECON method was not able to generate

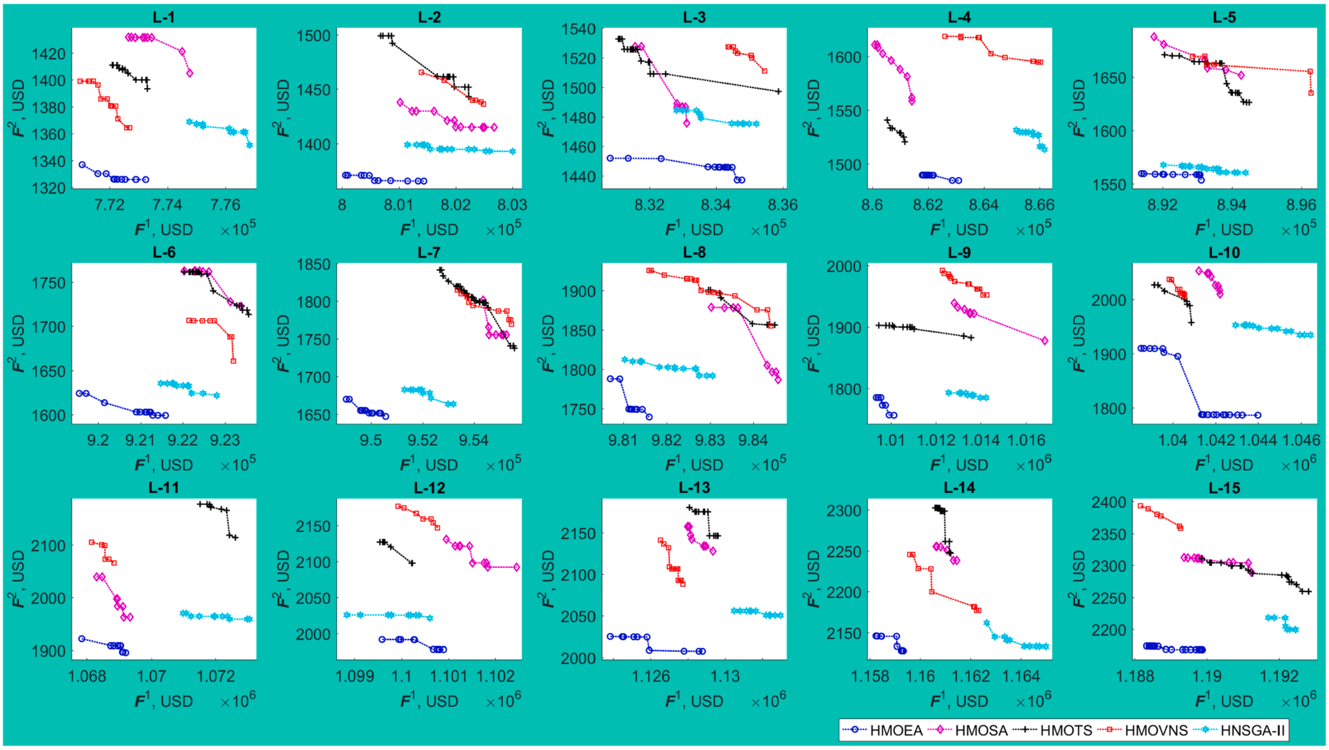


**Fig. 12.** PFs generated for the large-size problem instances.

**Table 8**
The *QM* values for the candidate algorithms and large-size problem instances.

| Problem Instance | Total Points | HMOEA | % | HMOSA | % | HMOTS | % | HMOVNS | % | HNSGA-II | % |
|---|---|---|---|---|---|---|---|---|---|---|---|
| L-1 | 12 | **11** | **91.67** | 0 | 0.00 | 0 | 0.00 | 1 | 8.33 | 0 | 0.00 |
| L-2 | 11 | **11** | **100.00** | 0 | 0.00 | 0 | 0.00 | 0 | 0.00 | 0 | 0.00 |
| L-3 | 15 | **15** | **100.00** | 0 | 0.00 | 0 | 0.00 | 0 | 0.00 | 0 | 0.00 |
| L-4 | 24 | **12** | **50.00** | 4 | 16.67 | 8 | 33.33 | 0 | 0.00 | 0 | 0.00 |
| L-5 | 12 | **12** | **100.00** | 0 | 0.00 | 0 | 0.00 | 0 | 0.00 | 0 | 0.00 |
| L-6 | 12 | **12** | **100.00** | 0 | 0.00 | 0 | 0.00 | 0 | 0.00 | 0 | 0.00 |
| L-7 | 14 | **14** | **100.00** | 0 | 0.00 | 0 | 0.00 | 0 | 0.00 | 0 | 0.00 |
| L-8 | 10 | **10** | **100.00** | 0 | 0.00 | 0 | 0.00 | 0 | 0.00 | 0 | 0.00 |
| L-9 | 7 | **7** | **100.00** | 0 | 0.00 | 0 | 0.00 | 0 | 0.00 | 0 | 0.00 |
| L-10 | 22 | **22** | **100.00** | 0 | 0.00 | 0 | 0.00 | 0 | 0.00 | 0 | 0.00 |
| L-11 | 10 | **10** | **100.00** | 0 | 0.00 | 0 | 0.00 | 0 | 0.00 | 0 | 0.00 |
| L-12 | 13 | **11** | **84.62** | 0 | 0.00 | 0 | 0.00 | 0 | 0.00 | 2 | 15.38 |
