

A BI-OBJECTIVE PRODUCTION-DISTRIBUTION PROBLEM IN A SUPPLY CHAIN NETWORK UNDER GREY FLEXIBLE CONDITIONS

FARIBA GOODARZIAN¹, DAVOOD SHISHEBORI^{1,*}, HADI NASSERI^{2,3} AND FARIDREZA DADVAR⁴

Abstract. One of the main topics discussed in a supply chain is the production-distribution problem. Producing and distributing the products plays a key role in reducing the costs of the chain. To design a supply chain, a network of efficient management and production-distribution decisions is essential. Accordingly, providing an appropriate mathematical model for such problems can be helpful in designing and managing supply chain networks. Mathematical formulations must be drawn close to the real world due to the importance of supply chain networks. This makes those formulations more complicated. In this study, a novel multi-objective formulation is devised for the production-distribution problem of a supply chain that consists of several suppliers, manufacturers, distributors, and different customers. Also, a Mixed Integer Linear Programming (MILP) mathematical model is proposed for designing a multi-objective and multi-period supply chain network. In addition, grey flexible linear programming (GFLP) is done for a multi-objective production-distribution problem in a supply chain network. The network is designed for the first time to cope with the uncertain nature of costs, demands, and capacity parameters. In this regard, due to the NP-hardness and complexity of problems and the necessity of using meta-heuristic algorithms, NSGA-II and Fast PGA algorithm are applied and compared in terms of several criteria that emphasize the quality and diversity of the solutions.

Received June 29, 2020. Accepted January 12, 2021.

1. INTRODUCTION

A supply chain includes a set of suppliers, manufacturers, distribution centers, and transfer channels. Each member plays a distinct role in manufacturing final products from raw materials according to the needs of the consumer [17,28,36]. In recent years, the globalization of trade, competition, and the integration of supply chains (SCs) have made organizations pay more attention to their production plans and the other related members in the SC [11,15,17,28]. Also, providing a production plan for the SC of an organization is one of the most significant decisions to make in the SC management. Therefore, supply chain management (SCM) should be able to plan

Keywords. Supply chain network design, production-distribution problem, grey flexible programming, meta-heuristic algorithms.

¹ Department of Industrial Engineering, Yazd University, Yazd, Iran.

² Department of Mathematics and Big Data, Foshan University, Foshan, P.R. China.

³ Department of Mathematics, Faculty of Mathematical Sciences, University of Mazandaran, Babolsar, Iran.

⁴ Department of Industrial Engineering, K. N. Toosi University of Technology, Tehran, Iran.

*Corresponding author: shishebori@yazd.ac.ir

all the activities involved in the chain from the suppliers to the final consumers; inappropriate management of SC can lead to the bankruptcy of the members and failure in global competition [16, 20, 22, 28, 53]. SCM is a key process through which competitive ability in the market is increased. Besides, managers can reduce the costs imposed on their organizations by practicing SCM. It provides a balance among the supplier, the manufacturing or service organization and the consumer(s) and ultimately guarantees the survival of the organization in the market [11, 22, 27, 37, 40, 49].

Therefore, SCM is a set of methods used to coordinate suppliers, manufacturers, warehouses, and retailers effectively so as to deliver products to customers at a specified time and place, minimize the costs of the whole SC, and keep customer requirements at high service levels [12, 22, 34, 37, 40]. The basic goal of supply chain network (SCN) problems is to plan a production and distribution model. The production planning area is the decisions which a producer should make in the case of the ordered product and its time and number to meet the customer's need [1]. The distribution planning area includes decisions to find a channel for delivering goods from a manufacturer to a distributor/customer [4, 12, 23]. These problems are interdependent, so they should be integrated to minimize the cost and to maximize the profit in the chain [25, 54].

Furthermore, in an SCN, the production and distribution problem is an optimization problem, and solving it by identifying the best model can be very cost-effective but time-consuming. To solve large-scale complex SCN models, meta-heuristic algorithms have been developed to offer approximated (generally good) solutions. These solutions apply to SC and logistics problems. Many researchers have studied the design and modeling of various components for a production-distribution problem.

Liu and Papageorgiou [31] presented a two-level production-distribution planning problem by paying simultaneous attention to the cost, response, and customer service levels. They used the ε -constraint and lexicography to solve the model. Cardona-Valdés *et al.* [9] designed a two-level production-distribution network with production factories, distribution zones and several warehouses. Their significant innovation is a solution provided to a two-tier random problem using the banned search algorithm along with the multi-objective adaptive memory programming framework. To solve the problem in small dimensions, the ε -constraint and the branch and leaf technique are used. Nazim *et al.* [44] proposed a multi-objective production-distribution under randomized fuzzy conditions. A formula was developed based on the purpose of the problem. To solve a multi-objective problem, a total weighting method was designed based on the genetic algorithm. Khalifehzadeh *et al.* [26] studied a multi-dimensional production-distribution problem. Their objectives included minimizing the total cost and maximizing the reliability of each transport system. To solve a large-scale problem, a new heuristic algorithm was developed. Rafiei *et al.* [47] developed an integrated production-distribution planning problem in a four-echelon SC with two main objectives including maximizing the service level and minimizing the total cost. Alavidoost *et al.* [3] developed a multi-objective MINLP model for multi-commodity tri-echelon SCNs. Then, some meta-heuristic algorithms were applied to solve the model. Nourifar *et al.* [45] proposed a production-distribution planning problem in a multi-period SCN with stochastic and fuzzy parameters. Then, a bi-level MILP model was formulated. Fakhrzad *et al.* [14] proposed a new production-distribution problem for SCN under uncertainty. Then, a multi-objective MILP model was formulated. Also, a NSGA-II algorithm was used to solve the model. Sakalli and Atabas [51] proposed a new multi-site integrated production-distribution plan in a fuzzy multi-objective optimization SC. In this regard, a multi-period and multi-product MILP model was formulated. The objective functions included the minimization of delivery time, total cost. Backorder levels were also taken into consideration. Rafie-Majd *et al.* [48] proposed a genetic algorithm (GA) and ant colony optimization for fuzzy stochastic production and distribution planning problems. Zhao and Dou [64] proposed a multi-objective integrated SC design for a single-product and four-echelon problem. Also, a MILP model was formulated. Then, a novel meta-heuristic algorithm called Multi-Objective Modified Particle Swarm Optimization (MOMPSO) was developed. Mohammadi *et al.* [38] developed a new distribution network by an adaptive multi-objective optimization method. Additionally, a new Heat Transfer Search (HTS) algorithm was used. Badhotiya *et al.* [6] developed a three-echelon SCN for an integrated inventory-location-routing problem. Also, the Lagrangian relaxation approach was used to solve the proposed problem. Then, a heuristic algorithm was developed to improve the feasibility of more solutions obtained from the Lagrangian relaxation algorithm. Billal

and Hossain [7] offered a new multi-objective optimization plan for multi-period multi-product four-echelon SC problems under uncertainty. The multi-objective GA, NSGA-II, and ε -constraint methods were applied in this case. Goodarzian and Hosseini-Nasab [19] devised a new fuzzy bi-objective mathematical formula for a four-echelon production and distribution problem in an SCN under uncertain conditions. Then, a novel self-adoptive evolutionary algorithm was worked out to solve the formula. Ghahremani-Nahr *et al.* [18] conducted robust fuzzy formulation for an SCN. They used a new Whale Optimization Algorithm (WOA) to solve the model on a large scale with the objective of minimizing the total cost. Fakhrazad and Goodarzian [13] proposed a new fuzzy multi-objective SCN in an uncertain environment. Then, they used a new Modification of Imperialist Competitive Algorithm (MICA) to solve the model in a large-size pattern. Zaidan *et al.* [62] applied a novel hybrid algorithm of simplex downhill and simulated annealing to solve a multi-objective linear programming aggregate production problem under fuzzy conditions.

There are several methods for solving multidisciplinary problems. They are categorized into numerical methods and Pareto methods. Using single-objective problems, numerical methods transform multi-objective problems via mathematical transformations. Pareto approaches use the concept of dominant sets to find Pareto's solutions. In practical situations, one may face large-scale problems which are difficult to resolve with exact and time-consuming methods. An important optimal solution to large-scale production-distribution problems in chain networks is the use of heuristic and meta-heuristic algorithms. Accordingly, in this paper, the NSGA-II and Fast-PGA algorithms have been utilized to solve large-scale problems whose results are then compared in terms of a series of indicators that emphasize the quality and development of the solutions. Thus, the important contributions of the current study can be summarized as follows:

- Formulating a new multi-product, multi-period, and multi-echelon production-distribution SCN problem as a MILP model.
- Achieving an applicable and effective Grey flexible modeling method for the developed SCN problem, considering various parameters of uncertainty as flexible constraints, Grey coefficients, and Grey purposes of the decision-maker(s).
- Considering cost and reliability objective functions as problem assessment criteria.
- Using multi-objective meta-heuristic algorithms to solve the model in large-scale problems.

The rest of the present paper is organized as follows. In Section 2, the programming model of the problem and the grey model are described. In Section 3, the proposed solution algorithms as well as the criteria for evaluating and comparing the methods are discussed. In Section 4, the computational results are analyzed. Finally, in Section 5, the conclusion and the suggestions for future research are presented.

2. DEFINITION AND FORMULATION OF THE PROBLEM

An SCN includes four levels of suppliers, manufacturers, distributors, and customers. Decisions are made for different products over several periods. The consumers are at the first level, while the second level is given to the distributors who transship a number of products to the customers. At the third level, there are the manufacturers which offer the products to the distributors. At the fourth level, the suppliers provide materials for the producers. The met demand of each customer for each product is assumed for each time period by re-ordering. However, the total customer's demand should be met in the last period. In this formulation, many sorts of fixed reliability rates are taken into account. The reliability value of every transport system in every route is achieved from the multiplication of the rate of every transport system by the rate of every route. On this basis, additional assumptions about the problem are given below. A quantity of each product can be generated by a producer over a time period. Every transport system can move many times in a given route at any time. Every transport system also has a limited capacity. The structure of this SCN is shown in Figure 1 and the general assumptions are as follows:

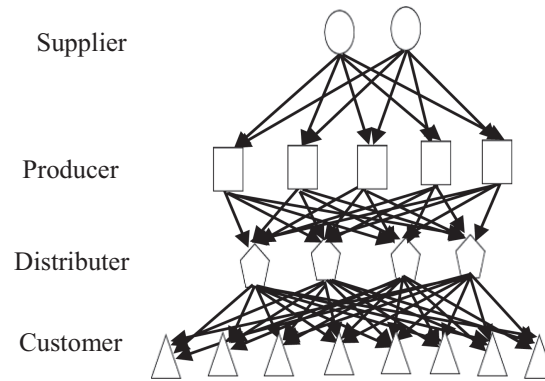


FIGURE 1. The structure of a four-level SCN.

- There are several suppliers, manufacturers, distributors, customers, products, periods, and transport systems.
- The costs of raw materials purchasing, products manufacturing, transport, maintenance, the shortage of each product, setup, and time of processing are taken into consideration.
- There are the capacity of raw materials and the rate of using them for product manufacturing, the available storage capacity of each manufacturer, distribution center and transport system, and the capacity of each producer.
- The amount of the customer demand in each time period, the available time for each manufacturer in each time period, and the reliability of each transport system are also taken into consideration.

Set of indices

- s Supplier ($s = 1, 2, 3, \dots, S$).
- p Producer ($p = 1, 2, 3, \dots, P$).
- k Distributor ($k = 1, 2, 3, \dots, K$).
- i Customer ($i = 1, 2, 3, \dots, I$).
- r Products ($r = 1, 2, 3, \dots, R$).
- t Time period ($t = 1, 2, 3, \dots, T$).
- v Transport system ($v = 1, 2, 3, \dots, V$).

Parameters

- rp_{st} Raw material cost which is prepared by supplier s at the t .
- pc_{rpt} Producing cost of products r from producer p at the t .
- pcr_{st} Producing cost of raw materials by supplier s at the t .
- st_{rp} Setup time for product r in producer p .
- k_{rp} The processing time of product r in producer p .
- \max_{rpt} Maximum available time of product r in producer p at the t .
- fts_{sp}^v The fixed sending cost of transport system v from supplier s to producer p .
- $vt s_{sp}^v$ Transport cost of raw materials from supplier s to producer p by using transport system v .
- $ft p_{pk}^v$ The fixed sending cost of transport system v from producer p to distributor k .
- $vt p_{pk}^v$ Transport cost of product r from producer p to distributor k by using transport system v .
- $ft d_{ki}^v$ The fixed cost of transporting transport system v from distributor k to customer i .
- $vt d_{ki}^v$ Sending cost of product r from distributor k to customer i by using transport system v .
- vp_r The amount of each product r .
- V_r The amount of raw material.

- ur_r The rate of using raw materials for producing product r .
- hp_p^G Holding cost of raw materials in producer p .
- hp_{rk}^G Holding cost of product r in distributor k .
- dem_{irt} Customer demand i for product r at the t .
- sc_{rp}^G Setup cost for a set of products r in producer p .
- scs_s^G Setup cost for a set of raw materials by supplier s .
- shc_{ri} Cost of shortage of product r for customer i .
- $capd_p^G$ Capacity of warehouse for producer p .
- $capd_k^G$ Capacity of warehouse for distributor k .
- $capd_v^G$ Capacity of each transport system v .
- ras_{sp}^v Reliability ratio of transport system v from route s to p .
- raj_{pk}^v Reliability ratio of transport system v from route p to k .
- rad_{ki}^v Reliability ratio of transport system v from route k to i .
- G A big positive number.

