Journal of Optimization in Industrial Engineering Vol.12, Issue 1, Winter and Spring 2019, 41- 52
DOI: 10.22094/joie.2017.738.1469

Solving the Fixed Charge Transportation Problem by New Heuristic Approach

Komeil Yousefi [a], Ahmad J. Afshari [a], Mostafa Hajiaghaei-Keshteli [b,*]

[a] *Department of Industrial Engineering, Shomal University, Amol, Iran*
[b] *Department of Industrial Engineering, University of Science and Technology of Mazandaran, Behshahr, Iran*

Received 11 March 2017; Revised 15 September 2017; Accepted 02 November 2017

Abstract

The fixed charge transportation problem (FCTP) is a deployment of the classical transportation problem in which a fixed cost is incurred, independent of the amount transported, along with a variable cost that is proportional to the amount shipped. Since the problem is considered as an *NP*-hard, the computational time grows exponentially as the size of the problem increases. In this paper, we propose a new heuristic along with well-known metaheuristic like Genetic algorithm (GA), simulated annealing (SA) and recently developed one, Keshtel algorithm (KA) to solve the FCTP. Contrary to previous works, we develop a simple and strong heuristic according to the nature of the problem and compare the result with metaheuristics. In addition, since the researchers recently used the priority-based representation to encode the transportation graphs and achieved very good results, we consider this representation in metaheuristics and compare the results with the proposed heuristic. Furthermore, we apply the Taguchi experimental design method to set the proper values of algorithms in order to improve their performances. Finally, computational results of heuristic and metaheuristics with different encoding approaches, both in terms of the solution quality and computation time, are studied in different problem sizes.

Keywords: Fixed charge transportation problem; Heuristic; Metaheuristic algorithms; Priority-based.

1. Introduction

The general transportation problem (TP) and its variations have been one of the attractive topics both in industries and academia. Especially, the researchers in this research area have been mostly focused on solution approaches and have been utilizing or developing several approaches. In any transportation problem, a basic assumption is that the cost of transportation is directly proportional to the number of units transported, while, in most real-world applications as shown in Figure 1, a fixed cost for developing the facilities or fulfilling the demand of customers from each source is also considered.

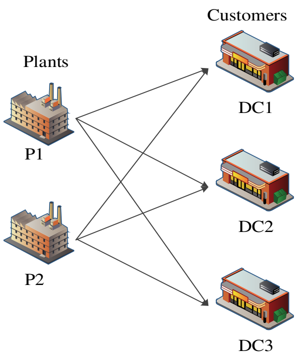

Fig. 1. A single-stage FCTP.

The fixed charge may represent the cost of renting a vehicle, toll charges on a highway, landing fees at an airport, set-up costs for machines in a manufacturing environment, the cost of building roads in transportation systems, time to locate a file in a distributed database system, etc. Fixed charge problems arise in a large number of production and transportation systems. Many practical transportation and distribution problems can be modeled as fixed charge transportation problems (FCTP). A fixed cost or charge in FCTP is incurred for every rout that is opened or used to send the demands. Two kinds of costs are considered in the fixed charge transportation problems: a variable cost that linearly increases with the amount transported between a source and a destination, and a fixed charge, which is incurred whenever a route is opened or used to send the demands to destination.

2. Literature Review

Hirsch and Dantzig (1954) firstly proposed the fixed charge problem and consequently, Balinski (1961) developed the FCTP for the first time in the literature. He studied the problem's structures and to solve the problem he developed an approximate algorithm. This problem is later discussed about its complexity by Klose (2008). He showed that solution time to the size of the problem increases exponentially, and proposed approximate algorithms to solve such problems. So, over the last two

*Corresponding author Email address: mostafahaji@mazust.ac.ir

decades, several heuristic and metaheuristic methods have been proposed by researchers to solve fixed charge transportation problems (see, e.g., heuristics (Aguado, 2009; Loch and Silva, 2014); tabu search (Sun *et al.*, 1998); simulated annealing (Jawahar *et al.* 2012; Yadegari *et al.* 2015); genetic algorithm (GA) (Hajiaghaei-Keshteli *et al.* 2010; Lotfi and Tavakkoli-Moghaddam, 2013; Tari and Hashemi, 2016;); artificial immune algorithm (Molla-Alizadeh-Zavardehi *et al*. 2011); hybrid particle swarm algorithm with artificial immune (El-Sherbiny and Alhamali, 2013); simplex-based simulated annealing (Yaghini *et al*. 2012); minimum cost flow-based genetic algorithm (Xie and Jia, 2012)).