| L-13 | 10 | **10** | **100.00** | 0 | 0.00 | 0 | 0.00 | 0 | 0.00 | 0 | 0.00 |
| L-14 | 11 | **11** | **100.00** | 0 | 0.00 | 0 | 0.00 | 0 | 0.00 | 0 | 0.00 |
| L-15 | 21 | **20** | **95.24** | 0 | 0.00 | 0 | 0.00 | 1 | 4.76 | 0 | 0.00 |
| **Mean:** | | **12.53** | **94.77** | 0.27 | 1.11 | 0.53 | 2.22 | 0.13 | 0.87 | 0.13 | 1.03 |

**Table 9**
The *SM* values for the candidate algorithms and large-size problem instances.

| Problem Instance | HMOEA | HMOSA | HMOTS | HMOVNS | HNSGA-II |
|---|---|---|---|---|---|
| L-1 | 0.636 | 0.749 | 0.801 | **0.532** | 1.009 |
| L-2 | **0.571** | 0.583 | 1.019 | 0.846 | 0.787 |
| L-3 | 1.056 | 1.081 | 1.370 | 1.009 | **0.707** |
| L-4 | 1.026 | 0.561 | 0.692 | 0.907 | **0.454** |
| L-5 | 0.817 | 0.724 | **0.599** | 1.216 | 0.799 |
| L-6 | 0.807 | **0.645** | 0.901 | 0.672 | 0.760 |
| L-7 | 0.721 | **0.548** | 0.799 | 0.864 | 0.772 |
| L-8 | **0.690** | 0.780 | 0.825 | 0.845 | 0.735 |
| L-9 | **0.314** | 1.339 | 1.110 | 0.813 | 0.856 |
| L-10 | 0.734 | 0.796 | 0.912 | 0.716 | **0.567** |
| L-11 | 1.089 | 0.852 | 0.722 | 0.805 | **0.717** |
| L-12 | 0.983 | 0.841 | 0.993 | **0.364** | 0.773 |
| L-13 | 0.672 | 1.101 | 0.540 | 0.686 | **0.516** |
| L-14 | 1.082 | 0.775 | **0.531** | 1.139 | 0.626 |
| L-15 | 0.741 | 0.852 | 0.810 | **0.582** | 0.782 |
| **Mean:** | 0.796 | 0.815 | 0.842 | 0.800 | **0.724** |

