Iranian Journal of Management Studies (IJMS) 2021, 14(3): 487-507
RESEARCH PAPER

A Location-Inventory-Pricing Supply Chain Network Design for Perishable Products Under Disruptions

Tooba Asghari, Soroush Aghamohamadi-Bosjin, Masoud Rabbani*

School of Industrial Engineering, College of Engineering, University of Tehran, Tehran, Iran

(Received: September 14, 2019; Revised: August 9, 2020 ; Accepted: August 11, 2020)

Abstract
In this study, we discuss a location-inventory-pricing model considering the capacity constraints of the warehouses, disruption, and multiple perishable products. We extend a model that assumes that warehouses may face disruption, failed warehouses cannot cover any service, and their customers are assigned to other warehouses. To decrease the risk of disruption, we examine the efficiency of markup pricing strategy and support services. The objective function of this MINLP is to maximize the total profit of warehouses. To solve this model, Genetic Algorithm (GA) and Grasshopper Optimization Algorithm (GOA) are used. To evaluate the recommended model, several sensitivity analyses are proposed. Finally, the results of numerical experiments implicate the high-performance of GOA in dealing with problems and achieving better results. According to the results, backup services and markup pricing strategies are very effective in reducing the damage caused by the disruption.

Keywords: Location-inventory, Perishability, Markup pricing, Disruption, Meta-heuristic algorithms.

Introduction

One of the significant issues in the supply chain is network design. Therefore, in creating a supply chain distribution network, the essential components of the supply chain such as transportation, equipment, inventory, and pricing should be examined together to promote the competitive strategy of a system and maximize the benefit of the supply chain (Li & Hai, 2019). Moreover, Location-inventory problems have attracted the attention of researchers over the past decade. One of the significant issues leading to an increase in the profit of the supply chain in a competitive situation is to design an effective supply chain network (Fahimi et al., 2018; Nemati et al., 2017). Location and inventory control decisions are in the category of strategic and operational/tactical decisions, respectively (Gzara et al., 2014). Traditionally, the storage and the distribution of products have been controlled independently. In the current situation, joint location-inventory problems are more developed.

Most of the research on location-inventory problem presumes that facilities are always ready to provide service for customers. Nonetheless, in recent years impressive consideration has been paid to location-inventory problems under disruption. Disruptions may happen at any time, for instance, due to natural disasters, fires, labor strikes, terrorist attacks, equipment failures, economic tension, etc. Generally, it can happen in any situation that in which there is a lack of access to facilities. Consequently, the consideration of the risk of disturbance leads to a more useful solution and makes the model more realistic (Farahani et al., 2017).

Basic supply chain models usually suppose that the life of goods is unlimited, while many products (e.g., meat, human blood, flowers, etc.) are perishable. Approximately 10% of the

* Corresponding Author, Email : mrabani@ut.ac.ir

perishable products are lost before they are bought by consumers. Therefore, the cost-induced deterioration should not be overlooked in real cases. In this study, the lost costs of perishable goods during transmission will be examined (Dai et al., 2018). It is necessary to mention that while few studies considered capacity constraints on their studies, most of real cases deal with capacity constraints.

Most of the studies on location-inventory problems have minimized the cost of the supply chain; only a few studies have investigated profit-maximization models with demand choice flexibility. In these kinds of models, the prices only affect consumers’ decisions to get service or not, and the quantity of demand is independent of the pricing strategy (Ahmadi-Javid & Hoseinpour, 2015a). In recent years, the extension of innovative pricing policies has been observed in various industries. Indeed, most of the companies have started using dynamic pricing to improve their systems (Kaya & Urek, 2016).

In this article, a location-inventory-pricing problem for multiple independent perishable products is proposed that investigates the markup and probability of a disruption in warehouses. This network consists of a plant, potential warehouses (in which inventory management decisions are included only), and retailers. When disruption occurs, the failed warehouses cannot provide any service, and their consumers are assigned to other warehouses. Therefore, warehouses attempt to reduce the impact of disruption. To fill these gaps, warehouses implement pricing strategies. Markup pricing is the strategy applied in this paper. Through markup pricing in each level of retailer assignment to a warehouse, the price will be increased to reduce the mentioned disruption. Our model's aim is to maximize the profit of warehouses. Based on the aforementioned information, the contributions of this study that distinguish it from other relevant research include:

- Analyzing a three-echelon location-inventory-pricing supply chain problem for perishable products
- Considering disruption and markup simultaneously
- Maximizing the profit of the warehouses
- Considering various perishable goods with various unit holding and ordering costs
- Considering warehouses with limited capacities
- Using a new metaheuristic algorithm called Grasshopper Optimization Algorithm (GOA) to solve LIP with large size.

Literature Review

In this part, the recent studies on joint location -inventory problem under facility disruption and price decision are reviewed. Tavakkoli-Moghaddam et al. (2018) improved the MINLP model for a two-echelon closed-loop supply. The model tried to minimize the supply chain costs for spare parts. They used particle swarm optimization (PSO) to solve the large instance in a reasonable time. Vahdani et al. (2017) used a genetic algorithm (GA) and simulated annealing (SA) to minimize total supply chain costs on a location-inventory problem that considered inventory shortages and correlated demand of the retailers. Orand et al. (2015) extended the inflationary inventory model under non-deterministic situations. Their model attempted to minimize the total discount cost of the inventory system. They used the classic numerical approach and Simpson approximation and particle swarm optimization (PSO) to solve the problem. Puga and Tancrez (2017) investigated a heuristic algorithm to solve large size LIP with non-deterministic demands. In their suggested model, three categories of decision-making were combined simultaneously that involved the number of open facilities, inventory management, and the location and allocation of the equipment and retailers. The purpose of solving this problem was to minimize the costs of the supply chain. Dai et al.

(2018) extended a MINLP model to formulate a location-inventory problem into a supply chain network with the consideration of perishable goods and fuzzy capacity constraints. Their proposed model was solved using a hybrid genetic algorithm (HGA) and hybrid harmony search (HHS) to minimize the costs. Hiassat et al. (2017) extended a genetic algorithm method to solve an LIP with perishable goods. In addition, they created a novel chromosome to design the individual structure of problem. Hamdan and Diabat (2019) presented a stochastic coding LIP that entailed the four levels of the RBC supply chain. They used CPLEX to solve the suggested model for minimizing the costs, the transfer time, and the number of outdated goods. Rafie-Majd et al. (2018) studied a joined location-allocation, routing, and inventory problem with multiple perishable goods under non-deterministic situations. They used a Lagrangian relaxation algorithm to solve this NP-HARD model. Guerrero et al. (2015) proposed the Lagrangian relaxation algorithm to solve the Location-Inventory-Routing Problem to optimize a supply chain design.

Ahmadi-Javid and Hoseinpour (2015a) offered a location-inventory-pricing model with the consideration of the capacity limitation. Each DC had a restricted warehouse place that affected consumer assignment to open DCs. They proposed a Lagrangian relaxation algorithm to obtain near-optimal solutions to this issue. Ghasemy Yaghin et al. (2017) proposed a fuzzy non-linear model that included pricing and inventory decisions to maximize return on inventory investment (ROII). Ahmadzadeh and Vahdani (2017) introduced a nonlinear programming model for a joint location-inventory-pricing issue in a three-echelon closed loop supply chain. In addition, they considered periodic review policy (R, T) for inventory management. They used three metaheuristic algorithms to solve it, including genetic algorithm (GA), imperialist competitive algorithm (ICA), and firefly algorithm (FA). Li and Hai (2019) investigated an inventory-pricing problem with the capacity constraint of the facility, and proposed a Lagrangian relaxation algorithm to solve the nonlinear integer programming model. Ahmadi-Javid and Hoseinpour (2015b) offered a location-inventory-pricing model by assuming continuous review inventory policy, multi-goods supply chain, price-sensitive demands, and capacity constraint. They proposed a Lagrangian relaxation approach to solve the large-size instances. Kaya and Urek (2016) introduced a location-inventory-pricing model in a closed loop supply chain to obtain optimal locations of the facilities, inventory quantities, and price of goods to maximize total profit. They integrated a group of used goods with the distribution of the new goods. Chen and Hu (2012) presented a pricing and inventory model of a single product with price regulation costs and deterministic demands. They determined an ordering volume and a price simultaneously at the beginning of each period. Furthermore, they developed polynomial-time algorithms to maximize profit. Chen and Zhou et al. (2011) investigated an inventory-pricing model that took into account price adjustments and multi-period to gain optimal supply chain profit. Etebari and Dabiri (2016) addressed a heuristic algorithm to solve a quadratic mixed-integer programming model and considered multi-period Inventory Routing and dynamic pricing strategy in their model. Ahmadi-Javid et al. (2018) offered a mixed-integer linear programming model for a location-routing problem by the consideration of price-sensitive demands. They used the branch-and-price algorithm to solve MINLP model for large-size instances. Smith and Agrawal (2017) simultaneously optimized the prices and inventory allocation across multiple retail locations in the presence of inventory dependent demand. Moreover, they analyzed how inventory dependence of demand affected the optimal pricing and distribution of inventory.

