Journal of Industrial and Systems Engineering
Vol. 12, No. 1, pp. 167-184
Winter (January) 2019

Addressing a fixed charge transportation problem with multi-route and different capacities by novel hybrid meta-heuristics

Mohammad Bagher Fakhrzad[1*], Fariba Goodarzian[1], A. M. Golmohammadi[1]

School of Industrial Engineering, Yazd University, Yazd, Iran

mfakhrzad@yazd.ac.ir, faribagoodarzian7070@gmail.com, golmohammadi@gmail.com

Abstract

In most real world application and problems, a homogeneous product is carried from an origin to a destination by using different transportation modes (e.g., road, air, rail and water). This paper investigates a fixed charge transportation problem (FCTP), in which there are different routes with different capacities between suppliers and customers. To solve such a NP-hard problem, four meta-heuristic algorithms include Red Deer Algorithm (RDA), Stochastic Fractal Search (SFS), Genetic Algorithm (GA), and Simulated Annealing (SA) and two new hybrid meta-heuristics include hybrid RDA & GA (HRDGA) algorithm and Hybrid SFS & SA (HSFSA) algorithm are utilized. Regarding the literature, this is the first attempt to employ such optimizers to solve a FCTP. To tune up their parameters of algorithms, various problem sizes are generated at random and then a robust calibration is applied by using the Taguchi method. The final output shows that Simulated Annealing (SA) algorithm is the better than other algorithms for small-scale, medium-scale, and large-scale problems. As such, based on the Gap value of algorithms, the results of LINGO software shows that it reveals better outputs in comparison with meta-heuristic algorithms in small-scale and simulated annealing algorithm is better than other algorithms in large-scale and medium-scale problems. Finally, a set of computational results and conclusions are presented and analyzed.

Keywords: Fixed-charge transportation problem, SA algorithm, GA algorithm, SFS algorithm, RDA algorithm, Taguchi method

1- Introduction

The world of business today is a world of uncertainty and the secret of the survival of companies with such conditions in their competitive power. In order for a company to effectively compete with other companies, it needs to supply chain management. In other words, the task of supply chain management can be summarized as follows: "Maximizing value added and minimizing total costs during business processes by focusing on speed and response to market requirements" (Chopra, 2010). Today, the supply chain management is a requirement, especially for the manufacturing industry, whose products are expected to be marketed at a competitive price and higher quality than their rivals.

It must be admitted that today, trade has quickly changed and more competitive than ever. A business firm today not only needs to operate at a lower cost to compete, but must also boost its competitive advantage to be featured in the market and among its rivals. Therefore, an important way for companies to differentiate themselves from others, as well as raise their profits, is in a highly competitive environment using service management, activities and interactions that result from the sale and purchase of a product (Chopra , 2010) and (Shoushtary et al. 2014).

*Corresponding author
ISSN: 1735-8272,

Transportation problem (TP) is a well-known and applied fundamental issue in the operation researches, which involves the transfer of goods from several origins or producers to several destinations or consumers. This is one of the most important linear programming applications that in logistics and supply chain management, display a major role in in reducing costs and improving service levels. A basic assumption on the transportation problems are related to the cost with the number of units transported, while in many real world problems, especially the distribution, with the exception of variable costs, the fixed cost for the development of the facility, as well as customer demands satisfaction is considered (Chopra, 2007).

FCTP is an extension of the transportation general problem, in which a number of products are shipped to demands satisfaction, while fixed costs apply in addition to variable costs. In practice, many problems of distribution and transportation can be formulated as FCTP. Also, the fixed cost problem is used in many problems of scheduling, location, vehicle allocation, solid waste management, process selection, and so on. In practice, this problem has been widely used in industrial, and has expanded widely in theory (Fakhrzad et al. 2013b).

In addition, there are several metaheuristics in the literature which are employed to solve the problem. Golmohamadi et al. (2017) developed a fuzzy fixed charge solid transportation problem using batch transferring by new approaches in meta-heuristic. They assumed that the products are transferred in batches in a fixed charge transportation problem. Also, the fuzzy values are applied according to the parameters value. They used six meta-heuristics to solve the presented model. In addition, they used for adjusting parameters of the Taguchi method. In future work, they suggested New hybrid algorithms can be presented using Red Deer algorithm for this problem. Another suggestion is considering several different batches to transfer commodities with different capacities. Also, a kind of discount state can be considered beside the batching transportation. Sadeghi-Moghaddam et al. (2017) developed new approaches in metaheuristics to solve the FCTP in a fuzzy environment. They presented to solve such a non-deterministic polynomial-time hard problem. They considered both fixed costs and variable costs as the fuzzy number, and developed several algorithms that included a single point-based, two population-based meta-heuristics, and Whale Optimization Algorithm (WOA). In addition, they are presented new methods to solve algorithms using both spanning tree-based Prufer number and priority-based. Besides, Taguchi approach is applied to adjust the parameters. El Idrissi et al. (2017) developed new crossover operator for genetic algorithm to resolve the fixed charge transportation problem. They studied efficiency of these operators on the performance of the GAs by making a comparative study to the FCTP. The results show that chooses of adequate crossover is necessary and important to solve. Also, the genetic algorithm with their developed crossover operator is more efficient. Baidya et al. (2017) developed four new fuzzy fixed charge solid transportation problems (FFCSTP). There are two objective function included maximize the total profit and minimize the total cost. Then, they used genetic algorithm and particle swarm optimization to solve the optimal transportation schedule for FFCSTP. Also, they proposed FFCSTP model can be formulate and solve a multi-objective STP to minimize total transportation cost, total delivery time, total deterioration of commodity during transportation and so on. Midya et al. (2017) analyzed the interval programming using interval and Rough Interval (RI). Also, they considered FCTP with uncertainty in terms of interval and RI. Interval programing is one the tools to uncertainty parameters in mathematical programming. Then, they used fuzzy programming method to solve crisp equivalent bi-objective FCTP, and this method is provided to non-dominated solution. In future work, this paper may be extended to multi-objective FCTP in rough set environment. Mingozzi et al. (2017) developed an exact algorithm for the FCTP based on matching source and sink patterns. In addition, described a new integer programming formulation that involves two sets of variables representing flow patterns from sources to sinks and from sinks to sources. There are two types of patterns to provide a valid FCTP solution. Zhang et al. (2016) presented fixed charge solid transportation problem in uncertain environment and its algorithm. There are three mathematical models included expected value mode, chance-constrained programming model, and measure-chance programming model, which uncertain variables are including supplies, demands, conveyance capacities, direct costs and fixed charges. Also, they developed a hybrid intelligent algorithm based on the uncertainty theory and Tabu Search (TS) algorithm to solve the model. In addition, they consider this problem in other more complex environment, such as uncertain random environment, and so on in future work. Pop et al. (2016) developed a hybrid based genetic algorithm for solving a capacitated two-stage FCTP in supply chains. Also, they proposed a collection of benchmark instances approaches. They compared a novel hybrid heuristic method with the state-of-the-art algorithms for solving capacitated two-stage FCTP. Saxena et al. (2016) developed a compromise approach for solving fuzzy multi-objective FCTP. They considered a transportation activity takes place between a source and destination pair-which. There are multiple and conflicting objectives in this paper. In addition, they obtained an interactive solution procedure for solving multi-objective FCTP, and there are fuzzy parameters for objective functions. In future work, they intend to look into the complexity and performance of this algorithm in terms of the processing time. Pramanik et al. (2015) developed a FCTP in two-stage supply chain network in Gaussian type-2 fuzzy environments. This paper presented two mathematical models. They used two algorithms for solve model, included both genetic algorithm based on Roulette wheel selection, arithmetic crossover with uniform mutation and modified particle swarm optimization. Hajiaghaei-Keshteli et al. (2010) considered the nonlinear fixed cost transportation problem and proposed a new method to design chromosomes in the genetic algorithm (GA) based on Prufer number spanning tree. Also, Lotfi and Tavakkoli-Moghaddam (2013) utilized a new chromosome based on priority in GA. An electromagnetism algorithm is employed for solving FCTP by Sanei et al. (2013). Molla-Alizadeh-Zavardehi et al. (2013) used three metaheuristics and hybrid VNS. Also various new neighborhood structures were proposed for the first time. In a recent research (Baidya et al. 2016) and (Fakhrzad and Heydari , 2008), a multi-purposes multi-stages problem is studied. Solving this problem using grey number theory is under conditions of uncertainty.