**Table 10**
The *DM* values for the candidate algorithms and large-size problem instances.

| Problem Instance | HMOEA | HMOSA | HMOTS | HMOVNS | HNSGA-II |
|---|---|---|---|---|---|
| L-1 | 0.388 | **0.441** | 0.264 | 0.438 | 0.393 |
| L-2 | 0.469 | 0.589 | **0.674** | 0.433 | 0.633 |
| L-3 | 0.798 | 0.625 | **1.021** | 0.274 | 0.485 |
| L-4 | 0.217 | 0.449 | 0.183 | **0.591** | 0.214 |
| L-5 | 0.357 | 0.583 | 0.601 | **0.744** | 0.490 |
| L-6 | **0.530** | 0.418 | 0.482 | 0.384 | 0.342 |
| L-7 | 0.261 | 0.277 | **0.695** | 0.398 | 0.308 |
| L-8 | 0.349 | 0.635 | 0.461 | **0.818** | 0.535 |
| L-9 | 0.161 | **0.597** | 0.554 | 0.315 | 0.225 |
| L-10 | **0.832** | 0.204 | 0.341 | 0.153 | 0.447 |
| L-11 | 0.278 | 0.336 | 0.309 | 0.195 | **0.400** |
| L-12 | 0.369 | 0.459 | 0.242 | 0.278 | **0.489** |
| L-13 | **0.551** | 0.227 | 0.259 | 0.333 | 0.275 |
| L-14 | 0.191 | 0.155 | 0.331 | **0.565** | 0.385 |
| L-15 | 0.335 | 0.416 | **0.676** | 0.284 | 0.182 |
| **Mean:** | 0.406 | 0.427 | **0.473** | 0.414 | 0.387 |

even one PF point for the considered large-size problem instances within the CPU time limit imposed; hence, its results are not reported. Fig. 12 demonstrates that HMOEA outperformed HNSGA-II in terms of both of the objective functions of the **MOFIBR** mathematical model in 14 out of the 15 large-size problem instances and in terms of the second objective function ($\boldsymbol{F}^2$) in instance L-12. In fact, HMOEA provided superior solutions with respect to all of the candidate metaheuristic algorithms in terms of one or both of the objective functions in all of the large-size problem instances. Such a finding can be also justified by the recorded values of the *QM* performance indicator. HMOEA on average contributed more than 90% of the total PF points for the considered large-size problem instances (see Table 8).

When considering other multi-objective optimization performance indicators (i.e., *SM* and *DM*), all the developed hybridized metaheuristic algorithms demonstrated similar performance. In particular, the average values of the *SM* indicator varied between ≈0.700 and ≈0.800 (see Table 9). On the other hand, the average values of the *DM* indicator ranged between ≈0.400 and ≈0.500 (see Table 10). As for the computational time, on average over 10 replications, HMOEA, HMOSA, HMOTS, HMOVNS, and HNSGA-II required 366.74 s, 217.14 s, 177.11 s, 173.23 s, and 406.58 s, respectively, per large-size problem instance. Hence, all of the metaheuristic algorithms were able to tackle the large-size problem instances in a reasonable amount of time. Therefore, based on the numerical experiments conducted for the developed large-size problem instances, HMOEA was found to be superior to HNSGA-II and other metaheuristic algorithms considered, taking into account the shapes of PFs obtained, recorded values of multi-objective optimization performance indicators, and average computational time incurred.

### 6.4. *Detailed analysis of solutions*

This section of the manuscript performs a comprehensive analysis of the solutions provided by the HMOEA algorithm, which was found to be the most promising hybridized metaheuristic algorithm based on the conducted numerical experiments, for large-size problem instances of the **MOFIBR** mathematical model. Table 11 shows the average values of the total travel time, the total early arrival time, the total late arrival time, and the total manufacturing time over 10 replications obtained by HMOEA for all the considered large-size problem instances and corner PF points (i.e., the points with the minimum $\boldsymbol{F}^1$ and $\boldsymbol{F}^2$ values). Based on the conducted numerical experiments, it can be generally noticed that there was a gradual increase in the total travel time, the total early arrival time, and the total manufacturing time after increasing the number of nodes. The late arrival times, on the other hand, remained fairly constant and close to zero. The total travel time for the corner PF points with the minimum $\boldsymbol{F}^1$ values was found to be smaller than that of the corner PF points with the minimum $\boldsymbol{F}^2$ values. The latter finding can be explained by the fact that $\boldsymbol{F}^1$ specifically minimizes the total travel cost, whereas $\boldsymbol{F}^2$ minimizes the total early arrival cost, the total late arrival cost, and the total compensation cost.