Dehghani et al. (2018) analyzed an integrated location-inventory problem in which facilities provisionally were not available. They also presented a method that included the Markov process and mathematical programming for designing the supply chain’s channels. They developed a simulated annealing algorithm to solve this issue. Zhang et al. (2016)

investigated a reliable location- inventory problem to analyze supply chain with heterogeneous disruption probabilities. They applied heuristics based on Lagrangian relaxation to minimize expected cost. Zhang et al. (2015) addressed a reliable capacitated location–routing problem in which depots were randomly faced with disruption. They also extended the simulated annealing algorithm to obtain optimum depot location, distribution routing, and backup systems. Chen and Li et al. (2011) investigated LRP under department disruption risk. They supposed that when a department breaks, its customers are diverted to other open departments to reduce the lost sale cost. They developed a Lagrangian relaxation solution to minimize the total cost. Farahani et al. (2017) introduced a multiple goods inventory-location problem under disruption in which facilities might fail partially and examined substitutable goods to decrease the damage of disruption. They proposed a hybrid algorithm based on Tabu Search (TS) and Variable Neighborhood Search (VNS). Asl-Najafi et al. (2015) addressed LRP for a multiproduct closed-loop supply chain in which facility disruption was taken into account. They developed a hybrid meta-heuristic algorithm consisting of Multi-Objective Particle Swarm Optimization (MOPSO) and Non-dominated Sorting Genetic Algorithm-II (NSGA-II).

According to Table 1, there are few papers that integrate the location-inventory problem with the price decisions. In addition, disruption in the system influences many elements such as changing the price of goods. This is a critical issue in real problems that have not been addressed before. Thus, it is a good idea to design a perishable product inventory location supply chain considering assumptions like disruption. Based on the mentioned gaps in location-inventory supply chain, assuming multi perishable goods and price decisions under the risk of disruptions will be a necessary step in the literature.

Table 1. Abstract Review of Selected Articles

Article	**chain echelon**		**location**	**inventory**	**routing**	**perishability**	**Price decision**	**disruption**	**Multi-product**	**Objective function**	**solution**
	bi-echelon	**three-echelon**									
Hamdan & Diabat (2019)			*	*		*			*	Cost/ out date / blood delivery time	CPLEX
Dehghani et al. (2018)		*	*	*				*		cost	SA
Tavakkoli-Moghaddam et al. (2018)	*		*	*					*	cost	PSO
Vahdani et al. (2017)		*	*	*						cost	GA & SA
Ahmadzadeh & Vahdani (2017)		*	*	*			*			profit	GA, ICA & FA
Zhang et al. (2016)			*	*				*		expected cost	Lagrangian relaxation algorithm
Ahmadi-Javid & Hoseinpour (2015a)	*		*	*			*		*	profit	Lagrangian relaxation algorithm
Zhang et al. (2015)	*		*		*			*		cost	SA
Asl-Najafi et al. (2015)		*	*	*				*		cost	MOPSO & NSGA-II
Chen and Li et al. (2011)	*		*	*				*		cost	Lagrangian relaxation algorithm
Current study		✓	✓	✓		✓	✓	✓	✓	profit	GOA & GA

Problem Statement and Formulation

Problem Description

This model examines a joint location-inventory supply chain and price decision for multiple independent perishable goods, which investigates markup and the probability of a disruption in the warehouse. This network consists of three echelons, namely a plant, warehouses, and retailers. The main goal of this model is to maximize the profit of warehouses. To this end, it considers the total revenue of the warehouses, the purchasing costs of products, the lost sale costs (when disruptions occur), warehouse holding and ordering costs, fixed costs, transportation costs, and loss costs of perishable products. The main decisions are described as follow:

- The optimal number of warehouses
- The locations of warehouses
- The optimal allocation of retailers to open warehouses
- Order-size decisions at open warehouses
- The retail price decisions of products offered at each warehouse

Assumptions:

- Inventory control decisions are considered only at warehouses, and the multi-product EOQ (Economic Order Quantity) strategy is assumed in this problem.
- The holding and ordering costs are different for each kind of perishable products.
- Products deteriorate only during the transition between warehouses and retailers. If transportation time surpasses the crucial time, the perishable products will be thrown away.
- The deterioration rate of perishable products is deterministic and known for each product.
- The lead time and stock shortages for warehouses are not considered in this model.
- Each retailer is allocated to only one warehouse at each assignment level.
- Fixed costs for opening warehouses are pre-designated.
- Transportation costs depend on the distance and number of perishable products.
- Warehouses are controlled with the capacity constraint.
- Each opened warehouse breaks independently with probability q. If a warehouse fails, it cannot provide any service and its original customers will be diverted to other warehouses. In addition, each retailer is allowed to get service from a sequence of $R \leq |J|$ warehouses.
- According to the previous assumptions, two scenarios are considered. Under the normal scenario (where no warehouses fail), a retailer is assigned to its level-1 warehouse. Under the second scenario, the probability for retailer i to get service from its level-r facility is $q^{r-1}(1-q)$. If all R assigned warehouses for each retailer fail, the probability will be q^R(Chen and Li et al., 2011). Further, when a disruption occurs, warehouses attempt to reduce the impact of that disruption. To fulfill this purpose, warehouses implement pricing strategies. Markup pricing is the strategy applied in this paper. Through markup pricing in each level of retailer assignment to a warehouse, the price will be increased to reduce the mentioned disruption.

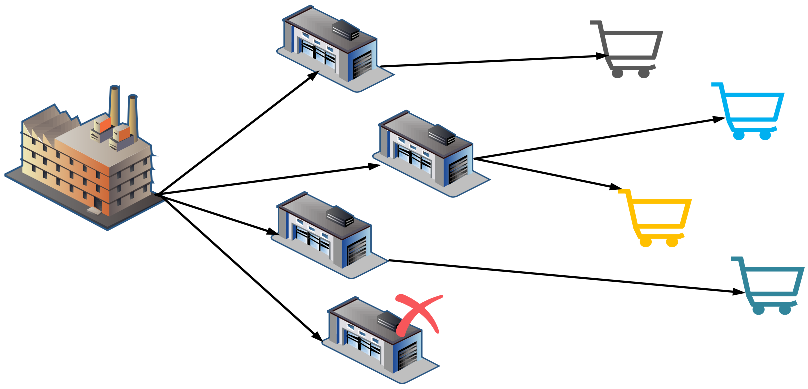

Figure 1. An Overview of Discussed Problem

Notation

Parameter	Definition
$\boldsymbol{k}$	index of perishable product, $k\epsilon\{1,\dots,K\}$
j	index of potential warehouses, $j\epsilon\{1,\dots,J\}$
i	index of retailers, $j\epsilon\{1,\dots,I\}$
R	The levels in the succession of the warehouse that a retailer should visit to gain the demanded product (index r)
q	disruption probabilities for a warehouse
$\boldsymbol{D_{ik}}$	the annual demand of retailer i for perishable product k
$\boldsymbol{f_j}$	annual fixed cost of warehouse j
$\boldsymbol{s_k}$	loss cost for per unit of perishable product k
$\boldsymbol{d_{pj}}$	distance between plant and warehouse j
$\boldsymbol{d_{ji}}$	distance between warehouse j and retailer i
$\boldsymbol{c_k}$	unit cost of transportation product k per km
$\boldsymbol{A_{jk}}$	ordering cost at warehouse j for per unit perishable product k
$\boldsymbol{h_{jk}}$	holding cost at warehouse j for per unit perishable product k
$\boldsymbol{T_k}$	the critical time of perishable product k
$\boldsymbol{\theta_k}$	deterioration rate of perishable product k
$\boldsymbol{w_j}$	annual capacity constraint for warehouse j
$\boldsymbol{V_k}$	the volume of per perishable product k
$\boldsymbol{v}$	speed of vehicle
$\boldsymbol{\pi}$	lost sale cost for per unit of demand
$\boldsymbol{b_{jr}}$	the markup percentage of product determined by warehouse j
$\boldsymbol{C_{jk}}$	the wholesale price of product k at warehouse j
$\boldsymbol{p_{jrk}}$	the retail price of product k at warehouse j in level r
Decision variables	
$\boldsymbol{x_{jir}}$	1, if retailer i is allocated to warehouse j at level r ; 0, otherwise
$\boldsymbol{y_j}$	1, if warehouse j is established; 0, otherwise
$\boldsymbol{Q_{jk}}$	ordering quantity of perishable product k at each warehouse j