Therefore, by studying the literature on the subject, so far, there have been many studies in the field of transportation. Some of these studies are limited to simple transportations and some others, in addition to fixed-cost transportation. In this research, we are dealing with fixed cost transportation problem, assuming that there are different routes with different capacities. And we have to choose a way to send the goods at least cost. With regard to our various solutions, we use modern methods (meta-algorithms) for a large-scale problem in order to get the best results.

This paper is organized as follows. In section 2 the proposed model is described. In section 3, our solution approaches are presented. Computational results are investigated in section 4. Finally, the results and suggestions for future works are implemented in section 5.

2- Modeling framework

We can develop a mathematical programming model for the FCTP. Considering, there are *m* types of suppliers and *n* types of customers. Therefore, if it is not shipped a good in a route, it costs zero. But there are two types of costs in case of carriage of good: for each transportation route, if using that route, there is a fixed cost independent of the number of shipped products. Also, the variable cost is proportional to the number of shipped products. According to the problem that a problem is FCTP with fixed cost, we will consider the existence of several routes with different capacities. So, there are two types of fixed and variable cost, which the routing transportation cost, is the sum of its variable cost of c_{ijlk} and fixed cost f_{ijlk} .We consider the transportation network to be coordinated. The number of resource points, distribution centers, and demand is clear and the input is the issue. The amount of available resources and demand is clear. The capacity of the vehicles is known. The capacity of the distribution centers is known. The capacity of the routes is known. We will consider a specific capacity for each route, according to the number of goods shipped to choose the desired route. Demand is certain, too.

2-1- Model and problem variables

2-1-1 Notations

m	supplier (warehouse or factory)
n	customer (destination or point of demand)
k	Possible routes for shipping goods
c_{ijlk}	Cost (variable) Send per unit from supplier i to customer j using route k
f_{ijlk}	Fixed cost to open the route from supplier i to customer j by route k
x_{ijlk}	an unknown quantity is to be transmitted by route (i, j) from supplier i to consumer j by route k.
y_{ijlk}	Each rout has a fixed cost, which if it is selected, the cost is considered equivalent.
a_i	Number of units in the facility *i*
b_j	Number of units demanded in customer place *j*
e_k	is the unit number of the product that can be by *k* different route of transportation
$L(i, j)$	Set of all routes from node *i* and *j*

$$\min\ z = \sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{k=1}^{k}\sum_{l\in L(i,j)}\left(c_{ijlk}\ x_{ijlk} + f_{ijlk}\ y_{ijlk}\right) \tag{1}$$

$$\sum_{j=1}^{n}\sum_{k=1}^{k}\sum_{l\in L(i,j)} x_{ijlk} \le a_i \qquad i = 1,2,...,m \tag{2}$$

$$\sum_{i=1}^{m}\sum_{k=1}^{k}\sum_{l\in L(i,j)} x_{ijlk} \ge a_i \qquad j = 1,2,...,n \tag{3}$$

$$\sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{l\in L(i,j)} x_{ijlk} \le e_i \qquad k = 1,2,...,k \tag{4}$$

$$x_{ijlk} \ge 0 \;\; i = 1,2,...,m \quad j = 1,2,...,n \quad k = 1,2,...,k \;\; \forall l \in L(i, j) \tag{5}$$

$$y_{ijlk} = 0 \;\; if \;\; x_{ijlk} = 0 \;\; i = 1,2,...,m \quad j = 1,2,...,n \quad k = 1,2,...,k \;\; \forall l \in L(i, j)$$

$$y_{ijlk} = 1 \;\; if \;\; x_{ijlk} > 0 \;\; i = 1,2,...,m \quad j = 1,2,...,n \quad k = 1,2,...,k \;\; \forall l \in L(i, j)$$

The objective function (1) is to determine which paths should be opened, and the size of the cargo is determined on those routes using the carriers, in such a way that the total cost of the requested application is minimized and at the same time supply constraints and path capacity are met. The constraint set (2) shows that the delivered goods should be less than or equal to the amount of available inventory available to the supplier. The constraint set (3) shows that the amount of goods shipped must be either equal to the customer's demand or more than that, the customer's demand must be fully met. The constraint set (4) specifies that the amount of goods in the program should be according to the capacity of the route (it may be equal to or less than the capacity of the route).

3- Solution approach

NP-hard problems need effective methods to find the best optimal solution. Therefore, the hybridization methods of some recent meta-heuristics give the opportunity to make trade-off between the exploration and exploitation phases. In this paper, researchers have some good plans to solve the proposed model.

