Annals of Operations Research
https://doi.org/10.1007/s10479-022-05005-7

S.I.: COMPUTATIONAL LOGISTICS IN FOOD AND DRINK INDUSTRY

Investigating a citrus fruit supply chain network considering CO_2 emissions using meta-heuristic algorithms

Fariba Goodarzian[1] · Vikas Kumar[2] · Peiman Ghasemi[3]

Accepted: 19 September 2022

Abstract

According to the increasing carbon dioxide released through vehicles and the shortage of water resources, decision-makers decided to combine the environmental and economic effects in the Agri-Food Supply Chain Network (AFSCN) in developing countries. This paper focuses on the citrus fruit supply chain network. The novelty of this study is the proposal of a mathematical model for a three-echelon AFSCN considering simultaneously CO_2 emissions, coefficient water, and time window. Additionally, a bi-objective mixed-integer non-linear programming is formulated for production–distribution-inventory-allocation problem. The model seeks to minimise the total cost and CO^+ emission simultaneously. To solve the multi-objective model in this paper, the Augmented Epsilon-constraint method is utilised for small- and medium-sized problems. The Augmented Epsilon-constraint method is not able to solve large-scale problems due to its high computational time. This method is a well-known approach to dealing with multi-objective problems. It allows for producing a set of Pareto solutions for multi-objective problems. Multi-Objective Ant Colony Optimisation, fast Pareto genetic algorithm, non-dominated sorting genetic algorithm II, and multi-objective simulated annealing are used to solve the model. Then, a hybrid meta-heuristic algorithm called Hybrid multi-objective Ant Colony Optimisation with multi-objective Simulated Annealing (HACO-SA) is developed to solve the model. In the HACO-SA algorithm, an initial temperature and temperature reduction rate is utilised to ensure a faster convergence rate and to optimise the ability of exploitation and exploration as input data of the SA algorithm. The computational results show the superiority of the Augmented Epsilon-constraint method in small-sized

✉ Vikas Kumar
Vikas.Kumar@uwe.ac.uk

Fariba Goodarzian
faribagoodarzian@us.es

Peiman Ghasemi
Peiman.Ghasemi@gutech.edu.om

[1] Engineering Group, School of Engineering, University of Seville, Camino de los Descubrimientos s/n, 41092 Seville, Spain

[2] Bristol Business School, University of the West of England, Bristol BS16 1QY, UK

[3] Department of Logistics, Tourism and Service Management, German University of Technology in Oman (GUtech), Muscat, Oman

Published online: 13 October 2022

problems, while HACO-SA indicates that is better than the suggested original algorithms in the medium- and large-sized problems.

Keywords Citrus fruit agri-food supply chain network · CO_2 emissions · Mathematical model · Meta-heuristic algorithms

1 Introduction

Agriculture is one of the most important sectors along with oil, automotive, and the pharmaceutical industries, etc., in the economy of any country, and plays an important role in its political and economic independence (Mousavi et al., 2015). The existence of many natural gifts and special climatic conditions have turned Asia countries into a four-season land, which provides the necessary basis for focusing this sector on the country's economy (Mogale et al., 2020). One of the most obvious challenges in the agricultural sector in the country is the lack of awareness of farmers about the balanced cultivation of agricultural products on demand. The disruption of this balance, on the one hand, causes the abundance of a product, a significant reduction in its price in one year and harms farmers. On the other hand, the reduction of other products increases the price and dissatisfaction among the people (Bottani et al., 2019; Keshavarz-Ghorbani & Pasandideh, 2021).

Iran is one of the world's largest fruit producers and provides the finest conditions for citrus fruit production (Cheraghalipour et al., 2019). Hence, the citrus supply chain network (CSCN) is one of the main parts of the agriculture-food supply chain in Iran. The definition of the CSCN illustrates the activities of production to distribution of agricultural and horticultural products from farms. One of the fundamental factors in the CSCN is to increase the quality and safety of foods and other variables relevant to weather conditions (Mousavi et al., 2015; Sharma et al., 2020). Supply chains are primarily responsible for the production or assembly of products and their distribution. In the production and distribution stages, there is always a need to check the inventory of products. The reason for this is to satisfy customer demand and prevent shortages of products. Previous studies have often focused on one aspect of production, distribution, and inventory, but integrating these three issues brings the issue closer to the real world. The integrated production and distribution problem is very complicated due to its many variables and constraints. Therefore, many techniques were applied to find the best solution for this problem within a reasonable time.

Today's world is facing a series of environmental challenges, many of which have led to environmental crises, inflicting irreparable damage that has often been impossible to prevent from spreading and, even if possible, there are many costs involved. The CO_2 emissions and greenhouse gases pose a major threat to the environment. An agri-food supply chain is a network of food-related business enterprises through which food products move from production through consumption (Kamble et al., 2020; Soto-Silva et al., 2016) and in the process contribute to CO_2 emissions.

Agriculture crops refer to the part of agricultural products in which the time from planting to flowering (end of life of the crop) is less than one year. In fact, their life cycle lasts less than a year, while agricultural products have a long life cycle and, after flowering, the plant does not end its life and can be harvested in successive periods without replanting. Another issue that needs to be addressed is irrigated and rainfed cultivation (Sellitto et al., 2018). In this regard, irrigated cultivation is a type of agricultural production in which farmers provide the water needed by the plant promptly, while, in rainfed cultivation, farmers are hoping to get the required water from rainfall during the growing, cultivating, and harvesting stages.

To tackle the current challenges in the citrus fruit supply chain, the Ministry of Agriculture Jihad (MAJ) in Iran is moving toward the citrus fruit supply chain network in terms of holding, transportation, and storage. In this network, citrus fruit is transported in the box using the truck as well as specially designed Nissans and stored in specially designed citrus fruit storage. Proper planning and coordination among all the entities of the citrus fruit supply chain network are essential to reduce transportation, inventory holding, and waste costs (Sgarbossa & Russo, 2017). Likewise, the determination of each type of capacitated vehicle used for transporting between various entities is also a crucial aspect of the citrus fruit supply chain problem (Rong et al., 2011). Sufficient availability of capacitated vehicles helps in the quick transfer of citrus fruit from producer to consumer cities (Roghanian & Cheraghalipour, 2019). In the existing literature, there is currently no comprehensive planning to produce citrus fruit (see for example Rong et al., 2011; Sgarbossa & Russo, 2017; Roghanian & Cheraghalipour, 2019; Mogale et al., 2020).

In this paper, therefore, a new bi-objective, multi-echelon, and multi-period agriculture food supply chain network (AFSCN) model for a production–distribution-inventory problem considering CO_2 emissions and time windows is developed. The main reason for using time windows is to prevent corruption of products. Therefore, in the proposed supply chain, the production process is performed in farms and inventory control is performed at fridges and farms levels. Distribution operations are also performed from farm to fridges and from fridges to citrus markets.

In this regard, a Mixed-Integer Non-Linear Programming (MINLP) model is formulated for the current problem. The main aim of this study is to minimise the total costs (transportation, inventory holding, warehouse, waste, and operational costs) and the released CO_2 emissions through transportation systems. In the proposed model, the optimal amount of the production of the agricultural foods (citrus fruits) in the farms according to the location of the fridges for citrus holding and the allocation of citrus fruits to them according to the shelf life of the citrus fruits are considered. In the AFSCN, the demand areas are citrus fruit markets (in urban areas), the supply areas are large and small farms can be considered as centralised areas in each urban area. Another novelty in this paper, based on the production planning problem, is a new method for monitoring the perishability of citrus fruit and combining locations. In addition, a performance coefficient for saving agricultural land and water consumption is considered for the first time. To solve the AFSCN model and find Pareto solutions, simulated data as test problems including small-, medium-, and large-sized problems are considered. Therefore, the CPLEX method using GAMS software is employed to solve the presented model in the small-sized problem. Then, meta-heuristic algorithms containing multi-objective ant colony optimisation (MOACO), Fast PGA, non-dominated sorting genetic algorithm II (NSGA-II), and multi-objective simulated annealing (MOSA) algorithms are used to solve the mathematical model in different size problems. Additionally, a novel hybrid heuristic method based on multi-objective ant colony optimisation (ACO) and multi-objective simulated annealing (SA) algorithms, called the HACO-SA algorithm, is developed in this paper. In summary, the present paper has the following novelties:

In summary, the present paper has the following key contributions:

- Designing a bi-objective mathematical model for production-allocation-inventory problem in the AFSCN for citrus fruit and analysing it,
- Formulation of an MINLP model considering time windows for servicing to citrus fruit markets in the AFSCN problem and analysing it,

- Considering CO_2 emissions as environmental effects through used vehicles in the proposed network and analysing it,
- Employing the performance coefficient of the citrus fruit and needed water for the production of the citrus fruit in the suggested model,
- Considering the capability to re-cultivate on the agricultural land after citrus fruit harvest,
- Developing a new heuristic method based on a meta-heuristic algorithm (HACO-SA algorithm) to solve the mathematical model and comparing it with other methods.

The rest of this work is as follows: Sect. 2 presents the literature review. The problem description and mathematical model along with notations, objective function, and constraints are demonstrated in Sect. 3. The solution methodologies to solve the AFSCN model are stated in Sect. 4. The implementation of the presented model along with the numerical results and sensitivity analysis of the models are proposed in Sect. 5. Finally, Sect. 6 presents the conclusions and future work.

2 Literature review

A review of existing studies shows that literature related to the agri-food supply chain networks is scant. This section, therefore, reviews the previous studies in the fields of (I) the agri-food supply chain network along with meta-heuristic algorithms and (II) the agri-food supply chain management.

2.1 Meta-heuristic algorithms in agri-food supply chain network

Osvald and Stirn (2008) presented an algorithm for the distribution of fresh vegetables. They provided a vehicle routing problem with time windows and time-dependent travel times where the travel times between two locations depended on both the distance and the time of the day. Validi et al. (2014a) designed an integrated low-carbon/green distribution system. The DoE-guided MOPSO approach was used to solve their proposed model. A scenario analysis of the realistic vehicle routes is provided by considering alternative possible outcomes for further guidance to DMs. In another work, Validi et al. (2014b) presented a sustainable multi-objective model for food supply chain management. They evaluated distribution and routing problems in a two layers green supply chain network. Their main goal was to minimise supply chain costs and carbon dioxide emissions. They investigated a real case study in the Irish dairy. Moreover, Validi et al. (2015) developed a sustainable distribution model to minimise costs and carbon dioxide emissions in the Irish dairy processing industry supply chain. In their study, experts prioritised realistic routes with alternate transportation scenarios. Masson et al. (2016) developed a two-stage approach according to an adaptive large neighbourhood search to solve the dairy supply chain network model. The first phase solved the transportation problem and the second phase ensured that the optimisation of plant assignment was performed. Mogale et al. (2017a) investigated the multi-period multi-modal bulk wheat transportation and storage problem in a two-stage supply chain network of public distribution system while seeking to minimise the total costs. They developed an MINLP after studying the Indian wheat supply chain scenario. In addition, they suggested and used two meta-heuristic algorithms. Mogale et al. (2017b) designed a multi-period inventory transportation model for tactical planning of the food grain supply chain. Then, they formulated an MINLP model, while their main aim was the minimisation of the cost of the food grain supply chain. Cheraghalipour et al. (2019) considered the rice supply chain and also presented a bi-level optimisation model for the rice supply chain. They designed an MILP mathematical model, while investigating minimising total cost concerning the two decision-makers' opinions. Thus, two meta-heuristic algorithms were presented and two hybrid algorithms developed. Finally, they studied a real case study in Iran. Sahebjamnia et al. (2020) developed a new multi-objective integer non-linear programming model for designing citrus fruit three-echelon supply chain network. They considered two objective functions including minimising network costs and maximising the network's profits. To solve their model, firstly, the model was converted to a linear programming model. Then, they used three multi-objective meta-heuristic algorithms for finding efficient solutions. The outcomes of their proposed algorithms were compared by several assessment criteria including a number of Pareto solutions, maximum spread, mean ideal distance, and diversification metric. Dwivedi et al. (2020) proposed an MINLP model for a two-echelon food grain supply chain along with sustainability aspects (carbon emissions). Their main aim was minimising the total transportation cost and carbon emission tax in gathering food grains from farmers to the hubs and later to the selected demand points (warehouses). They used two meta-heuristic algorithms to solve their model and also adopted simulated data to validate their model. Validi et al. (2020) proposed a robust mathematical model for routing and location in the three-echelon supply chain network. Their proposed chain included factories, distribution centres, and retailers. The DoE-guided Multiple-Objective Genetic Algorithm of type II (MOGA-II) was used to solve their proposed model. They considered a real case study in the east of Ireland. Liao et al. (2020) developed a new MILP model for a citrus fruit closed-loop supply chain along with minimising total costs and air pollution. They used meta-heuristic algorithms to solve their model. Assessment criteria were used to compare their proposed algorithms. To validate their model, they conducted a real case study. In this regard, a four-echelon citrus fruit supply chain including gardeners, distribution centres, citrus fruit storage, and fruit market was developed by Fakhrzad and Goodarzian, (2021). They formulated an MINLP model, which showed minimising the total cost and maximising the profit of the citrus fruit supply chain network. To solve their model, ant colony optimisation and simulated annealing algorithms were used. Gómez-Lagos et al. (2021) proposed an MILP model for a fresh fruit supply chain network based on supporting tactical decisions during the harvest season in order to reduce total costs. They used a greedy randomised adaptive search procedure meta-heuristic algorithm to solve their model. Finally, they provided a real case study in the fruit market field. Validi et al. (2021) presented a bi-objective mathematical model for location, routing, and distribution centres in a three-echelon sustainable supply chain. The main purpose of their proposed model was to minimise supply chain costs while minimising carbon dioxide emissions. DoE-guided meta-heuristic-based model was used to solve their proposed model. Due to the two objectives of their proposed model, they used the TOPSIS approach to prioritise Pareto points and meta-heuristic algorithms to solve their model.

2.2 Agri-food supply chain network

Rong et al. (2011) developed an optimisation method to manage fresh food quality in the food supply chain network. Additionally, they integrated food quality in decision-making on production and distribution in a food supply chain. Also, they designed a mixed-integer linear programming (MILP) model for production and distribution planning problems. Their resulting model is used in a real case study and is able to use to design and operate food distribution systems, utilising both food quality and cost criteria. Etemadnia et al. (2015) designed a fruit and vegetable supply chain network based on an optimal national wholesale or hub location network for serving food consumption markets through efficient connections with production sites. Therefore, they formulated an MILP model that decreases the total costs including transportation costs of products and the location of the facilities. To solve their model, heuristic and exact methods are presented. A scenario study is utilised to investigate their model's sensitivity to parameter changes. An application is made to the U.S. fruit and vegetable industry. Bortolini et al. (2016) suggested a three-objective distribution problem for fresh food network along with minimising carbon footprint, operating cost, and delivery time. They formulated a linear programming (LP) model considering the farmer production capacities, the food quality dependence on the delivery time and the geographically distributed market demand. To validate their model, they applied a real case study dealing with the distribution of fresh fruits and vegetables in Italy. Lamsal et al. (2016) proposed an integrated linear programming (ILP) model for planning the movement of the crop from the farm to the processing plant. They considered economically significant agricultural systems in the United States including sugarcane in Louisiana and sugar beets in the northern areas of the United States. Eventually, they demonstrated that their model is computationally tractable by introducing new datasets according to the sugarcane industry in Louisiana. Hyland et al. (2016) introduced conceptual and mathematical models of the domestic grain supply chain including elevator storage, rail transportation, and trucking. They compared conventional rail service supported by country elevators with shuttle service supported by terminal elevators with critical transportation service dimensions divided into three categories incorporating capacity, travel time, and cost. The time and cost model outcomes showed that shuttle service carried grain faster and diminished supply chain costs more than conventional service, even after considering trucking and elevator storage. Their rail capacity model outcomes demonstrated that shifting grain from conventional to shuttle service considerably raises rail capacity. Accorsi et al. (2016) considered the food supply chain as an ecosystem and defined more inclusive boundaries. They presented a design framework that supports strategic decision-making on agriculture and food distribution issues while addressing climate stability. They merged an original land-network problem with localised and large-sized decisions as land-use allocation and location-allocation problems in an agro-food network. A linear programming model that optimises infrastructure, agriculture, and logistics costs was formulated as well as carbon emissions within the agro-food ecosystem were balanced. They suggested a regional potato supply chain for the effectiveness of their model. Jabarzadeh et al. (2020) presented a closed-loop supply chain network optimisation problem for a perishable agricultural product to attain three pillars of sustainability. They formulated a multi-objective MILP model and used the LP-Metric and weighted Tchebycheff approaches to solve their model. A set of experiment problems was provided to validate their model. Motevalli-Taher et al. (2020) optimised a sustainable wheat supply chain network and formulated a multi-objective mathematical model in order to minimise network costs and water consumption and maximise job opportunities. They used the meta-goal programming approach for transforming the multi-objective to the single-objective problem. Thus, they employed a simulation approach to cope with the uncertainty of the demand for wheat flour. Finally, a real case study was provided to assess their proposed model. Mehrbanfar et al. (2020) presented an efficient network of greenhouses, agricultural lands, processing plants, and agricultural distribution centres (farmer's markets) according to sustainability concepts. Also, they used an augmented ε-constraint approach to solve the model and provided a real case study in Iran to investigate their model. Varas et al. (2020) proposed a new multi-objective MILP model to support wine grape harvesting. They considered the opposing nature of operational cost minimisation and grape quality maximisation, subject to several constraints, such as grape requirements and routing decisions. They developed a negotiation protocol that can lead to an agreed final harvest schedule according to the operations of a winery they worked with. Based on the augmented weighted Tchebycheff method, the protocol included an initial obtained Pareto optimal solution. To solve their model, they used the augmented ϵ -constraint approach. Yakavenka et al. (2020) extended a multi-objective model for sustainable supply chains. They formulated an MILP model for this network of perishable food products while minimising the total costs, social-time, and emissions. Finally, their model was implemented in the case of a fruit importer in the North-Eastern European region considering its geographical settings.

Table 1 shows the summary of the examined papers in the literature review.

According to the examined papers, the main gaps are explained as follows:

- In terms of mathematical modelling, they didn't pay attention to a multi-period and bi-objective model for designing an AFSCN considering CO_2 emission and time windows that are designed in this paper.
- Other differences between our paper and studied papers are, first of all, the considered citrus fruit is orange, which is a perishable product. For this reason, the waste cost is considered in this network. Secondly, some studies have examined a single-objective mathematical model (minimising total costs), but a bi-objective model including minimising total cost and minimising CO_2 emission is formulated and designed simultaneously in this paper. Thirdly, the performance coefficient and needed water for the production of the citrus fruit (orange) in the AFSCN model are considered. Additionally, the capability to re-cultivate in the agricultural land after the orange harvest is considered in this paper which the examined papers did not.
- Unlike previous models, which are defined as separate variables and constraints for staying in the fridge/cold storage and imposing perishability constraints, this variable considers simultaneously both the sent amount to the fridge/cold store and the duration of staying in the fridge/cold storage. In this paper, planning for the citrus fruit (orange) is performed with irrigated cultivation. Due to limited water resources in some regions, water resource constraints are considered that were ignored in the examined papers in the literature review.
- In terms of solution methodology, a significant gap is specified. Since the AFSCN model is an NP-hard problem, exact solvers cannot solve large-scale problems in a reasonable time. Several studied papers have used metaheuristic algorithms, including NSGA-II, MOGA, GA, MOPSO, CRO, TS, and SADE algorithms, to solve their model.

3 Problem description

In this paper, the proposed AFSCN problem considers three echelons including (I) citrus fruit (orange) farms that can be planted in different periods, (II) the fridges to hold citrus fruit, which must be selected among a number of candidate points, and (III) citrus fruit markets (demand points) where the location of supply the products to the final customer and then their location and demand for each period is determined in advance. In this regard, the orange harvest season is from late November to mid-February, which indicates the multi-period nature of the proposed model. The types of flow in the presented network are considered as follows (see Fig. 1):

- The flow of product transfer from farms to fridges and citrus fruit markets,
- The flow of product transfer from the fridge to the citrus fruit markets, including products that are stored in the fridge for at least a period, or products that are not stored, and only due to the lower cost of transportation through the fridge to the fruit market is transported.