Formulation of Model

According to the assumption that the wholesale price is predetermined, determining the retail price at each warehouse is equivalent to determining its proper markup. Retail price per unit of product k at warehouse j in level r equals:

$$p_{jrk} = \left(1 + b_{jr}\right) C_{jk} \tag{1}$$

Accordingly, the probability for retailer i to receive service from its level-r warehouse is $q^{r-1}(1-q)$. The annual demand for product k in warehouse j is calculated as follows:

$$D_{jk} = \sum_{i \epsilon I} \sum_{r=1}^{R} q^{r-1} \left(1-q\right) D_{ik} x_{jir} \tag{2}$$

If all R assigned warehouses for each retailer fail, the probability will be q^R and lost sale cost is not acceptable for warehouses:

$$LS = \pi \sum_{i \epsilon I} \sum_{k \epsilon K} q^R D_{ik} \tag{3}$$

According to the assumption that the wholesale price is predetermined, various perishable goods have different holding and ordering costs. So, the optimum ordering size(Q_{jk}) is represented as:

$$Q_{jk} = \sqrt{2 A_{jk} D_{jk} / h_{jk}} \tag{4}$$

Warehouses' total holding cost is:

$$WHC = \sum_{j \epsilon J} \sum_{k \epsilon K} Q_{jk} h_{jk} / 2 \tag{5}$$

And, warehouses' total ordering cost is formulated as:

$$WOC = \sum_{j \epsilon J} \sum_{k \epsilon K} Q_{jk} A_{jk} \tag{6}$$

Fixed costs of warehouses are shown by:

$$FC = \sum_{j \epsilon J} f_j y_j \tag{7}$$

The costs for transportation from warehouses to retailers and plant to warehouse are computed by:

$$TC = \sum_{j \epsilon J} \sum_{i \epsilon I} \sum_{k \epsilon K} \sum_{r=1}^{R} c_k q^{r-1} \left(1-q\right) D_{ik} x_{jir} d_{ji} + \sum_{j \epsilon J} \sum_{k \epsilon K} c_k d_{pj} Q_{jk} \tag{8}$$

Therefore, the loss costs of perishable products are:

$$LC = \sum_{j \epsilon J} \sum_{i \epsilon I} \sum_{k \epsilon K} \sum_{r=1}^{R} s_k q^{r-1} \left(1-q\right) D_{ik} x_{jir} \theta_{ji} \tag{9}$$

The purchasing costs of products for warehouses:

$$PC = \sum_{k \epsilon K} \sum_{j \epsilon J} C_{jk} Q_{jk} y_j \tag{10}$$

The total revenue of the warehouses is formulated as:

$$\sum_{j \epsilon J} \sum_{i \epsilon I} \sum_{k \epsilon K} \sum_{r=1}^{R} p_{jik} q^{r-1} \left(1-q\right) D_{ik} x_{jir} \tag{11}$$

The mathematical model can be expressed as follows:

$$\begin{aligned} \max & \sum_{j \epsilon J} \sum_{i \epsilon I} \sum_{k \epsilon K} \sum_{r=1}^{R} p_{jik} q^{r-1} \left(1-q\right) D_{ik} x_{jir} - \sum_{k \epsilon K} \sum_{j \epsilon J} C_{jk} Q_{jk} y_j - \pi \sum_{i \epsilon I} \sum_{k \epsilon K} q^R D_{ik} \\ & - \sum_{j \epsilon J} \sum_{k \epsilon K} \frac{Q_{jk} h_{jk}}{2} - \sum_{j \epsilon J} \sum_{k \epsilon K} Q_{jk} A_{jk} - \sum_{j \epsilon J} f_j y_j - \sum_{j \epsilon J} \sum_{i \epsilon I} \sum_{k \epsilon K} \sum_{r=1}^{R} c_k q^{r-1} \left(1-q\right) D_{ik} x_{jir} d_{ji} \\ & - \sum_{j \epsilon J} \sum_{k \epsilon K} c_k d_{pj} Q_{jk} - \sum_{j \epsilon J} \sum_{i \epsilon I} \sum_{k \epsilon K} \sum_{r=1}^{R} s_k q^{r-1} \left(1-q\right) D_{ik} x_{jir} \theta_{ji} \end{aligned} \tag{12}$$

s.t.

$$\sum_{j \epsilon J} x_{jir} = 1 \qquad \forall i \epsilon I,\ r \epsilon \left\{1, 2, \ldots, R\right\} \tag{13}$$

$$\sum_{r=1}^{R} x_{jir} \le y_j \qquad \forall i\epsilon I, j\epsilon J \tag{14}$$

$$x_{jir}\left(\frac{d_{ji}}{v} - T_k\right) \le 0 \qquad \forall i, j, k, r \tag{15}$$

$$\sum_{k\epsilon K} V_k Q_{jk} \le w_j y_j \qquad \forall j, k, r \tag{16}$$

$$Q_{jk} \le \sqrt{2A_{jk} D_{jk} / h_{jk}} \qquad \forall j, k, r \tag{17}$$

$$\sum_{j\epsilon J}\sum_{i\epsilon I}\sum_{k\epsilon K}\sum_{r=1}^{R} q^{r-1}(1-q) D_{ik} x_{jir} + \sum_{i\epsilon I}\sum_{k\epsilon K} q^R D_{ik} = \sum_{j\epsilon J}\sum_{k\epsilon K} Q_{jk} \qquad \forall i, j, k, r \tag{18}$$

$$x_{jir}, y_j \epsilon \{0,1\}, \quad \forall i, j, k, r$$
$$Q_{jk} \ge 0 \qquad , \qquad \forall j, k \tag{19}$$

The objective (12) maximizes the warehouses' profit (i.e., total revenue minus total cost, which includes the purchasing costs, lost sale cost, total holding and ordering cost, fixed costs, transportation cost, and loss costs of perishable products). Constraint (13) ensures that retailer i is only allocated to a warehouse at each allocation level. Constraint (14) guarantees that a retailer can only recourse to an established warehouse and retailer does not refer to a visited warehouse at each level. Constraint (15) states that if the time of delivering perishable products from a warehouse to retailers exceeds the critical time, it will lead to discarding them. Constraint (16) requires that the capacity of each warehouse should be respected. Constraint (17) states that according to warehouse capacity constraint, the ordering quantity of perishable product may be lower than the optimal ordering quantity. Constraint (18) controls perishable products to balance on warehouses. Constraint (19) defines variables.

Solution Approach

The presented LIP model considers two kinds of problems: inventory control and location-allocation problem with capacity constraint. The capacitated department location-allocation problem is an expansion of the incapacitated location problem. In addition, there is not an effective approach that ensures an optimal solution (Punyim et al., 2018). Amiri-Aref et al. (2018) developed an estimated solution method, according to Sample Average Approximation (SAA) method, to solve the location inventory problem that has equal NP-hardness trait like a simple location problem. Consequently, our offered MINLP model is an NP-Hard problem. On the other hand, due to the complexity of the proposed nonlinear model, such as Kuhnle and Lanza (2019), it is difficult to linearize the model. In addition, linearizing the model may make the problem more complicated. In this regard, Taleizadeh et al. (2011) and Farahani et al. (2015) proved that meta-heuristic algorithms have high efficiency to deal with nonlinear problems. Therefore, the Grasshopper Optimization Algorithm (GOA) is used to solve this model.

Genetic Algorithm

The genetic algorithm is an evolutionary algorithm. Studies in the literature have shown the high performance of GA in solving combinatorial optimization problems (Hiassat et al., 2017). The steps of Genetic algorithm based on Vahdani et al. (2017) are as follows.

1. Initial Population

The first step of the GA is the creation of solutions as chromosomes. This means generating the initial population for all decision variables (x_{jir} ,y_j ,Q_{jk}). The initialization process aims to generate the initial solutions for each decision variable randomly (for example, the optimal order quantity in the range [$q_{min,}$ q_{max}] is generated in random order). Generating random chromosomes may cause some restrictions and unfeasible solutions. To face these conditions, a penalty for the objective function is defined. Figure 2 shows a sample of the generated chromosome for Q_{jk} when $j = 5$ and $k = 1$.

8.33	9.15	2.14	9.22	6.69

Figure 2. Initial Chromosome for $\boldsymbol{Q_{jk}}$.