First of all, we presented two powerful recent meta-heuristics. These methods are used repeatedly in the recent papers to solve the NP-*hard* problems. As mentioned earlier, most of papers in FCTP used two successful methods (*i.e.* GA and SA). So, researchers also utilize these two well-known methods to compare with the new proposed methods. In addition, two new hybridization methods are developed from both groups in order to enrich the algorithms and used their advantages. In the following subsections, the proposed methods are detailed to address the problem.

3-1- Red Deer Algorithm (RDA)

Although many methods have been developed in the recent two decades, but just only a few of them considered and discussed on the two important phases; exploration and exploitation, and their trade-off. Red Deer algorithm Fathollahi Fard and Hajiaghaei-Keshteli (2016) is one of first methods in recent meta-heuristics to give the opportunity to a user to make a balance between intensification and diversification. This algorithm explored the Red Deer's characteristics in breeding season and simulated their main behaviors in this specially time of year. The Scottish Red Deer (*Cervus Elaphus Scoticus*) is a subspecies of Red Deer and lives in British Isles (Fakhrzad and Sadri Esfahani , 2013a). The males roar loudly and repeatedly during the breeding season and females prefer a high to a low roaring rate. The males want to increase their territory and the number of hinds in their harems. So, the course of fight is unavoidable. Although it is possible that a male has no territory and harem, hence, they prefer to mate with a handy hind. In a nutshell, RDA starts with an initial population, called Red Deers (RD). They are divided into two types: hinds and male RDs. Besides, a harem is a group of female RDs, and the competition of male RDs to get the harem with more hinds via roaring and fighting, and their mating behavior is the basis of the proposed evolutionary algorithm. In the continuous, the steps of the algorithm are detailed in the pseudo-code as shown in figure 1.

```
Initialize the Red Deers population.
Calculate the fitness and sort them and form the hinds (N_hind) and male RDs (N_male).
X*=the best solution.
while (t< maximum number of iteration)
    for each male RD
        A local search near his position.
        Update the position if better than the prior ones.
    end for
    Sort the males and also form the stags and the commanders.
    for each male commander
        Fight between male commander and stag.
        Update the position of male commander and stag.
    end for
    Form harems.
    for each male commander
        Mate male commander with the selected hinds of his harem randomly.
        Select a harem randomly and name it k.
        Mate male commander with some of the selected hinds of the harem.
    end for
    for each stag
        Calculate the distance between the stag and all hinds and select the nearest hind.
        Mate stag with the selected hind.
    end for
    Select the next generation with roulette wheel selection.
    Update the X* if there is better solution.
    t=t+1
end while
return X*
```

Fig 1. The pseudo-code of RDA

3-2-Stochastic Fractal Search (SFS)

Stochastic Fractal Search (SFS) is introduced by Salimi (2015) is one of population-based and stochastic optimization techniques and inspired by the natural phenomenon of fractal's growth. The characteristics of fractals in this algorithm are summarized as follows:

- ✓ Each particle has an electrical potential energy.
- ✓ Each particle diffuses, and causes some other random particles to be created, and the energy of the seed particle is divided among generated particles.

- ✓ Only few of the best particles remain in each generation, and the rest of the particles are disregarded.

Two main processes occurred in the SFS are: The diffusing process and the updating process. In the first process, similar to Fractal Search, each particle diffuses around its current position to satisfy intensification (exploitation) property. This process increases the chance of finding the global minima, and also prevents being trapped in the local minima. In the latter process, the algorithm simulates how a point in the group updates its position based on the position of other points in the group. Unlike the diffusing phase in FS which causes a dramatic increase in the number of participating points, we consider a static diffusion process for SFS. It means that the best generated particle from the diffusing process is the only particle that is considered, and the rest of the particles are discarded. In addition to efficient exploration of the problem space, SFS uses some random methods as updating processes. In other word, updating process in SFS leads us to diversification (exploration) properties in meta-heuristic algorithms. For more data, the pseudo-code of algorithm is depicted in figure 2.

```
Initialize random solutions.
Select the best solution X.
while (t<maximum number of iterations)
      for each fractal
         Do exploration phase by searching new position for new fractals.
         Calculate the fitness of these positions.
         if New fractal better the prior ones.
               Replace the new position.
         end if
      endfor
      t=t+1;
      Update the X* if there is better solution.
endwhile
Returan X
```

Fig 2. The pseudo-code of SFS

3-3-Genetic Algorithm (GA)

Evolutionary algorithms (EAs) were discovered to simulate some of the processes which are seen in nature evolution. (Goldberg and Holland, 1988) has developed the Genetic algorithm to solve the huge and complex problems, for the first time. GA is inspired by genetic evolutionary. GAs are the special type of EAs and include so many methods in this classification (*i.e*. Genetic programming (Baidya et al. 2016), Scatter search (Glover, 1977), and Differential evolution (Storn and Price, 1997). Chromosomes are the structure of cells in animals, plants and humans. In GA, we define an array of variables which called chromosome (Hajiaghaei-Keshteli , 2011) and (Kirkpatrick, et al. 1983). Chromosomes are altered by two operators: mutation and crossover. So, some new solutions are created by these two mentioned operators (Engin et al, 2011) and (Hajiaghaei-Keshteli and Aminnayeri , 2014) and (Fard et al. 2018).

In the Genetic algorithm, like other methods, in first step, some random solutions in feasible space are initialized. In mutation, one solution changes to a new solution by generating a neighbor of this solution. In crossover, the two chromosomes called parents are selected. They compose together and make two new solutions named offspring (Hajiaghaei-Keshteli and Fathollahi-Fard, 2018a) and (Fathollahi-Fard et al. 2018a) and (Hajiaghaei-Keshteli and Fathollahi Fard, 2018a) and (Fathollahi-Fard, and Hajiaghaei-Keshteli , 2018b). These two operators are so simple. Hence, user can utilize a creative way to do these operators. However, GA is so easy and simple to code, but it has not any special plans to explore the potential areas. As discussed earlier, the trade-off between the two phases is so significant. GA just does these two important phases by crossover and mutation operators which are blind in search space as mentioned before. As shown in figure 3, steps of algorithm are explained.

```
Generate random population.
Calculate the fitness of each individual chromosome.
X*=the best solution.
while (t< maximum number of iteration)
    Select a pair of chromosomes as parents.
    Perform crossover and mutation to generate new chromosomes.
    Merge the all chromosomes and select the new population.
    Update the X* if there is better solution.
    t=t+1
end while
return X*
```

Fig 3. The pseudo-code of GA

3-4- Simulated Annealing (SA)

SA introduced by Kirkpatrick et al. (1983), is based on the annealing process of metals. Researchers know that SA is an intelligent single-solution method. In addition, SA is a kind of a local search algorithm. In this probabilistic algorithm, SA starts with an initial random solution. The neighbor of this solution is made by some suitable techniques. These techniques are similarity to mutate operator in GA. In this regard, the objective function (OF) of this solution is approximated. If a diminution in the cost is reached, this solution is replaced the current solution.