On the other hand, smaller total early arrival times and total late arrival times were observed at the corner PF points with the minimum $\boldsymbol{F}^2$ values than those of the corner PF points with the minimum $\boldsymbol{F}^1$ values. However, the total manufacturing time remained the same for all the corner PF points, as it was pre-optimized through hybridization at the population initialization step. Hence, the developed HMOEA algo-

**Table 11**
The values of time components at the corner PF points.

| Instance | Travel Time | | Early Arrival Time | | Late Arrival Time | | Manufacturing Time | |
|---|---|---|---|---|---|---|---|---|
| | $min(\boldsymbol{F}^{1})$ | $min(\boldsymbol{F}^{2})$ | $min(\boldsymbol{F}^{1})$ | $min(\boldsymbol{F}^{2})$ | $min(\boldsymbol{F}^{1})$ | $min(\boldsymbol{F}^{2})$ | $min(\boldsymbol{F}^{1})$ | $min(\boldsymbol{F}^{2})$ |
| L-1 | 855.18 | 857.27 | 32,706.04 | 30,066.32 | 0.00 | 0.00 | 232.58 | 232.58 |
| L-2 | 887.80 | 889.45 | 29,250.56 | 27,983.75 | 0.00 | 0.00 | 241.53 | 241.53 |
| L-3 | 919.56 | 923.86 | 37,570.85 | 34,035.92 | 0.00 | 0.00 | 250.37 | 250.37 |
| L-4 | 953.25 | 954.67 | 35,565.33 | 34,404.95 | 0.00 | 0.00 | 259.40 | 259.40 |
| L-5 | 986.89 | 988.93 | 41,207.66 | 39,799.53 | 0.00 | 0.00 | 268.48 | 268.48 |
| L-6 | 1,018.94 | 1,021.32 | 43,214.94 | 39,719.89 | 1.00 | 0.00 | 277.39 | 277.39 |
| L-7 | 1,052.52 | 1,054.29 | 45,473.14 | 39,988.61 | 0.00 | 0.00 | 286.26 | 286.26 |
| L-8 | 1,086.12 | 1,086.94 | 62,750.36 | 51,070.36 | 0.00 | 0.00 | 295.22 | 295.22 |
| L-9 | 1,118.82 | 1,120.04 | 50,965.95 | 44,012.21 | 0.00 | 0.00 | 304.26 | 304.26 |
| L-10 | 1,149.30 | 1,156.87 | 69,288.63 | 39,894.81 | 0.00 | 0.00 | 313.27 | 313.27 |
| L-11 | 1,183.18 | 1,185.35 | 60,863.68 | 54,769.66 | 0.00 | 0.00 | 322.23 | 322.23 |
| L-12 | 1,216.87 | 1,218.67 | 66,452.19 | 63,292.65 | 0.00 | 0.00 | 331.15 | 331.15 |
| L-13 | 1,247.07 | 1,252.58 | 62,670.22 | 59,309.47 | 0.29 | 0.00 | 340.06 | 340.06 |
| L-14 | 1,283.39 | 1,284.75 | 80,913.95 | 76,726.48 | 0.00 | 0.00 | 349.03 | 349.03 |
| L-15 | 1,318.07 | 1,319.83 | 76,587.96 | 75,128.91 | 0.00 | 0.00 | 357.85 | 357.85 |
| **Mean** | **1,085.13** | **1,087.66** | **53,032.10** | **47,346.90** | **0.09** | **0.00** | **295.27** | **295.27** |

rithm would assist decision makers with an effective analysis of trade-offs between conflicting objectives throughout factory-in-a-box supply chain planning (i.e., minimize the total travel time of vehicles vs. minimize the total violation of the previously negotiated time windows at customer locations).