2. Crossover Operator

The uniform crossover operator is implemented in this study to create a new generation for each iteration of the algorithm. In this step, two parent chromosomes are considered and using the roulette wheel, two random numbers in the range of [1, length of the first chromosome] and [1, length of the second chromosome] are selected. Then the selected columns are replaced with each other. After that, a random number ($\alpha \in [0,1]$) is generated to explore the solution space. Based on Eq. 20 -21, a new offspring is obtained for transmission to the next generation. Figure 3 shows a sample of crossover operator when $\alpha = 0.6$.

$$Offspring1 = \alpha \times Parent1 + (1-\alpha) \times Parent2 \quad (20)$$

$$Offspring2 = \alpha \times Parent2 + (1-\alpha) \times Parent1 \quad (21)$$

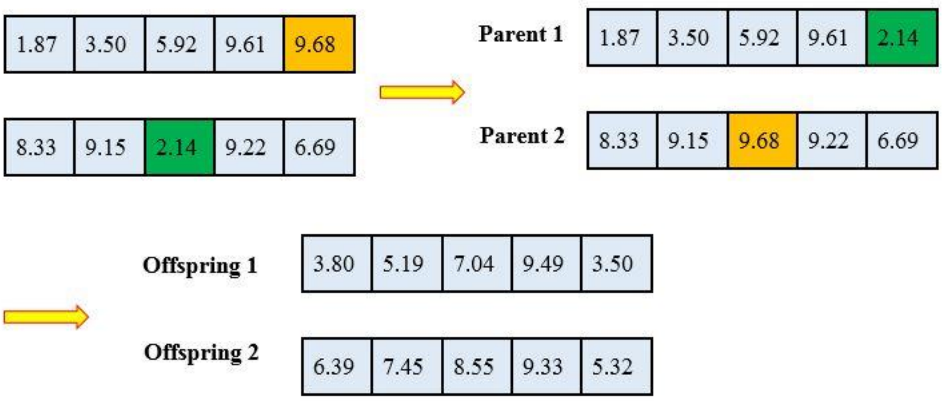

Figure 3. Crossover Operator

Figure 4. Swap Operator

3. Mutation Operator

In this step, the swap operator is utilized. To this end, a chromosome is randomly chosen from the population, and then the positions of the two randomly chosen genes are switched with each other. Figure 4 demonstrates a sample of mutation operator.

Grasshopper Optimization Algorithm

Grasshopper Optimization Algorithm (GOA) is a metaheuristic algorithm inspired by the swarming conduct of grasshoppers that is performed to find the optimum solution of models. Further, GOA analyzes the repulsion and attraction powers between the grasshoppers. Saremi et al. (2017) introduced the GOA algorithm with mathematically models that relate to the Swarm Intelligence methods. They proved the ability of GOA to solve real problems with unknown search spaces. Moreover, the results of other research in the literature explain the power of this algorithm in enhancing the quality of creating population, moving between exploration and exploitation, the top cover of search space, and the high-speed of convergence curve during repeat path (Saremi et al., 2020). The low number of control parameters, the flexible and exceptional exploratory search model, and a gradient-free mechanism are the basic advantages of this algorithm. For problems with constraint, Neve et al. (2017) used a penalty function to punish the grasshoppers that violate any of the limitations. Ahanch et al. (2017) applied GOA to solve the reconfiguration problem of a distribution system for minimizing active power loss.

The mathematical formulations suggested for this algorithm are given as follows (Saremi et al., 2017).

The flying route of a grasshopper is influenced by several elements, namely social interaction(S_i), gravity(G_i), and wind advection(A_i). The position of the $i-th$ grasshopper in the $i-th$ iteration ($X_i(t)$) is indicated as given in Eq. 22.

$$X_i(t+1)=G_i(t)+S_i(t)+A_i(t) \quad t=1,2,\dots,tMax \tag{22}$$

GOA simulates social interaction (S_i) to solve problems, which is defined as follows:

$$S_i=\sum_{j=1,j\neq i}^{nPop} s(d_{ij})\hat{d}_{ij} \tag{23}$$

$$d_{ij}=|x_i-x_j| \tag{24}$$

where d_{ij} is Euclidean distance between *i-th* and *j-th* grasshoppers. And $\hat{d}_{ij}=\frac{x_j-x_i}{d_{ij}}$ is a unit vector from *i-th* to *j-th* grasshoppers. The power of social force is shown by s, which defines the movement direction of a grasshopper in the swarm and is determined as follows:

$$s(r)=fe^{-\frac{r}{l}}-e^{-r} \tag{25}$$

where f and l indicate the intensity of attraction and the attractive length measure, respectively. When grasshoppers are interacting, the formulated model of search is defined as follows in Eq. 26.

$$X_i^d(t+1)=c\left\{\sum_{j=1,j\neq i}^{nPop} c\frac{ub_d-lb_d}{2}\, s\left(\left|x_i(t)-x_j(t)\right|\right)\frac{x_j-x_i}{d_{ij}}\right\}+\hat{T}_d(t) \tag{26}$$

where ub_d denotes the upper bound of $d-th$ dimension, lb_d indicates the lower bound of $d-th$ dimension, $\hat{T}_d$ is the best solution that has been obtained so far, and c is a subtractive ratio to decrease the comfort zone, refusal zone, and attraction zone.

In Eq. 26, the internal c contributes to the decrease in the degree of repulsion/attraction forces between grasshoppers, while the external c reduces the search coverage around the target. In the GOA, grasshoppers are tracking the target and if a better solution is found, the best solution and c are updated using Eq. 27.

$$c = c_{max} - t\frac{c_{max} - c_{min}}{tMax} \qquad (27)$$

where $tMax$ indicates the maximum number of iteration, t shows the current iteration, c_{max} = 1, and c_{min} = 0.00001. The pseudo-code of GOA algorithm is shown in Figure 5. Generally, the steps of the grasshopper algorithm are as follows.

Initial population: The first step of the GOA is to create a set of random initial solutions as grasshoppers. This means generating the initial population for all decision variables (x_{jir}, y_j, Q_{jk}) randomly. In addition, a penalty function is used to punish the grasshoppers that violate any of the limitations.

Evaluation of grasshoppers group: In this step, the fitness for each grasshopper is calculated and determined by the best search agent (T).

Updating the position of grasshoppers: In each iteration, first, the parameter c is updated using Eq. 27. Then the distances between grasshoppers are normalized in the range [1, 4]. Finally, the position of each search agent is updated by Eq.26. Moreover, the best search agent (T) is defined.

Updating the position of grasshoppers continues until the maximum iteration is reached; it is then stopped.

```
Initialize the swarm X_i (i = 1,2, ..., nPop)
Initialize c_max, c_min, and maximum number of interation
Calculate the fitness of each search agent
T = the best search agent
while (t < Max number of interaction)
    Update c using Eq. 27.
    for each search agent
        Normalize the distances between grasshopper in [1,4]

Update the position of the current search agent by the Eq. 26.

Bring the current search agent back if it goes outside the boundaries
    end for
    Update T if there is a better solution
    t = t + 1
end while
Return T
```

Figure 5. Pseudo Codes of Grasshopper Optimization Algorithm

In addition, the numerical results of the GOA, GA, and GAMES are compared in the next section to justify the performance of this offered algorithm.

Numerical Results

In this part, information about the numerical instances and the parameters of GA and GOA are determined. Then, instances with different dimensions and parameters are solved using GAMS, GA, and GOA. Finally, the impact of changing the major parameters of the model is discussed in the sensitivity analysis.

Parameters tuning

The outcomes of metaheuristic algorithms are deeply sensitive to the parameters of each algorithm. To attain better solutions, the parameters of the algorithms need calibrations. We use the Taguchi method for tuning the parameters. The Taguchi method is an effective parameter tuning method. To tune the parameters of GA, we consider four parameters, including Mutation Rate, Max-Iteration, Population, and Crossover Rate. For the GOA algorithm, these parameters are Max-Iteration, Population, C_{max}, and C_{min}. The Taguchi design selects the best value of the objective function. Minitab software is used to implement the numerical experiments. Table 2 and Figure 6 - 7 present the experimental results of parameters for GA and GOA.