Decision variables

- L_{rpt} The amount of product r produced in producer p at the t .
- LN_{st} The amount of raw materials produced by supplier s at the t .
- TRS_{spt}^v The amount of raw materials sent by using transport system v from supplier s to producer p at the t .
- TPP_{rpkt}^v The amount of product r sent by using transport system v from producer p to distributor k at the t .
- TPD_{rkit}^v The amount of product r sent by using transport system v from distributor k to customer i at the t .
- Z_{pt} The raw material inventory in producer p at the end of the t .
- W_{rkt} The inventory of product r in distributor k at the end of the t .
- M_{irt} The quality of the re-order of customer i of product r at the end of the t .
- NS_{spt}^v The quality of the re-order of supplier s of producer p by transport system v at the end of the t .
- NP_{pkt}^v The quality of the movement of transport systems v from producer p to distributor k at the t .
- NP_{kit}^v The quality of the movement of transport systems v from distributor k to customer i at the t .
- Q_{rpt} If producer p produces products r during the t , it is equal to 1, otherwise 0.
- Y_{st} If supplier s of the raw materials is provided at the t , it is equal to 1, otherwise 0.

$$\begin{aligned}
 \text{Objective 1 : } \min & \sum_s \sum_t (pcr_{st} \cdot N_{st} + scs_s^G \cdot Y_{st}) + \sum_v \sum_s \sum_p \sum_t (fts_{sp}^v \cdot NS_{spt}^v) \\
 & + \sum_v \sum_s \sum_p \sum_t (vts_{sp}^v \cdot TRS_{spt}^v) + \sum_v \sum_s \sum_p \sum_t (rp_{sp}^v \cdot TRS_{spt}^v) \\
 & + \sum_r \sum_p \sum_t (pcr_{rpt} \cdot L_{rpt} + sc_{rp}^G \cdot Q_{rpt}) + \sum_v \sum_p \sum_k \sum_t (ftp_{pk}^v \cdot NP_{pks}^v) \\
 & + \sum_v \sum_r \sum_p \sum_k \sum_t (vtp_{pk}^v \cdot TPP_{rpkt}^v) + \sum_p \sum_t (hp_p^G \cdot Z_{pt}) + \sum_v \sum_k \sum_i \sum_t (ftd_{ki}^v \cdot ND_{kit}^v) \\
 & + \sum_v \sum_r \sum_k \sum_i \sum_t (vtd_{rki}^v \cdot TPD_{rkit}^v) + \sum_r \sum_k \sum_t (hc_{rk}^G \cdot W_{rkt}) + \sum_r \sum_i \sum_t (shc_{ri} \cdot M_{rit})
 \end{aligned} \tag{2.1}$$

$$\begin{aligned}
 \text{Objective 2 : } \text{Max} & \sum_v \sum_r \sum_s \sum_p \sum_t (ras_{sp}^v \cdot TRS_{spt}^v) \\
 & + \sum_v \sum_r \sum_p \sum_k \sum_t (raj_{pk}^v \cdot TPP_{rpkt}^v)
 \end{aligned}$$

$$+ \sum_v \sum_r \sum_k \sum_i \sum_t (rad_{ki}^v \cdot TPD_{rkit}^v) \tag{2.2}$$

s.t.

$$\sum_r (st_{rp} + k_{rp} \cdot L_{rpt}) \cdot Q_{rpt} \leq \max_{rpt} \quad \forall p \cdot t \tag{2.3}$$

$$\sum_v \sum_p TPS_{spt}^v = N_{st} \quad \forall s \cdot t \tag{2.4}$$

$$\sum_v \sum_k TPP_{rpkt}^v = L_{rpt} \quad \forall r \cdot p \cdot t \tag{2.5}$$

$$Z_{p(t-1)} + \sum_v \sum_s TRS_{spt}^v - Z_{pt} - \sum_r ur_r \cdot L_{rpt} = 0 \quad \forall p \cdot t \tag{2.6}$$

$$W_{rk(t-1)} + \sum_v \sum_p TPP_{rpkt}^v - W_{rkt} - \sum_v \sum_i TPD_{rkit}^v = 0 \quad \forall r \cdot k \cdot t \tag{2.7}$$

$$Z_{pt} \cdot vr \leq capp_p^G \quad \forall p \cdot t \tag{2.8}$$

$$\sum_r W_{rkt} \cdot vp_r \leq capd_r^G \quad \forall k \cdot t \tag{2.9}$$

$$N_{st} \leq G \cdot Y_{pt} \quad \forall s \cdot p \cdot t \tag{2.10}$$

$$L_{rpt} \leq G \cdot Q_{rpt} \quad \forall r \cdot p \cdot t \tag{2.11}$$

$$vr \cdot TRS_{spt}^v \leq capd_v^G \cdot NS_{spt}^v \quad \forall v \cdot s \cdot p \cdot t \tag{2.12}$$

$$\sum_r (vp_r \cdot TPP_{rpkt}^v) \leq capd_v^G \cdot NP_{rkt}^v \quad \forall v \cdot p \cdot k \cdot t \tag{2.13}$$

$$\sum_r (vp_r \cdot TPD_{rkit}^v) \leq capd_v^G \cdot ND_{kit}^v \quad \forall v \cdot p, k \cdot i \cdot t \tag{2.14}$$

$$M_{rit} - M_{ri(t-1)} - dem_{rit}^G + \sum_v \sum_k (TPD_{rkit}^v) = 0 \quad \forall r \cdot i \cdot t \tag{2.15}$$

$$\sum_v \sum_k (TPD_{rkit}^v) - M_{ri(t-1)} = dem_{rit}^G \quad \forall r \cdot i \cdot t \tag{2.16}$$

$$L_{rpt} \cdot N_{st} \cdot TRS_{spt}^v \cdot TPP_{rpkt}^v \cdot TPD_{rkit}^v \cdot Z_{pt} \cdot W_{rkt} \cdot M_{rit} \geq 0. \\ NS_{spt}^v \cdot NP_{pkt}^v \cdot ND_{kit}^v \in N \quad Q_{rpt} \cdot Y_{st} \in \{0.1\}. \tag{2.17}$$

Objective (2.1) refers to the minimization of the total operating costs from suppliers, including delivery costs from suppliers to manufacturers, product costs including the purchase of primary materials, production of products, sales costs of products to distributors and maintenance costs from primary materials as well as total distributor and customer costs including costs from distribution to customers, inventory shortage costs and maintenance costs in customer locations. Objective (2.2) refers to the maximization of the reliability of supplying primary materials from suppliers to producers and the reliability of supplying products from suppliers to distributors and then to customers.

Constraint (2.3) indicates the total set-up and processing times to produce products, which are less than the maximum accessible time. Constraint (2.4) represents all the primary materials produced by each supplier which must be shipped to the producer within a similar time. Constraint (2.5) indicates all the products produced per manufacturer which must be sent from the distributor within the same time. Constraints (2.6) and (2.7) present the equilibrium equations of the inventory in manufacturers and distributors. Constraints (2.8) and (2.9) specify the remaining inventory control at every producer and distributor at the end of every period. Constraint (2.10) shows the supplier of raw materials at any given time. Constraint (2.11) emphasizes the producer of products manufactured at any given time. Constraints (2.12)–(2.14) indicate the capacity of the

transport system. Constraint (2.15) shows the equilibrium equation of the shortage in every customer’s location. Constraint (2.16) emphasizes the equilibrium equation of the shortage before the time period. Constraint (2.17) represents the state of decision variables.

2.1. Description of gray problem

This section presents some definitions and comments which are very useful for the further study of the GS theory and, in particular, for the calculation of grey numbers as a tool to solve GFLP problems.

2.2. Grey numbers and grey system

The Grey theory is a convenient tool, as the fuzzy sets theory is, with which to interpret the uncertainty of parameters. It can improve the mathematical analysis of systems in an uncertain environment. The rest of the details are represented in the Appendix A.

2.3. Grey flexible model

A grey linear optimization model in the form of GFLP is given to solve.

$$\begin{aligned}
 \text{Objective 1 : } \min & \sum_s \sum_t (pcr_{st} \cdot SR_{st} + scs_s^G \cdot Y_{st}) + \sum_m \sum_s \sum_j \sum_t (fts_{sj}^m \cdot NS_{sjt}^m) \\
 & + \sum_m \sum_s \sum_j \sum_t (vts_{sj}^m \cdot TRS_{sjt}^m) + \sum_m \sum_s \sum_j \sum_t (rp_{st} \cdot TRS_{sjt}^m) \\
 & + \sum_p \sum_j \sum_t (pc_{pjt} \cdot SP_{pjt} + sc_{pj}^G \cdot X_{pjt}) + \sum_m \sum_j \sum_d \sum_t (ftp_{jd}^m \cdot NP_{jds}^m) \\
 & + \sum_m \sum_p \sum_j \sum_d \sum_t (vtp_{jd}^m \cdot TPP_{pjdt}^m) + \sum_j \sum_t (hp_j^G \cdot IP_{jt}) \\
 & + \sum_m \sum_d \sum_c \sum_t (ftd_{dc}^m \cdot ND_{dct}^m) + \sum_m \sum_p \sum_d \sum_c \sum_t (vtd_{pdc}^m \cdot TPD_{pdct}^m) \\
 & + \sum_p \sum_d \sum_t (hc_{pd}^G \cdot ID_{pdt}) + \sum_p \sum_c \sum_t (shc_{pc} \cdot SH_{pct}) \tag{2.18}
 \end{aligned}$$

$$\begin{aligned}
 \text{Objective 2 : } \text{Max} & \sum_{\forall m} \sum_{\forall p} \sum_{\forall s} \sum_{\forall j} \sum_{\forall t} (ras_{sj}^m \cdot TRS_{sjt}^m) \\
 & + \sum_{\forall m} \sum_{\forall p} \sum_{\forall j} \sum_{\forall d} \sum_{\forall t} (raj_{jd}^m \cdot TPP_{pjdt}^m) \\
 & + \sum_{\forall m} \sum_{\forall p} \sum_{\forall d} \sum_{\forall c} \sum_{\forall t} (rad_{dc}^m \cdot TPD_{pdct}^m) \tag{2.19}
 \end{aligned}$$

s.t.

Constraints (2.3)–(2.7).

$$IP_{jt} \cdot vr \leq capp_j^G + p_i(1 - \alpha_i) \quad \forall j \cdot t \tag{2.20}$$

$$\sum_p ID_{pdt} \cdot vp \leq capd_d^G + p_i(1 - \alpha_i) \quad \forall d \cdot t. \tag{2.21}$$

$$\text{Constraints (2.10) and (2.11).} \tag{2.22}$$

$$vr \cdot TRS_{sjt}^m \leq (capt_m^G + p_i(1 - \alpha_i)) \cdot NS_{sjt}^m \quad \forall m \cdot i \cdot j \cdot s \cdot t \tag{2.23}$$

$$\sum_p (v_p \cdot \text{TPP}_{pdct}^m) \leq (\text{capt}_m^G + p_i (1 - \alpha_i)) \cdot \text{NP}_{dct}^m \quad \forall m \cdot j \cdot d \cdot t \quad (2.24)$$

$$\sum_p (v_p \cdot \text{TPD}_{pdct}^m) \leq (\text{capt}_m^G + p_i (1 - \alpha_i)) \cdot \text{ND}_{dct}^m \quad \forall m \cdot d \cdot c \cdot t \quad (2.25)$$

$$\text{SH}_{pct} - \text{SH}_{pc(t-1)} - (\text{dem}_{pct}^G + p_i (1 - \alpha_i)) + \sum_m \sum_d (\text{TPD}_{pdct}^m) = 0 \quad \forall p \cdot c \cdot t \quad (2.26)$$

$$\sum_m \sum_d (\text{TPD}_{pdct}^m) - \text{SH}_{pc(t-1)} = \text{dem}_{pct}^G + p_i (1 - \alpha_i) \quad \forall p \cdot c \cdot t. \quad (2.27)$$

Constraint (2.17).

3. SOLUTION METHODS

It has been proven that supply chain network models are NP-hard. Hence, the literature has seen several meta-heuristics that were ordered to solve these NP-hard problems [21, 55]. Besides, No Free Lunch theory says that there is no meta-heuristic to show a good performance for *all* optimization problems (Wolpert and Macready 1997). Accordingly, the recent decade has seen a rapid development of meta-heuristic methods [21]. Multi-objective problem solutions are divided into classical and evolutionary types. Classical methods are weak in that they provide only one optimal solution in each step and do not find all the optimal solutions in a multi-objective optimization process. To overcome this weak point, researchers have used evolutionary techniques that can find several optimal solutions in one run. Genetic algorithms make up a family of meta-heuristic algorithms based on random searches and serve as a powerful tool for solving large-scale optimization problems. Among the genetic algorithms in use, NSGA-II has high efficiency due to low computational complexity and the use of the congestion distance operator in solving multi-objective problems. Also, studies have shown that, for large-scale problems, the Fast PGA method performs better than the NSGA-II algorithm to solve the supply chain network model [14].

3.1. NSGA-II algorithm

Intelligent and evolutionary methods, unlike numerical processing methods, make it possible for the one-time solving of multi-objective optimization problems. In the case of multi-objective problems, since there is no possibility of optimizing a single solution for all the purposes simultaneously, the algorithm that offers some solutions on Pareto or near Pareto has a high practical value. In fact, sorting the solutions through multi-objective optimization is problematic. This problem can be solved with the NSGA algorithm. The algorithm converts a multi-objective optimization space, which is not an irreplaceable space, into a sortable space. The second version of the NSGA algorithm (NSGA-II) is also presented. In addition to considering the quality of solutions, NSGA-II addresses the diversity and variety of Pareto's optimal solutions too [46, 50].