Table 1 The summary of the examined papers (*S* sustainable, *SCN* supply chain network, *CLCN* closed-loop supply chain)

References	Single/Multi-period	Model	Objective			The type of problem	The type of network	Time window	Product	Solution methodology
			Min time	CO_2 emission	Min cost					
Osvald and Strin (2008)	Single	MILP	X	–		Distribution	SCN	X	Vegetables	Exact
Rong et al. (2011)	Multi	MILP	–	–	X	Production, Distribution	SCN	–	Fresh food	CPLEX
Validi et al. (2014a, 2014b)	Multi	MILP	–	X	X	Distribution	SCN	–	Dairy food	MOPSO
Validi et al. (2014a, 2014b)	Multi	MILP	–	X		Distribution	SCN	–	Food, milk	TOPSIS, NSGA–II, MOGAII, HYBRID
Validi et al. (2015)	Multi	MILP	–	X	X	Distribution	S	–	Dairy food	MOGA–II
Etemadnia et al. (2015)	single	MILP	–	–		Hub location	SCN	–	Fruit and vegetables	Heuristic, Exact
Masson et al. (2016)	Multi	MINLP	–	–	X	Transportation, Allocation	SCN	–	Dairy products	Adaptive large neighbored search

Table 1 (continued)

References	Single/Multi-period	Model	Objective			The type of problem	The type of network	Time window	Product	Solution methodology
			Min time	CO$_2$ emission	Min cost					
Accorsi et al. (2016)	Multi	MILP	–	X	X	Location, Allocation	SCN	–	Potato	CPLEX
Mogale et al. (2017a)	single	MINLP	–	–	X	Distribution	SCN	–	Wheat	CRO, TS hybrid CROTS
Cheraghalipour et al. (2019)	Multi	MILP	–	–	X	Production, Distribution	SCN	–	Rice	GA, PSO, HGA–PSO, HPSO–GA
Motevalli-Taher et al. (2020)	Multi	MILP	–	–	X	Production, Distribution	S	–	Wheat	Meta–goal programming
Validi et al. (2020)	Multi	MILP	–	X	X	Location, Routing	SCN	–	Dairy food	MOGA–II
Yakavenak et al. (2020)	Multi	MILP	X	X	–	Distribution		–	Perishable food	CPLEX
Sahebjamnia et al. (2020)	Multi	INLP		–	X	Production, Distribution	SCN	–	Citrus	CLPEX, LINGO, MOPSO, MOICA, NSGA–II
Dwivedi et al. (2020)	Multi	MINLP		X	X	Hub location	SCN	–	Grain	GA
Liao et al. (2020)	Multi	MILP		–	X	Distribution	CLSC	–	Citrus	Keshtel
Fakhrzad and Goodarzian (2021)	Multi	MINLP	–	–	X	Production	SCN	–	Citrus	ACO, SA

Table 1 (continued)

References	Single/Multi-period	Model	Objective			The type of problem	The type of network	Time window	Product	Solution methodology
			Min time	CO_2 emission	Min cost					
Gómez-Lagos et al. (2021)	Single	MILP	–	–	X	Production	SCN	–	Fresh fruit	Greedy randomized adaptive search procedure
Validi et al. (2021)	Multi	MILP	–	X	X	Location, Routing, Distribution	S	–	Food	Design of experiment
This paper	Multi	MINLP	–	X	X	Production, Distribution, Inventory	SCN	X	Citrus fruit	HACO–SA, MOSA, Fast PGA, NSGA–II, MOACO

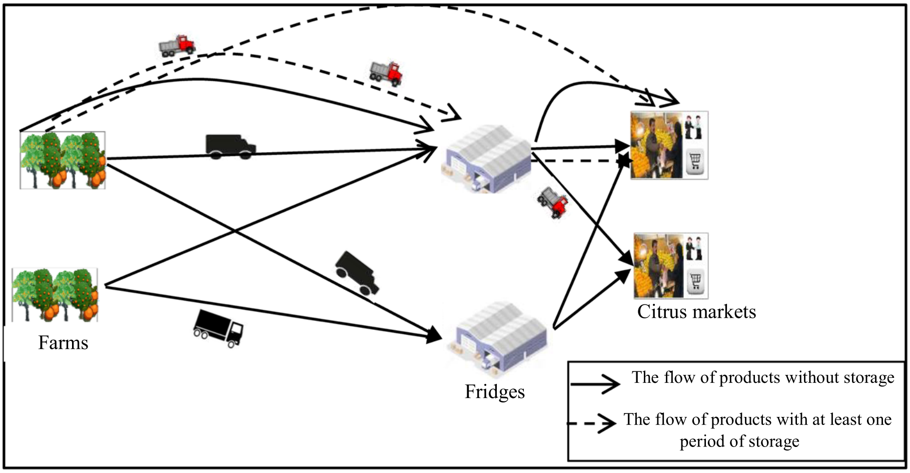

Farms
Fridges
Citrus markets
The flow of products without storage
The flow of products with at least one period of storage

Fig. 1 The structure of the proposed network in this paper

The main aims of the proposed model include minimising the total costs (production, transportation, fridge rental, and inventory holding) and the released carbon dioxide emissions through vehicles as environmental effects simultaneously, which are considered in this paper. The Ministry of Agriculture Jihad (MAJ), Iran, is considered as operators (managers) at all stages of the chain. Therefore, the production of the citrus fruit (orange) is the responsibility of the farmers, but it is assumed that the MAJ pays the farmers the cost of production and the profit of the farmers in the form of the guaranteed purchase at harvest time. Also, this is considered a production cost for the MAJ. In the presented model, the planning time evaluates for the cultivation and transportation of citrus fruit and considers other agricultural technical requirements. In this paper, some of them are mentioned as follows:

- Perishable concept: according to this concept, the shelf life of the citrus fruit (orange) from the harvest time to the time of reaching the demand points should be less than the perishable time of the product.
- Processing time: the time from cultivating to harvest is called the processing time of the product; therefore, post-cultivating, the products must pass the necessary processing time until harvest.
- Possibility of re-use of the citrus fruit garden: one of the characteristics of citrus fruit cultivating is the possibility of re-use of the citrus fruit garden, which based on post-harvest the citrus fruit from the citrus fruit garden can be again cultivated on the citrus fruit garden. All the papers related to the supply chain of perishable products, because of planning for a period of cultivation, have ignored this feature.
- Cultivation capability: cultivation of products in each region is allowed in certain periods; for example, for the citrus fruit (orange) product in Mazandaran province, Iran, there is the ability to cultivate during the autumn and winter seasons and there is no ability to cultivate in other seasons.
- Water resources: in this paper, planning for the citrus fruit (orange) is performed with irrigated cultivation. Due to limited water resources in some regions, water resource constraints are considered in the AFSCN model as a novelty.

- Fridge/cold storage: in the proposed model, there are a specific number of fridges, in which a number of them are to be selected. The capacity of each fridge is limited. Each fridge has a fixed cost called the fridge rental during the planning horizon.
- Accessible budget: the accessible budget for fridge rental is limited.
- Minimum income of farmers: in order to motivate farmers to follow the provided planning by the AFSCN model, earning the minimum income for farmers is considered.
- The minimum acceptable amount of cultivating: according to this limitation, cultivating the citrus fruit in each period will be unauthorised if less than a low limit.

The reasons for the f agricultural technical requirements in this network are divided into several categories including one aspect of the impact of fridges (cold storage) and its direct impact on reducing citrus fruit perishes/waste. Also, the existence of cold storage to store citrus fruit and processed products in one area can affect various economic and environmental conditions. Another influential and important parameter in the presented model is the available budget for fridge/cold storage rental, which controls the number of fridges/cold storages that can be rented. As water is the most essential agricultural requirement, it's important to consider this in the cultivation planning of citrus fruit.

The details of the assumptions of the proposed mathematical model are explained as follows:

- The cultivation of the agricultural products (citrus fruit/orange) is considered as an irrigated cultivation.
- For each amount of citrus fruit (orange) cultivated, after the processing period, it is harvested to the same required amount.
- A time window is considered to respond to citrus fruit market demands.
- After harvesting and before transferring the products to the citrus fruit market or fridge/cold storage, the decayed orange is removed.
- The planning horizon is included a number of specified periods.
- The products become completely corrupt and unusable after the end of the shelf life.
- If there is a cultivation or harvesting operation in a period, this operation is performed at the beginning of the period.
- The required time to transport the products is assumed to be negligible.
- The transportation cost of the oranges depends on their weight.
- The holding cost of the oranges depends on the weight and duration of holding.
- The transport fleet is considered homogeneous. The problem dealt with a homogeneous fleet and assumes that each vehicle has the same characteristics, such as capacities and cost.,
- In the proposed supply chain in the process of production and holding of inventory, CO_2 emission is not considered as it only occurs in the distribution process,. The reason for this is that the amount of carbon released during the production and holding phase is very small compared to the distribution process and can be ignored.

3.1 Notations

{Insert
notations}

3.2 The AFSCN model

$$MinZ_1 = \sum_m \sum_i \sum_t (CP_{mit}X_{mit} + CV_{mit}V_{mit}) + \sum_m \sum_i \sum_j CA_{mij}dA_{ij}\left(\sum_t xa_{mijt}\right)$$
$$+\sum_m \sum_j \sum_l CB_{mjl}dB_{jl}\left(\sum_t \left(xb_{mjlt} + \sum_n xbd_{mjl(t-n)n}\right)\right) + \sum_m \sum_i \sum_l CC_{mil}dC_{il}\left(\sum_t xc_{milt}\right)$$
$$+\sum_j \sum_T F_{jT}R_{jT} + \sum_m \sum_j \sum_t I_{mjt}CH_{mjt} \qquad (1)$$

The first objective function (1) refers to the minimisation of the total costs including fixed costs for cultivating produces, the transportation cost from farms to fridges, from fridges to citrus fruit markets, and from farms to citrus fruit markets, the cost of fridge rental, production cost, and holding cost of the citrus fruit in the fridge.

$$Min\ Z_2 = \zeta\left(\sum_m \sum_i \sum_j \sum_t dA_{ij} \times \frac{xa_{mijt}}{\sigma} \times \eta\right.$$
$$\left. + \sum_m \sum_j \sum_l \sum_t dB_{jl} \times \frac{xb_{mjlt}}{\sigma} \times \eta + \sum_m \sum_i \sum_l \sum_t dC_{il} \times \frac{xc_{milt}}{\sigma} \times \eta\right) \qquad (2)$$

The second objective function (2) seeks to minimise the total released CO_2 emissions by vehicles which are formulated based on distances between networks, required fuel for the vehicle at each kilometre of transporting, vehicle capacity, and the amount of transported citrus fruit.

$$Constraint K_{it} = K_{i(t-1)} - \sum_m X'_{mi(t-1)}g_{mi} + \sum_m X'_{mit}g_{mi} \quad \forall i, t \qquad (3)$$

Constraint (3) shows the cultivable level on the farm i to the period t. Therefore, this amount is equal to the cultivable level in the previous period minus the occupied level by cultivation in the previous period plus the released level from the harvested products at the beginning of the same period.

$$X_{mit} = X'_{mi(t-q_m)} \quad \forall i, m, t \qquad (4)$$

The amount of obtained product (X_{mit}) at the period t is equal to the amount of the planted product ($X^{'}{}_{mi(t-q_m)}$) at the period $t - q_m$ is indicated in constraint (4).

$$\sum_m X'_{mit}g_{mi} \le K_{it} \quad , t \qquad (5)$$

Constraint (5) indicates the cultivating capacity of the citrus fruit and also guarantees the cultivating rate of different products on each farm in each period should be less than the cultivable surface capacity.

$$X_{mit}(1 - \alpha_{mit}) = \sum_l xc_{milt} + \sum_j xa_{mijt} \quad \forall i, t, m \qquad (6)$$

Constraint (6) depicts that the produced product will be allocated to fridges or fruit markets after deducting the percentage of waste. As such, the balance of the product on the farm is determined.

$$I_{mjt} = I_{mj(t-1)} \sum_j xa_{mijt} - \left(\sum_l \left(xb_{mjlt} + \sum_n xbd_{mjl(t-n)n}\right)\right) \quad \forall j, t, m \qquad (7)$$

In addition, the amount of inventory of each fridge in each period and per each product is shown by constraint (7). This amount is equal to the inventory of the previous period plus the total input of the product in that period minus the total output of the product in that period.

$$\sum_i xa_{mijt} = \sum_l \left(xb_{mjlt} + \sum_n xbd_{mjl(t-n)n} \right) \quad \forall j, t, m \tag{8}$$

Constraint (8) proves a balance in fridges that, in each period, the total entered product from all farms into one fridge is equal to the amount transferred to the citrus fruit markets without storage plus the amount of products entering the fridge for holding.

$$\sum_l \sum_m (CP_{mit} X_{mit} + CV_{mit} V_{mit}) \geq MinP_i \quad \forall i \tag{9}$$

The minimum income of farmers is shown in constraint (9). In order to motivate farmers to follow the presented programme through this model, earning a minimum income for farmers is considered.

$$\sum_m I_{mjt} \leq S_j \sum_T R_{jT} \quad \forall j.t \tag{10}$$

Constraint (10) also states that the amount of storage of all products in the fridge j in each period must be less than the capacity of that fridge if the R_j variable equal is 1, and that fridge has been rented over the total planning horizon.

$$\sum_T R_{jT} F_{jT} \leq B \quad \forall j \tag{11}$$

Constraint (11) guarantees that the renting cost of selected fridges should not more than the maximum available budget (B). The maximum total budget is also determined based on the opinion of the government and distributors.

$$xa_{mijt} \leq \sum_T R_{jT} M \quad \forall i, j, t, m \tag{12}$$

Constraint (12) states that the product can be sent only if the fridge is selected. Based on the statement of MAJ and experts, citrus fruit (oranges) will be transported to the selected fridges/cold storages and, usually, only one fridge/cold storage is selected for a farm.

$$L_{mit} V_{mit} \lambda_{mit} \leq X'_{mit} \leq V_{mit} \lambda_{mit} M \quad \forall i, t, m \tag{13}$$

Constraint (13) ensures that the planting of the product per each farm and in each period based on the setting up of the planting was performed and the ability to plant the product. Additionally, if the planting of the product is performed, the amount must be greater than the specified minimum value. The minimum acceptable amount of the cultivating product on the farm is based on the opinion and agreement of the environmental organisation and the farmer.

$$\varphi_l = \max((\varpi_l - \tau_l), 0) \quad \forall l \tag{14}$$

$$\Omega_l = \max((\tau_l + \Delta_l - \mu_l), 0) \ \forall l \tag{15}$$

$$\sum_t \sum_m X'_{mit} w_m g_{mi} \leq TW_i \quad \forall i \tag{16}$$

Constraints (14) and (15) determine the rate of violation of the time window for servicing to citrus fruit markets. The logic of these two constraints (i.e., 14 and 15) is based on the upper and the lower limit of the time window. If the arrival time of the vehicle to the citrus fruit market is greater than the range of time window, the amount of violation is greater. Constraint (16) indicates the amount of accessible water per farm.

$$R_{jT}, V_{mit}, \lambda_{mit} \in \{0, 1\} \quad \forall j, T, m, i, t \tag{17}$$

$$X_{mit}, X'_{mit}, xa_{mijt}, xc_{milt}, xbd_{mjltn}, xb_{mjlt}, \emptyset_l, \Omega_l, I_{mjt}, K_{it} \quad \forall l \tag{18}$$

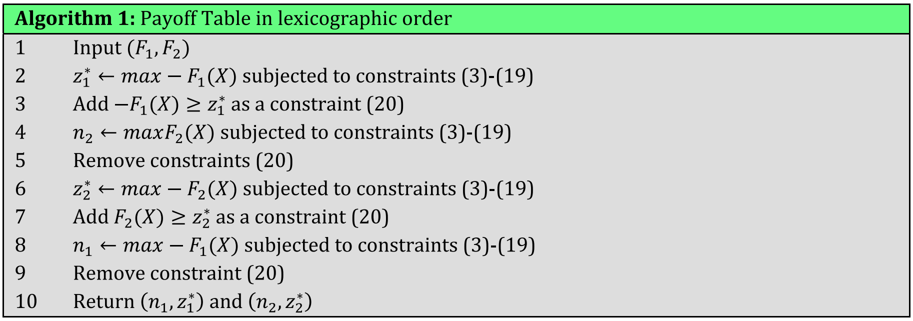
Constraints (17) and (18) display the binary and continuous variables in the AFSCN model, respectively.

4 Solution approach

It has been proven that supply chain models are NP-hard (Goodarzian et al.,). Thus, meta-heuristic algorithms are used to solve these NP-hard problems. In this regard, to solve the proposed model, four well-known meta-heuristic algorithms, including MOACO, MOSA, NSGA-II, and Fast PGA algorithms, along with a hybrid method based on multi-objective SA and multi-objective ACO, are suggested. Then, the multi-objective solution technique (augmented Epsilon-constraint method), encoding scheme, and mentioned algorithms are discussed. It should be noted that, in this research, the augmented Epsilon-constraint approach along with meta-heuristic approaches has been used to solve small- and medium-scale problems. Then, the accuracy of meta-heuristic algorithms is compared with the augmented Epsilon-constraint algorithm. If the accuracy of the model is proved, the proposed model will be solved in large sizes using meta-heuristic approaches. The explanations of the MOSA, MOACO, NSGA-II, and Fast PGA are stated in Appendix A–D. Then, there are several reasons to use the presented meta-heuristic algorithms including the proposed algorithms that use elitism and a crowded comparison operator to keep diversity without specifying any additional parameters. These algorithms can cope with continuous search spaces. In addition, distributed computation in the suggested algorithms evades premature convergence. All presented algorithms utilise a population-based evolutionary algorithm. The presented algorithms are especially useful when time and resource constraints permit only a small number of solution evaluations, which cause to boost the algorithm's efficiency in finding Pareto optimal solutions while decreasing computational effort. The represented algorithms can converge to a true global optimum and also are useful in scientific research and engineering. Additionally, the proposed meta-heuristic algorithms have a very fast rate of convergence and reduced computational time. For this reason, these algorithms are used for comparison together in this paper.