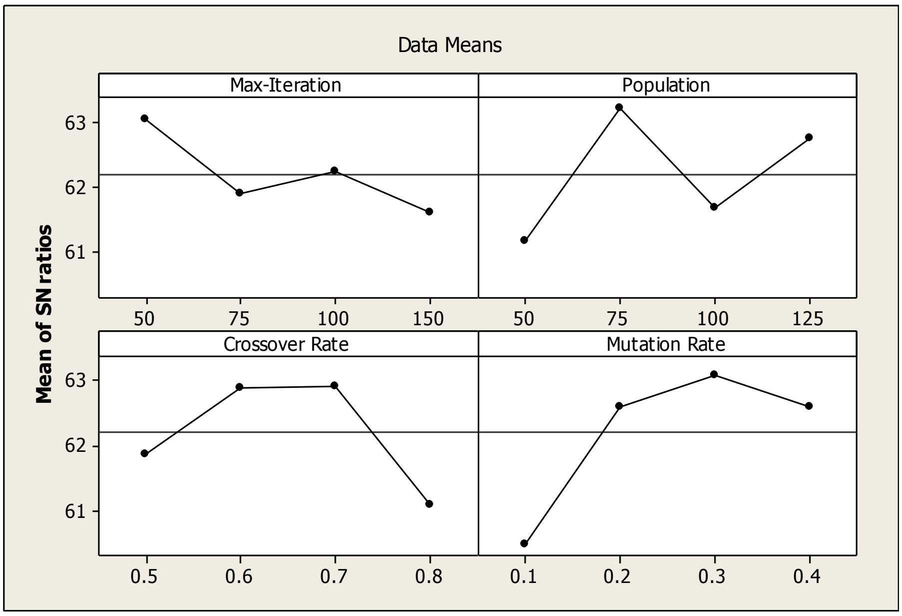

Figure 6. S/N Ratio Plot for GA Parameters

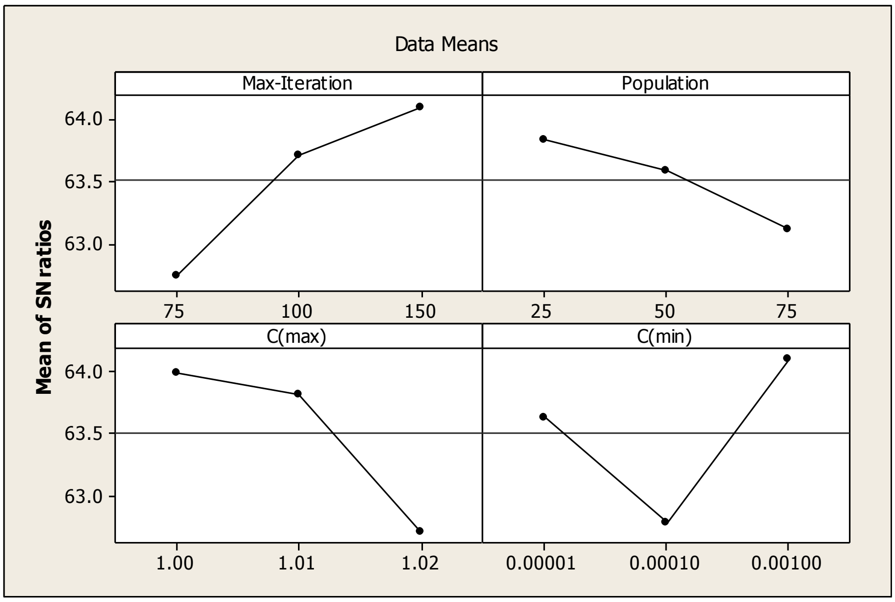

Figure 7. S/N Ratio Plot for GOA Parameters

Table 2. Tuned Parameters for GA and GOA

Algorithm	Parameters					
	Mutation Rate	Crossover Rate	Max-Iteration	Population	C_{max}	C_{min}
GA	0.3	0.7	50	75	-	-
GOA	-	-	150	25	1	0.001

Comparative Study

To illustrate the performance of the introduced algorithm, several different experiments and the Genetic Algorithm (GA) were used. The proposed mathematical model was coded in GAMS software. Besides, all metaheuristic algorithms were coded in MATLAB R2015b. Two kinds of test problems were considered, which consisted of small size and large size problems. Each test problem was illustrated by four figures, indicating the number of products (K), number of potential warehouses (J), number of retailers (I), and number of levels(R), respectively. The data used for analyzing the model and algorithms' performances were determined based on Dai et al. (2018). Moreover, Due to the scarcity of benchmark instances in the literature, expert opinions were used to define the limited area of all parameters (see Table 3). Then all test problems were generated randomly by MATLAB software. In addition, GAMS was used to solve small problems to ensure model performance. The results of the experiments are displayed in Table 4. Moreover, Eq. (28) is utilized to calculate the gap.

$$Gap = \frac{optimal\ solution - best\ found\ solution}{optimal\ solution} \times 100 \quad (28)$$

In Table 4, the gap between the objective function of each metaheuristic algorithm and exact method for small size instances are presented. Moreover, the optimality gap for the large-sized instances represents the gap between the objective function of GOA and GA. The accomplished results of the presented algorithm for distinctive tests illustrate that GOA has an effective execution versus GA in terms of the objective function, but GA has the highest performance in computation time. Figure 9 displays the distinction between the objective work esteem of GA, GOA, and GAMS. Figure 8 shows the difference between the computational time of GA and GOA.

Table 3. Source of Randomly Generated Parameters

Parameter	Value	Parameter	Value
q	$\sim U(0,0.6)$	T_k	$\sim U(100,150)$
D_{ik}	$\sim U(1,10)$	θ_k	$\sim U(0,0.5)$
f_j	$\sim U(500,1000)$	w_j	$\sim U(100,500)$
s_k	$\sim U(1,5)$	V_k	$\sim U(10,25)$
d_{pj}	$\sim U(15,100)$	v	$\sim U(4,8)$
d_{ji}	$\sim U(10,100)$	π	$\sim U(15,100)$
c_k	$\sim U(1,5)$	b_{jr}	$\sim U(0,3)$
A_{jk}	$\sim U(20,70)$	C_{jk}	$\sim U(200,500)$
h_{jk}	$\sim U(4,10)$		

Table 4. Obtained Results of GAMS, GA and GOA Algorithms

Problem instance	Dimension $K/R/J/I$	GAMS	GA			GOA		
		OFV	OFV	Time(s)	Gap	OFV	Time(s)	Gap
1	2 / 2 / 2 / 2	1813.496	1792.77	11.46	1.14%	1802.73	36.79	0.5%
2	2 / 2 / 3 / 3	11859.879	11395.3	11.96	3.91%	11558.47	36.95	2.54%
3	2 / 2 / 3 / 4	19103.957	17803.75	13.61	6.8%	18555.02	37.76	2.87%
4	3 / 1 / 4 / 5	24286.807	22491.65	15.07	7.39%	23236.63	40.23	4.32%
5	2 / 2 / 4 / 6	36574.965	34074.96	15.61	6.83%	35412.56	40.68	3.17%
6	2 / 3 / 4 / 6	25814.36	23698.82	18.68	8.19%	24118.54	42.7	6.56%
7	2 / 3 / 7 / 7	18173.446	16488.78	23.97	9.26%	17400.11	47.64	4.25%
8	3 / 2 / 8 / 9	36693.951	33408.91	24.6	8.95%	34871.18	48.25	4.96%
9	3 / 2 / 9 / 12	-	50770.92	32.27	-	57331.64	55.32	-
10	3 / 3 / 9 / 12	-	34548.04	39.92	-	40052.93	62.32	-
11	4 / 3 / 11 / 13	-	41465.5	53.42	-	49999.31	75.13	-
12	4 / 3 / 13 / 20	-	60294.08	76.8	-	69640.34	98.34	-
13	4 / 3 / 20 / 20	-	66355.06	108.12	-	77340.64	127.72	-
14	5 / 3 / 25 / 23	-	270335.07	158.99	-	322647.88	176.19	-
15	5 / 3 / 30 / 35	-	578056.41	254.28	-	680670.34	265.98	-
16	5 / 4 / 32 / 32	-	671004.61	320.48	-	812140.75	462.24	-
17	6 / 4 / 35 / 38	-	674711.94	478.7	-	814595.15	618.13	-
18	6 / 4 / 40 / 40	-	1047494.11	598.87	-	1331497.06	640.89	-
19	6 / 4 / 42 / 44	-	909824.06	642.47	-	1153358.49	751.51	-
20	7 / 5 / 45/ 50	-	1263647.13	1008.82	-	1668847.51	1282.95	-