The NSGA-II algorithm has two known phases. The first phase is related to the quality of solutions, and the second considers their order. In the first phase, the ranking of solutions is done, which is the common feature of different fronts. To this end, two parameters are determined, including (a) the number of times that a solution is dominated and (b) the set of solutions that the current solution overcomes. In this regard, the two parameters should be compared for all the solutions. If the number of times that the solutions are dominated is zero, they are non-dominated solutions approximate to the Pareto's front. Accordingly, they are called first front (F_1). In order to identify the solutions of the second front, at first, the number of times that all the solutions are lost is reduced, and then the number of times that the solutions have fallen to zero as the second front is taken into account. This process is repeated until all the fronts are identified (Fig. 2).

In the second phase, the distance of swarming criterion is used to represent the distance among all the solutions at the same level. If the swarming distance among the selective points (in under-populated areas) is large, these points contribute to the diversity. When comparing two different solutions, one may face two modes;

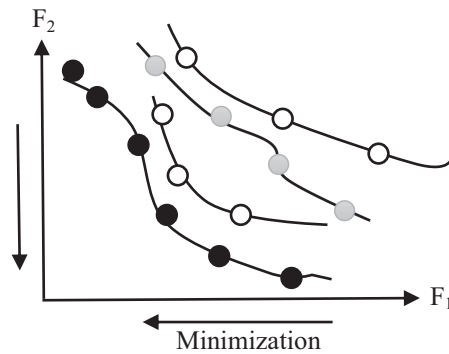


FIGURE 2. Different ranks of non-dominated solutions.

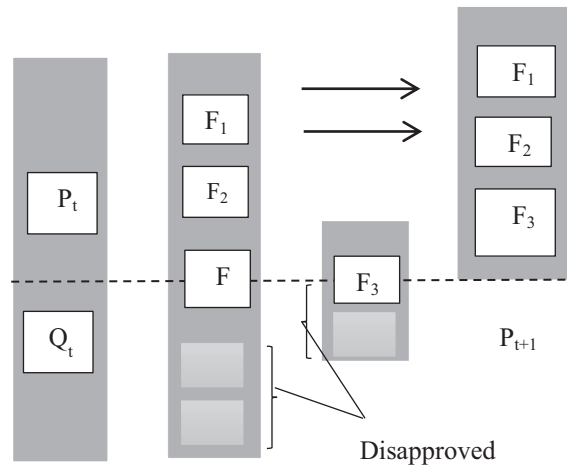


FIGURE 3. The overall mechanism of the NSGA-II performance in the next generation.

(a) of two solutions with different ranks, the superior one has the lower rank and (b) if two solutions belong to the same front, the solution with a greater distance is chosen.

If P_t is the population of the current generation and Q_t is the population of the children created by crossover and mutation operators, for the next generation or P_{t+1} , the P_t and Q_t solutions are first merged together. Then, this aggregate population undergoes ranking by means of the ranking operator. It is followed by the sorting of the distance from the crowd. Finally, the population size of the first-order solutions is transmitted directly to the next generation, and the remaining solutions are eliminated. In fact, the algorithm makes a balance between quality and order with respect to the high importance which it gives to solutions and the less relative importance assigned to distance [46, 63]. Figure 3 presents the general diagram for the next generation production in the NSGA-II algorithm.

3.2. Fast PGA algorithm

This algorithm is presented to simultaneously optimize several goals. It is cost-effective in terms of computational and financial considerations. The algorithm makes use of new genetic operators to improve the performance in terms of ranking, convergence and computation. It is to be noted that quick convergence is important in solving multi-objective optimization problems that have a large solution space. The purpose of

this algorithm is to find an optimum Pareto solution that provides suitable distribution and breadth in the solution space and is reasonably computational and cost-effective [33].

3.2.1. Rating of solutions and allocation of merits to them

The new ranking strategy introduced in this algorithm assigns two different ranks to the candidate's solutions based on the overcoming approach. First, all non-dominated solutions are viewed as the first-rank solutions, but dominated ones are recognized as the second-rank solutions. Then, merit values are allocated to the first-rank solutions through the comparison of those solutions using the swarm distance of coupling. In addition, every dominated solution in the second category is compared to all the other solutions (including non-dominated and dominated ones) in the population. The corresponding merit values are based on the number of the dominated solutions. Here, for each solution x_i in the population, a weak power value $S(x_i)$ is allocated, which indicates the number of the dominated solutions.

$$S(X_i) = |\{x_j \forall x_j \in CP_t \wedge x_i > x_j \wedge j \neq i\}|. \quad (3.1)$$

The cardinal of a set is shown by $|\cdot|$. CP_t is the set of solutions for generation t , and the expression $x_i > x_j$ means that the solution x_i dominates solution x_j . The merit value of each dominated solution is calculated by the following equation:

$$F(x_i) = \sum_{x_i > x_j} S(x_j) - \sum_{x_k > x_i} S(x_k) \forall x_j \cdot x_k \in CP_t \wedge j \neq i \neq k. \quad (3.2)$$

In other words, the merit value of every dominated solution x_i is equal to the difference between the sum of the power values of all the solutions that make up the dominated x_i and the sum of the power values of all the solutions that dominate x_i . Practicing this approach involves only a few computations, because no mechanism is used; maintaining other varieties among non-dominated solutions needs few computations.

Once the merit values of all the population solutions are calculated and compared, three scenarios should be noted as follows:

- (a) Two solutions are compared with different ranks. The solution to choose is the one with a better rank.
- (b) There are two equal-rank solutions, but there is a difference between the amounts of merit. In this case, the solution to choose is the one with a higher merit value.
- (c) There are two solutions with the same rank and merit. In this case, one of them is chosen randomly.

3.2.2. Elitism and population adjustment

In this section, the population of the previous generation and that of the offspring are combined. This provides the opportunity to maintain the superior solutions in the next generations and to ignore the weak ones. If the population is too large and is constant over different generations, it leads to a reduction in elitism in the early generations. Also, the existence of fluctuation in the number of non-dominated solutions over different generations requires an adaptive population size strategy to emphasize the intensity of elitism in non-dominated solutions. If the intensity of elitism is very low, convergence may be too late. This imposes a high computational cost. On the other hand, if the intensity of elitism is very high, early convergence may occur. Therefore, this algorithm dynamically uses a modifier operator to set the population size until it reaches a predetermined value. The modifier operator uses the following equation:

$$|P_t| = \min\{a_t + [b_t * |\{x_i | x_i \in CP_t \wedge x_i \text{ is nondominated} \cdot maxpopsize\}]\} \quad (3.3)$$

where $|P_t|$ is the population size in generation t , a_t and b_t are the positive and the integer positive values respectively, which may vary over different generations, x_i is the smallest integer greater than or equal to the real value of x , and *maxpopsize* represents the largest preset population size.

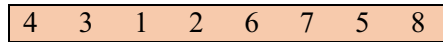


FIGURE 4. An example of how to represent the string of a chromosome.

The Fast PGA algorithm, unlike many other evolutionary optimization methods, has the advantage of using the size of the offspring generated by crossover and mutation operators dynamically. The size of the offspring in each generation is calculated as follows:

$$|O_t| = \min\{c_t + [d_t * |\{x_i | x_i \in CP_t \wedge x_i \text{ is nondominated} \cdot \text{maxsolved}\}]\} \quad (3.4)$$

where $|O_t|$ is the size of the population of the offspring generated in generation t , c_t and d_t are the integer positive and the true positive values respectively, and *maxsolved* represents the maximum number of the evaluated solutions in each generation. This dynamic Fast PGA algorithm allows the saving of a significant number of solutions at the beginning of the search and exploits extractions efficiently in later generations [33].

3.3. The used operators in the algorithm

3.3.1. How to display the solution?

This section presents several execution modes for each activity in the problem. Of these modes, only one is to be run. So, there is a need to display how a mode can be run for an activity. The most appropriate type of display for the category of integer problems involves a chromosome as a set of integer values (genes) that represent the selected execution mode for a given activity of a problem. Figure 4 displays a feasible solution (as a string of a chromosome) for a problem with eight activities.

3.3.2. Calculation of objective functions

This section is dedicated to the comparison of different solutions with one another and ranking of them according to the calculated values for the objective functions. In fact, as the dominance rule suggests, if none of the cost functions is worse than the others and at least one of the objectives is better than the others, one solution can be considered superior to another one.

3.3.3. Crossover operator

The crossover operator depends on the type of display, and, for different displays, different operators must be defined. Here, according to the type of display, the uniform crossover is used. It is the crossover operator related to integer representations [33, 35, 46, 50, 63]. In a uniform crossover, sequences of random variables are generated from the uniform distribution in $[0, 1]$ multiplied by the number of chromosome genes. In each position, if the value is less than parameter P_c , the gene will be generated from the first parent; otherwise, it will be inherited from the second parent. The second offspring is generated in an opposite process (Fig. 5).

3.3.4. Mutation operator

Basically, the type of operator used for mutation depends on the type of presentation. In this case, we use the random replacement operator, which is one of the common methods for numeric representations [33, 50]. In this operator, first, a string of random numbers with the length of the number of genes in the interval $[0, 1]$ is generated. In each position of the string, if the corresponding random number is lower than the mutation rate, a new value is randomly assigned to a set of allowable values in each position which serve to select and replace the number of the genes in that position. Otherwise, the number of the genes remains unchanged. For more explanation, see Figure 6.

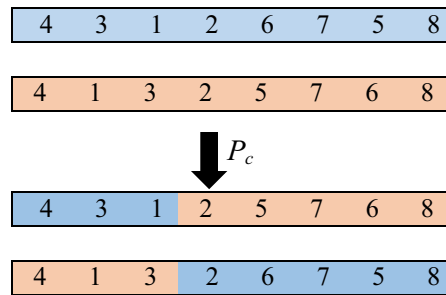


FIGURE 5. An example of a uniform crossover process.

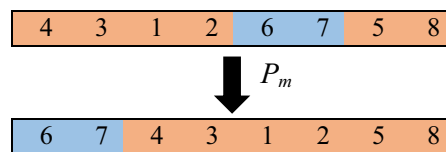


FIGURE 6. An example of a random replacement mutation.

3.3.5. Parent choice mechanism

In the general structure of the algorithm, there is a two-stage evolutionary cycle based on competition. The stages include (a) selection of individuals to participate in reproduction (parent selection) and (b) selection of individuals for survival in the next generation (choice of survivors).

In the genetic algorithm, the individuals with higher levels of fitness are more likely to be parental than those with lower fitness. Here, a binary variable is applied for parent's selection. In this type of choice, at each stage 2, choices are given, and the best individual is selected among them as a parent in terms of fitness. This is repeated until the intended number of parents is selected [50].

3.3.6. Primary population generation

In this section, each gene is formed by the assignment of a mode of execution to every activity. This mode is selected randomly from a set of possible execution modes. It is done to reach the primary population. The possible execution modes for an activity actually refer to a set of modes that have the necessary quality to perform that activity. Accordingly, before the production of the initial population, a preprocessor function should be used to identify the inefficient operating modes whose total weight of quality indices is lower than a certain limit. These modes should be avoided because they may disturb any stage of population generation. Therefore, all the generated solutions are always feasible solutions.

3.3.7. Computational analyses

In this section, the authors firstly generate the test problems for the presented model developed. According to the two conflicting objectives, four evaluating metrics are presented to assess the quality of non-dominated solutions of the meta-heuristics. Finally, a set of sensitivity analyses is utilized to examine the validity of the presented model developed. To the best of our knowledge and according to the novelty of the presented model, no existing study has treated a similar model in the literature. Then, the benchmarks existing in the literature is not available for the model and an approach is needed to design the test problems. In order to evaluate the efficiency of the presented algorithms, a number of standard examples are generated for the problem. Ten test problems including two classifications *i.e.* small: SP1 to SP4, medium sizes: MP5 to MP7, and large sizes: LP8

TABLE 1. The instances for test problem.

Classifications	Instance	s	p	k	i	r	v	t
Small	SP1	1	1	2	2	2	2	1
	SP2	2	2	3	3	4	3	2
	SP3	3	4	4	5	6	4	4
	SP4	4	5	6	6	8	6	8
Medium	LP5	6	8	8	8	10	8	9
	LP6	8	10	12	10	12	10	10
	LP7	10	12	14	12	14	12	12
Large	LP8	12	14	16	14	16	12	12
	LP9	14	16	18	16	18	14	12
	LP10	18	20	22	20	24	16	12

TABLE 2. Parameters for generated examples.

Problem	Number of activities	Number of run modes	Cost	Reliability	Number of reliability indicators	Reliability of each indicator in each run mode	The minimum total of reliability indicators in each run mode
1	7	[2, 9]	[15, 70]	[120, 190]	2	[60, 99]	[60, 75]
2	9	[2, 4]	[10, 50]	[10, 20]	5	[60, 99]	[60, 75]
3	15	[2, 7]	[10, 200]	[30, 60]	2	[60, 99]	[60, 75]
4	21	[2, 6]	[50, 400]	[250, 480]	3	[60, 99]	[60, 75]
5	31	[2, 6]	[150, 450]	[50, 140]	4	[60, 99]	[60, 75]
6	40	[2, 5]	[100, 160]	[50, 120]	4	[60, 99]	[60, 75]
7	50	[2, 5]	[30, 120]	[80, 180]	3	[60, 99]	[60, 75]
8	60	[2, 6]	[150, 450]	[100, 200]	5	[70, 99]	[60, 75]
9	80	[2, 5]	[200, 550]	[150, 250]	6	[80, 99]	[60, 75]
10	100	[2, 8]	[300, 650]	[200, 300]	8	[90, 99]	[60, 75]

to LP10 are presented. Table 1 introduces the sizes of problem instances. Table 2 presents the parameters of the generated examples.