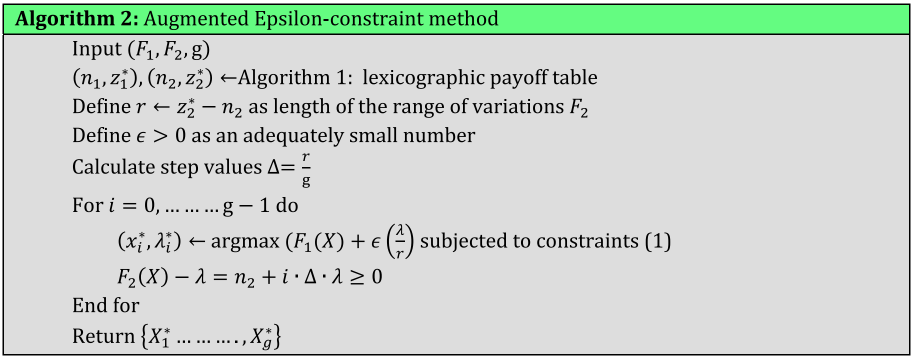
4.1 Augmented Epsilon-constraint method

In multi-objective design, a single solution does not necessarily optimise all-objective functions at a time. For this reason, non-dominated solutions that address Pareto optimal are used for the concept of optimality. The Pareto set refers to the generated feasible solution for a problem. The decision-makers choose a solution from this set according to the dominance principle. There is a necessity to generate these optimal fronts iteratively. This work utilises the augmented-constraint method to generate the optimal fronts (Mavrotas, 2009).



```
Algorithm 1: Payoff Table in lexicographic order
Input $(F_1, F_2)$
$z_1^* \leftarrow max - F_1(X)$ subjected to constraints (3)-(19)
Add $-F_1(X) \geq z_1^*$ as a constraint (20)
$n_2 \leftarrow maxF_2(X)$ subjected to constraints (3)-(19)
Remove constraints (20)
$z_2^* \leftarrow max - F_2(X)$ subjected to constraints (3)-(19)
Add $F_2(X) \geq z_2^*$ as a constraint (20)
$n_1 \leftarrow max - F_1(X)$ subjected to constraints (3)-(19)
Remove constraint (20)
Return $(n_1, z_1^*)$ and $(n_2, z_2^*)$
```

Fig. 2 The pseudo-code of the payoff Table in lexicographic order



```
Algorithm 2: Augmented Epsilon-constraint method
Input $(F_1, F_2, g)$
$(n_1, z_1^*), (n_2, z_2^*) \leftarrow$Algorithm 1:  lexicographic payoff table
Define $r \leftarrow z_2^* - n_2$ as length of the range of variations $F_2$
Define $\epsilon > 0$ as an adequately small number
Calculate step values $\Delta = \frac{r}{g}$
For $i = 0, \dots\dots\dots g - 1$ do
    $(x_i^*, \lambda_i^*) \leftarrow$ argmax $(F_1(X) + \epsilon\left(\frac{\lambda}{r}\right)$ subjected to constraints (1)
    $F_2(X) - \lambda = n_2 + i \cdot \Delta \cdot \lambda \geq 0$
End for
Return $\{X_1^* \dots\dots\dots, X_g^*\}$
```

Fig. 3 The pseudo-code of the augmented Epsilon-constraint method

The most important observation in this method is that one objective is kept as a constraint while optimising the other objective. This method starts with computing the Payoff Table in lexicographic order. The Augmented Epsilon-constraint approach solves many of the problems of the Epsilon-constraint approach. One of the advantages of this approach is the ease of value selection for the ε vector. Another advantage of this approach over the Epsilon constraint is that it solves the problem with a more logical number of iterations than the traditional epsilon constraint method (Cooper et al., 2017; Mavrotas & Florios, 2013; Yang et al., 2014, 2021; Zhang & Reimann, 2014;). The information of this method is shown in Fig. 2. z_1^* denotes the optimal value for maximising -F_1 and m_2 denote the maximum value of F_2 for constraint-$F_1(X) \geq z_1^*$. The same procedure is repeated for z_2^* and m_1. Here, $\left[n_2, z_1^*\right]$ and $\left[n_1, z_2^*\right]$ are ranges for F_1 and F_2 respectively. Figure 3 represents the process procedure (Pseudocode) of the augmented Epsilon-constraint method.

4.2 HACO-SA algorithm

In this section, to solve the proposed model, a heuristic method based on meta-heuristic algorithms called HACO-SA is developed for solving the bi-objective production–distribution-inventory holding problem. The details of the developed algorithm are proposed as follows:

In the developed algorithm, HACO-SA focuses on an initial temperature of the multi-objective SA operator in the main loop of the multi-objective ACO algorithm. Moreover, the initial temperature causes HACO-SA to have a much higher convergence speed and efficiency than other suggested algorithms. The multi-objective ACO algorithm is improved by using the initial temperature operator of the multi-objective SA algorithm. Furthermore, the proposed algorithm causes an ant that isn't limited to the pheromone trail which is already aggregated in its march process; however, it utilises both the search (exploration) of new routes and the information (exploitation) which has been detected. This increases the system's robustness significantly. On the other hand, the multi-objective ACO algorithm still falls into a local, at least in the trial process. In this regard, this process causes a route that doesn't lead the ant to expand the Pareto solution so that it provides the total system premature convergence. When the local minimum situation appears, we add an initial temperature in the main loop to avoid this premature convergence and run away from a local minimum. Generally, by adding initial temperature in the main loop of the multi-objective ACO algorithm, HACO-SA causes the near-optimal solutions to run away from the trap of a local minimum and to expand in the direction of the optimisation.

Hybrid meta-heuristic algorithms show an important role in improving the search capability of algorithms. In terms of hybridisation, the main goal of the hybridisation of the meta-heuristic algorithm is to decrease any significant disadvantage (Poorzahedy et al., 2007). Generally, the results of the hybridisation can perform several improvements according to the accuracy or computational speed. In a hybrid algorithm, two or more meta-heuristic algorithms are cooperatively and collectively solving a predefined problem (Poorzahedy et al., 2007). In order to find better solutions by searching for the optimal parameter for better performance, the hybrid meta-heuristic algorithm is developed in this paper. Another reason is that the presented hybrid algorithm improves the results of the overall convergence speed and accuracy and acts better than other suggested meta-heuristic algorithms. In the following, the details of the developed algorithm are explained.

Step 1 The parameters of the ACO and SA algorithms are adjusted (Appendix E and F) and the pheromone trails are initialised.
Step 2 Until the stop condition isn't met the below steps should be implemented:

- Start to generate the candidate solutions (construct ant solution stage). In this step, a set of ant m generates candidate solutions to the hybridised optimisation problem using elements of a finite set of candidate solution components available to $F = \{f_{rt}\}, r = 1, \ldots, n, t = 1, \ldots, |D_r|$. The generation phase of candidate solutions begins with the generation of a partial candidate $w^s = \phi$ solution. Then, in the next steps of generating candidate solutions, the partial candidate solution generated by w^s is expanded by adding a component from the sets of feasible neighbours $N(w^s) \subseteq F$.

  The process of generating candidate solutions can be considered as a path in the structural graph $G_F = (Q, U)$. In other words, the purpose of extending the optimal solution is to determine the possible paths for the ant in the structure graph of the pheromone model. In this way, the Neighborhood areas of the partial candidate solution are searched to determine the best path to the optimal global solution. Permitted paths in graph G_F are implicitly defined by the solution generation mechanism. The solution generation mechanism defines the set of possible neighbours $N(w^s) \subseteq F$ for each of the partial solutions separately.

  At each stage of the generation of candidate solutions, the method of selecting the components of the possible neighbour set to extend the partial candidate solution is performed quite possible. The rules for selecting a component from a set of possible neighbours differ in different implementations of the ant colony algorithm. However, the most well-known rule concerns the ant system algorithm, which is formulated as Eq. (19).

$$p(f_{rt}|w^s) = \frac{\delta_{rt}^{\alpha}.\mu(f_{rt})^{\beta}}{\sum_{f_{re\in N(w^s)}} \delta_{re}^{\alpha}.\mu(f_{re})^{\beta}}, \forall f_{rt} \in N(w^s) \tag{19}$$

In Eq. (19), δ_{rt} shows values of the pheromone to component f_{rt}. $\mu(0)$ is a function that assigns a so-called "heuristic" value to each candidate solution $f_{rt\in N(w^s)}$ at each stage of candidate solution generation. The heuristic values generated by the function $\mu(0)$ are called "heuristic information". Also, the parameters α and β indicate parameters with positive values whose values demonstrate the relative importance (weight) of pheromone information (values of the variables of a candidate solution), and heuristic information determines the generation of the potential value based on Eq. (19).

- Start local search stage. At this stage, local optimal solutions are used to determine which pheromones need to be updated.
- Do the pheromone update stage. The goal of the pheromone update stage is to increase the pheromone values related to the good and optimal candidate solutions and to decrease the pheromone values related to the bad solutions. This is done through two main processes:
  1. Decreasing the pheromone values related to all candidate solutions through the "Pheromone Evaporation" process
  2. Increasing the pheromone values related to the candidate solutions of the "good solution" or W_{upd}
- Therefore, the two above processes are controlled by Eq. (20). Equation (20) is called the rule of the pheromone update.

$$\delta_{rt} \leftarrow (1-\pi)\delta_{rt} + \pi \sum_{w\in W_{upd}|f_{rt\in w}} Z(w) \tag{20}$$

The first part of this equation controls the pheromone evaporation process (reducing the pheromone content of all candidate solutions). The second part only increases the pheromone values related to the candidate solutions of the good solutions or W_{upd}. In this regard, W_{upd} includes a set of candidate solutions that are highly appropriate; That is, they are closer to the optimal global solution. Parameter $\pi \in (0, 1]$ shows the "Evaporation Rate" and $Z : W \to R_0^+$ indicates the "Fitness Function".
Accordingly, this process leads to an increase in the number of pheromones related to the ants that are in the best available paths to the optimal solution (closer to the optimal solution) and have higher fitness (lower cost or higher profit). As a result, other ants converge in this direction.

Step 3 Algorithm MOSA uses a probabilistic function to accept neighbouring solutions that have dominated the current solution of the algorithm to escape the local optimisation. Hence, this function is shown in Eq. (21).

$$P(\nabla E, T) = e^{\frac{-\sum_i \nabla E(f_i)}{T}} \tag{21}$$

In Eq. (21), ∇E shows the difference between the objective function of the current solution and the neighbour solution, and T indicates a parameter called temperature. Equation (21) demonstrates the probability of accepting solutions that are dominated by the current solution. In other words, the worse solution is directly related to the temperature T and the sum of the changes in the objective functions. If the temperature of T is considered too high, the probability of accepting these types of solutions increases, and almost all the solutions are accepted and the algorithm transforms to a probabilistic algorithm. If T is considered too low, only solutions that do not dominate the current solution will be accepted and become a heuristic algorithm. Therefore, at the beginning of the algorithm, the initial temperature must be chosen so that a ratio of bad solutions is accepted. There are several methods to reduce the temperature and the dynamic geometric method is used in this paper (see Eq. 22).

$$T_{i+1} = \alpha T_i, 0 \nless \alpha < 1 \tag{22}$$

Step 4 In this study, in addition to the condition of stopping reaching the final temperature, another condition, called the number of times that the objective functions are calculated, is considered for termination. If one of these two conditions is met, the algorithm finishes. Otherwise, do the steps again.

The schematic steps and pseudo-code of the developed algorithm are shown in Appendix E and F.

The presence of the pheromone evaporation in TS and temperature in SA parameters is necessary to prevent the "Rapid and Premature Convergence" of the proposed algorithm. These two parameters provide a kind of "forgetting" mechanism in the optimisation process and cause more emphasis on searching and exploring new areas in the search space of implemented algorithm.

The proposed approach aims to enhance the exploitation of the multi-objective ACO algorithm. In the hybrid method, multi-objective SA is utilised as a component in the multi-objective ACO algorithm. Hence, when showing the optimal local minimum, initial temperature is applied to avoid this premature convergence and running away from a local minimum. Therefore, to find the best solution, the process of implementation of the multi-objective SA is applied in the main loop of the multi-objective ACO algorithm after the stop condition of this algorithm.

5 Computational experiments

In this section, first of all, the experiment instances for the AFSCN model are generated. Four evaluation metrics are proposed to evaluate the quality of Pareto or non-dominated solutions of the algorithms based on two objective functions. Additionally, the Taguchi technique is employed to set the appropriate values for the proposed algorithm's parameters. As a result, the best trade-off among two objective functions is addressed by a comparison of presented algorithms and the best algorithms are chosen. Also, various sorts of algorithms are compared and hybridised together to find the best strategy for the developed problem. Finally, a set of sensitivity analyses is employed to investigate the validation of the AFSCN model. It should be mentioned that the relevant code was written in MATLAB on a laptop with Corei7, 3.6 GHz in CPU, and 8 GB RAM and utilising Windows 10.

5.1 Numerical example

In this subsection, based on the novelty of the proposed AFSCN model, no available paper or research has used a similar problem in the examined studies. Moreover, the accessible benchmarks in the literature review are not available for the problem and a strategy is required to design the experiment instances. To evaluate the efficiency of the proposed algorithms, the proposed model is solved with several problems in different sizes. Table 2 shows the

Table 2 Data ranges of used parameters in the AFSCN model

Parameters	Range of values	Parameters	Range of values
CP_{mit}	~ *U(1S0000,30,000)*	kit	~ *U(15,30)Hectare*
CV_{mit}	~ *U(100,000,2S000)*	α_{mit}	~ *U(5,10)*
CH_{mit}	~ *U(100,000,2S000)*	^{S}J	~ *U (1,000,000,3,000,000)*
CA_{mit}	~ *U (200,000,800,000)*	σ	~ *U (1,000,000,3,000,000)*
CB_{mit}	~ *U (150,000,400,000)*	B	~ *U(15,000,000,30,000,000)*
CC_{mit}	~ *U (100,000,900,000)*	L_{mit}	~ *U(1000,40,000)*
$dAtj$	~ *U(15,30)km*	gmi	~ *U(500,5000)*
dB_n	~ *U(5,15)km*	q_m	~ *U(1,8)month*
dC_a	~ *U(5,30)km*	Δ_i	~ *U(2,5)*
F_{jT}	~ *U (15,000,000,3,000,000)*	τ_i	~ *U(5,15)*
Wm	~ *U(20,60)*	TW_t	~ *U(100,250)*
$MinP_t$	~ *U(200,300)*	M	*90,000,000,000,000*
ζ	~ *U(0.15,0.45)*	$[\varpi_l,\ \mu_l]$	~ *U(1,4)*
η	~ *U (15,20)*		

parameters of the AFSCN model. Each problem instance is characterised by the number of citrus fruit markets (l), number of farms (i), number of fridge/cold storage (j), number of products (m), the shelf life time of the citrus fruit in the fridge (n), and number of periods (t). The detailed delineation of all the problems is mentioned in Table 3. In addition, all the problem instances are classified into three groups according to the total number of decision variables of the problem instances. Furthermore, all the test instances are divided into three categories, including small, medium, and large. It is necessary to mention that available data in Tables 2 and 3 are shown according to the simulated data.

Table 3 The size of the test problems

Problem size	Instance no	Farms	Product/Orange	Fridges/Cold storages	Citrus fruit Markets	Period	Shelf lifetime
Small	SS1	3	2	3	2	3	3
	SS2	4	5	3	4	3	3
	SS3	7	8	6	5	3	3
Medium	MS4	11	9	7	8	2	2
	MS5	14	10	9	8	3	3
	MS6	17	13	11	10	4	4
Large	LS7	25	20	15	12	5	5
	LS8	30	40	20	18	6	6
	LS9	40	60	25	28	8	8

5.2 Evaluation metrics for Pareto optimum solutions

Generally, the comparison of multi-objective algorithms with each other is difficult. In this regard, researchers have proposed several metrics to evaluate the quality of Pareto fronts for the algorithms (Devika et al., 2014; Fathollahi-Fard et al., 2020). Therefore, four famous evaluation metrics including the number of Pareto solution (NPS) (Goodarzian et al., 2021b), mean ideal distance (MID) (Govindan et al., 2015), the spread of non-dominance solution (SNS) (Esmaeilikia et al., 2016; Govindan et al., 2015), and maximum spread (MS) (Fathollahi-Fard et al., 2020) are employed in this paper.

- ***NPS:*** The number of Pareto optimal solutions is ascertained using this metric. The maximum number of Pareto solutions is indicative of the algorithm's high performance.
- ***MID:*** MID measures the gap between the Pareto solution and the ideal solution. Lesser MID is always preferred to achieve better performance.

$$MID = \frac{\sum_{i=1}^{n}\sqrt{\left(\frac{f_{1i}-f_1^{best}}{f_{1,total}^{max}-f_{1,total}^{min}}\right)^2+\left(\frac{f_{2i}-f_2^{best}}{f_{2,total}^{max}-f_{2,total}^{min}}\right)^2}}{n} \tag{23}$$

where n represents a total non-dominated set, $f_{i,total}^{max}$ and $f_{i,total}^{min}$ shows the maximum and minimum values of the i^{th} objective function in non-dominated solutions obtained from algorithms, and f_i^{best} indicates the ideal solution of i^{th} objective function.

- ***SNS:*** SNS is used to measure the spread of ideal solutions and non-dominance solutions. It is calculated with the help of the average distance between the ideal solution and the optimal solution. The higher value of SNS ensures the spread of optimal solutions.

$$SNS = \sqrt{\frac{\sum_{i=1}^{n}(\overline{c}-c_i)^2}{n-1}} \tag{24}$$

- Here, $c_i = \|\overrightarrow{f_i} - \overrightarrow{f_{ideal}}\|, \overline{c} = \frac{c_i}{n}, \overrightarrow{f_{ideal}} = \{\min(f_1), \min(f_2)\ldots\ldots\ldots\min(f_k)\}$
- ***MS:*** MS measures the distance between solutions with respect to the true Pareto front in the objective space. The higher value of MS reflects larger area is covered by the true Pareto front. The expression for MS can be given as:

$$MS = \frac{1}{M}\sum_{m=1}^{M}\left(\frac{\min(F_{i,known}^{max}, F_{i,true}^{max}) - \max(F_{i,known}^{min}, F_{i,true}^{min})}{F_{i,true}^{max} - F_{i,true}^{min}}\right)^2 \tag{25}$$

where M shows the considered total objective, $F_{i,known}^{max}$ and $F_{i,known}^{min}$ represent the maximum and minimum of i^{th} function value in F_{known} respectively, and $F_{i,true}^{max}$ and $F_{i,true}^{min}$ are the maximum and minimum of i^{th} function value in F_{true} respectively.

5.3 Taguchi technique: parameter tuning

One of the goals of test design is to be able to observe and identify output changes by consciously making changes to process input variables. There are several ways to design an experiment using the Taguchi method. Since the proposed meta-heuristic algorithms have a set of control parameters, a plan should be provided to tune the algorithms' parameters. If the algorithms are not controlled well, the proposed algorithm behaviour would be inefficient. In this paper, the Taguchi technique is employed to control and tune the presented algorithm's parameters. The Taguchi method involves reducing the variation in a process through the robust design of experiments. Taguchi developed a method for designing experiments to investigate how different parameters affect the mean and variance of a process performance characteristic, which defines how well the process is functioning. The experimental design proposed by Taguchi involves using orthogonal arrays to organise the parameters affecting the process and the levels at which they should be varied. Instead of having to test all possible combinations like the factorial design, the Taguchi method tests pairs of combinations (Goodarzian et al., 2021a). Taguchi (1986) introduced this method to decrease the number of tests for quality engineering scopes. Then, several studies have suggested seeing the application of the Taguchi method and its definition and introduction for interested scholars (see Devika et al., 2014; Cheraghalipour et al., 2019; Fathollahi-Fard et al., 2020; Goodarzian et al., 2020, 2021a, 2021c). The characteristics of the comparison of the Taguchi approach are divided into two categories: control and noise factors. The approach considers calculating the variation value of response according to the signal-to-noise ratio to attain the aim to tune and control the proposed algorithm's parameters. Signal-to-noise ratio (SNR or S/N) is a measure used in science and engineering that compares the level of a desired signal to the level of background noise. SNR is defined as the ratio of signal power to the noise power, often expressed in decibels. A ratio higher than 1:1 (greater than 0 dB) indicates more signal than noise. Additionally, the Taguchi approach mechanism focuses on the response type. The presented response is divided into three categories including the nominal is better, the smaller is better, and the larger the better type. Whereas the presented response of this paper is a minimisation sort, "the smaller is better" is utilised to tune and control each algorithm's parameters. Then, the S/N ratio is formulated based on Eq. (26).

$$S/N = -10 \times \log\left(\frac{\sum_{i=1}^{n} Y_i^2}{n}\right) \tag{26}$$

where Y_i shows the value of the response for i th orthogonal array and n is the number of orthogonal arrays.

In this paper, the MOSA, MOACO, NSGA-II, and Fast PGA are used and HACO-SA is developed. The algorithms' parameters are considered as the factors for each algorithm. Furthermore, the proposed levels along with the factors are described in Table 4. Hence, a maximum of three levels is provided to the algorithms' factors. Accordingly, the Taguchi approach reduces the total number of experiments by suggesting a set of orthogonal arrays to control and calibrate the algorithms in a reasonable time. Thus, the Taguchi method suggests L8 for MOSA and L16 for HACO-SA, Fast PGA, NSGA-II, and MOACO using MINITAB 16 Statistical Software, as reported in Tables 5, 6, 7, 8. In Taguchi's approach, three levels are selected for the parameter change range. Then three numbers are randomly selected in the desired range. In each instance, these three combinations are repeated. Govindan et al. (2019) and Roghanian et al. (2019) have also used this approach.