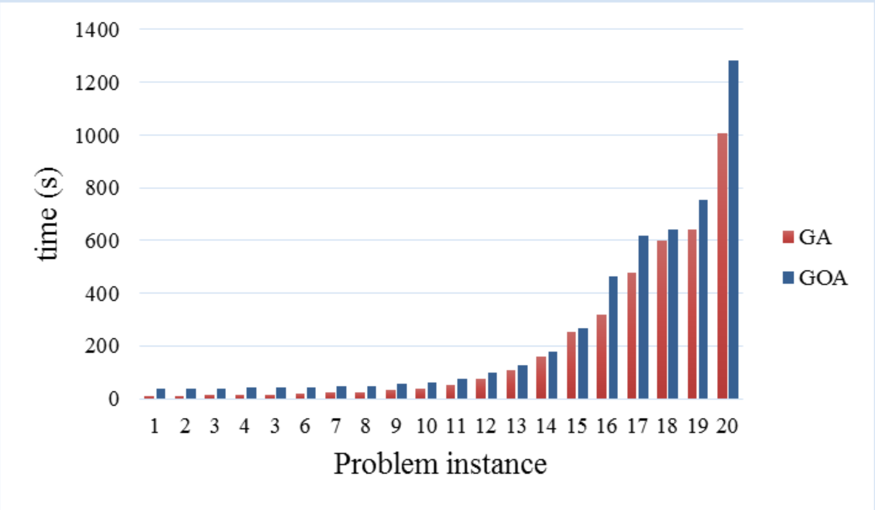

Figure 8. Comparison of GOA and GA in Terms of Computational Time

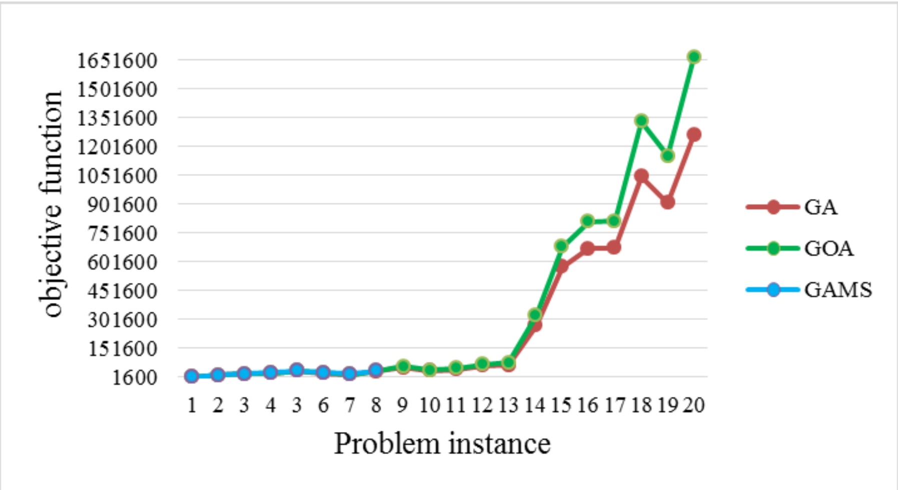

Figure 9. Comparison of GOA, GA and GAMS in Terms of the Objective Function

Sensitivity Analysis

In this section, sensitivity analysis was carried out to examine the influences of parameters on the objective function. Several sensitivity analyses were conducted based on the different parameter values to validate the proposed model for medium-size problems. The distribution functions of randomly generated test problems are summarized in Table 3. In addition, the GAMS software was used to conduct the sensitivity analysis.

One of important investigations was to analyze the influence of R-level sequence of warehouses on the profit of warehouses. To do so, we created a problem according to parameters in Table 3. This instance was to run with various amounts of q and R. The results of Figure 10 illustrate that the profit decreases with the increase in q, due to the extra lost sale cost undertaken by warehouses. Consequently, the profit when $R > 1$ is significantly higher than those with $R = 1$, indicating the the important advantage of rendering backup services.

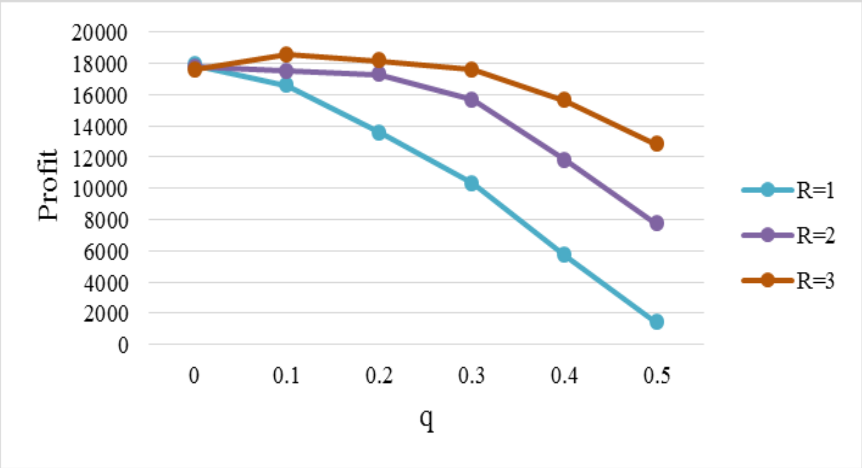

Figure 10. The Effect of q & R on the Profit

Figure 11 displays that the optimized number of open warehouses doesn't change with increasing q if R is small. It also shows that if R is large, the optimum number of open warehouses increases with growing q. This demonstrates that in a large failure probability, extra warehouses can afford good performance for quality services against disturbances. Consequently, when a retailer is re-allocated to the backup warehouse, the cost of an additional warehouse can be better than additional infrastructure investment, hence making extra warehouse is favored.

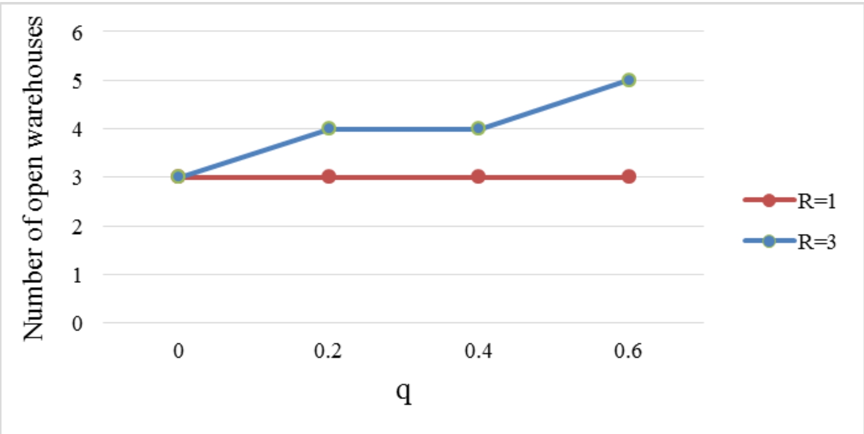

Figure 11. The Effect of q on the Number of Open Warehouses

Figure 12 shows that increasing the holding cost usually decreases the value of objective function. Figure 13 shows the relationship between the amount of profit and annual fixed cost of the warehouse in which increasing the fixed cost leads to the reduction of profits.

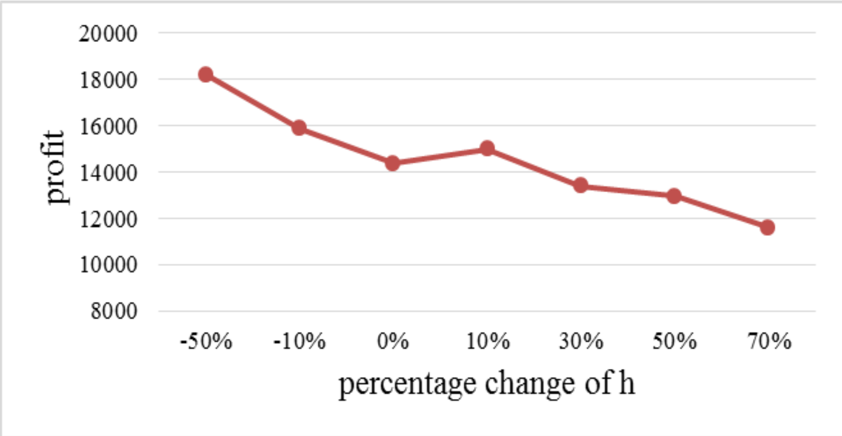

Figure 12. The Effect of Changing the Holding Cost on the Profit

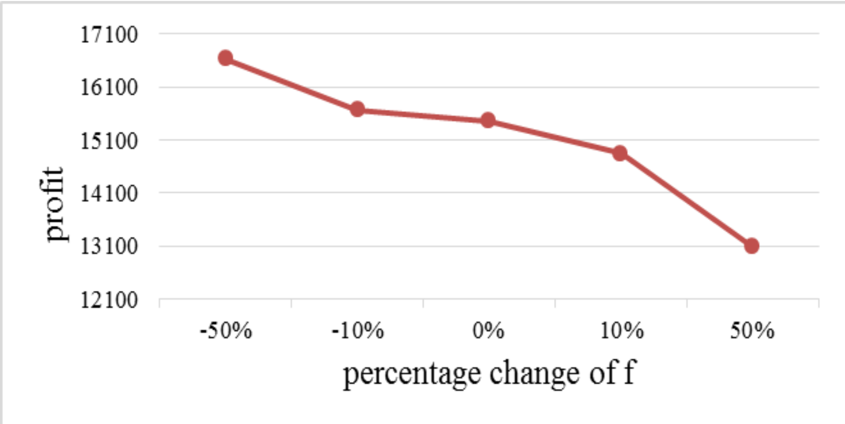

Figure 13. The Effect of Changing the Annual Fixed Cost of the Warehouse on Profit

One of the other important investigations is analyzing the impact of markup percentage (wholesale price) on the objective function. Figure 14 illustrates the impacts of changing markup for different levels (R>1) on the objective function, where a normal condition is compared to a condition with 30% disruption probability. It is obvious that retailers' demand is provided immediately on the first level when q=0. Moreover, changing b in R>1 has no influence on the amount of profit. It is also clear that with increasing the probability of failure (q), profit is sharply reduced. The best strategy to deal with probability disruption is increasing markup, which is obtained from Figure 14. Results show that implementing this strategy leads to offsetting the loss of profit and supporting other levels, simultaneously. Finally, an efficient supply chain that is resilience to disruptions is obtained.