3.4. Parameters setting

Meta-heuristic algorithms are very sensitive to their setting parameters, and a change in those parameters has a significant impact on the efficiency of the algorithms. The proposed algorithms are coded utilizing the MATLAB R2016b software. They are implemented in the computer system with the characteristics of the Core i5/6 GHz processor and the 6 GB side-mounted memory. Table 3 shows the proposed values for each parameter related to the NSGA-II and Fast PGA algorithms. As it is noted in the table, n represents the activities related to each problem. Therefore, some experiments were conducted on problems 2 and 6 to survey parameter setting for small-scale problems (*i.e.* problems 1, 2, 3, and 4), medium-scale (*i.e.* problems 5, 6, and 7), and large-scale problems (*i.e.* problems 8, 9 and 10). Tables 4 and 5 present the parameter setting values for two algorithms in which the value of each parameter is fixed by keeping the other parameters in their lowest value. This is done based on the number of the non-dominated points identified through a certain number of repetitions.

TABLE 3. The initial value of proposed algorithms parameters.

Type of algorithm	Parameter	Proposed values		
NSGA-II	Mutation rate	0.1	0.2	0.3
	Intersection rate	0.4	0.5	0.6
	Population number of each generation	$5n$	$7n$	$10n$
	a	$2n$	$3n$	$4n$
Fast PGA	b	0.5	1	1.5
	c	$2n$	$3n$	$4n$
	d	0	0.5	1
	Mutation rate	0.1	0.2	0.3
	Intersection rate	0.4	0.5	0.6
	Maximum population	$5n$	$7n$	$10n$
	Maximum number of children	$5n$	$7n$	$10n$

TABLE 4. Parameter setting values for Fast PGA algorithm parameters.

	a	b	c	d	The number of maximum	
					Population of offspring	Population
Small-scale examples	$4n$	0.5	$3n$	0.5	$5n$	$7n$
Medium-scale examples	$3n$	0.5	$3n$	1	$5n$	$7n$
large-scale examples	$6n$	0.9	$4n$	1	$6n$	$10n$

TABLE 5. Parameter setting values for the parameters of the NSGA-II algorithm.

	Mutation rate	Intersection rate	The population of each generation
Small-scale examples	0.2	0.4	$6n$
Medium-scale examples	0.3	0.5	$10n$
Large-scale examples	0.8	0.6	$12n$

3.5. Algorithms validation

In order to prove that the results of the formulation fully reach the Pareto front side of the problem and the solutions are necessary for dispersion, the efficiency of the methods used in the different parts of the algorithms as well as the validity of the obtained solutions should be measured. Thus, three examples are generated on a relatively small scale. Then, through a search for all the possible spaces in these real-world problems, they are identified and compared with the results of the algorithms. Table 6 presents the specifications of the example generation, the results related to the full search of the solution space and their solutions through two algorithms. For instance, after a complete review of the solution space of an example, as in Table 7, it has been found that there are 15 non-dominated points for that example. As it can be seen, the obtained solutions have no superiority to one another, and it is not possible to find a solution that is the best for all objects. The dispersion of the points in the solution space is indicated in Figure 7a. This problem has been obtained by NSGA-II and Fast PGA algorithms in about four seconds to identify 14 and 13 non-dominated points respectively. Regarding the results of the other two examples, it can be concluded that the validity and ability of the two algorithms to reach optimal solutions are acceptable. Figures 7b and 7c show the non-dominated points identified in the two algorithms.

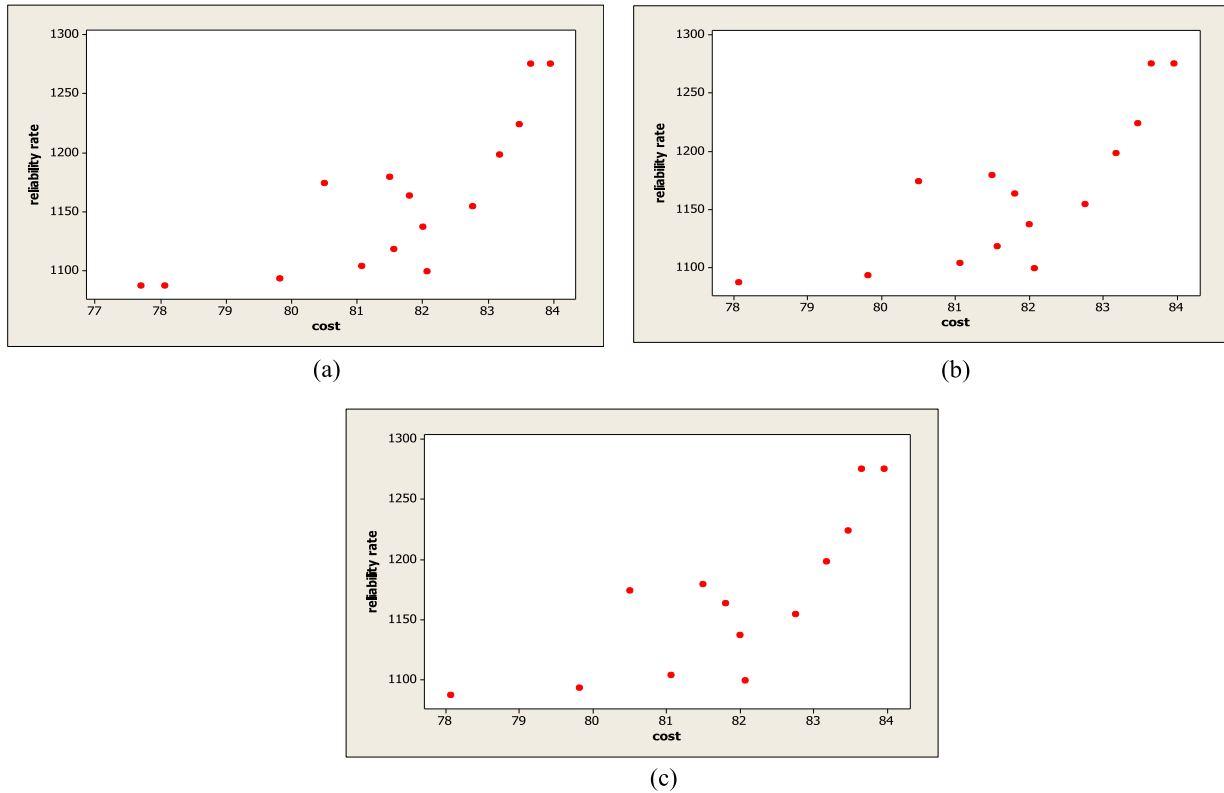


FIGURE 7. Distribution of non-dominated points. (a) Pareto points dispersion in the solution space using a full count method. (b) Non-dominated points dispersion identified in the NSGA-II algorithm. (c) Non-dominated points dispersion identified in the Fast PGA algorithm.

TABLE 6. Specifications and results related to the examples to validate algorithms.

Number of activities	Number of modes	Coefficient complexity	Total number of feasible points of solution space	Total number of non-dominated points	Total runtime (second)	The number of non-dominated points which discovered by the algorithms		Runtime of two algorithms (seconds)
						NSGA-II	Fast PGA	
8	[2, 5]	1.43	4850	17	274	15	13	5
9	[2, 5]	1.6	37 487	38	24 560	36	31	35
12	[2, 4]	1.53	196 534	13	23 730	14	12	40

3.6. The evaluating criteria and algorithms comparison

Through the comparison of two multi-objective algorithms in terms of efficiency, several useful indicators, or criteria, should be taken into account. These indicators are mainly divided into two categories. The first emphasizes the convergence and the quality of the solutions, and the second emphasizes the dispersion and the

TABLE 7. The values of the objective functions related to non-dominated points of the first example.

Non-dominated points	Reliability	Cost
1	1275	83.96
2	1224	83.66
3	1163	83.48
4	1198	81.82
5	1137	83.18
6	1137	82.35
7	1179	81.58
8	1154	82.76
9	1118	81.57
10	1104	81.07
11	1174	80.51
12	1099	82.07
13	1093	79.82
14	1087	78.07
15	1087	77.71

expansion of those solutions in the solution space. In this respect, there are four indicators for the comparison of two algorithms [50]:

- Number of Pareto solutions (NPS),
- Maximum spread or diversity (MS),
- Mean ideal distance (MID),
- Diversification metric (DM).

4. COMPUTATIONAL RESULTS

In this section, the examples generated in the previous section are solved with the NSGA-II and Fast PGA algorithms. In order to assess and compare the two algorithms, the criteria mentioned in the previous section are used. Also, each example is run for 10 times by each algorithm, and the non-dominated solutions of these 10 runs are used to compare the results. Therefore, the comparison of the algorithms is presented in small-scale examples (Examples 1–4), medium-scale examples (Examples 5–7), and large-scale examples (Examples 8–10).

4.1. Performance evaluation of proposed algorithms on small-scale problems

The performance and the efficiency of the NSGA-II and Fast PGA algorithms are compared in examples 1–4. In Table 8, the best results are executed in 10 times, and the average of the results obtained from this set of runs is presented for the two algorithms. Also, Table 9 represents the values of the comparative indices for both algorithms. The run time is taken into consideration for two identical algorithms.

Table 9 and Figure 8 indicate that the Fast PGA algorithm is more successful than the NSGA-II algorithm in finding the number of Pareto's points in all the problems, except problem 4. Regarding the mean distance from the ideal solution, there cannot be a decisive claim of superiority for one of the algorithms over the other. For most of the expanding and diverse criteria that imply the dispersion and proper distribution of the solutions, the Fast PGA algorithm is significantly superior to NSGA-II.

TABLE 8. Computation results of NSGA-II and Fast PGA algorithms for examples 1–4.

Problem	Algorithms	The best solution		The average of solutions	
		Reliability	Cost	Reliability	Cost
SP	NSGA-II	1843.56	87.54	1854.43	85.02
	Fast PGA	1843.93	84.22	1853.63	82.19
SP2	NSGA-II	700.21	91.52	710.77	89.32
	Fast PGA	711.23	87.79	723.34	86.63
SP3	NSGA-II	1431.54	86.32	1048.67	84.65
	Fast PGA	1429.33	85.24	1045.19	84.60
SP4	NSGA-II	11 455.22	88.86	12 203.45	87.45
	Fast PGA	11 458.12	88.63	11 981.44	87.23

TABLE 9. Comparative criteria for the NSGA-II and Fast PGA algorithms for examples 1–4.

Problem	Algorithms	Run time (S)	NPS	MS	MID	DM
SP1	NSGA-II	10	30	114.43	51.43	12.32
	Fast PGA	10	31	115.22	47.76	12.55
SP2	NSGA-II	10	24	87.63	34.37	10.06
	Fast PGA	10	25	90.54	40.07	10.24
SP3	NSGA-II	30	19	65.39	37.07	8.94
	Fast PGA	30	34	67.31	33.79	9.02
SP4	NSGA-II	150	132	396.14	202.73	22.96
	Fast PGA	150	126	440.19	214.42	24.01

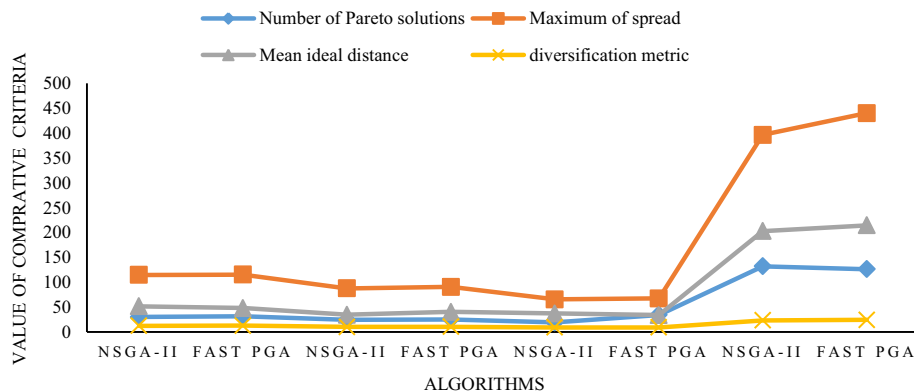


FIGURE 8. Comparative criteria for the NSGA-II and Fast PGA algorithms for examples 1–4.

4.2. Performance evaluation of the proposed algorithms on medium-scale and large-scale problems

As in the previous section, the performance of the proposed algorithms is examined here in relation to problems 5–7 for medium scale and 8–10 for large scale. Table 10 presents the best results and the average of the results achieved for the two algorithms. The values of the comparative indices for the two algorithms are also indicated in Table 11.

TABLE 10. Computation results of NSGA-II and Fast PGA algorithms for examples 5–7 and 8–10.

Problem	Algorithms	The best solution		The average of solutions	
		Reliability	Cost	Reliability	Cost
MP5	NSGA-II	5349.71	87.54	5390.82	87.13
	Fast PGA	5408.52	87.33	5462.63	87.12
MP6	NSGA-II	8254.76	82.76	8342.46	82.65
	Fast PGA	8354.53	82.73	8403.28	82.66
MP7	NSGA-II	8434.62	85.94	11477.33	85.32
	Fast PGA	9102.87	85.65	12477.33	85.54
LP8	NSGA-II	9431.25	93.21	9567.12	92.76
	Fast PGA	9780.23	93.13	9877.38	92.53
LP9	NSGA-II	10212.92	104.39	10516.44	104.21
	Fast PGA	13434.22	104.21	13849.17	104.34
LP10	NSGA-II	14555.67	156.83	16772.12	155.67
	Fast PGA	18939.26	156.67	19566.59	155.29

TABLE 11. Comparative criteria for the NSGA-II and Fast PGA algorithms for examples 5–7 and 8–10.