## 7. Conclusions

The frequency and severity of pandemics (e.g., COVID-19) is expected to increase under the existing projections. Pandemics warrant urgent production and distribution of medical supplies under disrupted supply chain conditions. An innovative logistics solution to meet the urgent demand during emergencies could be the factory-in-a-box manufacturing concept. Factory-in-a-box manufacturing could also be helpful to meet the urgent demand during natural disasters or for military applications (e.g., production of military supplies during wars). To obtain extensive flexibility and mobility, this manufacturing concept deploys vehicles to transport containers that are used to install production modules (i.e., factories). The vehicles travel to customer locations and perform on-site production. However, conventional manufacturing could also be useful in some situations due to faster production at manufacturer locations. Furthermore, throughout factory-in-a-box supply chain planning, decision makers may have to compromise conflicting objectives. For example, selection of particular routes may minimize the total travel cost but, in the meantime, cause violation of the previously negotiated time windows at customer locations. However, no study contrasted the options of factory-in-a-box manufacturing with those of conventional manufacturing in multi-objective settings.

To fulfill this gap in the state-of-the-art, this study proposed a novel multi-objective optimization model for the vehicle routing problem with a factory-in-a-box, which captures the options of factory-in-a-box manufacturing and conventional manufacturing for each customer. The objectives of the model were to minimize the total travel cost and to minimize the sum of the total early arrival cost, the total late arrival cost, and the total compensation cost. A customized multi-objective hybrid metaheuristic solution algorithm was developed to solve the model. The algorithm was hybridized with an exact optimization approach to optimize the locations of the final production. On the other hand, various evolutionary operators (i.e., Boltzmann selection, cycle crossover, custom mutation operator, and ranking selection) were employed for route generation. A case study was performed for a vaccination project involving factory-in-a-box manufacturing along with conventional manufacturing. The developed HMOEA metaheuristic algorithm was compared against the ECON method, which is a well-known exact optimization approach for multi-objective optimization problems, and some of the well-known metaheuristic algorithms, including HMOSA, HMOTS, HMOVNS, and HNSGA-II.

A set of numerical experiments revealed that hybridized metaheuristic algorithms consistently outperformed their non-hybridized versions with up to ≈5% objective improvements. The analysis of small-size instances indicated that HMOEA obtained the PFs that were close to the optimal ones produced by ECON. However, ECON was not able to handle large-size problem instances because of the computational complexity of the proposed mathematical model. On the other hand, HMOEA and other developed metaheuristic algorithms demonstrated consistent performance in terms of the computational time not only for small-size problem instances but for large-size problem instances as well. Moreover, based on the numerical experiments conducted for the developed large-size problem instances, HMOEA was found to be superior to HNSGA-II and other metaheuristic algorithms considered, taking into account the shapes of PFs obtained, recorded values of multi-objective optimization performance indicators, and average computational time incurred. Last but not least, a detailed analysis of HMOEA solutions revealed that the proposed algorithm can assist with an effective analysis of trade-offs between conflicting objectives throughout factory-in-a-box supply chain planning (i.e., minimize the total travel time of vehicles vs. minimize the total violation of the previously negotiated time windows at customer locations).

This research may be extended further in several ways. First, determination of raw materials for suppliers, decomposition of sub-assembly, assigning manufacturers to sub-assembly modules, and task-manufacturer assignment could be studied. Second, the developed optimization model was tested in deterministic settings. Various sources of uncertainties could be captured and evaluated as a part of the future research (e.g., uncertainties in vehicle travel times due to roadway closures, traffic congestion, inclement weather conditions, etc.). Third, multiple depots could be considered for more flexibility. Fourth, alternative multi-objective solution methodologies (e.g., hybrid versions of the ECON method, Multi-Objective Social Engineering Optimizer, Multi-Objective Red Deer Algorithm, Strength Pareto Evolutionary Algorithm, Multi-Objective Bacterial Swarm Optimization, Pareto Archived Evolution Strategy, Multi-Objective Keshtel Algorithm) could be developed to solve the proposed model [52–56].

## Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgments

This work has been partially supported by the National Science Foundation grant CMMI-1901109. The opinions, findings, and conclusions, expressed in this publication, are those of the authors and do not necessarily reflect the views of the National Science Foundation.

## References

[1] UNESCO, Pandemics to increase in frequency and severity unless biodiversity loss is addressed, 2020. Accessed on 10/18/2021 from <https://en.unesco.org/news/pandemics-increase-frequency-and-severity-unless-biodiversity-loss-addressed>.