In SA, to escape from local optimum, if the new solution is worse than the current solution, we give a chance to this new solution by a probabilistic function in chemistry engineering. The chance of reception or rejection the new solution is identified by setting random numbers, but this process is controlled by a function named Boltzmann. This probability of reception a move which causes an enhancement δ in OF is named the Acceptance Function (AF) and is normally set to $exp(-\delta/T)$ where T is a control parameter which corresponds to temperature in the analogy with physical annealing. AF explained that small increases in OF are more similarly to be accepted than large increases, also that when T is high most moves will be accepted, while T comes close to zero most difficult moves will be rejected. So in SA, the method is started with a high rate of T, to avoid being ensnared in a local optimum before the due time. Figure 4 shows the pseudo-code of SA.

```
Select a random solution X*.
Initialize the parameters.
while (t< maximum number of iteration)
      sub=0;
      while (sub< maximum number of sub-iteration)
        Create a neighbor of this solution.
        if the function value of the new solution is better than prior
          Replace the new solution as old solution.
        else
          Calculate δ, δ = |f_old − f_new|.
            if rand< exp(−δ/T)
              Replace the new solution.
            endif
        endif
      sub=sub+1;
      endwhile
      Update T.
      Update the X* if there is better solution.
      t=t+1;
endwhile
return X*
```

Fig 4. The pseudo-code of SA

3-5-Hybridized RDA & GA (HRDGA)

In this section, by hybridized RDA and GA, a new meta-heuristic is developed. This algorithm obtains RDA as main loop and GA as a local search. It seems that RDA is very good at intensification phase by two different operators to perform it. In this method, roaring and fighting process are saved and instead of mating process, algorithm obtains the GA by using crossover operator. In order to code this, each commander and all hinds in his harem are mated by crossover operator. This modified of

RDA can reduce the process time and does the diversification phase better than the general of RDA in these special steps about mating process. As illustrated in figure 5, the pseudo-code of the proposed hybridized algorithm is presented. This idea is probed to solve the problem in comparison of its original algorithms.

```
Initialize the Red Deers population.
Calculate the fitness and sort them and form the hinds (N_hind) and male RDs (N_male).
X*=the best solution.
While (t< maximum number of iteration)
    for each male RDs.
     A local search near his position.
     Update the position if better than the prior ones.
   end for
   Sort the males and also form the stags and the commanders.
   for each male commanders
   Fight between male commanders and stags.
   Update the position of male commanders and stags
  end for
    for each male commanders
        Select a hind with roulette wheel selection.
          Specify this commander and mentioned hind as parents.
          Perform crossover and generate two new solutions.
   end for
  Select the next generation with roulette wheel selection.
   Update the X* if there is better solution.
   t=t+1
end while
return X*
```

Fig 5. The pseudo-code of RDGA

3-6-Hybridized SFS & SA (HSFSA)

As mentioned in SFS, this algorithm has two main steps to do exploitation and exploration phases. It seems that this algorithm has not any special plan to escape from local optima. In order to improve the SFS, this method is hybridized with SA to cover the disadvantages. So, a new approach is proposed. This approach uses SA to evaluate the new generation of fractals. As detailed in figure 6, the pseudo-code of proposed algorithm is explained.

```
Initialize random solutions.
Select the best solution X.
while (t<maximum time number of iteration)
        for each fractal
          Do exploration phase by searching new position for new fractals.
          Calculate the fitness of these positions.
          if New fractal better the prior
                 Replace the new position
            else
                 Calculate tetta.
                 Create a random probability by rand
                      if the random probability is lower than exp(-tetta/T)
                           Replace this new solution instead of prior
                      endif
          end if
          Update X.
        endfor
        T=T*(1-alpha);
        t=t+1;
endwhile
Returan X
```

Fig 6. The pseudo-code of SFSA

4- Computational experiments

4-1- Instances

In order to analyze and study the performance of algorithms in this paper, we must have a plan to generate tests problems. The problems are divided into three classes (*i.e*. small, mediate and large). In each class, four random solutions are initialized to design the tests problems. Table 1 shows the experimental design.

Table 1. Experimental design of tests problem

Size of problems	No. of problems	No. of shipments	No. of containers	No. of shipping routes	Total of demands	Volume of shipments		Volume of containers	
						Lower limit	Upper limit	Lower limit	Upper limit
Small	P1	10	3	2	1000	10	30	10000	50000
	P2	15	4	2	1200				
	P3	15	5	3	1500				
	P4	20	5	3	2500				
Medium	P5	40	10	5	2500				
	P6	50	10	6	3000				
	P7	55	12	6	4000				
	P8	60	15	8	5000				
Large	P9	70	20	10	10000				
	P10	80	25	12	15000				
	P11	90	25	14	20000				
	P12	100	30	16	30000				

4-2- Parameter setting

The parameters and their levels for the algorithms are shown in table 2. Generally, the effectiveness of meta-heuristic algorithms depends on the correct choice of the parameters. So, we study the behavior of the different parameters of the proposed algorithms. One of the methods widely used in the most researches is the full factorial design, which tests all possible combinations of factors (Fathollahi-Fard and Hajiaghaei-Keshteli, 2018b) and (Fathollahi-Fard et al. 2018c) and (Sahebjamnia et al. 2018) and (Fathollahi-Fard et al. 2018d). When the number of factors significantly increases, this method does not seem to be effective. For instance, in RDA, there are 6 parameters and three levels for them. In addition, we run each algorithm for thirty times. So, the number of runs for algorithm is equal to $6 \times 3 \times 30 = 540$ times. And it is not possible to perform this work in each algorithm.