Problem	Algorithms	Run time (S)	NPS	MS	MID	DM
MP5	NSGA-II	600	52	354.51	158.09	20.05
	Fast PGA	600	49	566.84	183.76	23.49
MP6	NSGA-II	900	104	546.23	271.32	22.67
	Fast PGA	900	53	753.56	214.08	27.16
MP7	NSGA-II	1000	114	494.14	299.95	22.45
	Fast PGA	1000	43	654.54	158.09	26.07
LP8	NSGA-II	1000	128	788.55	201.21	28.45
	Fast PGA	1000	56	467.84	167.45	23.12
LP9	NSGA-II	1000	132	891.21	271.32	34.67
	Fast PGA	1000	69	582.98	213.13	21.32
LP10	NSGA-II	1000	139	932.75	321.81	41.92
	Fast PGA	1000	88	682.29	218.34	26.48

According to the results presented in Table 11 and Figures 9 and 10, it can be claimed that the NSGA-II algorithm yields better results with the number of non-dominated solutions, while the Fast PGA algorithm performs better in terms of solution variety. As expected, if the size of the problem and the solution space of the NSGA-II algorithm are increased, compared with the Fast PGA, the number of Pareto's points will be found. This is due to the fast convergence of the Fast PGA algorithm to reach non-dominated points. In this algorithm, higher iterations lead to the generation of feasible optimal solutions. As it was observed, for small-scale examples, both algorithms have roughly equal performance, and the differences in their results can be ignored. When the size of problems increases, however, the differences become more and more evident.

Of course, one cannot ignore the role of runtime in achieving the best solutions. Obviously, the higher the runtime, the better the NSGA-II algorithm functions than the other one. The shorter the runtime, however, the more favorable the Fast PGA algorithm is to the other one. Of course, as mentioned before, due to its high convergence, the Fast PGA algorithm is used in cases where the evaluation of the solutions is either computational or financially cost-effective, or good solutions are to be found in a short time.

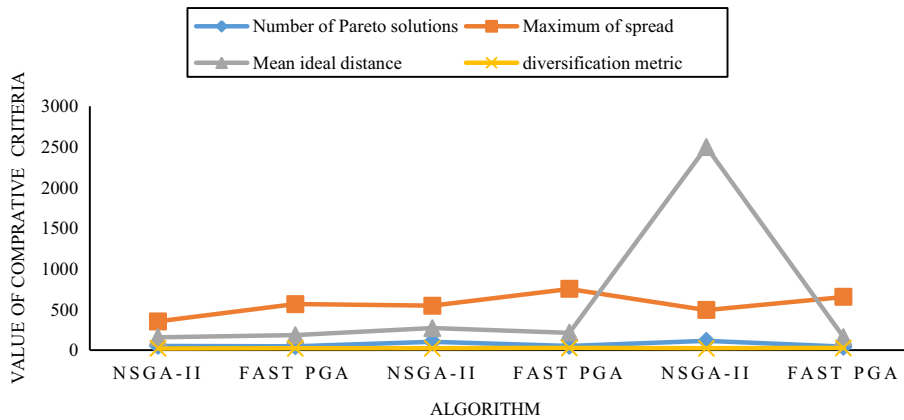


FIGURE 9. Comparative criteria for the NSGA-II and Fast PGA algorithms for examples 5–7.

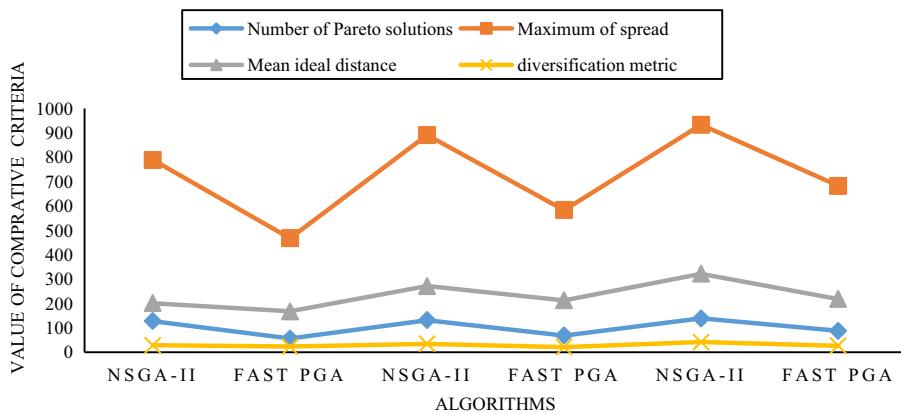


FIGURE 10. Comparative criteria for the NSGA-II and Fast PGA algorithms for examples 8–10.

Moreover, to find out the best meta-heuristic decisively, this study conducts a set of statistical comparisons among meta-heuristics based on Pareto optimal analyses taken by measurement metrics. Accordingly, the results which were reported by Tables 8–11 are transformed into a well-known metric, namely, Relative Deviation Index (RDI) as the following formula [50]:

$$RDI = \frac{|Alg_{sol} - Best_{sol}|}{Max_{sol} - Min_{sol}} \times 100 \tag{4.1}$$

where Alg_{sol} is the objective value obtained by a given measurement metric of algorithm, Max_{sol} and Min_{sol} are respectively the maximum and the minimum values among all values outputted by algorithms. $Best_{sol}$ is the best solution among methods; in another word, it is one of the Max_{sol} and Min_{sol} according to the nature of metrics. It is evident that a lower value of RDI brings a higher quality of algorithms. Consequently, the means plot and Least Significant Difference (LSD) for the proposed modified and hybrid algorithms and their individual ones have been resulted. The results run by *Minitab* 16 Statistical Software are shown in Figure 11. According to Figure 11, based on NPS, MID, MS, and DM metrics, the Fast PGA algorithm shows the best performance.

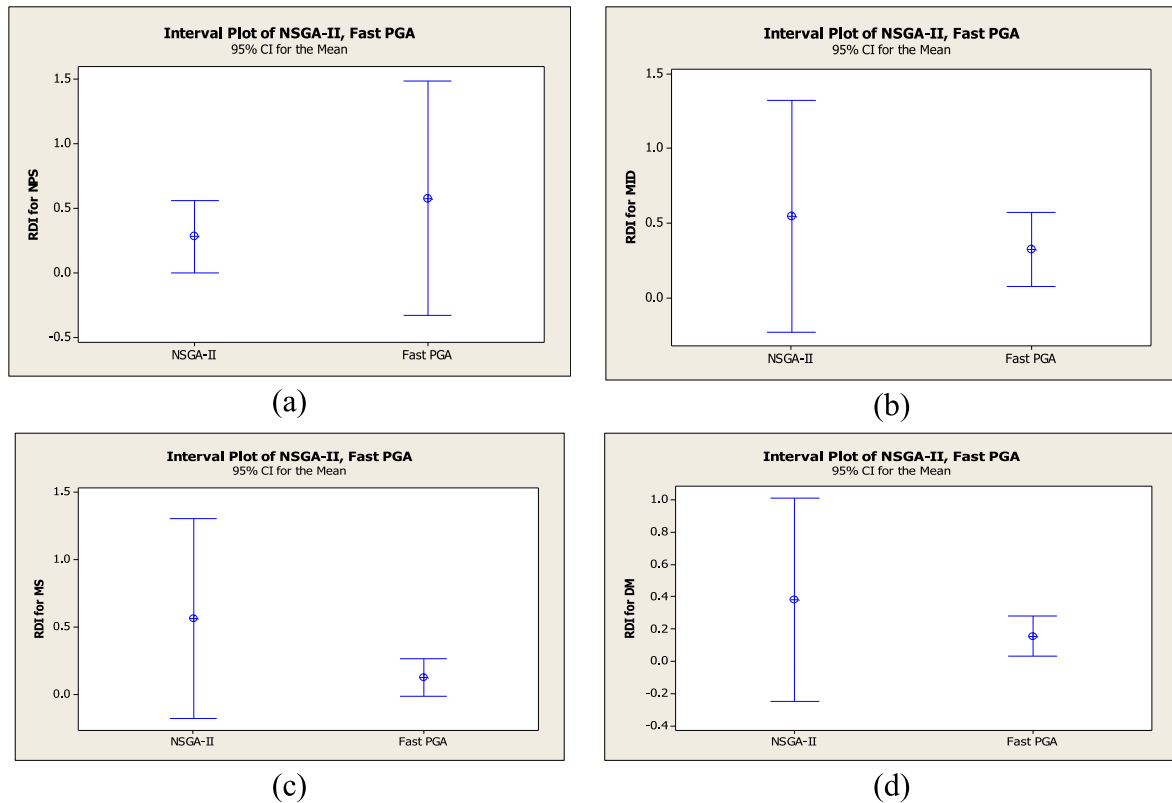


FIGURE 11. ANOVA plots for the assessment metrics in term of RDI for meta-heuristics (*i.e.* (a) for NPS, (b) for MID, (c) for MS and (d) for DM).

4.3. Sensitivity analyses

This section contains the results of three tests applied to assess the sensitivity analysis of the impact of the parameters on the values of the decision variables and the objective function of the presented model. Since the Fast PGA algorithm proved to be the most effective one in the comparisons, we used it for this sensitivity analysis, applied to the large-sized experimental problem LP6. The results of each test include the value of the total cost (TC) and the reliability (R).

The first experiment considers modifications in the parameter of the raw material cost (rp_{st}). The second experiment focuses the parameter of the reliability ratio of transport system (ras_{sp}^{ts} , raj_{pk}^{ts} , rad_{ki}^{ts}). We designed four experiments for each parameter and analyzed the changes in the objective functions. The relevant results are presented in Tables 12 and 13, and the trade-off between objective functions of total cost and reliability are presented as normalized values in Figures 12 and 13.

As indicated in Table 12 and Figure 11 with an increase in raw material cost, in the four experiments the value of the objective function increases slowly with TC and then R is fixed and without change. Additionally, by increasing reliability ratio of transport systems, the first objective function is remained fixed but the second objective function is raised according to the Table 13 and Figure 12.

4.4. Management insight

This study attempts to develop a new solution methodology in a supply chain network (SCN) design under grey flexible condition. Hence, by optimizing the network configurations, managerial capabilities in elevating

TABLE 12. The results of the sensitivity analysis related to the first experiment.

The number of cases	$\#rp_{st}$	TC	R
C1	10000	854.32	312.41
C2	20000	945.34	312.41
C3	30000	997.21	312.41
C4	40000	1023.12	312.41

TABLE 13. The results of the sensitivity analysis related to the second experiment.

The number of cases	$\#ras_{sp}^{ts}$	$\#raj_{pk}^{ts}$	$\#rad_{ki}^{ts}$	TC	R
C1	#10000	#15000	#20000	854.32	312.41
C2	#15000	#20000	#25000	854.32	378.23
C3	#25000	#30000	#35000	854.32	467.27
C4	#35000	#40000	#45000	854.32	678.29

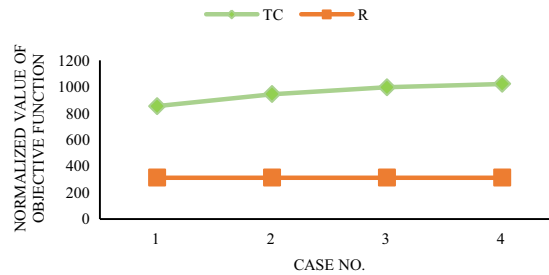


FIGURE 12. The behavior of normalized objective functions for sensitivity analysis in the first experiment.

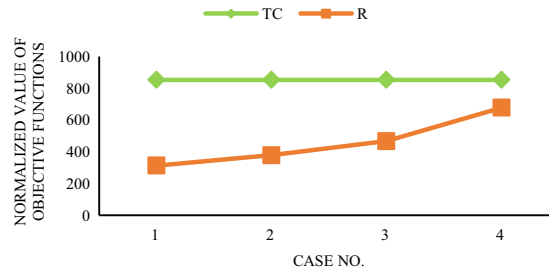


FIGURE 13. The behavior of normalized objective functions for sensitivity analysis in the second experiment.

their SCN, and also solving real-world problems are improving. Such improvements can help many industries, *e.g.* the pharmaceutical and the car industries in applying theoretical developments for real cases. On the other side, when managers decide to apply such models in their organization, problems result in most cases, which general exact solvers are unable to cope with. Additionally, the outcomes of the model have tremendous effects on management decisions for the production and distribution of the SCN. Therefore, elevating the capabilities of solving such vital problems and proposing new methodologies which can solve real cases in appropriate time with reliable solutions, are important steps to the applicability of SCN problems in real situations for managers and industries. Therefore, the results of this paper present powerful and reliable methods for managers to cope with supply chain network design and planning issues. It also encourages managers to revise the production-distribution decisions of the SCN of their organizations.

5. CONCLUSION

In this paper, a novel multi-echelon, multi-objective, multi-product, and multi-period production-distribution model is developed in the SCN design problem. This model is based on two objective functions including minimizing the operating costs of suppliers, manufacturers, distributors, and customers and maximizing the reliability of the system. The proposed model is formulated in the form of MILP. In the context of Grey flexible linear mathematical programming, the model deals with Grey/flexible constraints jointly; that is, the Grey coefficients for the lack of knowledge and the Grey purpose of the decision maker(s). Because the problem belongs to the category of NP-hard problems, and an increase in the problem size makes its solution time increase exponentially, the model makes use of the NSGA-II and Fast PGA algorithms, which are among the algorithms used in multi-objective problems. After the corresponding parameters were set in this study, the two algorithms were compared in terms of several indices for several examples generated in different sizes. Finally, the computational results were analyzed for those examples.