With the increase in the deterioration rate (θ_k) and, consequently, the loss of more products, the loss costs of perishable products increases, which leads to a reduction in total profits. As shown in Figures 15 and 16, with increasing the deterioration rate, the loss costs of perishable products decreases and the total profit of warehouses increases.

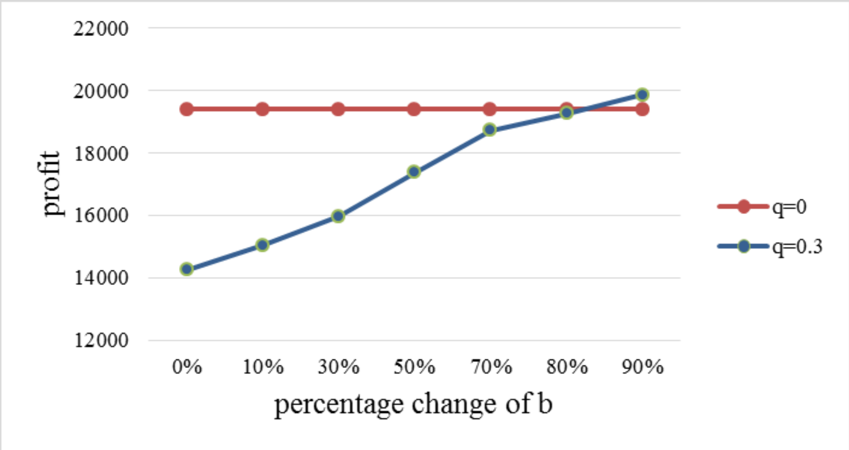

Figure 14. The Effect of Changing the Markup Percentage on profit

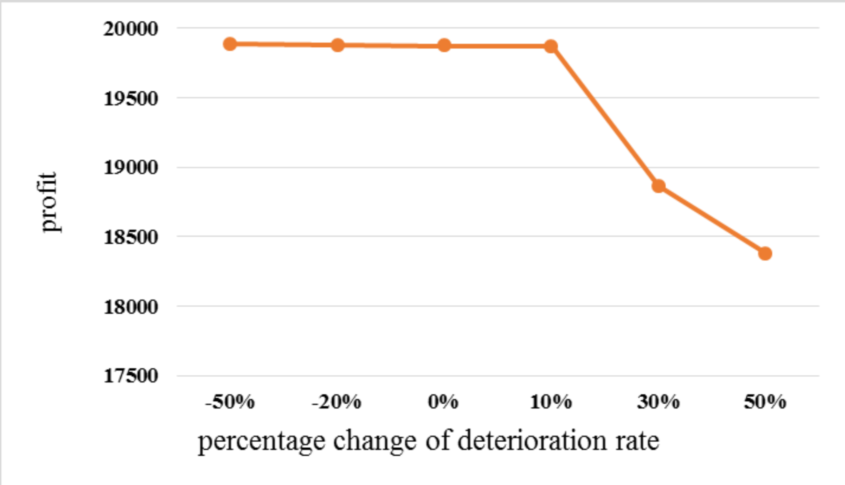

Figure 15. The Impact of Deterioration Rate on Profit

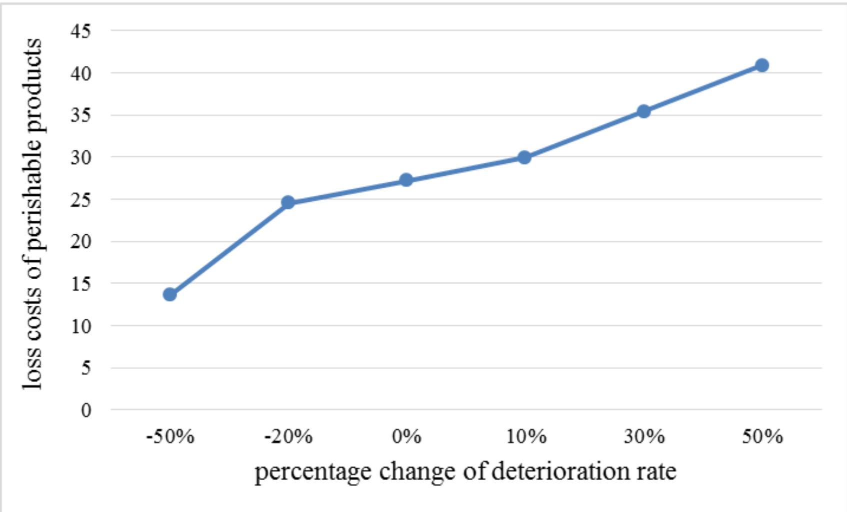

Figure 16. The Impact of Deterioration Rate on LC

Conclusion

This study proposed a three-echelon location-inventory supply chain problem with pricing decision for multiple independent perishable products, which investigated markup and the probability of a disruption in the warehouse. To analyze the problem, a mixed integer non-linear programming (MINLP) model was presented which aimed to maximize the warehouses' profit.

Our model focused on several decisions, including inventory control decisions, multi-echelon location-allocation problem, and pricing decision. Finally, it determined the number and locations of warehouses, the assignment of retailers to warehouses, the optimal order quantity in any established warehouse, and the retail price of each product at each warehouse and each level. The model was solved by two metaheuristic algorithms, including a Genetic Algorithm (GA) and Grasshopper Optimization Algorithm (GOA). To compare the results of algorithms, some test problems were designed in different sizes. Numerical results indicated that GOA approach has a better performance in terms of objective function, but the GA was able to solve the problem in a shorter span of time. There are some noticeable directions for future studies such as considering price-sensitive demands in problem, formulating the problem in fuzzy conditions, and integrating this problem with other pricing strategies.

References

Asasi, M. S., Ahanch, M., & Amiri, M. S. (2017, December). A Grasshopper Optimization Algorithm to solve Optimal Distrbution System Reconfiguration and Distributed Generation Placement Problem. In *2017S IEEE 4th International Conference on Knowledge-Based Engineering and Innovation (KBEI) Dec* (Vol. 22). Tehran, Iran, pp. 0659-0666…https://doi.org/10.1109/KBEI.2017.8324880

Ahmadi-Javid, A., & Hoseinpour, P. (2015a). A location-inventory-pricing model in a supply chain distribution network with price-sensitive demands and inventory-capacity constraints. *Transportation Research Part E: Logistics and Transportation Review*, *82*, 238-255. https://doi.org/10.1016/j.tre.2015.06.010

Ahmadi-Javid, A., & Hoseinpour, P. (2015b). Incorporating location, inventory and price decisions into a supply chain distribution network design problem. *Computers & Operations Research*, *56*, 110-119. https://doi.org/10.1016/j.cor.2014.07.014

Ahmadi-Javid, A., Amiri, E., & Meskar, M. (2018). A profit-maximization location-routing-pricing problem: A branch-and-price algorithm. *European Journal of Operational Research*, *271*(3), 866-881. https://doi.org/10.1016/j.ejor.2018.02.020

Ahmadzadeh, E., & Vahdani, B. (2017). A location-inventory-pricing model in a closed loop supply chain network with correlated demands and shortages under a periodic review system. *Computers & Chemical Engineering*, *101*, 148-166. https://doi.org/10.1016/j.compchemeng.2017.02.027

Amiri-Aref, M., Klibi, W., & Babai, M. Z. (2018). The multi-sourcing location inventory problem with stochastic demand. *European Journal of Operational Research*, *266*(1), 72-87. https://doi.org/10.1016/j.ejor.2017.09.003

Asl-Najafi, J., Zahiri, B., Bozorgi-Amiri, A., & Taheri-Moghaddam, A. (2015). A dynamic closed-loop location-inventory problem under disruption risk. *Computers & Industrial Engineering*, *90*, 414-428. https://doi.org/10.1016/j.cie.2015.10.012