Therefore, the results of the S/N ratio should be analysed to find the best levels of each algorithm. The results of the S/N are shown in Figs.4, 5, 6, 7, 8. Based on the signal-to-noise figures, the level with the lowest value will be selected. According to Figs.4, 5, 6, 7, 8, A2, B3, C1, D2, and E2 for NSGA-II, A3, B2, C1, D2, and E1 for MOACO, A2, B3, C1, and D3 for MOSA, A3, B3, C3, D2, E3, and F1 for Fast PGA, and A1, B3, C2, D3, E3, F3, G3, and H2 for HACO-SA are selected to run of the proposed algorithms in different size problems based on S/N.

Table 4 The values of parameter setting of the presented algorithms

Algorithms	Factors	Levels		
		1	2	3
NSGA-II	Max Iteration	50	100	150
	Pop size	50	100	150
	Crossover rate	0.1	0.2	0.3
	Mutation rate	0.2	0.3	0.4
	Tournament size	1	1.5	2
MOACO	Max Iteration	50	100	150
	Number of ants	25	50	75
	Pheromone factor α	0.25	0.35	0.4
	Evaporation rate ρ	0.3	0.4	0.45
	Heuristic factor β	1	1.5	2
MOSA	Max Iteration	50	100	150
	Sub Iteration	25	50	75
	Mutation rate	0.2	0.3	0.4
	Initial temperature	0.15	0.25	0.35
Fast PGA	Max Iteration	50	100	150
	Intersection rate	0.25	0.35	0.4
	Mutation rate	0.15	0.25	0.35
	Pop size	50	100	150
	Maximum number of children	75	85	95
HACO-SA	Max Iteration	50	100	150
	Number of ants	35	60	75
	Pheromone factor α	0.2	0.3	0.4
	Evaporation rate ρ	0.3	0.4	0.45
	Heuristic factor β	1	1.5	2
	Sub Iteration	25	50	75
	Mutation rate	0.2	0.3	0.4
	Initial temperature	0.1	0.15	0.25

5.4 Pareto optimum analysis: comparison of metaheuristics

In this section, the MOACO, NSGA-II, Fast PGA, MOSA, HACO-SA algorithms are employed to solve the Pareto solutions and the AFSCN model as well as to find a good and suitable solution. To evaluate the performance of the proposed algorithms, the obtained solutions from these three algorithms are investigated in three different sizes. Also, the optimal response values, the best response in three different sizes, the average, and their standard deviation (SD) of objective functions are provided in Table 9.

According to the proposed four evaluation metrics of Pareto optimum analyses, meta-heuristic algorithms are compared with each other. Finally, to improve the reliability of meta-heuristic algorithms, the average of the results for thirty run times is considered to utilise in this section. The behaviour of meta-heuristic algorithms based on computational

Table 5 The orthogonal array L16 for the NSGA-II and MOACO

A	B	C	D	E
1	1	1	1	1
1	1	1	1	1
1	2	2	2	2
1	2	2	2	2
2	1	1	2	2
2	1	1	2	2
2	2	2	1	1
2	2	2	1	1
3	1	2	1	2
3	1	2	1	2
3	2	1	2	1
3	2	1	2	1
4	1	2	2	1
4	1	2	2	1
4	2	1	1	2
4	2	1	1	2

Table 6 The orthogonal array L8 for the MOSA

A	B	C	D
1	1	1	1
1	2	2	2
2	1	1	2
2	2	2	1
3	1	2	1
3	2	1	2
4	1	2	2
4	2	1	1

(CPU) time is indicated in Fig. 9. From Fig. 9 and Table 9, it is clear that the HACO-SA meta-heuristic is quicker and more robust than other suggested algorithms. Thus, the CPU time of HACO-SA is less than other proposed meta-heuristics on different problems. Therefore, HACO-SA has a minimum average of CPU time (378.2967 s), but the MOSA has the maximum rate of this item (623.3133 s). As can be seen in Table 9, the CPU time in the Augmented Epsilon-constraint approach increases exponentially as the sizes of the problem increase. Therefore, according to the CPU time, it can be understood that the proposed model is NP-hard.

As it is seen, in using the MOSA algorithm, all of the suggested problems have been able to achieve the Pareto solutions in different problems. Moreover, the expected states of problem size, SDs increase as the problem size increases. It is also clear that the SD of the

Table 7 The orthogonal array L16 for the Fast PGA

A	B	C	D	E	F
1	1	1	1	1	1
1	1	1	1	1	2
1	2	2	2	2	1
1	2	2	2	2	2
2	1	1	2	2	1
2	1	1	2	2	2
2	2	2	1	1	1
2	2	2	1	1	2
3	1	2	1	2	1
3	1	2	1	2	2
3	2	1	2	1	1
3	2	1	2	1	2
4	1	2	2	1	1
4	1	2	2	1	2
4	2	1	1	2	1
4	2	1	1	2	2

Table 8 The orthogonal array L16 for the HACO-SA

A	B	C	D	E	F	G	H
1	1	1	1	1	1	1	1
1	1	1	1	1	2	2	2
1	2	2	2	2	1	1	1
1	2	2	2	2	2	2	2
2	1	1	2	2	1	1	2
2	1	1	2	2	2	2	1
2	2	2	1	1	1	1	2
2	2	2	1	1	2	2	1
3	1	2	1	2	1	2	1
3	1	2	1	2	2	1	2
3	2	1	2	1	1	2	1
3	2	1	2	1	2	1	2
4	1	2	2	1	1	2	2
4	1	2	2	1	2	1	1
4	2	1	1	2	1	2	2
4	2	1	1	2	2	1	1

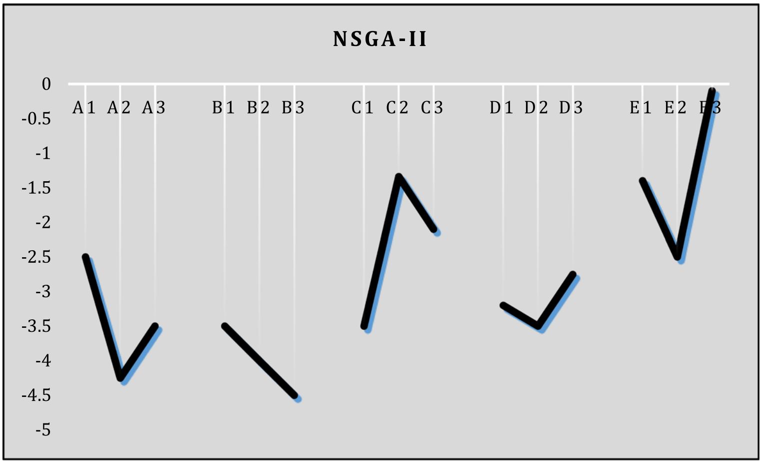


Fig. 4 The results of the S/N ratio of the NSGA-II

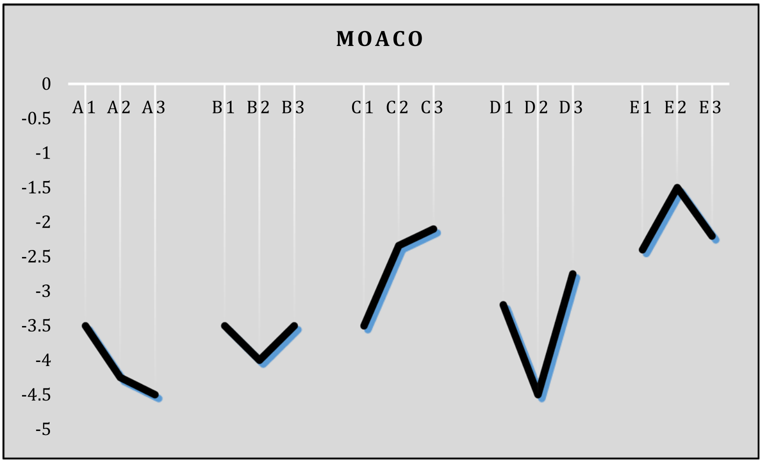


Fig. 5 The results of the S/N ratio of the MOACO

first problem (SS1) in the small size equal is zero, which indicates algorithms have reached the optimal Pareto solutions.

The MOACO and HACO-SA algorithms have achieved the optimal Pareto solutions in proposed test problems. Likewise, the HACO-SA has achieved the optimal Pareto solution, with the difference that, in three problems (SS1, MS4 and LS7), the SD is equal to zero. This means that in three categories of these experiment problems, exactly each run is optimised for the solutions, while for the MOSA, only in the first problem (SS1) have all the repetitions reached the optimal Pareto solutions. Therefore, HACO-SA is more reliable than other proposed algorithms. The NSGA-II and Fast PGA algorithms have also been able to achieve optimal Pareto solutions in proposed test problems, but only in two samples of these problems, which are optimised for each run.

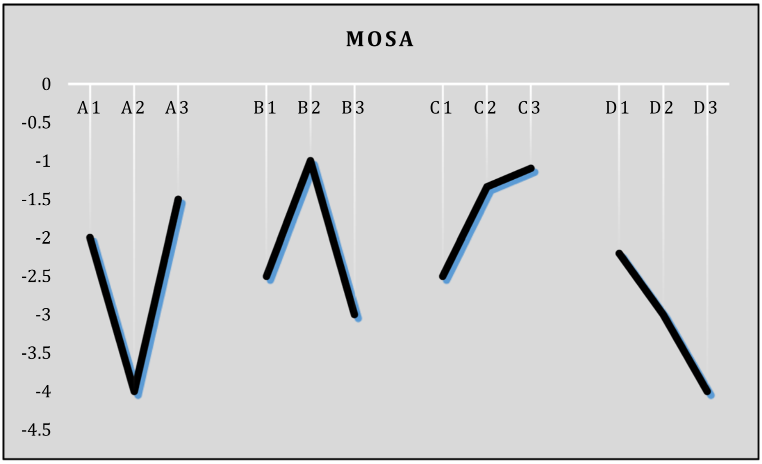


Fig. 6 The results of the S/N ratio of the MOSA

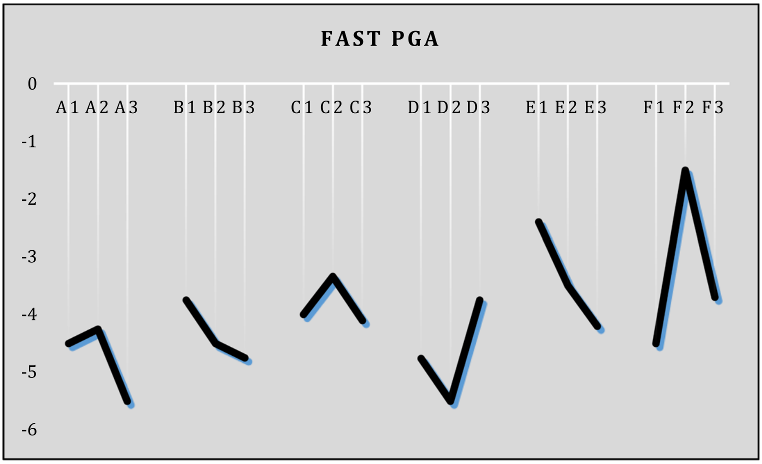


Fig. 7 The results of the S/N ratio of the Fast PGA

The presented meta-heuristic algorithms were also evaluated based on CPU time. It is clear that, in Table 9, the minimum CPU time is related to the HACO-SA algorithm, in which the MOSA, NSGA-II, MOACO, and Fast PGA algorithms need more CPU time than the HACO-SA algorithm. Then, according to the measure of the average, the standard deviation, and the CPU time, the HACO-SA algorithm has high efficiency and better performance than the other suggested algorithms in the different sizes. According to Fig. 9, it is clear that the CPU times of the NSGA-II and Fast PGA are very similar and close together. The trend of the HACO-SA indicates that, in the small size, there is almost a routine, but, by increasing size, the CPU time increases significantly. In addition, as shown in Fig. 9, as the size of the problem increases, the CPU time of the Augmented Epsilon-constraint approach increases exponentially. It should be noted that all CPU times for the Augmented Epsilon-constraint approach are divided by 10. Also, the Augmented Epsilon-constraint approach is not able to

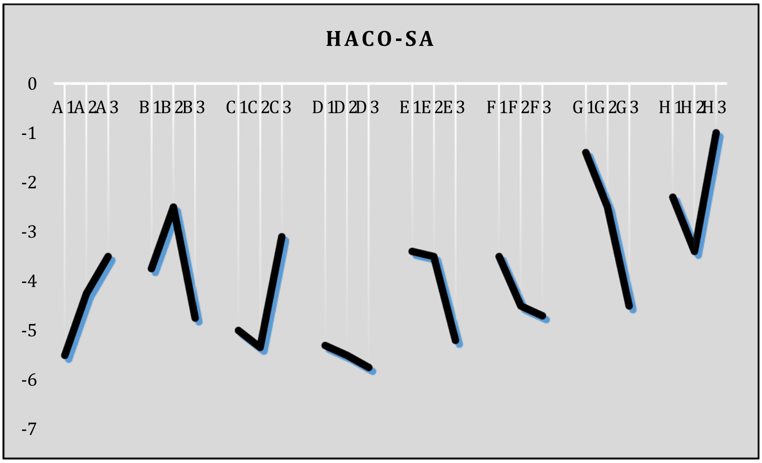


Fig. 8 The results of the S/N ratio of the HACO-SA

solve problems in large size due to its long CPU time. Therefore, the NP-hard of the proposed model is proved.

The performance of proposed algorithms is examined by four evaluation metrics including NPS, MID, MS, and SNS as the comparison metrics for obtained Pareto sets under every experiment problem. Then, the outputs of the assessment metrics are indicated in Tables 10, 11, 12, 13 where it is clear that the HACO-SA algorithm is more reliable, better, and more powerful than other presented algorithms in this paper. In terms of NPS, SNS, and MS, a higher value of these three criteria indicates a greater ability of algorithm solutions, efficiency and performance, and a higher quality of non-dominated solutions. It should be noted that the lowest value of the MID criterion shows the better quality of the algorithm's solutions. As a result, in the four assessment metrics, HACO-SA has the best quality and more efficiency than other suggested meta-heuristics.

Figure 10 indicates non-dominated (Pareto solutions) solutions of proposed algorithms in different experiment problems. The Pareto performance of the solutions is examined to assess the efficiency of the presented algorithms and their "number of Pareto solutions" results. Here, non-dominated solutions of algorithms in SS1, MS4, and LS8 (in Table 10) test problems based on the number of Pareto solutions (NPS) are indicated. The Pareto frontiers from the NSGA-II, Fast PGA, MOACO and MOSA algorithms were also compared with HACO-SA. In terms of efficiency, the HACO-SA has the more effective Pareto front than the other presented algorithms. The obtained outcomes based on NPS for the HACO-SA algorithm follow the Pareto optimality and are effective. In terms of the selected results in the NPS assessment metric, the number of Pareto front solutions is computed for each algorithm. It is clear that whatever the number of Pareto solution in each algorithm is further, it is more favourable. Therefore, the results of the objective functions based on the NPS metric are examined in Fig. 10. Here it is evident that iNSGA-II shows the worst performance, and HACO-SA is mostly overcome by the other algorithms. Consequently, HACO-SA indicates a more powerful Pareto efficiency than other suggested algorithms to solve the AFSCN problem and to find Pareto frontier solutions.

Thus, this paper performs a set of statistical comparisons between the suggested algorithms according to the Pareto optimal analyses by assessment metrics to detect the better algorithm.

Table 9 The results obtained are the objective functions and CPU time (based on seconds) of the methods

Instance		Augmented Epsilon-constraint		
	Objectives	Optimum	SD	Time (s)
SS1	Z_1	1974.04	0	16.75
	Z_2	2332.24	23.12	
SS2	Z_1	2471.20	42.64	367.20
	Z_2	2704.60	51.78	
SS3	Z_1	3415.19	88.24	663.11
	Z_2	3712.45	101.24	
MS4	Z_1	5731.31	0	1235.91
	Z_2	5910.50	203.50	
MS5	Z_1	6351.49	205.64	3921.93
	Z_2	6889.29	201.98	
MS6	Z_1	8008.21	212.14	8011.67
	Z_2	8464.19	234.80	
LS7	Z_1	-	-	–
	Z_2	–	–	
LS8	Z_1	–	–	-
	Z_2	–	–	
LS9	Z_1	–	–	–
	Z_2	-	–	
Instance	NSGA-II			
	Objectives	Optimum	SD	Time (s)
SS1	Z_1	2252.13	0	34.65
	Z_2	2467.17	61.15	
SS2	Z_1	2865.52	84.74	97.33
	Z_2	3045.67	98.45	
SS3	Z_1	3791.81	167.43	107.66
	Z_2	3967.02	188.38	
MS4	Z_1	6320.53	165.77	194.33
	Z_2	7104.34	199.23	
MS5	Z_1	6773.30	0	274.07
	Z_2	7356.23	211.67	
MS6	Z_1	8582.53	227.04	697.54
	Z_2	9105.81	237.89	
LS7	Z_1	46,940.71	855.33	870.45
	Z_2	49,677.56	2330.32	
LS8	Z_1	58,250.48	1104.76	958.65
	Z_2	6122.06	3012.64	
LS9	Z_1	87,511.81	1997.85	1634.55
	Z_2	96,604.66	4566.18	

Table 9 (continued)

Instance	NSGA-II			
	Objectives	Optimum	SD	Time (s)
MOACO				
SS1	Z_1	2012.21	0	32.21
	Z_2	2203.19	45.2	
SS2	Z_1	2579.23	75.12	78.54
	Z_2	2798.18	81.56	
SS3	Z_1	3576.41	132.07	89.33
	Z_2	3877.19	143.78	
MS4	Z_1	5731.31	0	167.21
	Z_2	5988.16	168.81	
MS5	Z_1	6251.46	58.44	196.93
	Z_2	6678.28	201.33	
MS6	Z_1	7956.53	112.67	365.07
	Z_2	8233.35	256.8	
LS7	Z_1	34,346.21	652.43	541.42
	Z_2	37,445.87	766.32	
LS8	Z_1	52,721.40	766.54	766.41
	Z_2	57,887.24	899.56	
LS9	Z_1	85,819.21	1322.98	1456.19
	Z_2	88,498.79	1899.49	
Fast PGA				
SS1	Z_1	2059.71	0	31.43
	Z_2	2251.29	58.11	
SS2	Z_1	2687.43	80.54	88.42
	Z_2	2841.63	92.41	
SS3	Z_1	3540.55	154.70	99.01
	Z_2	3760.16	169.33	
MS4	Z_1	5949.32	0	167.01
	Z_2	6574.21	178.16	
MS5	Z_1	6454.34	167.78	230.32
	Z_2	7154.27	198.31	
MS6	Z_1	8231.62	154.32	487.33
	Z_2	8631.31	218.32	
LS7	Z_1	37,542.70	733.09	691.04
	Z_2	42,671.17	1230.32	
LS8	Z_1	54,367.02	988.12	883.22
	Z_2	56,182.31	2912.61	
LS9	Z_1	85,866.61	1522.26	1403.98
	Z_2	93,621.68	3668.13	

Table 9 (continued)

Instance	NSGA-II			
	Objectives	Optimum	SD	Time (s)
MOSA				
SS1	Z_1	2851.36	0	44.67
	Z_2	3227.17	89.12	
SS2	Z_1	3168.32	98.72	122.45
	Z_2	3547.16	167.43	
SS3	Z_1	3877.82	188.42	143.67
	Z_2	4122.67	256.31	
MS4	Z_1	6766.51	201.71	233.56
	Z_2	7323.78	288.27	
MS5	Z_1	6988.31	298.56	301.45
	Z_2	7786.24	310.61	
MS6	Z_1	8782.98	238.29	867.12
	Z_2	9275.88	341.42	
LS7	Z_1	47,749.32	1055.21	988.67
	Z_2	50,673.52	2457.31	
LS8	Z_1	62,251.44	1254.51	1023.56
	Z_2	64,562.01	3452.67	
LS9	Z_1	89,812.45	2997.82	1884.67
	Z_2	97,804.62	5666.71	
HACO-SA				
SS1	Z_1	1982.04	0	29.43
	Z_2	2013.67	45.2	
SS2	Z_1	2486.23	75.12	67.22
	Z_2	2677.18	81.56	
SS3	Z_1	3276.41	132.07	78.33
	Z_2	3677.15	143.78	
MS4	Z_1	5731.31	0	141.32
	Z_2	5988.16	168.81	
MS5	Z_1	6251.46	58.44	184.93
	Z_2	6678.28	201.33	
MS6	Z_1	7956.53	112.67	365.07
	Z_2	8233.35	256.8	
LS7	Z_1	34,346.21	0	541.42
	Z_2	37,445.87	766.32	
LS8	Z_1	52,721.40	766.54	766.41
	Z_2	57,887.24	899.56	
LS9	Z_1	81,838.18	1322.98	1230.54
	Z_2	86,455.77	1899.49	