In this study, all model parameters are considered as deterministic parameters. However, the proposed model could be extended to consider the uncertainty in demand, costs, and capacity for future work. Additionally, the performance of the NSGA-II and Fast PGA can be compared with other meta-heuristic algorithms such as the Tabu, particle swarm optimization, and social engineering optimization algorithms. In addition, other approaches such as simulation-based optimization can be applied instead of a mathematical programming approach, which can lead to interesting findings. On the other hand, the bi-objective approach can be elevated to multi-objective approaches, which let the network consider some essential objectives such as minimizing CO₂ emissions and shortage of products and maximizing the number of customer respondents.

APPENDIX A. RELATED MATERIALS ABOUT GREY SYSTEM

A.1. Definitions, remarks, and theorems

Definition A.1. A grey system is defined as a system which includes uncertain information presented by grey variables and grey numbers.

Definition A.2. Suppose X is a universal set. Here, $X = \mathbb{R}$ is considered as the set of all real numbers. Then, the grey system, G , of X is determined by the two maps $\overline{\mu G}$ and $\underline{\mu G}$, where $\overline{\mu G} : X \rightarrow [0, 1]$, $\underline{\mu G} : X \rightarrow [0, 1]$, $\overline{\mu G} \geq \underline{\mu G}$. The $\underline{\mu G}$ and $\overline{\mu G}$ are the lower and the upper membership functions in G respectively. The Grey Systems (GS) theory introduces the concept of interval grey numbers. Grey numbers are considered as the basic unit of grey systems to apply in the formulation of a grey model. It can be a significant factor in the grey system theory [60]. Let X specify a bounded and closed set of real numbers.

Definition A.3. A grey number is a number with clear lower and upper boundaries, but it has an unknown position within the boundaries. In the system, a grey number is represented mathematically as:

$$\otimes x = [\underline{x}, \overline{x}] = \{t \in x | \underline{x} \leq t \leq \overline{x}\} \quad (\text{A.1})$$

where t is information, $\otimes x$ or x^G is a grey number, \underline{x} and \bar{x} are the lower and the upper limits of the characteristic [56].

Remark A.4. Note that the notations $\otimes x$ and x^G are the same grey number, and, for simplicity in modeling, the notation x^G is sometimes applied for the mentioned grey number.

There are different kinds of grey numbers (see them in [2, 29, 30, 32, 39]). Among them, interval grey numbers are applied here as appropriate ones reported in the literature.

Definition A.5. An interval grey number is the one with both upper limit \bar{x} and lower limit $\underline{x} : \otimes \in x[\underline{x}, \bar{x}]$.

Definition A.6 (White and black numbers). When $\otimes \in x(-\infty, +\infty)$, *i.e.* when $\otimes x \in [\underline{x}, \bar{x}]$ and $\underline{x} = \bar{x}$, $\otimes x$ is called a white number. When x has neither a lower limit nor an upper limit, the limits are all grey numbers; in this case, $\otimes x$ is called a black number.

The rest of the current study deals with interval grey numbers. For simplicity, they are briefly called grey numbers.

Remark A.7. For any real number a , if $\otimes = a[a, a]$, we say $\otimes a$ is the corresponding grey number. In reality, any real number is a white number. Thus, without loss of generality, throughout the study, it can be $\otimes = 0[0, 0]$ as the zero grey number. For simplicity, it is shown with $\otimes 0$ or 0 .

Remark A.8. The set of all grey numbers is denoted by $R(\otimes)$ which can represent an element as $\otimes x \in [\underline{x}, \bar{x}]$. It belongs to (\otimes) by $\otimes x \in [\underline{x}, \bar{x}]$.

Definition A.9. Let $L(\otimes x) = |\bar{x} - \underline{x}|$ represent the length of the grey number $\otimes x \in [\underline{x}, \bar{x}]$. It is clear that $L(\otimes x) : R(\otimes) \rightarrow R^+ \cup \{0\}$.

Let two grey numbers be defined as $\otimes x_1 \in [\underline{x}_1, \bar{x}_1]$ and $\otimes x_2 \in [\underline{x}_2, \bar{x}_2]$. The arithmetic operations are defined as follows:

$$\otimes x_1 + \otimes x_2 = [\underline{x}_1 + \underline{x}_2, \bar{x}_1 + \bar{x}_2] \tag{A.2}$$

$$\otimes x_1 - \otimes x_2 = [\underline{x}_1 - \underline{x}_2, \bar{x}_1 - \bar{x}_2]. \tag{A.3}$$

The following theorems are quoted from Xie and Liu [60].

Theorem A.10. *The result of the self-minus of a grey number is zero. That is, $\otimes x - \otimes x = 0$.*

Lemma A.11. *If $k \in \mathbb{R}$, $\otimes x \in [\underline{x}, \bar{x}]$ is a grey number, we will have:*

$$k \cdot (\otimes x) = \otimes(kx) \in [k\underline{x}, k\bar{x}] \quad \text{if } k \geq 0, \tag{A.4}$$

$$k \cdot (\otimes x) = \otimes(kx) \in [k\bar{x}, k\underline{x}] \quad \text{if } k < 0. \tag{A.5}$$

Definition A.12. The whitening number of a grey value, $\otimes x$, is determined as a deterministic number with its value lying between the lower and the upper bounds of $\otimes x$ (see [59]):

$$\underline{x} \leq \widetilde{\otimes x} \leq \bar{x}$$

where $\widetilde{\otimes x}$ is the whitening value of $\otimes x$. This relationship can be represented as:

$$\widetilde{\otimes x} = \underline{x} + \alpha (\bar{x} - \underline{x})$$

where the grey coefficient is specified by the α .

Theorem A.13. *The set of all the grey numbers establishes a field.*

Proof. It is straight forward as given in Shi *et al.* [52]. □

A.2. Ranking of grey numbers

There are several approaches to rank the grey numbers. Some of them are mentioned here.

A.2.1. Possibility degree

Definition A.14. For the two grey numbers $\otimes y \in [y, \bar{y}]$ and $\otimes x \in [x, \bar{x}]$, the possibility degree of $\otimes y$ is less, or greater) than or equal to $\otimes x$. It can be expressed as follows [29]:

$$\rho = \frac{\max(0, L^* - \max(0, \bar{x} - y))}{L^*} \tag{A.6}$$

where $L^* = L(\otimes x) + L(\otimes y)$.

The relation between $\otimes x$ and $\otimes y$ can be defined as follows:

- (I) If $\underline{x} = \underline{y}$ and $\bar{x} = \bar{y}$, it can be said that $\otimes x$ is equal to $\otimes y$, which is shown as $\otimes x = \otimes y$, when $\rho = 0.5$.
- (II) If there is an intersection, when $\rho > 0.5$, it can be clarified that $\otimes y$ is greater than $\otimes x$, which is determined as $\otimes x < \otimes y$.

Also, when $\rho < 0.5$, it can be said that $\otimes y$ is less than $\otimes x$, shown as $\otimes x > \otimes y$. An interval number refers to a specific one in grey number conception terms [52]. It is easy to verify the following results.

Theorem A.15. A set of grey numbers based on the possibility degree is a totally ordered set.

Lemma A.16. For the two grey numbers $\otimes y \in [y, \bar{y}]$ and $\otimes x \in [x, \bar{x}]$, we have $\otimes x \geq \otimes y$ if and only if $-\otimes x \leq -\otimes y$.

Lemma A.17. A grey number $\otimes x \in [x, \bar{x}]$ is said to be nonnegative if the whitening value of $\otimes x$ is nonnegative.

A.2.2. Kernel and greyness degrees

In this subsection, the concepts and definitions are taken from Xie *et al.* [59].

Definition A.18. Take the grey number $\otimes x \in [x, \bar{x}]$, where $\underline{x} \leq \bar{x}$ and there is no distributing information for the values of the grey number $\otimes x$. If the grey number $\otimes x$ is not discrete, then $\otimes \hat{x} = \frac{1}{2}(\underline{x} + \bar{x})$ is called the kernel of that grey number.

Definition A.19. Suppose the background which makes a grey number $\otimes x$ come into being is Ω , and $\mu(\otimes x)$ is the measure of Ω . Then $g^\circ(\otimes x) = \mu(\otimes x)/\mu(\Omega)$ is called the greyness degree of the grey number $\otimes x$ (specified as g° for short).

Definition A.20. Let $\otimes \hat{x}$ and $g^\circ(\otimes x)$ be respectively the kernel and the greyness degree of a grey number $\otimes x$. Then, $\otimes x = \otimes \hat{x}_{(g^\circ)}$ can be defined as the regular form of that grey number.

Proposition A.21. For grey numbers, there is a one-to-one correspondence between the grey numbers $\otimes x \in [x, \bar{x}]$ and the simplified forms $\otimes x = \otimes \hat{x}_{(g^\circ)}$, where $\underline{x} < \bar{x}$.

Regarding the development of the GS theory in different scientific areas and the request for the comparison of grey numbers in several fields, ranking of grey numbers plays a key role in some grey system applications and decision making. So far, different strategies have been developed to rank grey numbers. Here, one of the popular methods in the literature is discussed.

Definition A.22. Suppose $\otimes x$ and $\otimes y$ are two grey numbers, $\otimes_1 \hat{x}$ and $\otimes_2 \hat{x}$ are the kernels of $\otimes x$ and $\otimes y$, respectively, and $g^\circ(\otimes_1 x)$ and $g^\circ(\otimes_2 x)$ are the greyness degrees of $\otimes x$ and $\otimes y$, respectively. Accordingly,

- if $\otimes \hat{x} < \otimes \hat{y}$, then $\otimes x < \otimes y$;
- if $\otimes \hat{x} = \otimes \hat{y}$, then

- (1) if $g^\circ x(\otimes x) = g^\circ(\otimes y)$, then $\otimes x = \otimes y$;
- (2) if $g^\circ x(\otimes x) < g^\circ(\otimes y)$, then $\otimes x > \otimes y$;
- (3) if $g^\circ x(\otimes x) > g^\circ(\otimes y)$, then $\otimes x < \otimes y$.

In this way, different interval grey numbers can be compared.

A.2.3. Ranking of grey numbers based on whitenization values

An appropriate method for solving GLP problems is to compare grey numbers in terms of the corresponding whitenization values. An efficient method to order the elements of $R(\otimes)$ is to assign a whitenization value to every grey number. The whitenized function to $R^G : R(\otimes) \rightarrow \mathbb{R}$ maps each grey number in a genuine line, where a natural order exists. In fact, for every $\otimes x \in R(\otimes)$, there is the relationship $R^G(\otimes x) = \widetilde{\otimes}x$. Therefore, the order of $R(\otimes)$ can be defined based on whitenization values as follows:

- $\widetilde{\otimes}x \geq \widetilde{\otimes}y$ if and only if $\otimes x \geq \otimes y$;
- $\widetilde{\otimes}x > \widetilde{\otimes}y$ if and only if $\otimes x > \otimes y$;
- $\widetilde{\otimes}x = \widetilde{\otimes}y$ if and only if $\otimes x = \otimes y$;

where $\otimes x$ and $\otimes y$ belong to (\otimes) . Also, there is $\otimes x \leq \otimes y$ if and only if $\widetilde{\otimes}x \geq \widetilde{\otimes}y$. By the above definition, $\otimes x > \otimes 0$ holds true when $\widetilde{\otimes}x \geq 0$. A corresponding lemma can be also described simply as follows.

Lemma A.23. *Let R^G be any whitenized function. Then,*

- (1) $\otimes x \geq \otimes y$ if and only if $\otimes x - \otimes y \geq \otimes 0$, and if and only if $-\otimes y \geq -\otimes x$;
- (2) If $\otimes x \geq \otimes y$ and $\otimes u \geq \otimes v$, then $\otimes x + \otimes u \geq \otimes y + \otimes v$;

where $\otimes x, \otimes y, \otimes u$, and $\otimes v$ belong to (\otimes) and $\otimes 0$. The latter is a grey zero number which is defined in Remark A.8.

A.3. Grey flexible linear programming

An appropriate fuzzy linear programming model is the Grey Flexible Linear Programming (GFLP), which practically involves various formulations of flexible linear programming [5, 24]. In the following subsection, a definition is given for this type of problems. Whatever discussion that follows is based on this definition.

A.3.1. GFLP problem with a linear membership function

Suppose a decision maker faces a linear programming problem in which s/he can endure violation in completing the constraints; that is, s/he allows the constraints to be held as well as possible. When there is a set of constraints, the assumptions $a_i x \leq^F b_i$ and $i = 1, \dots, n$ can be made for each constraint and formulated by the use of a membership function as follows:

$$\mu_i(x) = \begin{cases} 1, & a_i x \leq b_i \\ f(a_i x)_i, & b_i \leq a_i x \leq b_i + p_i \\ 0, & a_i x \geq b_i + p_i \end{cases} \tag{A.7}$$

where $f_i(0)$ is continuous and strictly decreasing for $a_i x$, $f_i(b_i) = 1$ and $f_i(b_i + p_i) = 0$.