[2] Knowable Magazine, Pandemics in recent history, 2020. Accessed on 10/18/2021 from <https://knowablemagazine.org/article/health-disease/2020/pandemics-recent-history>.

[3] A.M. Anter, D. Oliva, A. Thakare, Z. Zhang, AFCM-LSMA: new intelligent model based on Lévy slime mould algorithm and adaptive fuzzy C-means for identification of COVID-19 infection from chest X-ray images, Adv. Eng. Inf. 49 (2021), 101317.

[4] J. Li, Y. Horiguchi, T. Sawaragi, Counterfactual inference to predict causal knowledge graph for relational transfer learning by assimilating expert knowledge–Relational feature transfer learning algorithm, Adv. Eng. Inf. 51 (2022), 101516.

[5] J. Pasha, M.A. Dulebenets, A.M. Fathollahi-Fard, G. Tian, Y.Y. Lau, P. Singh, B. Liang, An integrated optimization method for tactical-level planning in liner shipping with heterogeneous ship fleet and environmental considerations, Adv. Eng. Inf. 48 (2021), 101299.

[6] WHO, WHO Coronavirus (COVID-19) dashboard, 2022. Accessed on 01/12/2022 from <https://covid19.who.int/>.

[7] GE, KUBio modular biomanufacturing environments, 2021. Accessed on 10/18/2021 from <https://www.gelifesciences.com/en/us/solutions/bioprocessing/products-and-solutions/enterprise-solutions/kubio>.

[8] Nokia, Nokia showcases factory in a box 2.0 with global IoT connectivity, improved reliability and security at Hannover Messe 2019, 2019. Accessed on 10/18/2021 from <https://www.nokia.com/about-us/news/releases/2019/04/01/nokia-showcases-factory-in-a-box-20-with-global-iot-connectivity-improved-reliability-and-security-at-hannover-messe-2019/>.

[9] Z. Jiang, H. Wang, Q. Tian, W. Guo, Co-design of supply chain network and subassembly planning considering the reconfiguration of supply chain structure for factory-in-a-box manufacturing, in: ASME 2018 13th International Manufacturing Science and Engineering Conference, College Station, TX, USA, June 18-22, 2018.

[10] J. Pasha, M.A. Dulebenets, M. Kavoosi, O.F. Abioye, H. Wang, W. Guo, An optimization model and solution algorithms for the vehicle routing problem with a "factory-in-a-box", IEEE Access 8 (2020) 134743–134763.

[11] M. Bengtsson, S.W. Elfving, M. Jackson, The factory-in-a-box concept and its maintenance application, in: 19th International Conference on Condition Monitoring and Diagnostic Engineering Management, Luleå, Sweden, 2006.

[12] M. Hedelind, M. Jackson, P. Funk, J. Stahre, R. Söderberg, J. Carlsson, M. Björkman, M. Winroth, Factory-in-a-box – solutions for availability and mobility of flexible production capacity, in: The Swedish Production Symposium, 2007.

[13] M. Jackson, A. Zaman, Factory-in-a-box – mobile production capacity on demand, Int. J. Modern Eng. 8 (1) (2007) 12–26.

[14] E. Olsson, M. Hedelind, M. Ahmed, P. Funk, Experience reuse between mobile production modules – an enabler for the factory-in-a-box concept, in: The Swedish Production Symposium, Gothenburg, Sweden, 2007.

[15] M. Winroth, M. Jackson, Manufacturing competition through the factory in a box concept, in: POMS 18th Annual Conference, Dallas, TX, USA, May 4-7, 2007.

[16] M. Jackson, M. Wiktorsson, M. Bellgran, Factory-in-a-box – demonstrating the next generation manufacturing provider, in: M. Mitsuishi, K. Ueda, F. Kimura (Eds.), Manufacturing Systems and Technologies for the New Frontier, Springer, London, 2008.

[17] A. Granlund, M. Hedelind, M. Wiktorsson, A. Hällkvist, M. Jackson, Realizing a factory-in-a-box solution in a local manufacturing environment, in: 42nd CIRP Conference on Manufacturing Systems Sustainable Development of Manufacturing Systems, Grenoble, France, June 3-5, 2009.