Japanese quality consultant Genichi Taguchi popularized a cost-efficient approach, known as robust parameter design. Taguchi assumed that there are two types of factors which operate on a process: control factors and noise factors. Due to unpractical and often impossible omission of the noise factors, the Taguchi tends to both minimize the impact of noise and also find the best level of the influential controllable factors on the basis of robustness (Sahebjamnia et al. 2018) and (Fathollahi-Fard et al. 2018d) and (Fathollahi-Fard et al. 2018e) and (Samadi et al. 2018). Moreover, Taguchi determines the relative importance of each factor with respect to its main impacts on the performance of the algorithm. A transformation of the repetition data to another value which is the measure of variation is developed by Taguchi. The transformation is the signal-to-noise (S/N) ratio, which explains why this type of parameter design is called a robust design, Here, the term ‘signal’ denotes the desirable value (response variable) and ‘noise’ denotes the undesirable value (standard deviation). So the S/N ratio indicates the amount of variation present in the response variable. The aim is to maximize the signal-to-noise ratio. In the Taguchi method, the S/N ratio of the minimization objectives is as such:

$$\text{S/N ratio} = -10 \log_{10}(\text{objective function})^2 \tag{6}$$

Table 2. Parameters and their levels for algorithms

	nPop	MaxT	Sub-it	reducT	init T	P_C	P_M	nMale	alpha	betta	gamma	walk	nDiff
GA	100	5				0.5	0.02						
	150	10				0.6	0.05						
	200	15				0.7	0.1						
SA		5	20	0.9	200								
		10	30	0.99	300								
		15	50	0.999	500								
RDA	100	5						7	0.7	0.4	0.6		
	150	10						10	0.8	0.5	0.7		
	200	15						15	0.9	0.6	0.8		
SFS	100	5										0.3	2
	150	10										0.5	5
	200	15										0.7	10
RDGA	100	5						7			0.6		
	150	10						10			0.7		
	200	15						15			0.8		
SFSA	100	5		0.9	200							0.3	2
	150	10		0.99	300							0.5	5
	200	15		0.999	500							0.7	10

For GA, SA, SFS and RDGA, we have only 4 parameters to tune. Table 3 shows the modified orthogonal array L9, where control factors are assigned to the columns of the orthogonal array and the corresponding integers in these columns indicate the actual levels of these factors. This table is used for four mentioned algorithms. In addition, table 4 displays the modified orthogonal array L27 which is used for tuning RDA and SFSA which have 6 parameters to tune.

Table 3. The modified orthogonal array L9 for the GA, SA, SFS and RDGA.

Trial	A	B	C	D
1	1	1	1	1
2	1	2	2	2
3	1	3	3	3
4	2	1	2	3
5	2	2	3	1
6	2	3	1	2
7	3	1	3	2
8	3	2	1	3
9	3	3	2	1

Table 4. The modified orthogonal array L27 for the RDA and SFSA

Trial	A	B	C	D	E	F
1	1	1	1	1	1	1
2	1	1	1	1	2	2
3	1	1	1	1	3	3
4	1	2	2	2	1	1
5	1	2	2	2	2	2
6	1	2	2	2	3	3
7	1	3	3	3	1	1
8	1	3	3	3	2	2
9	1	3	3	3	3	3
10	2	1	2	3	1	2
11	2	1	2	3	2	3
12	2	1	2	3	3	1
13	2	2	3	1	1	2
14	2	2	3	1	2	3
15	2	2	3	1	3	1
16	2	3	1	2	1	2
17	2	3	1	2	2	3
18	2	3	1	2	3	1
19	3	1	3	2	1	3
20	3	1	3	2	2	1
21	3	1	3	2	3	2
22	3	2	1	3	1	3
23	3	2	1	3	2	1
24	3	2	1	3	3	2
25	3	3	2	1	1	3
26	3	3	2	1	2	1
27	3	3	2	1	3	2

Twelve test problems, with different sizes, are solved to evaluate the performance of the presented algorithms. The experiments on the algorithms were based on the *L9* and *L27* orthogonal array, therefore 9 and 27 different combinations of control factors were considered, respectively. Due to stochastic nature of meta-heuristics, thirty replications were performed for each trial to achieve the more reliable results. Because the scale of objective functions in each instance is different, they could not be used directly. To solve this problem, the relative percentage deviation (*RPD*) is used for each instance.

$$\mathrm{RPD} = \frac{\mathrm{Alg_{sol}} - \mathrm{Min_{sol}}}{\mathrm{Min_{sol}}} \times 100 \qquad (7)$$

Where $\mathrm{Alg_{sol}}$ and $\mathrm{Min_{sol}}$ are obtained from objective value for each replication of trial in a given instance and the obtained best solution respectively.

After converting the objective values to RPDs, the mean RPD is calculated for each trial. To do according Taguchi parameter design instructions, these mean RPDs, are transformed to S/N ratios. The S/N ratios of trials are averaged in each level. As shown in figure 7 to 12, the best values for parameters in all algorithms are identified. In SA, figure 7 shows that A2, B1, C2 and D3 are the best performance among all parameters for SA. In addition, in SFS, as seen in figure 8, the best values for parameters are A3, B3, C3 and D2, respectively. Besides, in RDA, A2, B2, C3, D2, E1 and F1 are the most suitable parameters for this method as displayed in figure 9. Furthermore, A3, B3, C3, D2, E3, F3 are the best parameters set for the proposed SFSA as shown in figure 10. For GA, A3, B3, C3 and D2 are the most suitable parameters set as seen in figure 11. At the least, the best performance values for RDGA are obviously specified in the figure 12 as A1, B1, C1 and D3.

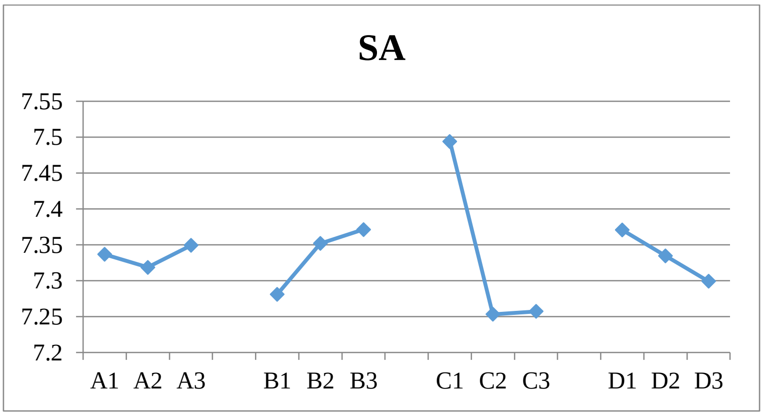

Fig 7. Mean RPD plot for each level of the factors in SA.

Fig 8. Mean S/N ratio plot for each level of the factors in SFS.