Equation (A.6) emphasizes that the decision maker can tolerate violation in the accomplishment of constraints i up to the value $b_i + d_i$. Considering the assumptions, the associated GFLP problem is represented as:

$$\begin{aligned} & \max z^G(x) = c^G x \\ & \text{s.t. } Ax \leq^F b \\ & \quad x \geq 0. \end{aligned} \tag{A.8}$$

Or,

$$\begin{aligned} & \max z^G(x) = c^G x \\ & \text{s.t. } H_i(z, a_i) = \sum_{j=1}^n a_{ij} x_j - b_i \leq^F 0, \quad i = 1, \dots, m, \\ & \quad x_j \geq 0, \quad j = 1, \dots, n, \end{aligned}$$

where $x^T = (x_1, x_2, \dots, x_n)$ is a n -dimensional real decision, and $c^G = (c_1^G, c_2^G, \dots, c_n^G)$ is a n -dimensional vector of grey parameters involved in the grey objective Z . Also the notation ‘‘max’’ maximizes the objective function

in a fuzzy sense and “ \preceq^F ” represents a fuzzy extension of “ \leq ” on \mathbb{R} which is applied to compare the left side of fuzzy constraints with the right side.

In general, the main model is not well-defined due to the following reasons:

- (i) The grey quantity $z(c^G, x)$ cannot be maximized based on the current method.
- (ii) The constraint $H_i(x, a_i) = \sum_{j=1}^n a_j x - b_i \leq^F 0$ does not result in a crisp feasible set.

An appropriate approach to state the crisp optimal solution preference of an alternative is comparing the grey quantities using the whitenization function $R^G : R(\otimes) \rightarrow \mathbb{R}$ that maps each grey quantity to a real line which exists a natural order (for more detail, refer to [10, 52]).

In addition, if a deterministic feasible set is to be specified, a confidence level α_i should be provided so that a desirable corresponding i -th fuzzy constraint can hold. Therefore, in order to remove those mentioned restrictions, the following problem is introduced:

$$\begin{aligned} \max z(x) &= R(c^G)x \\ \text{s.t. } \mu_i \{H_i(x, a_i) \leq^F 0\} &\geq \alpha_i, \\ x \geq 0, \alpha_i &\geq \alpha_i^D, 0 \leq \alpha_i \leq 1, \quad i = 1, \dots, m, \end{aligned} \tag{A.9}$$

where $z(x) = R(c^G)x$ means the corresponding crisp value of grey function $z^G(x)$ based on linear ranking function. To choose a suitable membership function for each fuzzy constraint, it is argued that, if $H_i(x, a_i) \leq 0$, then the i -th constraint is fully met. However, if $H_i(x, a_i) \geq p_i$, then the i -th constraint is perfectly violated. In the latter formula, p_i is the predefined maximum tolerance from zero, as defined by the decision maker. For $H_i(x, a_i) \in (0, 1)$, it is recommended by Liu and Zhang [33] that the membership function of the i -th constraint be applied as:

$$\mu(H_i(x, a_i) \leq 0) = \begin{cases} 1, & H_i(x, a_i) \leq 0 \\ 1 - \frac{H_i(x, a_i)}{p_i}, & 0 \leq H_i(x, a_i) \leq p_i \\ 0, & H_i(x, a_i) \geq 0. \end{cases} \tag{A.10}$$

And this is equal to:

$$\mu_i(x) = \begin{cases} 1, & x, a_i \leq b_i \\ \frac{b_i + p_i - a_i x}{p_i}, & b_i \leq a_i x \leq b_i + p_i \\ 0, & a_i x \geq b_i + p_i. \end{cases} \tag{A.11}$$

And the (2.27) gets:

$$\begin{aligned} \max z(x) &= R(c^G)x \\ \text{s.t. } a_i x &\leq b + p_i(1 - a_i) \\ x \geq 0, \alpha_i^D &\leq \alpha_i, 0 \leq \alpha_i \leq 1, \quad i = 1, \dots, m. \end{aligned} \tag{A.12}$$

The above problem is called Multi-Parametric Linear Programming problem (MPLP1).

Now, a feasible solution is made to the grey linear programming problem as presented in (A.13).

Definition A.24. Let $\bar{\alpha} = (\alpha_1, \dots, \alpha_m) \in (0, 1]^m$, then $X_{\bar{\alpha}} = \cap_{i=1}^m X_{\alpha_i}^i$, where

$$X_{\alpha_i} = \{x \in \mathfrak{R}^n \mid x \geq 0, a_i x \leq b_i + p_i(1 - a_i), a_i \geq \alpha_i^D\}, i = 1, \dots, m. \tag{A.14}$$

For $i = 1, \dots, m$ (namely, X_{α}^i is the α -cut of the i -th fuzzy constraint).

Proof. For any $\bar{\alpha} = (a_1, \dots, a_m) \in (0, 1]^m$, let $x \in X_{\bar{\alpha}}$. Therefore, $a_i \geq \alpha_i^D, a_i x \leq b_i + p_i(1 - a_i)$. With regard to equation (3.4), we have $x \in X_{\alpha_i}^i, i = 1, \dots, m$. Therefore, $X \in \cap_{i=1}^m X_{\alpha_i}^i$. Also, if $X \in \cap_{i=1}^m X_{\alpha_i}^i$, we have $x \in X_{\alpha_i}^i$, for all $i = 1, \dots, m$. Therefore, $a_i \geq \alpha_i^D, a_i x \leq b_i + p_i(1 - a_i)$ and hence $x \in X_{\bar{\alpha}}$. This completes the proof. \square

Proposition A.25. Let $\bar{\alpha}' = (\alpha'_1, \dots, \alpha'_m)$ and $\bar{\alpha}'' = (\alpha''_1, \dots, \alpha''_m)$, where $\alpha'_i \leq \alpha''_i$ holds for all i , then $\bar{\alpha}''$ -feasibility of x implies the $\bar{\alpha}'$ -feasibility.

Proof. The proof is straightforward. □

For a given $\alpha \in (0, 1]$, let $x \in \mathbb{R}^n$ be a usual α -feasible solution to (A.10). It is to be noted that this solution provides the same degree of satisfaction for all the constraints. It has the meaning of $a_i x \leq b_i + p_i (1 - \alpha_i)$, $\alpha_i \geq \alpha_i^D$ or equivalently $x \in X_\alpha^i$, for all $i = 1, \dots, m$.

If $\bar{\alpha} = (\alpha, \dots, \alpha)$, then $x \in X_\alpha$ which implies that the α -feasibility of (A.10) can be understood as a special case of the α -feasibility. Thus, the following result can be obtained.

Remark A.26. If the problem (A.10) is not infeasible, then X_α is not empty.

Proof. The proof is straightforward. □

Definition A.27. Let “ \prec^F ” be a fuzzy extension of binary relation “ \leq ” and let $x = (x_1, \dots, x_n)^T \in \mathbb{R}^n$ be an $\bar{\alpha}$ -feasibility solution to (A.10), where $\bar{\alpha} = (\alpha_1, \dots, \alpha_m) \in (0, 1]^m$. Also, let $c^G x$ be a grey objective. The vector $n x \in \mathbf{R}$ is an α -efficient solution to (A.10) with maximization of the objective. In this case, if there is no $x' \in X_{\bar{\alpha}}$, the relation $c^G x < c^G x'$ holds true.

Similarly, an α -efficient solution with the optimization of the objective can be defined.

It is to be noted that any α -efficient solution to the GFLP problem is an α -feasible solution to that problem with some extra properties. In the following theorem, a necessary and sufficient condition is postulated for a α -efficient solution to (3.2).

Theorem A.28. Let $\bar{\alpha} = (\alpha_1, \dots, \alpha_m) \in (0, 1]^m$ and $x^* = (x_1^*, \dots, x_n^*)^T$, $x_j^* \geq 0$, $j = 1, \dots, n$ be a $\bar{\alpha}$ -feasible solution to (A.10). Then, the vector $x^* \in \mathbb{R}^n$ is an $\bar{\alpha}$ -efficient solution to the problem (A.10) with the optimization of the objective if and only if x^* is optimal in the following mathematical model:

$$\begin{aligned} \max z(x) &= R(c^G)x \\ \text{s.t. } a_i x &\leq b_i + p_i(1 - \alpha_i) \\ x &\geq 0, \alpha_i^D \leq \alpha_i, 0 \leq \alpha_i \leq 1, \quad i = 1, \dots, m \end{aligned} \tag{A.15}$$

where p_i is the predefined maximum tolerance.

Proof. Let $\bar{\alpha} = (\alpha_1, \dots, \alpha_m) \in (0, 1]^m$ and $x^* = (x_1^*, \dots, x_n^*)^T$, $x_j^* \geq 0$, $j = 1, \dots, n$ be an $\bar{\alpha}$ -feasible solution to (A.11) with optimization of the objective. By “Definition A.24” and equation (A.10), we have $a_i x^* \leq b_i + p_i(1 - \alpha_i)$, $\alpha_i \geq \alpha_i^D$ for $i = 1, \dots, m$. Therefore, x^* is a feasible solution to (A.13). In this case, x^* is obviously an $\bar{\alpha}$ -feasible solution to problem (A.11). Thus, by “Definition A.27”, the optimality of x^* implies the $\bar{\alpha}$ -efficiency of x^* .

In Theorem A.28, we have provided a computational method to solve fuzzy flexible linear programming problem (A.11). Thus, when a specific $\bar{\alpha}$ is assigned by a decision maker, the α_j can be replaced in the corresponding constraint of (A.13), and the resulted problem can be solved. An $\bar{\alpha}$ -efficient solution is, thus, made to problem (A.11). Such a solution has two characteristics:

- (i) It provides various satisfaction degrees depending on the constraints.
- (ii) It is an optimal solution.

This solution permits decision makers to achieve more flexibility and compatibility by assigning desired preferences, especially in online optimization for more desirable cases.

In Theorem A.28, a method is introduced to obtain an $\bar{\alpha}$ -efficient solution to a GELP problem. If the resulting problem (A.13) has only one optimal solution, the theorem is confirmed. This solution is, thus, an $\bar{\alpha}$ -efficient one for a given fuzzy problem. In this case, problem (A.13) has some multiple optimal solutions. In order to

find a maximum efficient solution, *i.e.* an $\bar{\alpha}'$ -efficient solution with $\alpha' \geq \alpha$, $i = 1, \dots, m$, a two-phase approach is applied as follows. In Phase (I), problem (A.13) is solved. In Phase (II), a solution is obtained with a higher satisfaction degree than the previous one. So by the use of a two-phase approach, the available resources can be utilized better. Also, the solution resulting by this two-phase approach is always an $\bar{\alpha}'$ -efficient solution.

Problem (A.13) is called a “Phase (I) problem”. Let $\bar{\alpha}^0 = (\alpha_1^0, \dots, \alpha_m^0)$ and $(x^*, z(c^G, x^*))$ be the optimal solution of Phase (I) with the $\bar{\alpha}^0$ degree of efficiency. Set $\alpha_i^* = \mu_i \{H_i(x^*, \alpha_i) \prec^F 0\} \geq \alpha_i^0$. In Phase (II), the following problem is solved as follows:

$$\begin{aligned} \max W_\alpha &= \sum_{i=1}^m \alpha_i \\ R(c^G)x &\geq R(c^G)x^* \\ a_i x &\leq b_i + p_i(1 - a_i) \\ \text{s.t. } 0 &\leq \alpha_i \leq 1, \alpha_i^* \leq \alpha_i, \quad i = 1, \dots, m \\ x &\geq 0. \end{aligned} \tag{A.16}$$

The above problem is named a Multi-Parametric Linear Programming (MPLP) problem. Let $(x^{**}, \alpha_1^{**}, \dots, \alpha_m^{**})$ be an optimal solution to problem (A.16), *i.e.* (MPLP). Then, we have the following theorem. \square

Theorem A.29. *In the optimal solution x^{**} to the problem (A.16), or (MPLP), x^{**} is a maximum $\bar{\alpha}$ -efficient solution to the problem (A.11).*

Proof. In problem (A.16), there is the relationship $\alpha_i^* \geq \alpha_i^0$. Therefore, using Proposition (A.25), x^{**} is an $\bar{\alpha}^0$ -feasible solution to problem (A.13). This implies that it is feasible in problem [42]. Now, by the optimality of x^* in problem (A.9) and $z(x^*) = c^G x^* \leq z(x^{**}) = c^G x^{**}$, there is the optimality of x^{**} in problem [43]; hence, $z(x^*) = c^G x^* = z(x^{**}) = c^G x^{**}$. This means that x^{**} is also an $\bar{\alpha}^0$ -efficient solution to problem (A.15). Because $(x, \alpha_1^*, \dots, \alpha_m^*)$ is optimal and the coefficients in the objective function of problem (A.16) are positive, we have $\alpha_i^{**} = \mu_i \{H_i(x^*, \alpha_i) \prec^F 0\}$, $i = 1, \dots, m$. Now, if possible, let x^{**} be not a maximum $\bar{\alpha}^0$ -efficient solution to problem [43]. Then, there exists an $\bar{\alpha}^0$ -efficient x' to problem (A.15) such that $\alpha'_i > \alpha_i^{**}$, $i = 1, \dots, m$ and for some k , $\alpha'_k > \alpha_k^{**}$. In this case, $\alpha'_i = \mu_i \{H_i(x, a_i) \prec^F 0\}$, $i = 1, \dots, m$ and $z(x') = c^G x' \geq z(x^*) = cx$. Therefore, $(x', \alpha'_1, \dots, \alpha'_m)$ serves as a feasible solution to problem (A.16) and $\sum_{i=1}^m \alpha_i^{**} = \sum_{i=1, i \neq k}^m \alpha_i^{**} + \alpha_k^{**} < \sum_{i=1, i \neq k}^m \alpha'_i + \alpha'_k = \sum_{i=1}^m \alpha'_i$. But this implies that $(x^{**}, \alpha_1^{**}, \dots, \alpha_m^{**})$ is not a solution to problem (A.16), which is a clear contradiction.