Several heuristics can be found in the literature for the FCTP. Relaxation methods have been proposed by Guignard (1988) and developed by Wright *et al*. (1989, 1991). Most of the proposed heuristics search for good primal solutions (see Balinski, 1961; Dwyer, 1966; Steinberg, 1970; Walker, 1976; Shetty, 1990; Sun and McKeown, 1993). A Tabu search approach based on recency-based and frequency-based memories, along with two strategies for each of the intermediate and long-term memory processes, was proposed by Sun *et al*. (1998). Later, A simple heuristic is also developed and studied for small sizes by Adlakha and Kowalski (2003). Also, Glover *et al*. (2005) developed a parametric ghost image process and tested their heuristic on the FCTP and Aguado (2009) presented a heuristic approach based on Lagrangean relaxation techniques, decomposition methods, and branch and cut algorithms for solving a sequence of core problems. Loch and Silva (2014) proposed a heuristic, comparing the quality solution and computational time with the widely used solver CPLEX. Buson *et al*. (2014) proposed an ILS heuristic with the restart phase guided by a sequence of non-decreasing lower bounds that are computed using a novel three-index mathematical formulation, based on discretization, with additional valid inequalities.

Prüfer number, as one of the effective methods in network problems, has been initially introduced by Gen and Chang (1997). Encoding based on Prüfer number has been used successfully in the spanning tree-based representation (e.g. see Mahmoodi-Rad *et al*, 2014; Hajiaghaei-Keshteli, 2011; Molla-Alizadeh-Zavardehi *et al*, 2011). Gen and Li (1999) provided a genetic algorithm by using spanning tree-based representation and Prüfer number to solve two objective transportation problems with fixed cost. Gen and Cheng (2000) examined the feasibility of chromosomes produced by using Prüfer number and showed that this method does not lead to produce feasible chromosome in some cases. In the production of random chromosome based on spanning tree, there is the possibility that the production chromosome does not match with transportation network graph. Therefore, Jo *et al*. (2007) investigated the feasibility of chromosomes before decoding and converting it to spanning tree and provided a criterion to evaluate the feasibility and then modifying infeasible chromosomes. But this proposed method may allocate much time for modifications to itself. Finally, Hajiaghaei-Keshteli *et al*. (2010) offered feasible chromosomes generation without the need of modification. In fact, they corrected the procedure developed by Jo *et al*. (2007) and their procedure has been utilized in this research area by researchers until now.

Priority-based encoding is also a strong method which is mostly used in recent years in the related research areas. Gen *et al*. (2006) considered the two-stage transportation problem and used the genetic algorithm by using priority-based encoding and provided a new method to design operators. Hwang *et al*. (2008) compared system of direct and U-shaped assembly lines and used the genetic algorithm and priority-based representation to solve assembly line balancing problem. Lee *et al*. (2009) raised reverse logistics network problem in three stages and used a genetic algorithm with priority-based representation which has a new operator for better search in solution space. They also provided an innovative method for the third stage which is the transportation of materials from distribution center to the factory. Lotfi and Tavakkoli-Moghaddam (2013) presented a genetic algorithm using priority-based encoding for linear and nonlinear transportation problem with fixed cost in which the new operators have been provided for better search in solution space. They compared the problem with two priority representation and spanning tree without setting the parameter of proposed algorithm. Tari and Hashemi (2016) used a priority based genetic algorithm to solve the real world size problems in an allocation problem in supply chain.