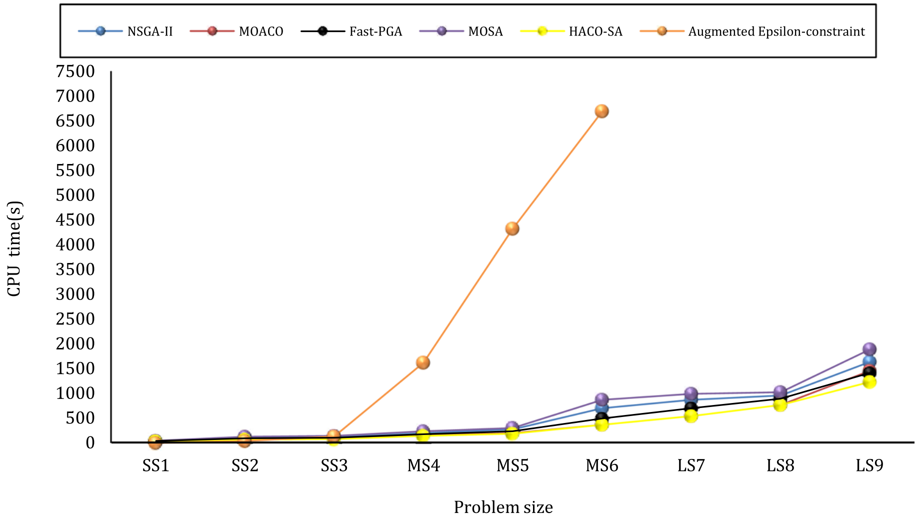


Fig. 9 The behavior of the proposed methods based on CPU time

Table 10 NPS's computational outputs for proposed algorithms

Experiment problem	NSGA-II	MOSA	MOACO	Fast PGA	HACO-SA
SS1	6	8	5	7	10
SS2	7	11	6	8	12
SS3	9	14	7	11	14
MS4	11	14	6	11	14
MS5	13	15	6	14	16
MS6	12	15	8	13	16
LS7	10	13	5	11	15
LS8	8	11	5	9	13
LS9	8	14	7	9	15

Table 11 MID's computational outputs for proposed algorithms

Experiment problem	NSGA-II	MOSA	MOACO	Fast PGA	HACO-SA
SS1	9.231	10.166	8.8092	9.3629	6.3242
SS2	6.1811	9.9431	8.2322	8.4192	5.8654
SS3	7.7026	9.8749	6.1617	7.314	6.1237
MS4	8.2929	11.046	7.3422	8.2063	6.4326
MS5	9.5263	14.293	8.8964	8.2343	7.6786
MS6	10.984	13.014	9.6898	10.762	8.2543
LS7	14.1412	18.278	11.171	13.304	9.6781
LS8	15.2146	21.035	14.907	16.705	10.6712
LS9	23.801	25.749	16.962	25.913	13.5691

Table 12 MS's computational outputs for metaheuristic algorithms

Experiment problem	NSGA-II	MOSA	MOACO	Fast PGA	HACO-SA
SS1	53,271	66,945	49,711	52,965	76,281
SS2	63,433	81,515	52,855	55,626	81,092
SS3	67,115	103,505	55,413	58,575	85,677
MS4	98,864	141,482	75,109	81,156	89,776
MS5	135,370	182,178	105,899	124,713	94,671
MS6	186,499	233,753	135,592	145,448	98,807
LS7	241,840	297,840	248,145	152,781	113,979
LS8	296,909	398,663	265,612	211,089	156,702
LS9	333,097	445,856	276,008	242,705	189,043

Table 13 SNS's computational outputs for proposed algorithms

Experiment problem	NSGA-II	MOSA	MOACO	Fast PGA	HACO-SA
SS1	8285.4	10,468	7775.5	8415.7	12,783
SS2	4805	9740.7	7009.7	7232.9	13,897
SS3	6157.1	9175.4	5362.8	3958.5	15,679
MS4	6827.9	13,226	5615.7	5364.2	16,897
MS5	8145.8	19,720	7515.4	7059.6	17,835
MS6	11,295	20,279	9874.3	9039.4	18,765
LS7	2481.6	33,537	13,661	17,966	19,532
LS8	3320.2	47,962	20,360	28,787	20,765
LS9	11,451	51,199	26,894	45,429	23,567

Hence, the outputs that were reported in Tables 10, 11, 12, 13 are converted to a well-known metric called Relative Deviation Index (RDI) using the following formula (Devika et al., 2014):

$$RDI = \frac{\left|Alg_{sol} - Best_{sol}\right|}{Max_{sol} - Min_{sol}} \tag{27}$$

where Min_{sol} and Max_{sol} illustrate the minimum and maximum values between all the obtained values of the presented algorithms. $Best_{sol}$ shows the best solution between the suggested algorithms and also Alg_{sol} indicates the value of the obtained objective function by an assessment metric of the meta-heuristic algorithms.

All approaches to calculating the quality of metaheuristic solutions measure the Pareto point distance obtained to the ideal point. Therefore, the quality of the solutions is not related to the parameters and variables of the problem. In the RDI approach, the distance of the calculated Pareto points from the Pareto points obtained from the exact solution is calculated. It is clear that a lower value of RDI shows a higher quality of meta-heuristics. Accordingly, the confidence interval of 95% for the performance assessment metrics in the proposed algorithms to statistically analyse the effectiveness of algorithms is performed

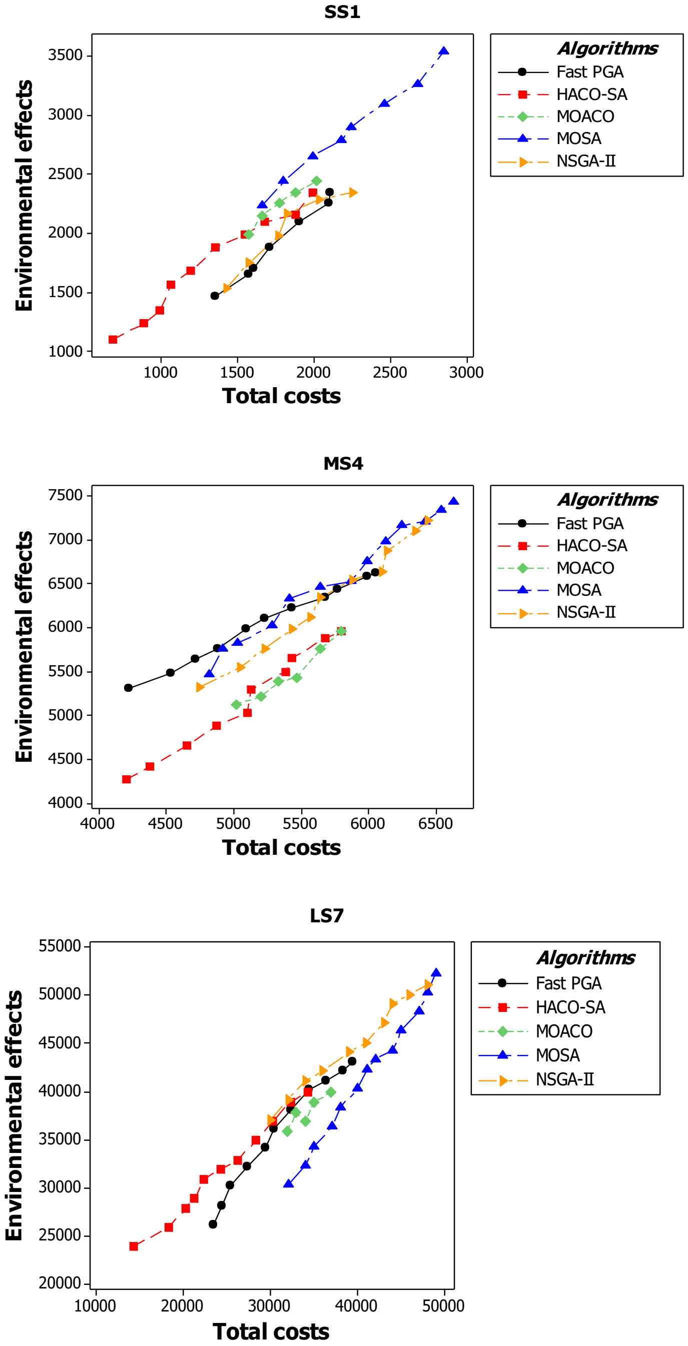


Fig. 10 Pareto frontier of the presented meta-heuristics algorithms in the sizes SS1, MS4, and LS8

(Naderi & Ruiz, 2010). Accordingly, Least Significant Difference (LSD) and the means plot are considered for the used meta-heuristics. The run results by Minitab 16 software are presented in Fig. 11.

According to Fig. 11, in terms of the NPS, MID, MS, and SNS assessment metrics, the developed meta-heuristic algorithm, i.e., HACO-SA, is more successful, of high quality, and more effective than the other suggested algorithms, while MOSA has worse performance than the NSGA-II, MOACO, Fast PGA and HACO-SA in all suggested metrics.

5.5 Performance analysis of the meta-heuristic algorithms

The optimisation performance related to the convergence of the proposed algorithms is evaluated based on convergence plots. Then, the convergence plots for NSGA-II, MOSA, MOACO, Fast PGA and HACO-SA algorithms, respectively, according to two objective functions in 50 iterations are shown in Figs. 12, 13, 14, 15, 16. All presented algorithms converge in a stable and fixed trend. The convergence of all of the suggested meta-heuristic algorithms is evaluated and, accordingly, their final outcomes are compared. The maximum iteration is generated based on Table 4 for the proposed algorithms, in which the maximum iteration is investigated the same for the presented algorithms. It is clear that, in Figs. 12, 13, 14, 15, NSGA-II, MOSA, MOACO and Fast PGA algorithms are converging in 50 iterations, but the HACO-SA algorithm converges after 43 iterations in a steady line. Consequently, HACO-SA has more efficiency and high performance than the other proposed algorithms. The purpose of showing the convergence of meta-heuristic algorithms was to prove the superiority of the hybrid algorithm over other algorithms. Therefore, due to the fact that the hybrid algorithm has reached convergence in less than 55 iterations, the convergence display of other algorithms has been omitted.

5.6 Sensitivity analysis

In this section, sensitivity analysis is carried out to demonstrate the effects of the convergence behaviour on the main parameters of the AFSCN model on the values of the objective functions. According to the HACO-SA algorithm that is the best algorithm based on the obtained results, then this algorithm is utilised to perform the sensitivity analysis.

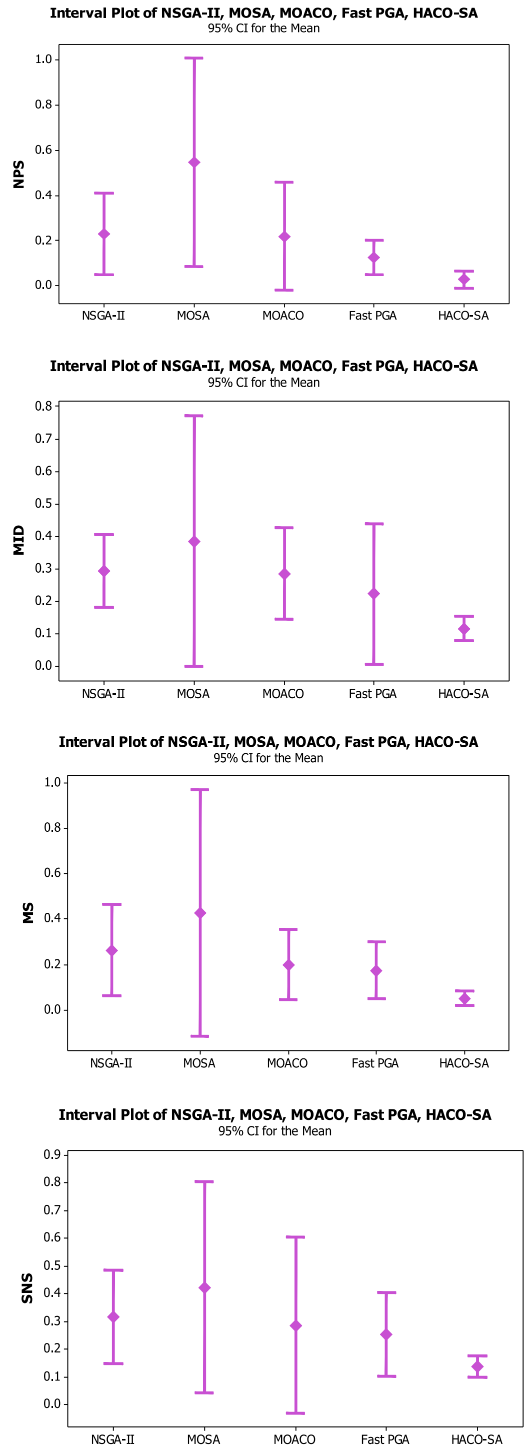
5.6.1 Changing the behaviour of the objective functions based on transportation costs

In terms of transportation costs, sensitivity analyses have been carried out by raising the amount of this parameter. To recognise the behaviour of both objective functions (total cost and environmental effects), the obtained values are provided in this comparison as depicted in Fig. 17. The outcomes illustrate that, by raising the amount of this parameter, the first objective function is raised slightly. Also, the value of the second objective decreased steadily.

5.6.2 Changing the behaviour of the objective functions based on parameter ζ

In this subsection, in order to analyse the effect of the amount of the CO_2 emissions coefficient through vehicle fuel on the objective function values, various values of the ζ are employed. The outcomes of Fig. 18 and Table 14 indicate that the rise in the amount of the $\mathrm{CO_2}$ emissions coefficient of vehicle fuel lead to increase gradually the second objective function. The value

Fig. 11 ANOVA plots for the assessment metrics in term of RDI for presented algorithms



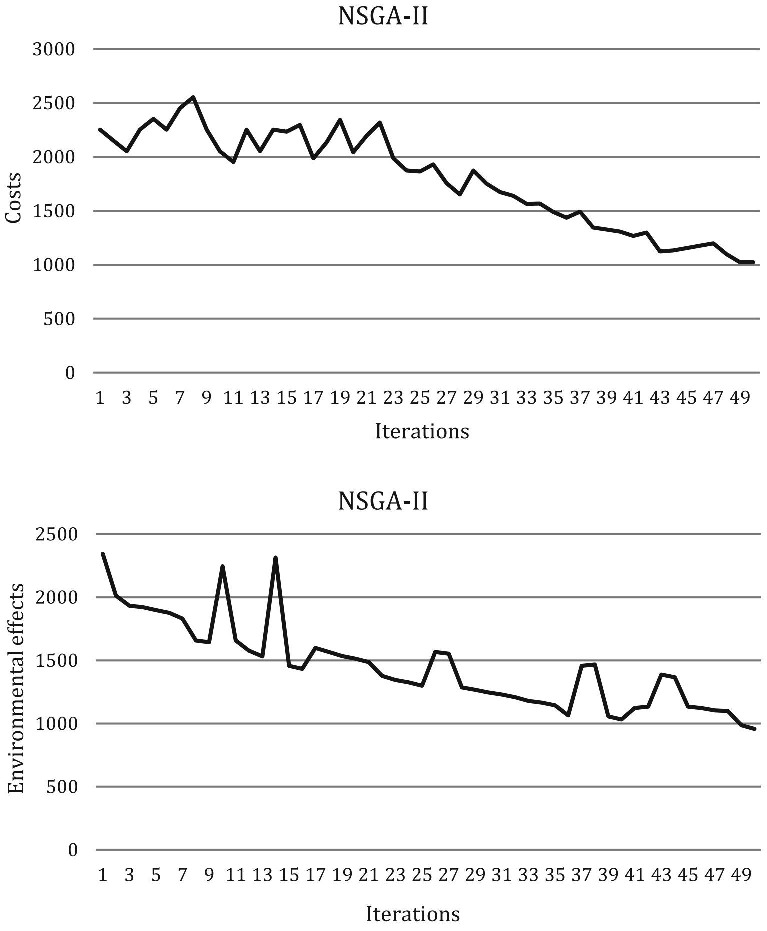


Fig. 12 Comparative convergence of the NSGA-II for costs and environmental effects

of the first objective function also increases. The slope of the increase in total costs is less than the slope of the environmental impact chart.

5.6.3 Changing the behaviour of the objective functions based on performance coefficient

In this subsection, in order to analyse the effect of the performance coefficient on the objective function values, various values of the g_{mi} are employed. The outcomes of Fig. 19 and Table 15 indicate that the rise in the amount of the g_{mi} lead to increase gradually the first objective function. The value of the second objective function also increases. The slope of the increase in environmental impact is less than the slope of the total costs chart.

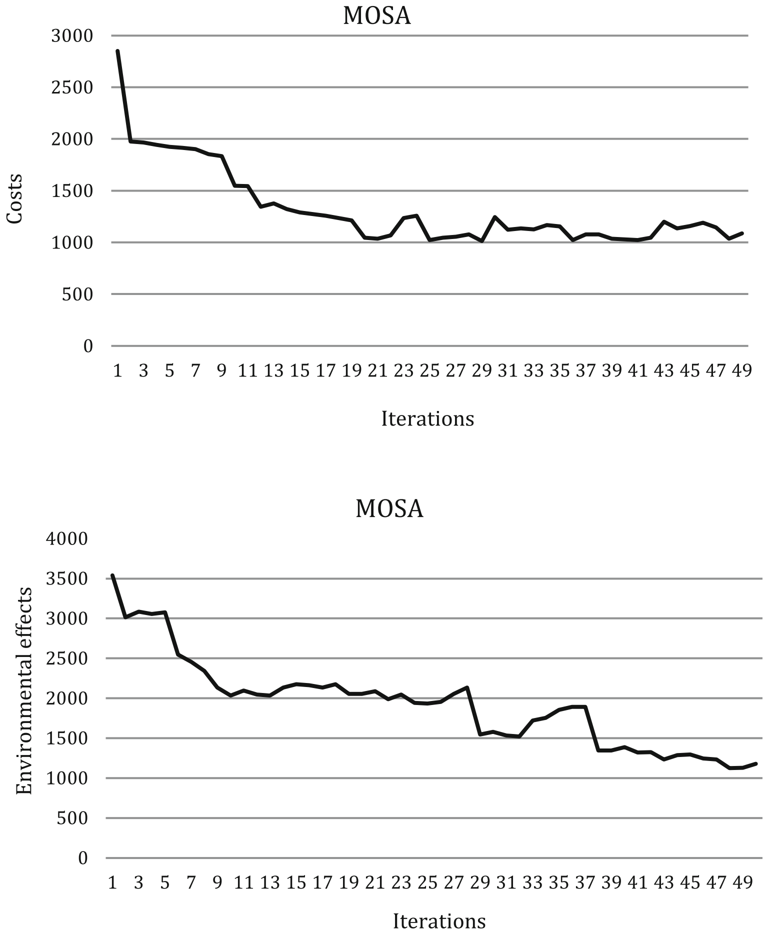


Fig. 13 Comparative convergence of the MOSA for costs and environmental effects

5.6.4 Changing the behaviour of the objective functions based on capability to re-cultivate

In this subsection, in order to analyse the effect of capability to re-cultivate on the objective function values, various values of the λ_{mit} are employed. As can be seen in Table 16 and Fig. 20, with increasing capability of cultivating product, the values of objective functions increase. Also, the first objective function increases faster than the second objective function.