[18] L. McHauser, C. Schmitz, M. Hammer, Model-factory-in-a-box: a portable solution that brings the complexity of a real factory and all the benefits of experiential-learning environments directly to learners in industry, Procedia Manuf. 45 (2020) 246–252.

[19] K. Braekers, K. Ramaekers, I.V. Nieuwenhuyse, The vehicle routing problem: state of the art classification and review, Comput. Ind. Eng. 99 (2016) 300–313.

[20] R. Elshaer, H. Awad, A taxonomic review of metaheuristic algorithms for solving the vehicle routing problem and its variants, Comput. Ind. Eng. 140 (2020), 106242.

[21] A. Mor, M. Speranza, Vehicle routing problems over time: a survey, 4OR – Q. J. Oper. Res. 18 (2020) 129–149.

[22] J. Brandão, A memory-based iterated local search algorithm for the multi-depot open vehicle routing problem, Eur. J. Oper. Res. 284 (2) (2020) 559–571.

[23] J. Sánchez-Oro, A. López-Sánchez, J. Colmenar, A general variable neighborhood search for solving the multi-objective open vehicle routing problem, J. Heuristics 26 (2020) 423–452.

[24] E. Lalla-Ruiz, M. Mes, Mathematical formulations and improvements for the multi-depot open vehicle routing problem, Optim. Lett. 15 (2021) 271–286.

[25] G. Li, J. Li, An improved tabu search algorithm for the stochastic vehicle routing problem with soft time windows, IEEE Access 8 (2020) 158115–158124.

[26] K. Zhang, Z. Zhang, X. Lin, M. Li, Multi-vehicle routing problems with soft time windows: a multi-agent reinforcement learning approach, Transp. Res. Part C: Emerging Technol. 121 (2020), 102861.

[27] M. Keskin, B. Çatay, G. Laporte, A simulation-based heuristic for the electric vehicle routing problem with time windows and stochastic waiting times at recharging stations, Comput. Oper. Res. 125 (2021), 105060.

[28] B. Pan, Z. Zhang, A. Lim, Multi-trip time-dependent vehicle routing problem with time windows, Eur. J. Oper. Res. 291 (1) (2021) 218–231.

[29] C. Chen, E. Demir, Y. Huang, An adaptive large neighborhood search heuristic for the vehicle routing problem with time windows and delivery robots, Eur. J. Oper. Res. 294 (3) (2021) 1164–1180.

[30] F. Mühlbauer, P. Fontaine, A parallelised large neighbourhood search heuristic for the asymmetric two-echelon vehicle routing problem with swap containers for cargo-bicycles, Eur. J. Oper. Res. 289 (2) (2021) 742–757.

[31] M. Abdirad, K. Krishnan, D. Gupta, A two-stage metaheuristic algorithm for the dynamic vehicle routing problem in industry 4.0 approach, J. Manage. Anal. 8 (1) (2021) 69–83.

[32] P. Stodola, Hybrid ant colony optimization algorithm applied to the multi-depot vehicle routing problem, Nat. Comput. 19 (2020) 463–475.

[33] J. Euchi, A. Sadok, Hybrid genetic-sweep algorithm to solve the vehicle routing problem with drones, Phys. Commun. 44 (2021), 101236.

[34] M. Islam, Y. Gajpal, T. ElMekkawy, Hybrid particle swarm optimization algorithm for solving the clustered vehicle routing problem, Appl. Soft Comput. 110 (2021), 107655.

[35] M. Mojtahedi, A.M. Fathollahi-Fard, R. Tavakkoli-Moghaddam, S. Newton, Sustainable vehicle routing problem for coordinated solid waste management, J. Ind. Inf. Integr. 23 (2021), 100220.

[36] A.E. Eiben, J.E. Smith, Introduction to Evolutionary Computing, Springer-Verlag, Berlin Heidelberg, 2015.

[37] M.A. Dulebenets, An adaptive polyploid memetic algorithm for scheduling trucks at a cross-docking terminal, Inf. Sci. 565 (2021) 390–421.

[38] E. Zitzler, M. Laumanns, S. Bleuler, A tutorial on evolutionary multiobjective optimization, in: X. Gandibleux, M. Sevaux, K. Sörensen, V. T'kindt (Eds.), Metaheuristics for Multiobjective Optimisation. Lecture Notes in Economics and Mathematical Systems, vol. 535, 2004, Springer, Berlin, Heidelberg.