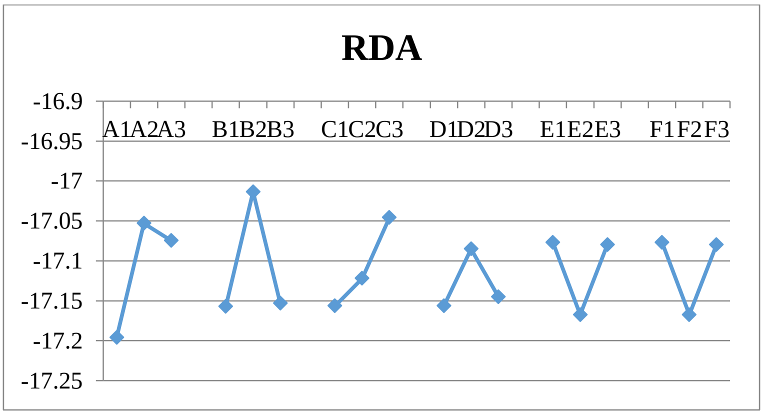

Fig 9. Mean S/N ratio plot for each level of the factors in RDA

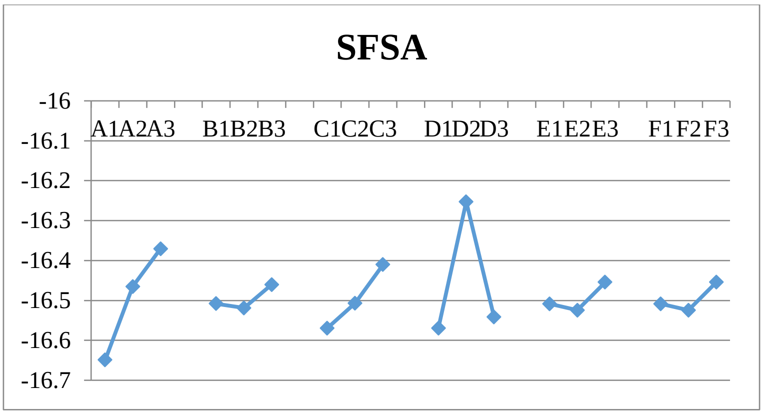

Fig 10. Mean S/N ratio plot for each level of the factors in SFSA

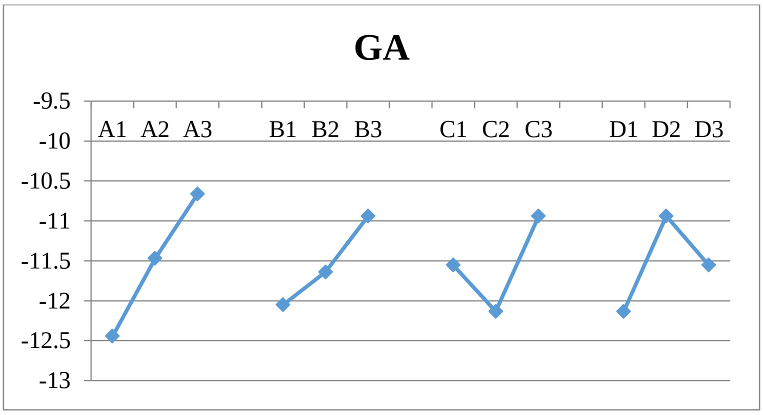

Fig 11. Mean S/N ratio plot for each level of the factors in GA

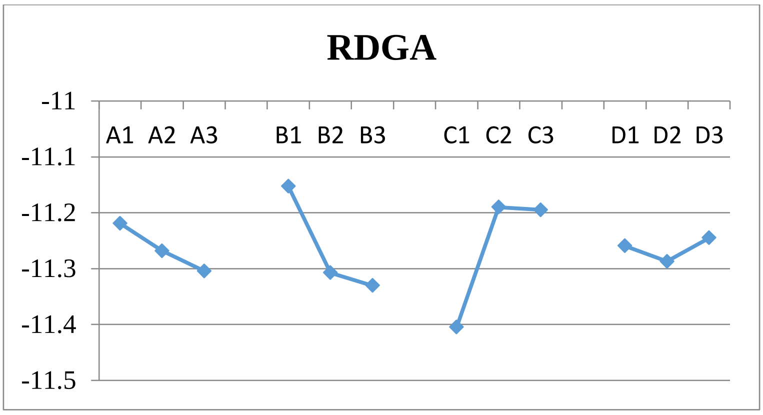

Fig 12. Mean S/N ratio plot for each level of the factors in RDGA

4-3--Experimental results

In this section, a comprehensive analysis is done on algorithms. It should be noted that each algorithm is run for thirty times. Hence, the results are based on the best value among thirty runs. We also obtain an exact approach by LINGO software to compare the outputs of algorithms. This method is checked the satisfying the outputs of metaheuristics. Table 5 shows the results of the experiments.

Table 5. The final outputs for the methods (G=Global optimum, L=Local optimum)

P_i	SA	GA	RDA	SFS	RDGA	SFSA	LINGO	
P1	10640	11590	10870	11340	11250	11730	8710	G
P2	12490	11780	11840	12490	12630	11920	10430	G
P3	15420	15280	15130	15360	14590	14580	15320	G
P4	18970	17460	17240	18470	18210	18190	16940	G
P5	23560	22840	22580	22690	21870	22960	22370	L
P6	27510	28690	26540	27160	27390	27430	26540	G
P7	33260	32710	33720	33450	33180	34710	32460	L
P8	41960	42560	40830	41590	42510	41620	41730	L
P9	57690	56480	55420	54620	54170	55640	55490	L
P10	63840	66120	64320	65640	64570	65910	63410	L
P11	74580	76870	73620	72660	74320	77420	72295	L
P12	85490	86740	84630	85910	84370	85490	84370	L

In addition, we use Gap to show the performance of proposed algorithms as illustrated in equation (20). Where Alg_{sol} is the final value of objectives and Best_{sol} is the best solution among all methods. Gap explains the deviation of solutions from the best solution. In order to achieve this purpose, table 6 is provided. Also, to illustrate this fact clearly figure 13 shows the Gap behavior of proposed meta-heuristics. As obviously in the figure, the proposed RDA and its hybridized meta-heuristic shows the better performance among all algorithms in this study.

$$\text{Gap} = \frac{\text{Alg}_{\text{sol}} - \text{Best}_{\text{sol}}}{\text{Best}_{\text{sol}}} \tag{8}$$

Table 6. The Gap value for each approach.