5.6.5 Changing the behaviour of the objective functions based on total amount of accessible water resources

In this subsection, in order to analyse the effect of the total amount of accessible water resources for cultivating the desired products on the farm on the objective function values, various values of the TW_i are employed. As can be seen in Table 17 and Fig. 21, with

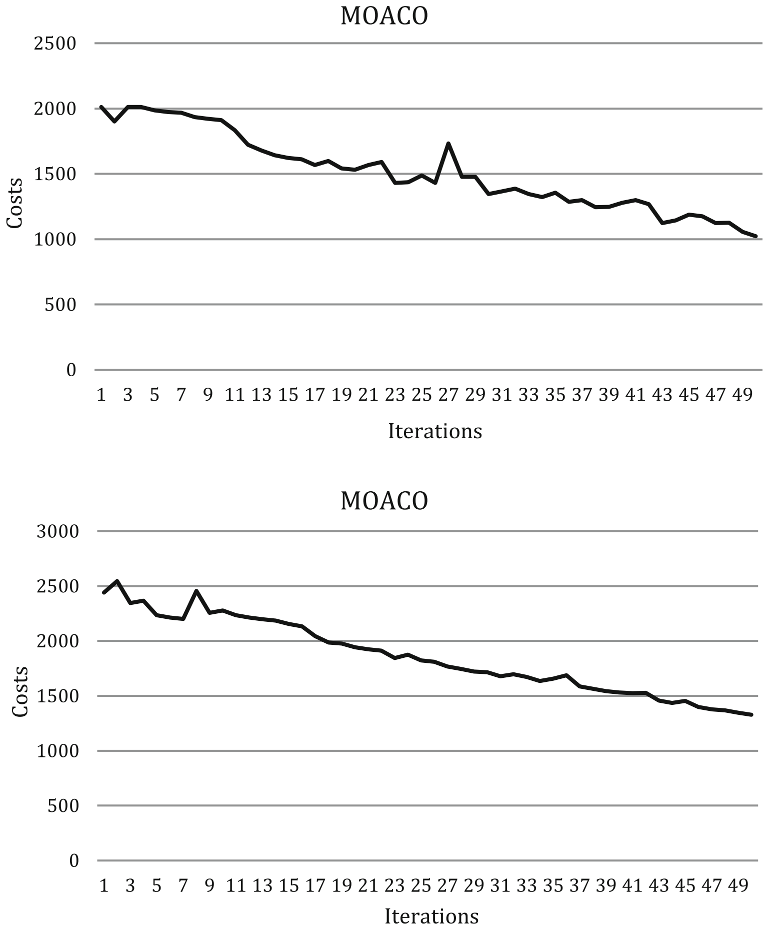


Fig. 14 Comparative convergence of the MOACO for costs and environmental effects

increasing total amount of accessible water resources for cultivating the desired products on the farm, the total cost decreases and the environmental effect increases.

5.7 Managerial insight

The different groups in the AFSCN including farmers, citrus fruit markets, farms, fridges/cold storage, and transportations get several insights from this paper. According to the increase of citrus fruit stock and the addition of the capacity, decision-makers or policymakers have to evaluate the storage capacity gap by establishing sufficient cold storage across the country. The proposed mathematical model can be utilised for the feasibility analysis of the different potential locations that support avoiding the loss of huge capital investment. The establishment of fridges/cold storage in surplus centres of the country will be useful for the brisk transition of citrus fruit (orange) from farms to citrus fruit markets. This will lead to an increase in procurement from farmers and can be used for the benefit of MSP, which

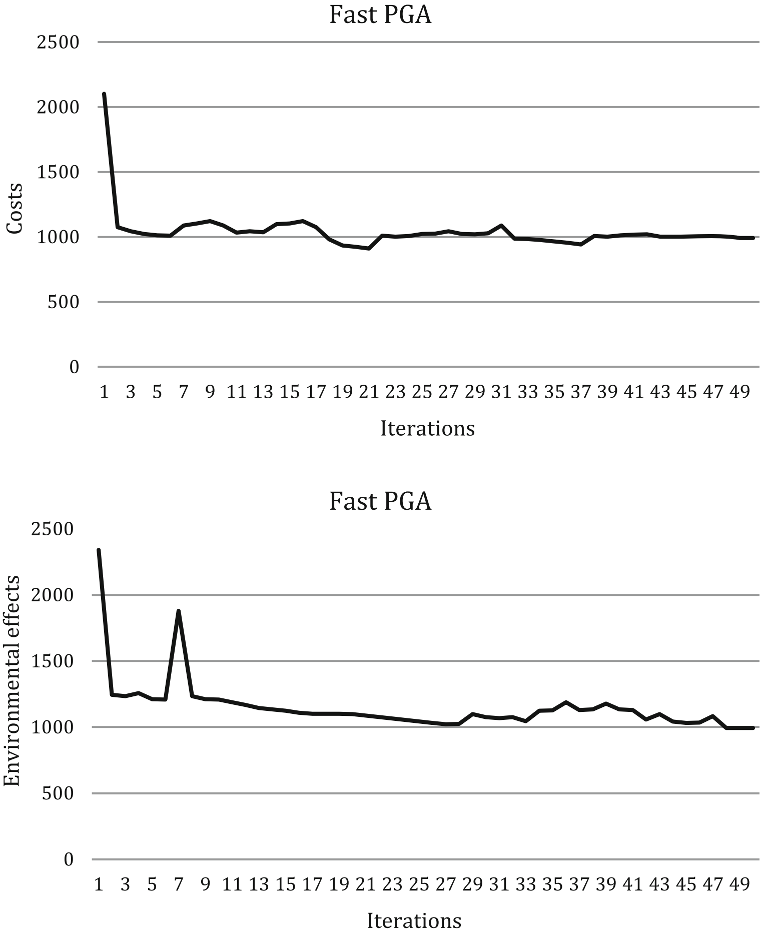


Fig. 15 Comparative convergence of the Fast PGA for costs and environmental effects

improves their welfare and economic growth. Likewise, the construction of state and district-level fridges/cold storage will be useful for the holding of citrus fruit by satisfying the demand of the people in shortage states and decreasing malnutrition. Due to the establishment of new fridges/cold storage, farmers and other actors traverse fewer distances to reach the nearby fridges/cold storage. This leads to a decrease in transportation costs and related emissions among various stages. Moreover, less emission of CO_2 will be instrumental to reduce the carbon tax on transportation activities. The holding losses of citrus fruit/orange by holding it in the surplus fridge/cold storage will be considerably decreased because of the establishment of new fridges/cold storage. All in all, the majority of the problems relevant to the holding, transportation, post-harvest losses, and carry overcharges can resolve after the accessibility of adequate storage capacity. Also, decision-makers can control and limit the release of CO_2 emission according to the holding of the citrus fruit and activities related to transportation by keeping the optimal citrus fruit/oranges in various fridges/cold storages. The obtained Pareto optimal solutions are beneficial for the decision-makers to maintain the suitable trade-off between cost and released CO_2 emission. Also considering the capability to re-cultivate the

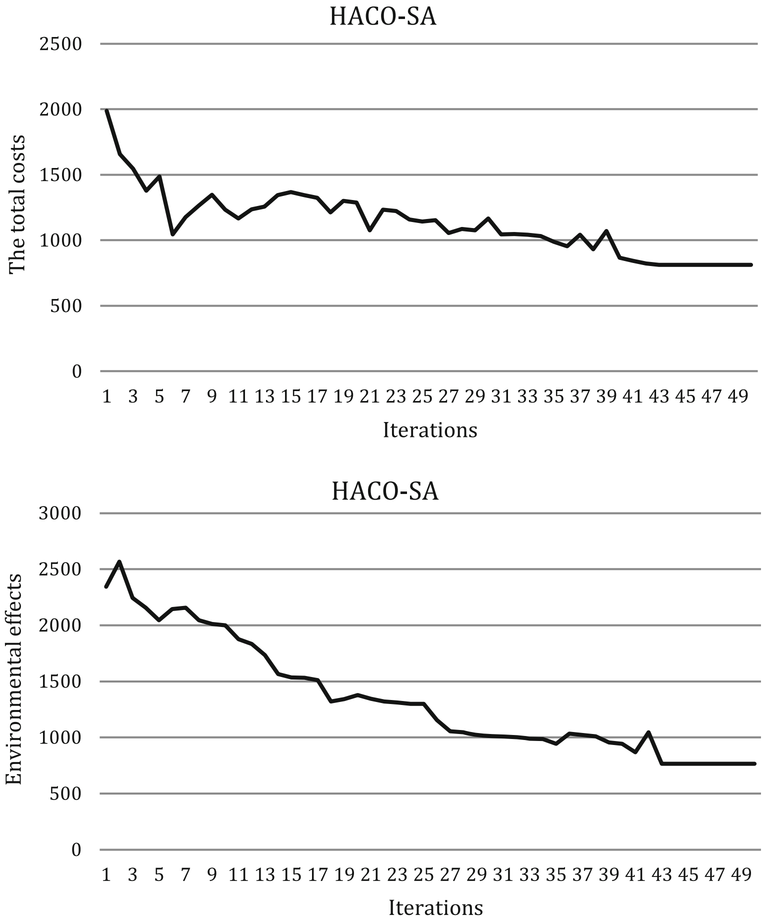


Fig. 16 Comparative convergence of the HACO-SA for costs and environmental effects

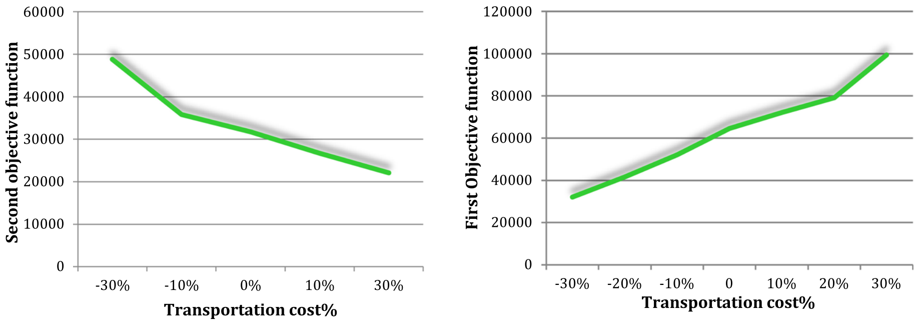


Fig. 17 The behavior of objective functions on transportation costs

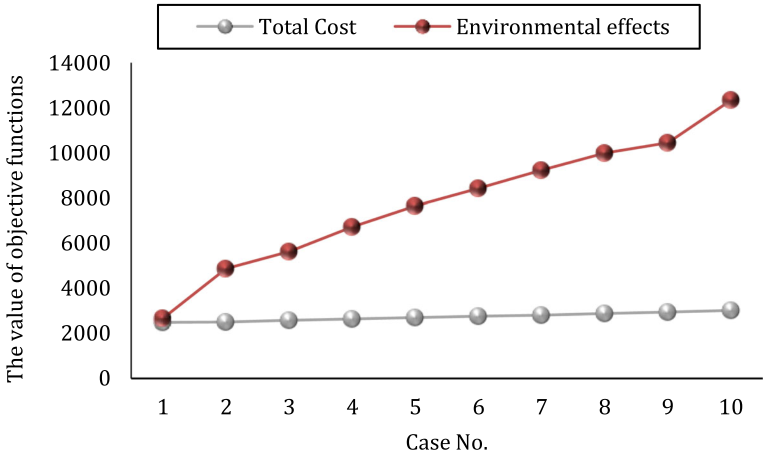


Fig. 18 The trend of objective functions based on changing ζ

Table 14 The results of the sensitivity analysis related to the number of vehicles

Cases no	ζ	Total cost	Environmental effects
1	0.15	2486.23	2677.18
2	0.25	2507.75	4873.43
3	0.35	2584.14	5633.81
4	0.45	2639.36	6732.21
5	0.55	2697.74	7654.65
6	0.65	2755.03	8432.32
7	0.75	2806.21	9231.19
8	0.85	2888.29	9987.69
9	0.95	2941.11	10,453.43
10	0.99	3015.70	12,348.28

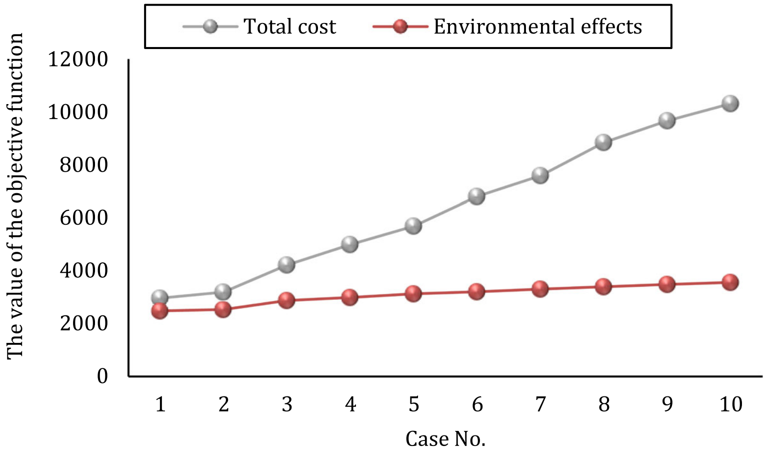


Fig. 19 The trend of objective functions based on changing g_{mi}

Table 15 The results of the sensitivity analysis related to the performance coefficient

Cases no	g_{mi}	Total cost	Environmental effects
1	0.15	2967.20	2478.78
2	0.25	3193.79	2533.11
3	0.35	4225.55	2876.65
4	0.45	4991.39	2987.00
5	0.55	5688.53	3123.60
6	0.65	6809.77	3199.42
7	0.75	7594.14	3304.16
8	0.85	8845.16	3396.60
9	0.95	9666.10	3487.41
10	0.99	10,316.11	3558.28

Table 16 The results of the sensitivity analysis related to the capability of cultivating product

Cases no	λ_{mit}	Total cost	Environmental effects
1	1	2264.87	2497.64
2	2	2490.85	2573.61
3	3	2725.57	2774.94
4	4	3267.53	2812.04
5	5	3888.17	3011.62
6	6	4009.85	3342.40
7	7	4394.43	3448.54
8	8	5045.10	3563.03
9	9	5586.17	3683.91
10	10	5996.12	3848.00

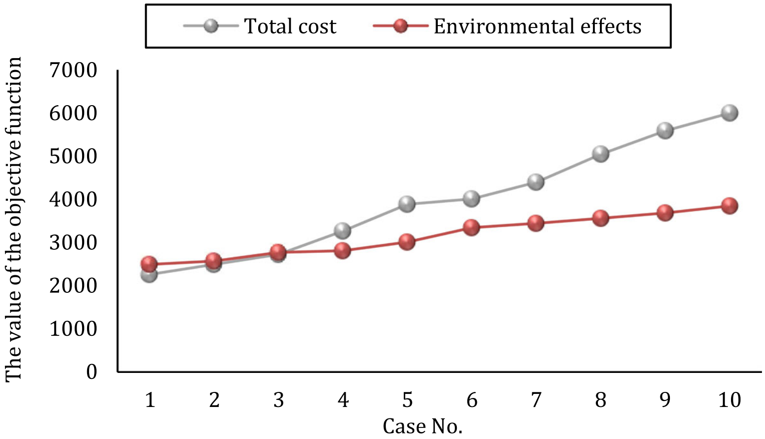


Fig. 20 The trend of objective functions based on changing λ_{mit}

Table 17 The results of the sensitivity analysis related to the total amount of accessible water resources

Cases no	TW_i	Total cost	Environmental effects
1	0.1	3768.76	2255.68
2	0.2	3456.98	2365.01
3	0.3	3094.25	2547.43
4	0.4	2870.10	2619.45
5	0.5	2611.72	2815.66
6	0.6	2341.90	2942.31
7	0.7	2099.32	3348.50
8	0.8	1976.89	3567.41
9	0.9	1890.59	3790.98
10	1	1745.60	3943.02

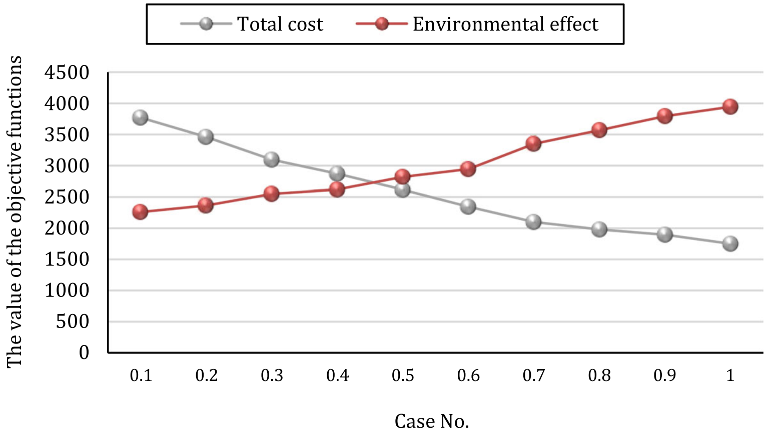


Fig. 21 The trend of objective functions based on changing TW_i

agricultural land after citrus fruit harvest, and developing a new heuristic method based on a meta-heuristic to solve the mathematical model are the other contributions of this paper.

Examination of the amount of released CO_2 through vehicles shows that there are usually two peaks per year, one in winter and the other in summer. The maximum winter incidence is usually in October, November, or December and the maximum summer incidence is in June, July, or August. The existence of maximum in the winter season can occur due to increased ignition consumption of fuels caused by thermal functions and vehicles along with meteorological conditions such as increased air pressure, air stability, and temperature inversion. Also, the maximum presence of released CO_2 in summer can be due to several factors including surface temperature inversion, reduced rain and failure to wash pollutants. Therefore, in this study, the released CO_2 by vehicles in summer is considered.

6 Conclusions and future research directions

In this paper, a bi-objective mathematical model for a multi-period, three-echelon, single product (orange) AFSCN is designed. Additionally, an MINLP model under time windows for servicing citrus fruit markets in the AFSCN along with minimising the total costs and the CO_2 emissions are provided. The water resource gap, considering simultaneously both the sent amount of orange to the fridge/cold storage and the duration of staying in the fridge/cold storage and time windows to service in citrus fruit market are some of the main motivations for this paper. In the current model, the capability to re-cultivate the agricultural land after citrus fruit harvest is considered. The formulated mathematical model is competent enough to demonstrate the trade-offs between cost and CO_2 emission. Small-, medium-, and large-sized problem examples stimulated from the agri-food supply chain are solved utilising four meta-heuristic algorithms including NSGA-II, MOSA, MOACO and Fast PGA. Additionally, a new hybrid algorithm called HACO-SA is developed to solve the presented model. Based on the obtained results, HACO-SA is superior compared to other proposed algorithms. Sensitivity analysis results show that the decision-makers should establish an adequate number of fridges/cold storage and the amount of water source in the production, holding, and consumption processes by maintaining a proper balance among the two objective functions.

Therefore, a fuzzy, stochastic, or robust multi-objective model can be formulated in the near future to obtain the uncertainty in cost parameters. The proposed model integrates the economic (total costs) aspects and environmental effects (CO_2 emissions) considering time windows. Therefore, interested researchers can use social effects such as the responsibility of farmers and workers in the proposed model. The current paper considered a single product (orange) and future research can look into the multi-product scenario. The quantification of the post-harvest losses can be another area to explore. The presented meta-heuristic algorithms can be applied to other problems such as location-routing, allocation-location-inventory, and scheduling-routing problems in the other supply chain network and healthcare network to assess their effectiveness. Finally, interested scholars can utilise other improved or hybrid meta-heuristic algorithms, exact methods such as the Lagrangian approach, and heuristic methods. The limitations of the research are as follows: As there was no official database for some parts of cost elements, the drivers' estimations were asked to help. The questions about the transportation costs have been categorised and the estimated costs have been entered into the mathematical model. Meta-heuristic algorithms are not able to calculate the global optimum and calculate the local optimum. The final solution in NSGA-II, MOSA, MOACO and Fast PGA algorithms depends on the coder's skill in defining chromosomes and the initial value of its parameters.