[39] D. Grygar, R. Fabricius, An efficient adjustment of genetic algorithm for pareto front determination, Transp. Res. Procedia 40 (2019) 1335–1342.

[40] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Trans. Evol. Comput. 6 (2) (2002) 182–197.

[41] S. Emde, N. Boysen, Berth allocation in container terminals that service feeder ships and deep-sea vessels, J. Oper. Res. Soc. 67 (4) (2016) 551–563.

[42] J. Cordeau, G. Laporte, P. Legato, L. Moccia, Models and tabu search heuristics for the berth allocation problem, Transp. Sci. 39 (4) (2005) 526–538.

[43] P. Hansen, C. Oguz, N. Mladenovic, Variable neighborhood search for minimum cost berth allocation, Eur. J. Oper. Res. 191 (3) (2008) 636–649.

[44] G. Mavrotas, Effective implementation of the ε-constraint method in multi-objective mathematical programming problems, Appl. Math. Comput. 213 (2) (2009) 455–465.

[45] M.A. Dulebenets, A comprehensive multi-objective optimization model for the vessel scheduling problem in liner shipping, Int. J. Prod. Econ. 196 (2018) 293–318.

[46] A.M. Fathollahi-Fard, M.A. Dulebenets, M. Hajiaghaei-Keshteli, R. Tavakkoli-Moghaddam, M. Safaeian, H. Mirzahosseinian, Two hybrid meta-heuristic algorithms for a dual-channel closed-loop supply chain network design problem in the tire industry under uncertainty, Adv. Eng. Inf. 50 (2021), 101418.

[47] H. Gholizadeh, H. Fazlollahtabar, A.M. Fathollahi-Fard, M.A. Dulebenets, Preventive maintenance for the flexible flowshop scheduling under uncertainty: a waste-to-energy system, Environ. Sci. Pollut. Res. (2021) 1–20.

[48] O. Theophilus, M.A. Dulebenets, J. Pasha, Y.Y. Lau, A.M. Fathollahi-Fard, A. Mazaheri, Truck scheduling optimization at a cold-chain cross-docking terminal with product perishability considerations, Comput. Ind. Eng. 156 (2021), 107240.

[49] A. Amini, R. Tavakkoli-Moghaddam, A bi-objective truck scheduling problem in a cross-docking center with probability of breakdown for trucks, Comput. Ind. Eng. 96 (2016) 180–191.

[50] A. Goodarzi, N. Nahavandi, S. Zegordi, A multi-objective imperialist competitive algorithm for vehicle routing problem in crossdocking networks with time windows, J. Ind. Syst. Eng. 11 (1) (2018) 1–23.

[51] M. Mohammadi, F. Jolai, R. Tavakkoli-Moghaddam, Solving a new stochastic multi-mode phub covering location problem considering risk by a novel multi-objective algorithm, Appl. Math. Model. 37 (24) (2013) 10053–10073.

[52] L. Caldas, Generation of energy-efficient architecture solutions applying GENE_ARCH: an evolution-based generative design system, Adv. Eng. Inf. 22 (1) (2008) 59–70.

[53] M.A. Dulebenets, Multi-objective collaborative agreements amongst shipping lines and marine terminal operators for sustainable and environmental-friendly ship schedule design, J. Cleaner Prod. 342 (2022), 130897.

[54] M.D. Shieh, Y. Li, C.C. Yang, Comparison of multi-objective evolutionary algorithms in hybrid Kansei engineering system for product form design, Adv. Eng. Inf. 36 (2018) 31–42.

[55] Z. Jiang, Y. Chen, X. Li, B. Li, A heuristic optimization approach for multi-vehicle and one-cargo green transportation scheduling in shipbuilding, Adv. Eng. Inf. 49 (2021), 101306.

[56] X. Gao, X. Jin, P. Zheng, C. Cui, Multi-modal transportation planning for multi-commodity rebalancing under uncertainty in humanitarian logistics, Adv. Eng. Inf. 47 (2021), 101223.