P_i	SA	GA	RDA	SFS	RDGA	SFSA	LINGO
P1	0.221584	0.330654	0.247991	0.301952	0.291619	0.346728	0
P2	0.197507	0.129434	0.135187	0.197507	0.21093	0.142857	0
P3	0.057613	0.048011	0.037723	0.053498	0.000686	0	0.050754
P4	0.119835	0.030697	0.01771	0.090319	0.07497	0.07379	0
P5	0.077275	0.044353	0.032465	0.037494	0	0.04984	0.022862
P6	0.036549	0.08101	0	0.023361	0.032027	0.033534	0
P7	0.024646	0.007702	0.038817	0.030499	0.022181	0.069316	0
P8	0.027676	0.042371	0	0.018614	0.041146	0.019349	0.022043
P9	0.064981	0.042644	0.023076	0.008307	0	0.027137	0.024368
P10	0.006781	0.042738	0.014351	0.035168	0.018294	0.039426	0
P11	0.026424	0.057941	0.01101	0	0.022846	0.065511	0.003991
P12	0.013275	0.028091	0.003082	0.018253	0	0.013275	0
Average	0.072845	0.073804	0.046784	0.067914	0.059558	0.073397	0.010335

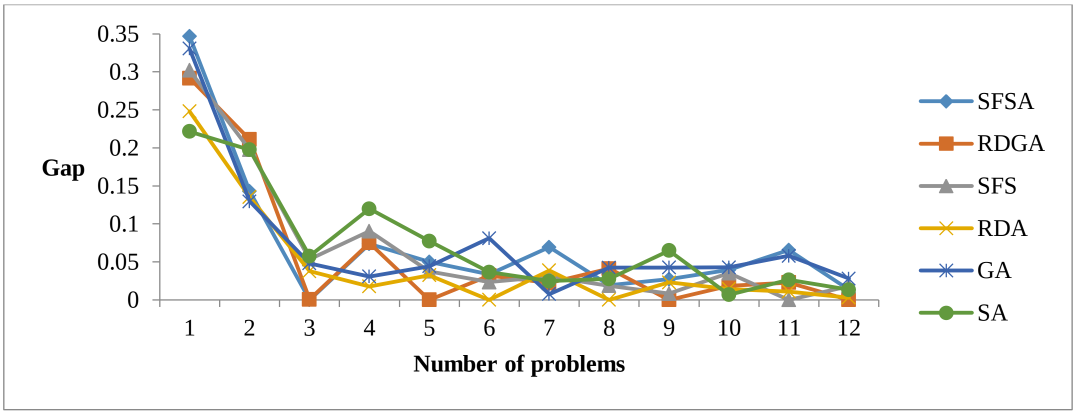

Fig 13. Gap behavior of algorithms

In order to verify the statistical validity of the results, we have performed an analysis of variance (ANOVA) to accurately analyze the results. The results demonstrate that there is a clear statistically significant difference between performances of the algorithms. The means plot and LSD intervals (at the 95% confidence level) for six algorithms are shown in figure 14.

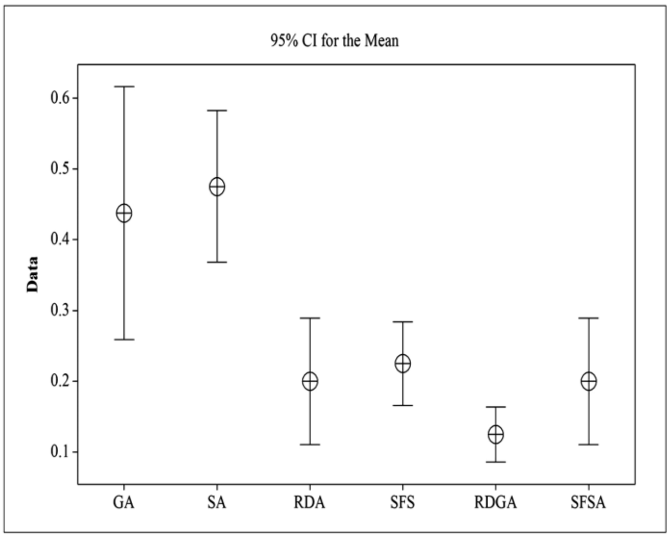

Fig 14. Means plot and LSD intervals for the algorithms.

5-Conclusion and future works

This study aims to explore the freight consolidation and containerization problem with supposition variable sizes for containers as an NP-*hard* problem. In order to solve the problem, the two novel meta-heuristics are used firstly in an engineering problem. Besides, the two traditional algorithms are obtained to compare the novel ones. In addition, this paper presents the two hybridized meta-heuristics based on novel proposed approach for the first time as well. All of parameters of algorithms are tuned by Taguchi experimental design method. Finally, the performance and efficiency of algorithms are investigated and some important analyses are created to show the fact. The results explain that RDA and its proposed hybridized method have the better performance among all algorithms.

For future studies, to explore the algorithms exactly, more comprehensive analysis may be needed. In addition, some other real constraints can propose to develop the problems. Moreover, some real study cases can test on our model. At the end, more real scale optimization problems can be obtained to evaluate the two new hybridized algorithms.

Reference

Baidya, A., Bera, U. K., & Maiti, M. (2017). Models for solid transportation problems in logistics using particle swarm optimisation algorithm and genetic algorithm. *International Journal of Logistics Systems and Management*, *27*(4), 487-526.

Baidya, A., Bera, U. K., & Maiti, M. (2016). The grey linear programming approach and its application to multi-objective multi-stage solid transportation problem. *OPSEARCH*, *53*(3), 500-522.

Chopra S. Meindl P., Supply Chain Management, Pearson, 4th edition, (2010), 978-0-130608040-4.

Chopra, S., Meindel, P. (2007). Supply chain management: Strategy, planning and operation, Pearson Prentice Hall.

EL IDRISSI, A. L., TAJANI, C., & SABBANE, M. (2017). NEW CROSSOVER OPERATOR FOR GENETIC ALGORITHM TO RESOLVE THE FIXED CHARGE TRANSPORTATION PROBLEM. *Journal of Theoretical & Applied Information Technology*, *95*.

Engin, O., Ceran, G., & Yilmaz, M. K. (2011). An efficient genetic algorithm for hybrid flow shop scheduling with multiprocessor task problems. *Applied Soft Computing*, *11*(3), 3056-3065.

Fakhrzad, M.B., A. Sadri Esfahani, (2013a). Modeling the time windows vehicle routing problem in cross-docking strategy using two meta-heuristic algorithms. *International Journal of Engineering-Transactions A: Basics* 27 (7), 1113.

Fakhrzad, M. B., Sadeghieh, A., & Emami, L. (2013b). A new multi-objective job shop scheduling with setup times using a hybrid genetic algorithm.

Fakhrzad, M.B., Heydari, M., (2008). A Heuristic Algorithm for Hybrid Flow-shop Production Scheduling to Minimize the Sum of The Earliness ANDF Tardiness Costs. *Journal of the Chinese Institute of Industrial Engineers*. 25 (2), 105-115.

Fard, A. M. F., & Hajiaghaei-Keshteli, M. (2018). A bi-objective partial interdiction problem considering different defensive systems with capacity expansion of facilities under imminent attacks. *Applied Soft Computing*, *68*, 343-359.