Chen, Q., Li, X., & Ouyang, Y. (2011). Joint inventory-location problem under the risk of probabilistic facility disruptions. *Transportation Research Part B: Methodological*, *45*(7), 991-1003. https://doi.org/10.1016/j.trb.2011.04.004

Chen, X., & Hu, P. (2012). Joint pricing and inventory management with deterministic demand and costly price adjustment. *Operations Research Letters*, *40*(5), 385-389. https://doi.org/10.1016/j.orl.2012.05.011

Chen, X., Zhou, S. X., & Chen, Y. (2011). Integration of inventory and pricing decisions with costly price adjustments. *Operations Research*, *59*(5), 1144-1158. https://doi.org/10.1287/opre.1110.0946

Dai, Z., Aqlan, F., Zheng, X., & Gao, K. (2018). A location-inventory supply chain network model using two heuristic algorithms for perishable products with fuzzy constraints. *Computers & Industrial Engineering*, *119*, 338-352. https://doi.org/10.1016/j.cie.2018.04.007

Dehghani, E., Pishvaee, M. S., & Jabalameli, M. S. (2018). A hybrid Markov process-mathematical programming approach for joint location-inventory problem under supply disruptions. *RAIRO-Operations Research*, *52*(4), 1147-1173. https://doi.org/10.1051/ro/2018012

Etebari, F., & Dabiri, N. (2016). A hybrid heuristic for the inventory routing problem under dynamic regional pricing. *Computers & Chemical Engineering*, *95*, 231-239. https://doi.org/10.1016/j.compchemeng.2016.09.018

Fahimi, K., Seyedhosseini, S. M., & Makui, A. (2018). Dynamic competitive supply chain network design with price dependent demand and Huff utility function. *Iranian Journal of Management Studies*, *11*(2), 271-305. https://doi.org/10.22059/IJMS.2018.241299.672813

Farahani, M., Shavandi, H., & Rahmani, D. (2017). A location-inventory model considering a strategy to mitigate disruption risk in supply chain by substitutable products. *Computers & Industrial Engineering*, *108*, 213-224. https://doi.org/doi.org/10.1016/j.cie.2017.04.032

Farahani, R. Z., Rashidi Bajgan, H., Fahimnia, B., & Kaviani, M. (2015). Location-inventory problem in supply chains: A modelling review. *International Journal of Production Research*, *53*(12), 3769-3788. https://doi.org/10.1080/00207543.2014.988889

Ghasemy Yaghin, R., Fatemi Ghomi, S. M. T., & Torabi, S. A. (2017). Incorporating return on inventory investment into joint lot-sizing and price discriminating decisions: A fuzzy chance constraint programming model. *Iranian Journal of Management Studies*, *10*(4), 929-959. https://doi.org/10.22059/IJMS.2017.230829.672615

Guerrero, W. J., Prodhon, C., Velasco, N., & Amaya, C. A. (2015). A relax-and-price heuristic for the inventory-location-routing problem. *International Transactions in Operational Research*, *22*(1), 129-148. https://doi.org/10.1111/itor.12091

Gzara, F., Nematollahi, E., & Dasci, A. (2014). Linear location-inventory models for service parts logistics network design. *Computers & Industrial Engineering*, *69*, 53-63. https://doi.org/10.1016/j.cie.2013.12.014

Hamdan, B., & Diabat, A. (2019). A two-stage multi-echelon stochastic blood supply chain problem. *Computers & Operations Research*, *101*, 130-143. https://doi.org/10.1016/j.cor.2018.09.001

Hiassat, A., Diabat, A., & Rahwan, I. (2017). A genetic algorithm approach for location-inventory-routing problem with perishable products. *Journal of Manufacturing Systems*, *42*, 93-103. https://doi.org/10.1016/j.jmsy.2016.10.004

Kaya, O., & Urek, B. (2016). A mixed integer nonlinear programming model and heuristic solutions for location, inventory and pricing decisions in a closed loop supply chain. *Computers & Operations Research*, *65*, 93-103. https://doi.org/10.1016/j.cor.2015.07.005

Kuhnle, A., & Lanza, G. (2019). Investigation of closed-loop supply chains with product refurbishment as integrated location-inventory problem. *Production Engineering*, *13*(3-4), 293-303. https://doi.org/10.1007/s11740-019-00885-4

Li, Z., & Hai, J. (2019). A capacitated location-inventory model with demand selection. *Journal of Advanced Transportation*, *2019*. https://doi.org/10.1155/2019/2143042

Nemati, Y., Madhoushi, M., & Safaei Ghadikolaei, A. (2017). Towards supply chain planning integration: Uncertainty analysis using fuzzy mathematical programming approach in a plastic forming company. *Iranian Journal of Management Studies*, *10*(2), 335-364. https://doi.org/10.22059/IJMS.2017.218842.672334

Neve, A. G., Kakandikar, G. M., & Kulkarni, O. (2017). Application of grasshopper optimization algorithm for constrained and unconstrained test functions. *International Journal of Swarm Intelligence and Evolutionary Computation*, *6*(3), 1-7. https://doi.org/10.4172/2090-4908.1000165

Orand, S. M., Mirzazadeh, A., Ahmadzadeh, F., & Talebloo, F. (2015). Optimization of the inflationary inventory control model under stochastic conditions with Simpson approximation: Particle swarm optimization approach. *Iranian Journal of Management Studies*, *8*(2), 203-220. https://doi.org/10.22059/IJMS.2015.52631

Puga, M. S., & Tancrez, J. S. (2017). A heuristic algorithm for solving large location–inventory problems with demand uncertainty. *European Journal of Operational Research*, *259*(2), 413-423. https://doi.org/10.1016/j.ejor.2016.10.037

Punyim, P., Karoonsoontawong, A., Unnikrishnan, A., & Xie, C. (2018). Tabu search heuristic for joint location-inventory problem with stochastic inventory capacity and practicality constraints. *Networks and Spatial Economics*, *18*(1), 51-84. https://doi.org/10.1007/s11067-017-9357-y

Rafie-Majd, Z., Pasandideh, S. H. R., & Naderi, B. (2018). Modelling and solving the integrated inventory-location-routing problem in a multi-period and multi-perishable product supply chain with uncertainty: Lagrangian relaxation algorithm. *Computers & Chemical Engineering*, *109*, 9-22. https://doi.org/10.1016/j.compchemeng.2017.10.013

Saremi, S., Mirjalili, S., & Lewis, A. (2017). Grasshopper optimisation algorithm: Theory and application. *Advances in Engineering Software*, *105*, 30-47. https://doi.org/10.1016/j.advengsoft.2017.01.004

Saremi S., Mirjalili S., Mirjalili S., & Song Dong J. (2020). Grasshopper Optimization Algorithm: Theory, literature review, and application in hand posture estimation. In: Mirjalili S., Song Dong J., & Lewis A. (Eds.), *Nature-inspired optimizers: Studies in computational intelligence*, vol. 811 (pp. 107-122). Springer, Cham. https://doi.org/10.1007/978-3-030-12127-3_7

Smith, S. A., & Agrawal, N. (2017). Optimal markdown pricing and inventory allocation for retail chains with inventory dependent demand. *Manufacturing & Service Operations Management*, *19*(2), 290-304. https://doi.org/10.1287/msom.2016.0609

Taleizadeh, A. A., Niaki, S. T. A., & Barzinpour, F. (2011). Multiple-buyer multiple-vendor multi-product multi-constraint supply chain problem with stochastic demand and variable lead-time: A harmony search algorithm. *Applied Mathematics and Computation*, *217*(22), 9234-9253. https://doi.org/10.1016/j.amc.2011.04.001

Tavakkoli-Moghaddam, R., Yadegari, M., & Ahmadi, G. (2018). Closed-loop supply chain inventory-location problem with spare parts in a multi-modal repair condition. *International Journal of Engineering*, *31*(2), 346-356. https://doi.org/10.5829/ije.2018.31.02b.20

Vahdani, B., Soltani, M., Yazdani, M., & Mousavi, S. M. (2017). A three level joint location-inventory problem with correlated demand, shortages and periodic review system: Robust meta-heuristics. *Computers & Industrial Engineering*, *109*, 113-129. https://doi.org/10.1016/j.cie.2017.04.041

Zhang, Y., Qi, M., Lin, W. H., & Miao, L. (2015). A metaheuristic approach to the reliable location routing problem under disruptions. *Transportation Research Part E: Logistics and Transportation Review*, *83*, 90-110. https://doi.org/10.1016/j.tre.2015.09.001

Zhang, Y., Snyder, L. V., Qi, M., & Miao, L. (2016). A heterogeneous reliable location model with risk pooling under supply disruptions. *Transportation Research Part B: Methodological*, *83*, 151-178. https://doi.org/10.1016/j.trb.2015.11.009