The following section of the study is dedicated to the main steps of a desired method to solve a GFLP problem based on the above theoretical discussions. \square

REFERENCES

- [1] O. Abdolazimi, M.S. Esfandarani, M. Salehi and D. Shishebori, Robust design of a multi-objective closed-loop supply chain by integrating on-time delivery, cost, and environmental aspects, case study of a Tire Factory. *J. Cleaner Prod.* **121566** (2020) 1–15.
- [2] M. Al Hail, W. Elkassem, A. Hamad, P. Abdulrouf, B. Thomas and D. Stewart, Overview of pharmacovigilance practices at the largest academic healthcare system in the State of Qatar. *Int. J. Clinical Pharmacy* **40** (2018) 769–774.
- [3] M.H. Alavidoost, M. Tarimoradi and M.F. Zarandi, Bi-objective mixed-integer nonlinear programming for multi-commodity tri-echelon supply chain networks. *J. Intel. Manuf.* **29** (2018) 809–826.
- [4] Z. Asim, S.A. Jalil and S. Javaid, An uncertain model for integrated production-transportation closed-loop supply chain network with cost reliability. *Sustainable Prod. Consumption* **17** (2019) 298–310.
- [5] H. Attari and S.H. Nasser, New concepts of feasibility and efficiency of solutions in fuzzy mathematical programming problems. *Fuzzy Inf. Eng.* **6** (2014) 203–221.
- [6] G.K. Badhotiya, G. Soni and M.L. Mittal, Fuzzy multi-objective optimization for multi-site integrated production and distribution planning in two echelon supply chain. *Int. J. Adv. Manuf. Technol.* **102** (2019) 635–645.

- [7] M.M. Billal and M. Hossain, Multi-objective optimization for multi-product multi-period four Echelon supply chain problems under uncertainty. *J. Optim. Ind. Eng.* **13** (2020) 1–17.
- [8] T. Cal, Post Optimal Analysis, Parametric Programming and Related Topics. De Gruyter, New York (1995).
- [9] Y. Cardona-Valdés, A. Álvarez and J. Pacheco, Meta-heuristic procedure for a bi-objective supply chain design problem with uncertainty. *Transp. Res. Part B: Methodological* **60** (2014) 66–84.
- [10] D. Darvishi, S. Liu and S.H. Nasseri, A new approach in animal diet using grey system theory. *Grey Syst.: Theory App.* **8** (2018) 167–180.
- [11] A.K. Das and D.K. Pratihari, A novel approach for neuro-fuzzy system-based multi-objective optimization to capture inherent fuzziness in engineering processes. *Knowl.-Based Syst.* **175** (2019) 1–11.
- [12] A. Diabat, A. Jabbarzadeh and A. Khosrojerdi, A perishable product supply chain network design problem with reliability and disruption considerations. *Int. J. Prod. Econ.* **212** (2019) 125–138.
- [13] M.B. Fakhrazad and F. Goodarzian, A fuzzy multi-objective programming approach to develop a green closed-loop supply chain network design problem under uncertainty: modifications of imperialist competitive algorithm. *RAIRO:OR* **53** (2019) 963–990.
- [14] M.B. Fakhrazad, P. Talebzadeh and F. Goodarzian, Mathematical formulation and solving of green closed-loop supply chain planning problem with production, distribution and transportation reliability. *Int. J. Eng.* **31** (2018) 2059–2067.
- [15] M.B. Fakhrazad, F. Goodarzian and A.M. Golmohammadi, Addressing a fixed charge transportation problem with multi-route and different capacities by novel hybrid meta-heuristics. *J. Ind. Syst. Eng.* **12** (2019) 167–184.
- [16] A.M. Fathollahi-Fard, A. Ahmadi, F. Goodarzian and N. Cheikhrouhou, A bi-objective home healthcare routing and scheduling problem considering patients' satisfaction in a fuzzy environment. *Appl. Soft Comput.* **93** (2020) 106385.
- [17] C. Feng, Y. Ma, G. Zhou and T. Ni, Stackelberg game optimization for integrated production-distribution-construction system in construction supply chain. *Knowl.-Based Syst.* **157** (2018) 52–67.
- [18] J. Ghahremani-Nahr, R. Kian and E. Sabet, A robust fuzzy mathematical programming model for the closed-loop supply chain network design and a whale optimization solution algorithm. *Expert Syst. App.* **116** (2019) 454–471.
- [19] F. Goodarzian and H. Hosseini-Nasab, Applying a fuzzy multi-objective model for a production-distribution network design problem by using a novel self-adoptive evolutionary algorithm. *Int. J. Syst. Sci.: Oper. Logistics* (2019) 1–22.
- [20] F. Goodarzian, H. Hosseini Nasab and M.B. Fakhrazad, A multi-objective sustainable medicine supply chain network design using a novel hybrid multi-objective metaheuristic algorithm. *Int. J. Eng.* **33** (2020) 1986–1995.
- [21] F. Goodarzian, H. Hosseini-Nasab, J. Muñuzuri and M.B. Fakhrazad, A multi-objective pharmaceutical supply chain network based on a robust fuzzy model: a comparison of meta-heuristics. *Appl. Soft Comput.* **92** (2020) 106331.
- [22] S. Jalali, M. Seifbarghy, J. Sadeghi and S. Ahmadi, Optimizing a bi-objective reliable facility location problem with adapted stochastic measures using tuned-parameter multi-objective algorithms. *Knowl.-Based Syst.* **95** (2016) 45–57.
- [23] S.F. Ji, X.S. Peng and R.J. Luo, An integrated model for the production-inventory-distribution problem in the Physical Internet. *Int. J. Prod. Res.* **57** (2019) 1000–1017.
- [24] S. Karmakar and P. Mujumdar, Grey fuzzy optimization model for water quality management of a river system. *Adv. Water Res.* **29** (2006) 1088–1105.
- [25] S. Khalifehzadeh and M.B. Fakhrazad, A modified firefly algorithm for optimizing a multi stage supply chain network with stochastic demand and fuzzy production capacity. *Comput. Ind. Eng.* **133** (2019) 42–56.
- [26] S. Khalifehzadeh, M. Seifbarghy and B. Naderi, Solving a fuzzy multi objective model of a production–distribution system using meta-heuristic based approaches. *J. Intell. Manuf.* **28** (2017) 95–109.
- [27] A. Khanna, A. Kishore, B. Sarkar and C.K. Jaggi, Inventory and pricing decisions for imperfect quality items with inspection errors, sales returns, and partial backorders under inflation. *RAIRO:OR* **54** (2020) 287–306.
- [28] J. Li, X. Zeng, C. Liu and X. Zhou, A parallel Lagrange algorithm for order acceptance and scheduling in cluster supply chains. *Knowl.-Based Syst.* **143** (2018) 271–283.
- [29] S.F. Liu and Y. Lin, Grey Information, Theory and Practical Applications. Springer (2006).
- [30] S.F. Liu and Y. Lin, Grey Systems: Theory and Applications. Springer-Verlag, Berlin (2011).
- [31] S. Liu and L.G. Papageorgiou, Multi objective optimization of production, distribution and capacity planning of global supply chains in the process industry. *Omega* **41** (2013) 369–382.
- [32] S.F. Liu and N. Xie, Grey Systems Theory and its Applications. The Science Press of China, Beijing (2013).
- [33] X. Liu and D. Zhang, An improved SPEA2 algorithm with local search for multi-objective investment decision-making. *Appl. Sci.* **9** (2019) 1675.
- [34] Y. Liu, H. Qin, L. Mo, Y. Wang, D. Chen, S. Pang and X. Yin, Hierarchical flood operation rules optimization using multi-objective cultured evolutionary algorithm based on decomposition. *Water Res. Manage.* **33** (2019) 337–354.
- [35] Z. Liu, S. Qu, M. Goh, R. Huang and S. Wang, Optimization of fuzzy demand distribution supply chain using modified sequence quadratic programming approach. *J. Intel. Fuzzy Syst.* **36** (2019) 6167–6180.
- [36] Q. Long, A multi-methodological collaborative simulation for inter-organizational supply chain networks. *Knowl.-Based Syst.* **96** (2016) 84–95.
- [37] A. Megahed and M. Goetschalckx, A modeling framework and local search solution methodology for a production-distribution problem with supplier selection and time-aggregated quantity discounts. *Appl. Math. Model.* **68** (2019) 198–218.
- [38] M.R. Mohammadi, M. Abedini and B. Rashidi, An adaptive multi-objective optimization method for optimum design of distribution networks. *Eng. Optim.* **52** (2020) 194–217.
- [39] R.E. Moore, R.B. Kearfott and M.J. Cloud, Introduction to Interval Analysis. SIAM Press, Philadelphia (2009).

- [40] B. Naderi, K. Govindan and H. Soleimani, A Benders decomposition approach for a real case supply chain network design with capacity acquisition and transporter planning: wheat distribution network. *Ann. Oper. Res.* **291** (2020) 685–705.
- [41] S.H. Nasserri and S. Bavandi, Amelioration of Verdegay’s approach for fuzzy linear programs with stochastic parameters. *Iran. J. Manage. Stud. (IJMS)* **11** (2018) 71–89.
- [42] S.H. Nasserri and D. Darvishi, Diet modeling in uncertainty conditions using Grey Systems approach. *J. Oper. Res. App.* **12** (2016) 29–45.
- [43] S.H. Nasserri and D. Darvishi, Duality Results on grey linear programming problems. *J. Grey Syst.* **30** (2018) 127–142.
- [44] M. Nazim, M. Hashim and J. Xu, Multi objective optimization of production-distribution problem under fuzzy random environment. *Global J. Technol. Optim.* **5** (2014) 161.
- [45] R. Nourifar, I. Mahdavi, N. Mahdavi-Amiri and M.M. Paydar, Optimizing decentralized production-distribution planning problem in a multi-period supply chain network under uncertainty. *J. Ind. Eng. Int.* **14** (2018) 367–382.
- [46] M. Rabbani, R. Heidari and R. Yazdanparast, A stochastic multi-period industrial hazardous waste location-routing problem: Integrating NSGA-II and Monte Carlo simulation. *Eur. J. Oper. Res.* **272** (2019) 945–961.
- [47] H. Rafiei, F. Safaei and M. Rabbani, Integrated production-distribution planning problem in a competition-based four-echelon supply chain. *Comput. Ind. Eng.* **119** (2018) 85–99.
- [48] Z. Rafie-Majd, S.H.R. Pasandideh and B. Naderi, Modelling and solving the integrated inventory-location-routing problem in a multi-period and multi-perishable product supply chain with uncertainty: Lagrangian relaxation algorithm. *Comput. Chem. Eng.* **109** (2018) 9–22.
- [49] S.J. Sadjadi and A. Alirezaee, Impact of pricing structure on supply chain coordination with cooperative advertising. *RAIRO:OR* **54** (2020) 1613–1629.
- [50] N. Sahebjamnia, F. Goodarzian and M. Hajiaghahi-Keshteli, Optimization of multi-period three-echelon citrus supply chain problem. *J. Optim. Ind. Eng.* (2020) 41–50.
- [51] U. Sakalli and I. Atabas, Ant colony optimization and genetic algorithm for fuzzy stochastic production-distribution planning. *Appl. Sci.* **8** (2018) 2042.
- [52] J.R. Shi, S.Y. Liu and W.T. Xiong, A new solution for interval number linear programming. *J. Syst. Eng. Theory Pract.* **2** (2005) 101–106.
- [53] D. Shishebori and A. Ghaderi, An integrated approach for reliable facility location/network design problem with link disruption. *Int. J. Supply Oper. Manage.* **2** (2015) 640–661.
- [54] D. Shishebori and M.S. Jabalameli, A new integrated mathematical model for optimizing facility location and network design policies with facility disruptions. *Life Sci. J.* **10** (2013) 1896–1906.
- [55] D. Shishebori, A.Y. Babadi and Z. Noormohammadzadeh, A Lagrangian relaxation approach to fuzzy robust multi-objective facility location network design problem. *Sci. Iran. Trans. E Ind. Eng.* **25** (2018) 1750–1767.
- [56] W. Wang, Study on grey linear programming. *J. Grey Syst.* **9** (1997) 41–46.
- [57] Z.X. Wang, Correlation analysis of sequences with interval grey numbers based on the kernel and greyness degree. *Kybernetes* **42** (2013) 309–317.
- [58] D. H. Wolpert, and W. G. Macready, No free lunch theorems for optimization. *IEEE Trans. Evol. Comput.* **1** (1997) 67–82.
- [59] N.M. Xie, On computational algorithms of grey numbers based on information background. *Grey Syst.: Theory App.* **3** (2013) 177–190.
- [60] N.M. Xie and S.F. Liu, A novel grey relational model based on grey number sequences. *Grey Syst.: Theory App.* **1** (2011) 117–128.
- [61] X. Yan and Z. Song, The portfolio models of contained grey profit under uncertainty. *Grey Syst.: Theory Appl.* **4** (2014) 487–494.
- [62] A.A. Zaidan, B. Atiya, M.A. Bakar and B.B. Zaidan, A new hybrid algorithm of simulated annealing and simplex downhill for solving multiple-objective aggregate production planning on fuzzy environment. *Neural Comput. App.* **31** (2019) 1823–1834.
- [63] K. Zhang, Y. Cai, S. Fu and H. Zhang, Multi objective memetic algorithm based on adaptive local search chains for vehicle routing problem with time windows. *Evol. Intel.* (2019) 1–12. DOI: [10.1007/s12065-019-00224-7](https://doi.org/10.1007/s12065-019-00224-7).
- [64] X. Zhao and J. Dou, Bi-objective integrated supply chain design with transportation choices: a multi-objective particle swarm optimization. *J. Ind. Manage. Optim.* **15** (2019) 1263–1288.