Fard, A. F., & Hajiaghaei-Keshteli, M. (2016). Red deer algorithm (RDA); a new optimization algorithm inspired by red deer's mating. In *International Conference on Industrial Engineering, IEEE., (2016 e)* (pp. 33-34).

Fathollahi-Fard, A. M., Hajiaghaei-Keshteli, M., & Tavakkoli-Moghaddam, R. (2018a). The Social Engineering Optimizer (SEO). *Engineering Applications of Artificial Intelligence*, *72*, 267-293.

Fathollahi-Fard, A. M., & Hajiaghaei-Keshteli, M. (2018b). A stochastic multi-objective model for a closed-loop supply chain with environmental considerations. *Applied Soft Computing*, *69*, 232-249.

Fathollahi-Fard, A. M., Hajiaghaei-Keshteli, M., & Mirjalili, S. (2018c). Hybrid optimizers to solve a tri-level programming model for a tire closed-loop supply chain network design problem. *Applied Soft Computing*.

Fathollahi-Fard, A. M., Hajiaghaei-Keshteli, M., & Mirjalili, S. (2018d). Multi-objective stochastic closed-loop supply chain network design with social considerations. *Applied Soft Computing*, *71*, 505-525.

Fathollahi-Fard, A. M., Hajiaghaei-Keshteli, M., & Tavakkoli-Moghaddam, R. (2018e). A bi-objective green home health care routing problem. *Journal of Cleaner Production*, *200*, 423-443.

Glover, F. (1977). Heuristics for integer programming using surrogate constraints. *Decision Sciences*, *8*(1), 156-166.

Goldberg, D. E., & Holland, J. H. (1988). Genetic algorithms and machine learning. *Machine learning*, *3*(2), 95-99.

Golmohamadi, S., Tavakkoli-Moghaddam, R., & Hajiaghaei-Keshteli, M. (2017). Solving a fuzzy fixed charge solid transportation problem using batch transferring by new approaches in meta-heuristic. *Electronic Notes in Discrete Mathematics*, *58*, 143-150.

Hajiaghaei-Keshteli, M., & Fard, A. M. F. (2018a). Sustainable closed-loop supply chain network design with discount supposition. *Neural Computing and Applications*, 1-35.

Hajiaghaei-Keshteli, M., & Fathollahi-Fard, A. M. (2018b). A set of efficient heuristics and metaheuristics to solve a two-stage stochastic bi-level decision-making model for the distribution network problem. *Computers & Industrial Engineering*, *123*, 378-395.

Hajiaghaei-Keshteli, M., & Aminnayeri, M. (2014). Solving the integrated scheduling of production and rail transportation problem by Keshtel algorithm. *Applied Soft Computing*, *25*, 184-203.

Hajiaghaei-Keshteli, M. (2011). The allocation of customers to potential distribution centers in supply chain networks: GA and AIA approaches. *Applied Soft Computing*, *11*(2), 2069-2078.

Hajiaghaei-Keshteli, M., Molla-Alizadeh-Zavardehi, S., & Tavakkoli-Moghaddam, R. (2010). Addressing a nonlinear fixed-charge transportation problem using a spanning tree-based genetic algorithm. *Computers & Industrial Engineering*, *59*(2), 259-271.

Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *science*, *220*(4598), 671-680.

Lotfi, M. M., & Tavakkoli-Moghaddam, R. (2013). A genetic algorithm using priority-based encoding with new operators for fixed charge transportation problems. *Applied Soft Computing*, *13*(5), 2711-2726. (14)

Midya, S., & Roy, S. K. (2017). Analysis of interval programming in different environments and its application to fixed-charge transportation problem. *Discrete Mathematics, Algorithms and Applications*. (7)

Molla-Alizadeh-Zavardehi, S., Nezhad, S. S., Tavakkoli-Moghaddam, R., & Yazdani, M. (2013). Solving a fuzzy fixed charge solid transportation problem by metaheuristics. *Mathematical and Computer Modelling*, *57*(5), 1543-1558. (16)

Mingozzi, A., & Roberti, R. (2017). An Exact Algorithm for the Fixed Charge Transportation Problem Based on Matching Source and Sink Patterns. *Transportation Science*. (8)

Pop, P., Matei, O., Sitar, C. P., & Zelina, I. (2016). A hybrid based genetic algorithm for solving a capacitated fixed-charge transportation problem. *Carpathian Journal of Mathematics*, 225-232. (10)

Pramanik, S., Jana, D. K., Mondal, S. K., & Maiti, M. (2015). A fixed-charge transportation problem in two-stage supply chain network in Gaussian type-2 fuzzy environments. *Information Sciences*, *325*, 190-214. (12)

Salimi, H. (2015). Stochastic fractal search: a powerful metaheuristic algorithm. *Knowledge-Based Systems*, *75*, 1-18. (19)

Samadi, A., Mehranfar, N., Fathollahi Fard, A. M., & Hajiaghaei-Keshteli, M. (2018). Heuristic-based metaheuristics to address a sustainable supply chain network design problem. *Journal of Industrial and Production Engineering*, *35*(2), 102-117. (37)

Sadeghi-Moghaddam, S., Hajiaghaei-Keshteli, M., & Mahmoodjanloo, M. (2017). New approaches in metaheuristics to solve the fixed charge transportation problem in a fuzzy environment. *Neural Computing and Applications*, 1-21. (4)

Sahebjamnia, N., Fard, A. M. F., & Hajiaghaei-Keshteli, M. (2018). Sustainable tire closed-loop supply chain network design: Hybrid metaheuristic algorithms for large-scale networks. *Journal of Cleaner Production*. (34)

Saxena, R. R., & Upmanyu, M. (2016). A compromise method for solving fuzzy multi objective fixed charge transportation problem. *Lecture Notes in Management Science*, *8*, 9. (11)

Shoushtary, M.A., Hoseini Nasab, H., Fakhrzad, M.B.(2014), Team robot motion planning in dynamics environments using a new hybrid algorithm (honey bee mating optimization-tabu list).*Chinese Journal of Engineering*, 1-8. (38)

Sanei, M., Mahmoodirad, A., & MOLLA, A. Z. S. (2013). An electromagnetism-like algorithm for fixed charge solid transportation problem. (15)

Storn, R., & Price, K. (1997). Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, *11*(4), 341-359. (23)

Zhang, B., Peng, J., Li, S., & Chen, L. (2016). Fixed charge solid transportation problem in uncertain environment and its algorithm. *Computers & Industrial Engineering*, *102*, 186-197. (9)