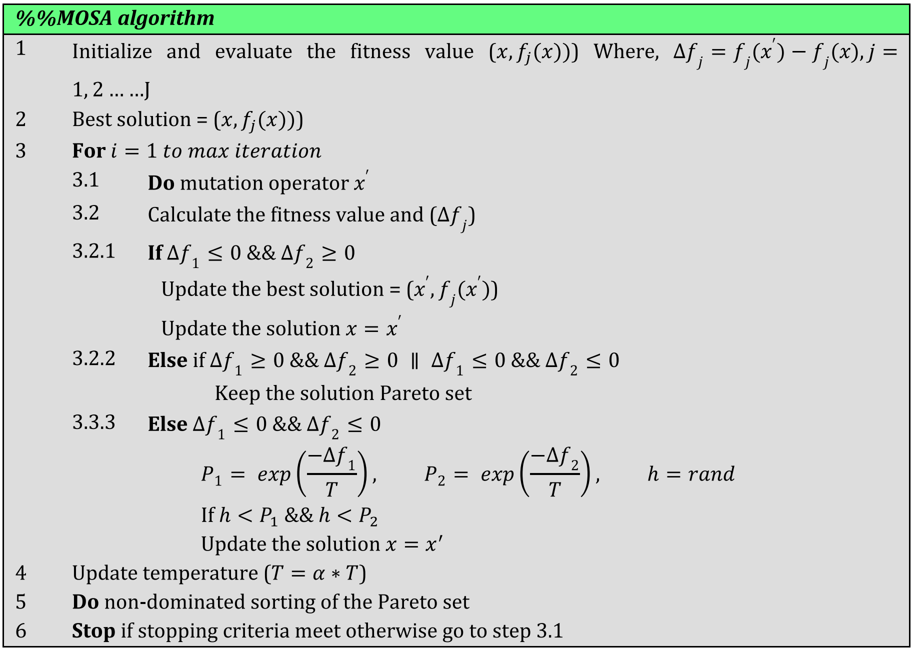
Similar to the other studies, the current paper has a few limitations, as follows:

- One of the limitations of the research is the lack of access to accurate information about transportation costs. Moreover, in this study, only the information of drivers and experts about transportation costs has been considered.
- In order to implement HACO-SA, NSGA-II, MOSA, MOACO and Fast PGA algorithms, high RAM and CPU hardware and software features such as MATLAB software are required.

Appendix A

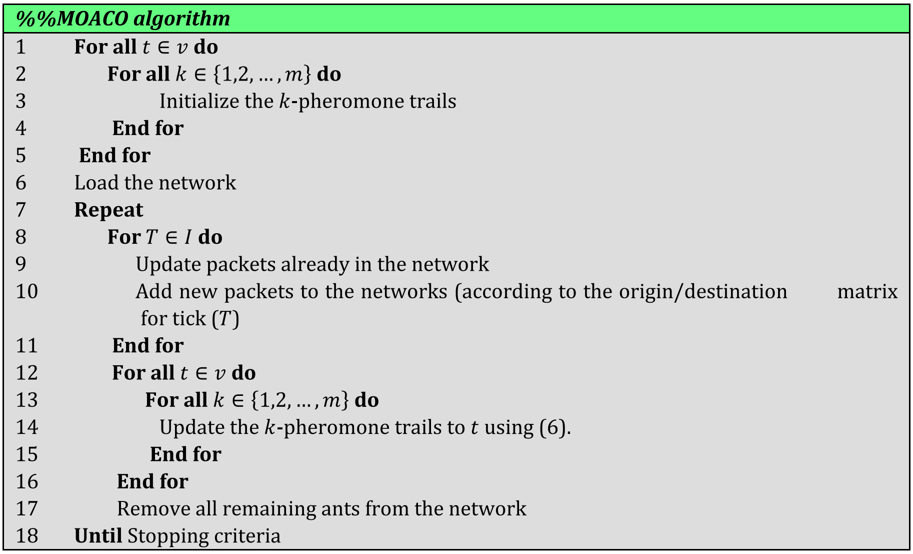
MOSA

Krickpatrick et al. (1983) proposed the basic version of the algorithm. The idea of the algorithm was analogous to the annealing process of physics. The heat bath is maintained at the higher temperature at which solid melts. Due to this, the particles rearrange themselves, and then this temperature decreases slowly. Finally, the solid arranges in the lattice with minimum energy. The search in the algorithm is based on random moves. Firstly, the bad moves are picked instead of good moves if the solution refines it is accepted else the algorithm generates a random solution based on the probability of less than one. This probability helps the algorithm to avoid trapping into local optima. This probability keeps on decreasing with the selected bad moves. This accepting rule is known as Metropolis criteria. Therefore, interested scholars can refer to Suman and Kumar (2006); for more information about MOSA algorithm. The pseudo-code of the MOSA algorithm is indicated in Fig. 22.



```
%%MOSA algorithm
1      Initialize and evaluate the fitness value $(x, f_j(x))$ Where, $\Delta f_j = f_j(x') - f_j(x), j = 1, 2 \ldots\ldots J$
2      Best solution = $(x, f_j(x))$
3      For $i = 1$ to max iteration
       3.1      Do mutation operator $x'$
       3.2      Calculate the fitness value and $(\Delta f_j)$
       3.2.1    If $\Delta f_1 \leq 0$ && $\Delta f_2 \geq 0$
                  Update the best solution = $(x', f_j(x'))$
                  Update the solution $x = x'$
       3.2.2    Else if $\Delta f_1 \geq 0$ && $\Delta f_2 \geq 0$ || $\Delta f_1 \leq 0$ && $\Delta f_2 \leq 0$
                         Keep the solution Pareto set
       3.3.3    Else $\Delta f_1 \leq 0$ && $\Delta f_2 \leq 0$
                    $P_1 = exp\left(\frac{-\Delta f_1}{T}\right)$,    $P_2 = exp\left(\frac{-\Delta f_2}{T}\right)$,    $h = rand$
                    If $h < P_1$ && $h < P_2$
                    Update the solution $x = x'$
4      Update temperature $(T = \alpha * T)$
5      Do non-dominated sorting of the Pareto set
6      Stop if stopping criteria meet otherwise go to step 3.1
```

Fig. 22 The pseudo-code of the MOSA (Loukil et al., 2007)



```
%%MOACO algorithm
For all $t \in v$ do
    For all $k \in \{1,2,\dots,m\}$ do
        Initialize the $k$-pheromone trails
    End for
End for
Load the network
Repeat
    For $T \in I$ do
        Update packets already in the network
        Add new packets to the networks (according to the origin/destination matrix
        for tick ($T$)
    End for
    For all $t \in v$ do
        For all $k \in \{1,2,\dots,m\}$ do
            Update the $k$-pheromone trails to $t$ using (6).
        End for
    End for
    Remove all remaining ants from the network
Until Stopping criteria
```

Fig. 23 The pseudo-code of the MOACO (Cardoso et al., 2003a)

Appendix B

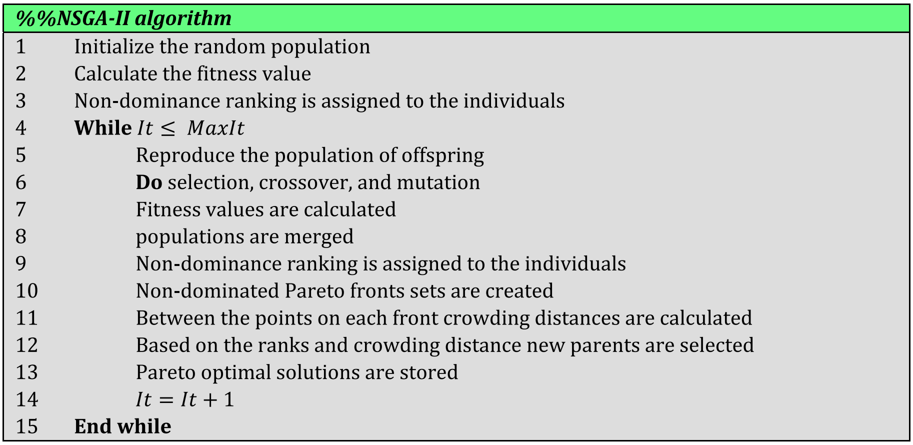
MOACO

This section outlines an efficient meta-heuristic inspired by the communication through pheromone trails found in ants proposed by Dorigo et al. (1999). These ants use the experience accumulated by previous ants (i.e., pheromone information) to solve a particular problem. Initially, parameters and pheromone information are first initialized. Possible solutions are constructed by adding solution components during the main loop. Local search is commonly used to improve these solutions. Each iteration presents new solutions. However, the best solution is selected from many possible solutions to updating pheromone information. Pheromone trails are evaporated by the factor ρ before the pheromone information is updated. A condition is proposed for termination of the loop, which may be termination criterion, computation time, or the number of solution constructions. Finally, when the main loop ends, the algorithm returns the best solution. Additionally, for more details about MOACO algorithm, you can refer to Cardoso et al., (2003a, 2003b) and García-Martínez et al. (2007) that introduced by Cardoso et al., (2003a, 2003b) for the first time. Additionally, for more details about MOACO algorithm, you can refer to Doerner et al. (2004) and Yagmahan and Yenisey (2010). Figure 23 shows the pseudo-code of the MOACO algorithm.

Appendix C

NSGA-II

NSGA-II is among the most popular multi-objective algorithm. It stands for the non-dominated sorting genetic algorithm proposed by Deb Et Al. (2002). It is a multi-objective



```
%%NSGA-II algorithm
Initialize the random population
Calculate the fitness value
Non-dominance ranking is assigned to the individuals
While $It \leq MaxIt$
    Reproduce the population of offspring
    Do selection, crossover, and mutation
    Fitness values are calculated
    populations are merged
    Non-dominance ranking is assigned to the individuals
    Non-dominated Pareto fronts sets are created
    Between the points on each front crowding distances are calculated
    Based on the ranks and crowding distance new parents are selected
    Pareto optimal solutions are stored
    $It = It + 1$
End while
```

Fig. 24 The pseudo-code of the NSGA-II

genetic algorithm that uses the basic genetic operators' selection, crossover, and mutation. Initially, the algorithm generates the candidate solution, and sorting is done based on the non-dominance criteria. This algorithm makes use of crowding distance-based selection to generate the offspring. It is different from a multi-objective genetic algorithm in respect to maintaining the elitism and diversity properties. Elitism property allows the algorithm to copy the best solution from the previous generation to the next generation. To maintain the diversity crowding distance-based sorting allows selecting solutions residing in the less crowded area (Validi et al., 2021). Therefore, the pseudo-code of the NSGA-II algorithm is displayed in Fig. 24.

Appendix D

Fast PGA

Fast PGA is a population-based evolutionary algorithm that introduced by Eskandari and Geiger (2008) for the first time. It includes a novel ranking strategy for the MOEA. The structure includes a novel ranking strategy and search mechanism to improve the convergence behavior of the algorithm. The initial population of candidate solutions is generated heuristically. The population size is considered equal to the size of the set of non-dominated solutions. For each generation evaluate the fitness functions for all objectives and make increment to the generation. Upon getting the fitness values parents are selected randomly using the roulette wheel method. The crossover and mutation operators are performed to generate the offspring and fitness values are evaluated. The composite populations are combined and ranked based on the novel ranking strategy. Finally, the population is regulated by the number of non-dominated solutions, and new populations are generated based on the composite population by removing the dominated solution (Goodarzian et al., 2021c). The algorithm is terminated when the stopping criterion is met. Figure 25 represents the pseudo-code of the Fast PGA algorithm.

```
%%Fast PGA algorithm
Initialize the algorithm parameters
Initialize random population of candidate solution vector $P_t = \{x_1^t, x_2^t, x_3^t \ldots \ldots \ldots\}$
Calculate fitness value for $m$ objectives
$t = 0$
While $It \leq MaxIt$
    $t = t + 1$
    $P_t^{'} = select(P_{t-1})$        // Select a pair of solution and perform selection operation
    $O_t = crossover(P_t^{'})$        // Perform crossover operation on generated offspring $(O_t)$
    $O_t = mutation(O_t)$        // Perform mutation operation on generated offspring $(O_t)$
    Evaluate $(O_t)$        // Evaluate fitness values for $m$ objectives and record them
    $CP_t = P_{t-1} \cup O_t$        // Merge the generated candidate solution $O_t$ with the previous population $P_{t-1}$ to generate the composite population $CP_t$
    Rank $(CP_t)$        // Rank the composite population based on the new ranking strategy
    Truncate $(CP_t)$        //Truncate the composite population $CP_t$ only with non-dominated solutions
    $P_t$ = generate $(CP_t)$        // Generate new solutions for a population to cover the loss of discarded inferior solutions.
    $It = It + 1$
End while
```

Fig. 25 The pseudo-code of the Fast PGA

Appendix E

See Fig. 26.

Appendix F

See Fig. 27.

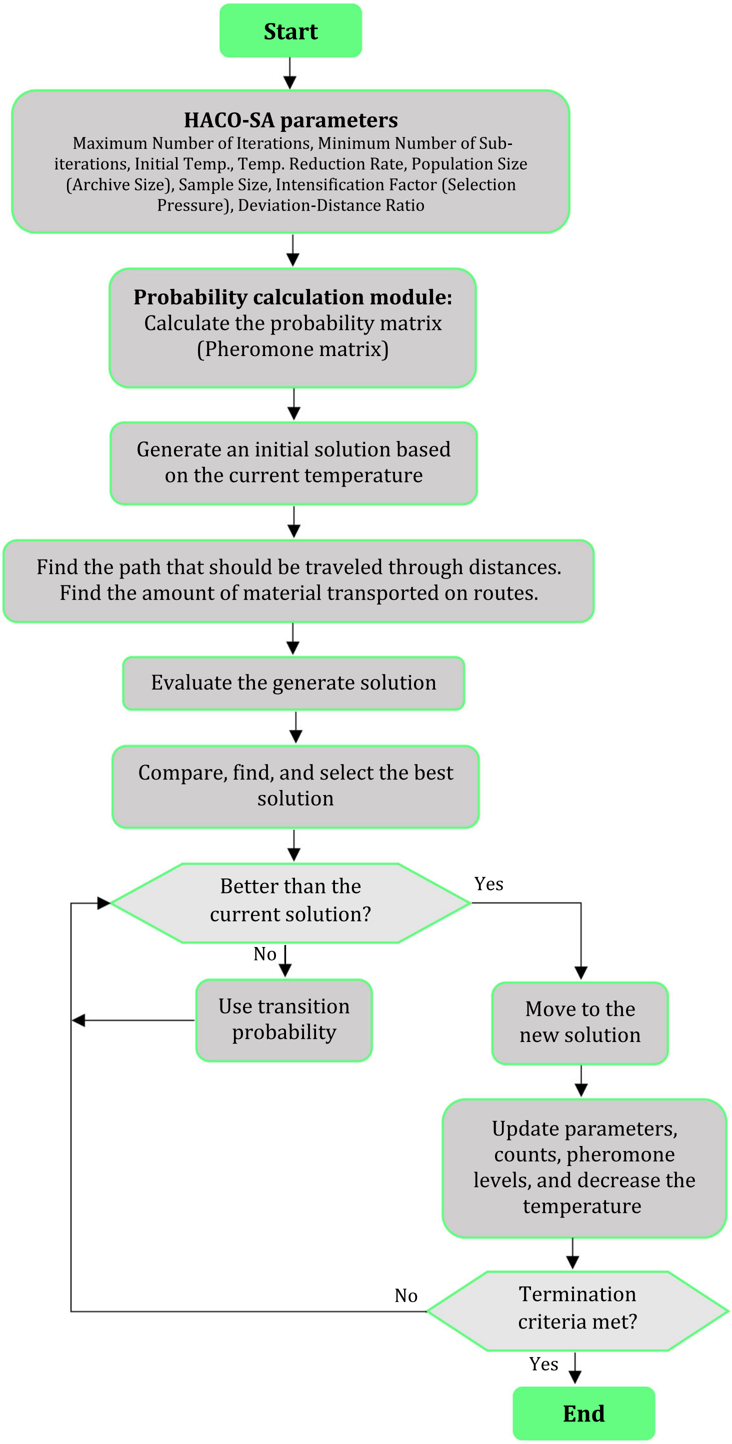
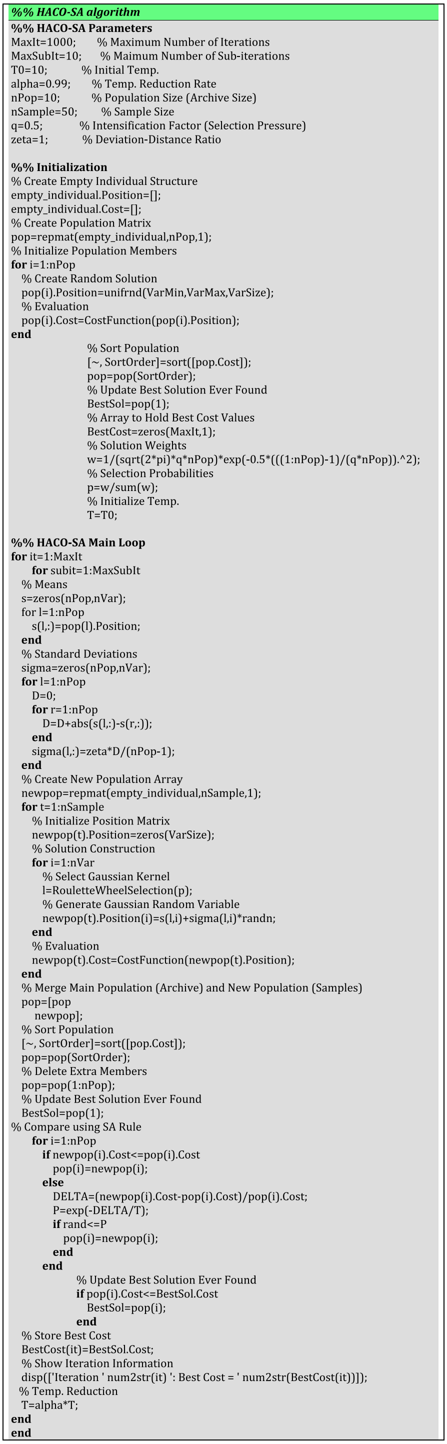

Start
HACO-SA parameters
Maximum Number of Iterations, Minimum Number of Sub-iterations, Initial Temp., Temp. Reduction Rate, Population Size (Archive Size), Sample Size, Intensification Factor (Selection Pressure), Deviation-Distance Ratio
Probability calculation module: Calculate the probability matrix (Pheromone matrix)
Generate an initial solution based on the current temperature
Find the path that should be traveled through distances. Find the amount of material transported on routes.
Evaluate the generate solution
Compare, find, and select the best solution
Better than the current solution?
Yes
No
Use transition probability
Move to the new solution
Update parameters, counts, pheromone levels, and decrease the temperature
No
Termination criteria met?
Yes
End

Fig. 26 The flowchart of the HACO-SA algorithm

Fig. 27 The pseudo-code of the HACO-SA



```
%% HACO-SA algorithm
%% HACO-SA Parameters
MaxIt=1000;       % Maximum Number of Iterations
MaxSubIt=10;      % Maimum Number of Sub-iterations
T0=10;            % Initial Temp.
alpha=0.99;       % Temp. Reduction Rate
nPop=10;          % Population Size (Archive Size)
nSample=50;       % Sample Size
q=0.5;            % Intensification Factor (Selection Pressure)
zeta=1;           % Deviation-Distance Ratio

%% Initialization
% Create Empty Individual Structure
empty_individual.Position=[];
empty_individual.Cost=[];
% Create Population Matrix
pop=repmat(empty_individual,nPop,1);
% Initialize Population Members
for i=1:nPop
  % Create Random Solution
  pop(i).Position=unifrnd(VarMin,VarMax,VarSize);
  % Evaluation
  pop(i).Cost=CostFunction(pop(i).Position);
end
              % Sort Population
              [~, SortOrder]=sort([pop.Cost]);
              pop=pop(SortOrder);
              % Update Best Solution Ever Found
              BestSol=pop(1);
              % Array to Hold Best Cost Values
              BestCost=zeros(MaxIt,1);
              % Solution Weights
              w=1/(sqrt(2*pi)*q*nPop)*exp(-0.5*(((1:nPop)-1)/(q*nPop)).^2);
              % Selection Probabilities
              p=w/sum(w);
              % Initialize Temp.
              T=T0;

%% HACO-SA Main Loop
for it=1:MaxIt
    for subit=1:MaxSubIt
  % Means
  s=zeros(nPop,nVar);
  for l=1:nPop
    s(l,:)=pop(l).Position;
  end
  % Standard Deviations
  sigma=zeros(nPop,nVar);
  for l=1:nPop
    D=0;
    for r=1:nPop
      D=D+abs(s(l,:)-s(r,:));
    end
    sigma(l,:)=zeta*D/(nPop-1);
  end
  % Create New Population Array
  newpop=repmat(empty_individual,nSample,1);
  for t=1:nSample
    % Initialize Position Matrix
    newpop(t).Position=zeros(VarSize);
    % Solution Construction
    for i=1:nVar
      % Select Gaussian Kernel
      l=RouletteWheelSelection(p);
      % Generate Gaussian Random Variable
      newpop(t).Position(i)=s(l,i)+sigma(l,i)*randn;
    end
    % Evaluation
    newpop(t).Cost=CostFunction(newpop(t).Position);
  end
  % Merge Main Population (Archive) and New Population (Samples)
  pop=[pop
     newpop];
  % Sort Population
  [~, SortOrder]=sort([pop.Cost]);
  pop=pop(SortOrder);
  % Delete Extra Members
  pop=pop(1:nPop);
  % Update Best Solution Ever Found
  BestSol=pop(1);
% Compare using SA Rule
    for i=1:nPop
      if newpop(i).Cost<=pop(i).Cost
        pop(i)=newpop(i);
      else
        DELTA=(newpop(i).Cost-pop(i).Cost)/pop(i).Cost;
        P=exp(-DELTA/T);
        if rand<=P
          pop(i)=newpop(i);
        end
      end
             % Update Best Solution Ever Found
             if pop(i).Cost<=BestSol.Cost
               BestSol=pop(i);
             end
  % Store Best Cost
  BestCost(it)=BestSol.Cost;
  % Show Iteration Information
  disp(['Iteration ' num2str(it) ': Best Cost = ' num2str(BestCost(it))]);
 % Temp. Reduction
  T=alpha*T;
end
end
```

References

Accorsi, R., Cholette, S., Manzini, R., Pini, C., & Penazzi, S. (2016). The land-network problem: Ecosystem carbon balance in planning sustainable agro-food supply chains. *Journal of Cleaner Production, 112*, 158–171. https://doi.org/10.1016/j.jclepro.2015.06.082

Bortolini, M., Faccio, M., Ferrari, E., Gamberi, M., & Pilati, F. (2016). Fresh food sustainable distribution: Cost, delivery time and carbon footprint three-objective optimization. *Journal of Food Engineering, 174*, 56–67. https://doi.org/10.1016/j.jfoodeng.2015.11.014

Bottani, E., Murino, T., Schiavo, M., & Akkerman, R. (2019). Resilient food supply chain design: Modelling framework and metaheuristic solution approach. *Computers & Industrial Engineering, 135*, 177–198. https://doi.org/10.1016/j.cie.2019.05.011

Cardoso, P., Jesus, M., & Marquez, A. (2003a). MONACO-multi-objective network optimisation based on an ACO. *Proc. X Encuentros de Geometrıa Computacional, Seville, Spain*.

Cardoso, P., Jesus, M., & Márquez, A. (2003b). Multiple Objective TSP based on ACO. *III Encuentro Andaluz de Matemáticas Discretas, Universidad de Almeria, Almeria*.

Cooper, K., Hunter, S. R., & Nagaraj, K. (2017). An epsilon-constraint method for integer-ordered bi-objective simulation optimization. In *2017 Winter Simulation Conference (WSC)* (pp. 2303–2314). IEEE. https://doi.org/10.1109/WSC.2017.8247961

Cheraghalipour, A., Paydar, M. M., & Hajiaghaei-Keshteli, M. (2019). Designing and solving a bi-level model for rice supply chain using the evolutionary algorithms. *Computers and Electronics in Agriculture, 162*, 651–668. https://doi.org/10.1016/j.compag.2019.04.041

Deb, K., Pratap, A., & Agarwal, S. (2002). Meyarivan TJItoec. *A Fast and Elitist Multiobjective Genetic Algorithm, 6*, 182–197. https://doi.org/10.1109/4235.996017

Devika, K., Jafarian, A., & Nourbakhsh, V. (2014). Designing a sustainable closed-loop supply chain network based on triple bottom line approach: A comparison of metaheuristics hybridization techniques. *European Journal of Operational Research, 235*(3), 594–615. https://doi.org/10.1016/j.ejor.2013.12.032

Dorigo, M. and Di Caro, G. (1999). Ant colony optimization: a new meta-heuristic. In *Proceedings of the 1999 congress on evolutionary computation-CEC99 (Cat. No. 99TH8406)* (Vol.2, pp.1470–1477). IEEE. https://doi.org/10.1109/CEC.1999.782657

Doerner, K., Gutjahr, W. J., Hartl, R. F., Strauss, C., & Stummer, C. (2004). Pareto ant colony optimization: A metaheuristic approach to multiobjective portfolio selection. *Annals of Operations Research, 131*(1), 79–99. https://doi.org/10.1023/B:ANOR.0000039513.99038.c6

Dwivedi, A., Jha, A., Prajapati, D., Sreenu, N., & Pratap, S. (2020). Meta-heuristic algorithms for solving the sustainable agro-food grain supply chain network design problem. *Modern Supply Chain Research and Applications*. https://doi.org/10.1108/MSCRA-04-2020-0007

Eskandari, H., & Geiger, C. D. (2008). A fast Pareto genetic algorithm approach for solving expensive multi-objective optimization problems. *Journal of Heuristics, 14*(3), 203–241. https://doi.org/10.1007/s10732-007-9037-z

Esmaeilikia, M., Fahimnia, B., Sarkis, J., Govindan, K., Kumar, A., & Mo, J. (2016). Tactical supply chain planning models with inherent flexibility: Definition and review. *Annals of Operations Research, 244*(2), 407–427. https://doi.org/10.1007/s10479-014-1544-3

Etemadnia, H., Goetz, S. J., Canning, P., & Tavallali, M. S. (2015). Optimal wholesale facilities location within the fruit and vegetables supply chain with bimodal transportation options: An LP-MIP heuristic approach. *European Journal of Operational Research, 244*(2), 648–661. https://doi.org/10.1016/j.ejor.2015.01.044

Fathollahi-Fard, A. M., Ahmadi, A., Goodarzian, F., & Cheikhrouhou, N. (2020). A bi-objective home health-care routing and scheduling problem considering patients' satisfaction in a fuzzy environment. *Applied Soft Computing*. https://doi.org/10.1016/j.asoc.2020.106385

Fakhrzad, M. B., & Goodarzian, F. (2021). A new multi-objective mathematical model for a Citrus supply chain network design: Metaheuristic algorithms. *Journal of Optimization in Industrial Engineering, 14*(2), 127–144. https://doi.org/10.22094/JOIE.2020.570636.1571

García-Martínez, C., Cordón, O., & Herrera, F. (2007). A taxonomy and an empirical analysis of multiple objective ant colony optimization algorithms for the bi-criteria TSP. *European Journal of Operational Research, 180*(1), 116–148. https://doi.org/10.1016/j.ejor.2006.03.041

Govindan, K., Jafarian, A., & Nourbakhsh, V. (2015). Bi-objective integrating sustainable order allocation and sustainable supply chain network strategic design with stochastic demand using a novel robust hybrid multi-objective metaheuristic. *Computers & Operations Research, 62*, 112–130. https://doi.org/10.1016/j.cor.2014.12.014

Govindan, K., Jafarian, A., & Nourbakhsh, V. (2019). Designing a sustainable supply chain network integrated with vehicle routing: A comparison of hybrid swarm intelligence metaheuristics. *Computers & Operations Research, 110*, 220–235. https://doi.org/10.1016/j.cor.2018.11.013

Goodarzian, F., Hosseini-Nasab, H., Muñuzuri, J., & Fakhrzad, M. B. (2020). A multi-objective pharmaceutical supply chain network based on a robust fuzzy model: A comparison of meta-heuristics. *Applied Soft Computing*. https://doi.org/10.1016/j.asoc.2020.106331

Goodarzian, F., Abraham, A., & Fathollahi-Fard, A. M. (2021a). A biobjective home health care logistics considering the working time and route balancing: A self-adaptive social engineering optimizer. *Journal of Computational Design and Engineering, 8*(1), 452–474. https://doi.org/10.1093/jcde/qwaa089

Goodarzian, F., Taleizadeh, A. A., Ghasemi, P., & Abraham, A. (2021b). An integrated sustainable medical supply chain network during COVID-19. *Engineering Applications of Artificial Intelligence, 100*, 104188. https://doi.org/10.1016/j.engappai.2021.104188

Goodarzian, F., Shishebori, D., Nasseri, H., & Dadvar, F. (2021c). A bi-objective production-distribution problem in a supply chain network under grey flexible conditions. *RAIRO-Operations Research, 55*, S1287–S1316. https://doi.org/10.1051/ro/2020111

Gómez-Lagos, J. E., González-Araya, M. C., Soto-Silva, W. E., & Rivera-Moraga, M. M. (2021). Optimizing tactical harvest planning for multiple fruit orchards using a metaheuristic modeling approach. *European Journal of Operational Research, 290*(1), 297–312. https://doi.org/10.1016/j.ejor.2020.08.015

Hyland, M. F., Mahmassani, H. S., & Mjahed, L. B. (2016). Analytical models of rail transportation service in the grain supply chain: Deconstructing the operational and economic advantages of shuttle train service. *Transportation Research Part e: Logistics and Transportation Review, 93*, 294–315. https://doi.org/10.1016/j.tre.2016.06.008

Jabarzadeh, Y., Yamchi, H. R., Kumar, V., & Ghaffarinasab, N. (2020). A multi-objective mixed-integer linear model for sustainable fruit closed-loop supply chain network. *Management of Environmental Quality: An International Journal, 31*(5), 1351–1373.

Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science, 220*(4598), 671–680. https://doi.org/10.1126/science.220.4598.671

Kamble, S. S., Gunasekaran, A., & Sharma, R. (2020). Modeling the blockchain enabled traceability in agriculture supply chain. *International Journal of Information Management, 52*, 101967. https://doi.org/10.1016/j.ijinfomgt.2019.05.023

Keshavarz-Ghorbani, F., & Pasandideh, S. H. R. (2021). A Lagrangian relaxation algorithm for optimizing a bi-objective agro-supply chain model considering CO 2 emissions. *Annals of Operations Research*. https://doi.org/10.1007/s10479-021-03936-1

Loukil, T., Teghem, J., & Fortemps, P. (2007). A multi-objective production scheduling case study solved by simulated annealing. *European Journal of Operational Research, 179*(3), 709–722. https://doi.org/10.1016/j.ejor.2005.03.073

Lamsal, K., Jones, P. C., & Thomas, B. W. (2016). Harvest logistics in agricultural systems with multiple, independent producers and no on-farm storage. *Computers & Industrial Engineering, 91*, 129–138. https://doi.org/10.1016/j.cie.2015.10.018

Liao, Y., Kaviyani-Charati, M., Hajiaghaei-Keshteli, M., & Diabat, A. (2020). Designing a closed-loop supply chain network for citrus fruits crates considering environmental and economic issues. *Journal of Manufacturing Systems, 55*, 199–220. https://doi.org/10.1016/j.jmsy.2020.02.001

Mavrotas, G. (2009). Effective implementation of the ε-constraint method in multi-objective mathematical programming problems. *Applied Mathematics and Computation, 213*(2), 455–465. https://doi.org/10.1016/j.amc.2009.03.037

Mavrotas, G., & Florios, K. (2013). An improved version of the augmented ε-constraint method (AUGMECON2) for finding the exact pareto set in multi-objective integer programming problems. *Applied Mathematics and Computation, 219*(18), 9652–9669. https://doi.org/10.1016/j.amc.2013.03.002

Mousavi, S. M., Alikar, N., Niaki, S. T. A., & Bahreininejad, A. (2015). Optimizing a location allocation-inventory problem in a two-echelon supply chain network: A modified fruit fly optimization algorithm. *Computers & Industrial Engineering, 87*, 543–560. https://doi.org/10.1016/j.cie.2015.05.022

Masson, R., Lahrichi, N., & Rousseau, L. M. (2016). A two-stage solution method for the annual dairy transportation problem. *European Journal of Operational Research, 251*(1), 36–43. https://doi.org/10.1016/j.ejor.2015.10.058

Mogale, D. G., Kumar, S. K., Márquez, F. P. G., & Tiwari, M. K. (2017a). Bulk wheat transportation and storage problem of public distribution system. *Computers & Industrial Engineering, 104*, 80–97. https://doi.org/10.1016/j.cie.2016.12.027

Mogale, D. G., Dolgui, A., Kandhway, R., Kumar, S. K., & Tiwari, M. K. (2017b). A multi-period inventory transportation model for tactical planning of food grain supply chain. *Computers & Industrial Engineering, 110*, 379–394. https://doi.org/10.1016/j.cie.2017.06.008

Mehrbanfar, M., Bozorgi-Amiri, A., & Nasiri, M. M. (2020). A mathematical programming model for sustainable agricultural supply chain network design under uncertainty. *Journal of Quality Engineering and Production Optimization, 5*(1), 189–220. https://doi.org/10.22070/JQEPO.2020.5666.1164

Mogale, D. G., Kumar, S. K., & Tiwari, M. K. (2020). Green food supply chain design considering risk and post-harvest losses: A case study. *Annals of Operations Research, 295*, 257–284. https://doi.org/10.1007/s10479-020-03664-y

Motevalli-Taher, F., Paydar, M. M., & Emami, S. (2020). Wheat sustainable supply chain network design with forecasted demand by simulation. *Computers and Electronics in Agriculture, 178*, 105763. https://doi.org/10.1016/j.compag.2020.105763

Naderi, B., & Ruiz, R. (2010). The distributed permutation flowshop scheduling problem. *Computers & Operations Research, 37*(4), 754–768. https://doi.org/10.1016/j.cor.2009.06.019

Osvald, A., & Stirn, L. Z. (2008). A vehicle routing algorithm for the distribution of fresh vegetables and similar perishable food. *Journal of Food Engineering, 85*(2), 285–295. https://doi.org/10.1016/j.jfoodeng.2007.07.008

Poorzahedy, H., & Rouhani, O. M. (2007). Hybrid meta-heuristic algorithms for solving network design problem. *European Journal of Operational Research, 182*(2), 578–596. https://doi.org/10.1016/j.ejor.2006.07.038

Roghanian, E., & Cheraghalipour, A. (2019). Addressing a set of meta-heuristics to solve a multi-objective model for closed-loop citrus supply chain considering CO_2 emissions. *Journal of Cleaner Production, 239*, 118081. https://doi.org/10.1016/j.jclepro.2019.118081

Rong, A., Akkerman, R., & Grunow, M. (2011). An optimization approach for managing fresh food quality throughout the supply chain. *International Journal of Production Economics, 131*(1), 421–429. https://doi.org/10.1016/j.ijpe.2009.11.026

Suman, B., & Kumar, P. (2006). A survey of simulated annealing as a tool for single and multiobjective optimization. *Journal of the Operational Research Society, 57*(10), 1143–1160. https://doi.org/10.1057/palgrave.jors.2602068

Soto-Silva, W. E., Nadal-Roig, E., González-Araya, M. C., & Pla Aragones, L. M. (2016). Operational research models applied to the fresh fruit supply chain. *European Journal of Operational Research, 251*(2), 345–355. https://doi.org/10.1016/j.ejor.2015.08.046

Sgarbossa, F., & Russo, I. (2017). A proactive model in sustainable food supply chain: Insight from a case study. *International Journal of Production Economics, 183*, 596–606. https://doi.org/10.1016/j.ijpe.2016.07.022

Sellitto, M. A., Vial, L. A. M., & Viegas, C. V. (2018). Critical success factors in Short Food Supply Chains: Case studies with milk and dairy producers from Italy and Brazil. *Journal of Cleaner Production, 170*, 1361–1368. https://doi.org/10.1016/j.jclepro.2017.09.235

Sharma, R., Kamble, S. S., Gunasekaran, A., Kumar, V., & Kumar, A. (2020). A systematic literature review on machine learning applications for sustainable agriculture supply chain performance. *Computers & Operations Research*. https://doi.org/10.1016/j.cor.2020.104926

Sahebjamnia, N., Goodarzian, F., & Hajiaghaei-Keshteli, M. (2020). Optimization of multi-period three-echelon citrus supply chain problem. *Journal of Optimization in Industrial Engineering, 13*(1), 39–53. https://doi.org/10.22094/JOIE.2017.728.1463

Taguchi, G. (1986). *Introduction to quality engineering: designing quality into products and processes* (No. 658.562 T3).

Validi, S., Bhattacharya, A., & Byrne, P. J. (2014a). Integrated low-carbon distribution system for the demand side of a product distribution supply chain: A DoE-guided MOPSO optimiser-based solution approach. *International Journal of Production Research, 52*(10), 3074–3096. https://doi.org/10.1080/00207543.2013.864054

Validi, S., Bhattacharya, A., & Byrne, P. J. (2014b). A case analysis of a sustainable food supply chain distribution system—A multi-objective approach. *International Journal of Production Economics, 152*, 71–87. https://doi.org/10.1016/j.ijpe.2014.02.003

Validi, S., Bhattacharya, A., & Byrne, P. J. (2015). A solution method for a two-layer sustainable supply chain distribution model. *Computers & Operations Research, 54*, 204–217. https://doi.org/10.1016/j.cor.2014.06.015

Validi, S., Bhattacharya, A., & Byrne, P. J. (2020). Sustainable distribution system design: A two-phase DoE-guided meta-heuristic solution approach for a three-echelon bi-objective AHP-integrated location-routing model. *Annals of Operations Research, 290*(1), 191–222. https://doi.org/10.1007/s10479-018-2887-y

Varas, M., Basso, F., Maturana, S., Osorio, D., & Pezoa, R. (2020). A multi-objective approach for supporting wine grape harvest operations. *Computers & Industrial Engineering, 145*, 106497. https://doi.org/10.1016/j.cie.2020.106497

Validi, S., Bhattacharya, A., & Byrne, P. J. (2021). An evaluation of three DoE-guided meta-heuristic-based solution methods for a three-echelon sustainable distribution network. *Annals of Operations Research, 296*(1), 421–469. https://doi.org/10.1007/s10479-020-03746-x

Yang, Z., Cai, X., & Fan, Z. (2014). Epsilon constrained method for constrained multiobjective optimization problems: some preliminary results. In *Proceedings of the companion publication of the 2014 annual conference on genetic and evolutionary computation* (pp. 1181–1186). https://doi.org/10.1145/2598394.2610012

Yakavenka, V., Mallidis, I., Vlachos, D., Iakovou, E., & Eleni, Z. (2020). Development of a multi-objective model for the design of sustainable supply chains: The case of perishable food products. *Annals of Operations Research, 294*(1), 593–621. https://doi.org/10.1007/s10479-019-03434-5

Yagmahan, B., & Yenisey, M. M. (2010). A multi-objective ant colony system algorithm for flow shop scheduling problem. *Expert Systems with Applications, 37*(2), 1361–1368. https://doi.org/10.1016/j.eswa.2009.06.105

Yang, X., Leng, Z., Shaoping, X., Yang, C., Yang, L., Liu, K., Song, Y., & Zhang, L. (2021). Multi-objective optimal scheduling for CCHP microgrids considering peak-load reduction by augmented ε-constraint method. *Renewable Energy, 172*, 408–423. https://doi.org/10.1016/j.renene.2021.02.165

Zhang, W., & Reimann, M. (2014). A simple augmentedϵ-constraint method for multi-objective mathematical integer programming problems. *European Journal of Operational Research, 234*(1), 15–24. https://doi.org/10.1016/j.ejor.2013.09.001