In this paper, we consider the FCTP and propose a new heuristic along with well-known metaheuristic like Ga, SA and recently developed one, Keshtel algorithm (KA). First of all, to the best of our knowledge, no one has considered (KA) for any kind of FCTPs. Hence, we firstly develop and use this strong algorithm for solving the problem. Besides, since the researchers recently used the priority-based representation to encode the transportation graphs and achieved very good results, we consider this representation in metaheuristics and compare the results with the proposed heuristic. Proposed heuristic is developed based on the nature of the problem which is detailed in related section. Furthermore, Taguchi experimental design method is utilized to set and estimate the proper values of the algorithms' parameters to improve their performance.

Finally, for evaluated and compared the performance of proposed solutions method in terms of solution quality and computation time, we use not only the test data from Hajiaghaei-Keshteli *et al.* (2010) but also we added two new different problem sizes to previous data and reach to nine different problem sizes in three levels; small, medium and large.

Five sections follow this Introduction. The next section describes the problem's mathematical formulation. The proposed metaheuristics and heuristic are detailed in Sections 4 and 5 respectively. Section 6, describes the Taguchi experimental design and compares the computational results. At the end, in Section 7, the paper is concluded and some areas of further research are proposed.

3. Mathematical Model and Descriptions

The problem is considered as a distribution problem with m suppliers and n customers. A supplier can respond to a customer's demand and send its products with the cost of c_{ij} for each unit as shipping cost. Besides, there is a fixed-cost of f_{ij} considered for opening of a route. The a_i and b_j are the value of capacity and demand of each supplier and customer, respectively. The objective minimizes the both variable and fixed costs.

3.1. Notations

Index list

i	*source index*
j	*depot index*

Parameters list

m	*number of sources*
n	*number of depots*
a_i	*capacity of source*
b_j	*demand of depot*
c_{ij}	*variable transportation cost*
f_{ij}	*fixed transportation cost*

Decision variables list

x_{ij}	*unknown quantity to be transported from source i to depot j*
y_{ij}	*a binary variable which is 1 if* $x_{ij}>0$

3.2. Mathematical model

Mathematically this problem may be stated as follows:

$$Minimize\ Z = \sum_{i=1}^{m}\sum_{j=1}^{n}\left(c_{ij}.x_{ij} + f_{ij}.y_{ij}\right) \quad (1)$$

s.t.

$$\sum_{j=1}^{n} x_{ij} = a_i\ ;\ i = 1,2,\dots,m \quad (2)$$

$$\sum_{i=1}^{m} x_{ij} = b_j\ ;\ j = 1,2,\dots,n \quad (3)$$

$$x_{ij} \geq 0;\ \forall i,j \quad (4)$$

$$y_{ij} = \begin{cases} 1 x_{ij} > 0 \\ 0\ otherwise \end{cases}\ ;\ \forall i,j \quad (5)$$

The objective function (1) asks for minimizing the total variable and fixed cost. Constraints (2) require that all goods available at each origin $(i \in 1,2,\dots,m)$ be delivered. Constraints (3) force, in any feasible solution, delivery b_j units of goods to each destination$(j \in 1,2,\dots,n)$. Finally, constraints (4) and (5) set the ranges of variables x_{ij} and y_{ij} , respectively.

In the problem raised in this study, we have two decision variables.x_{ij} which represents the amount of sent goods from supplier i to the customer j and the binary variable y_{ij} which shows reopening or lack of reopening the route ij and if $x_{ij} > 0$, its value is.

4. Proposed Metaheuristics

To explain the proposed metaheuristics, at first we describe priority-based encoding scheme. We employ this representation method into the developed GA, SA and KA.

4.1. Encoding scheme

The first step in solving the problem model is to link it with metaheuristic algorithm structure, i.e., making a communication bridge between the original problem and solution space in which evolution occurs. So choosing an appropriate representation method is one of the most important parts of designing an algorithm. Although most evolutionary algorithms use a random procedure to generate a set of initial solutions, priority representation has been used to achieve initial feasible solutions in this study.

4.2. Priority-based representation

Priority-based encoding is a new type of representation and is considered as strong method used in recent works. Initially Gen *et al*. (2006) used this type of representation in transportation problems and then Lotfi and Tavakkoli-Moghaddam (2013) employed it for FCTP. In this paper the modified decoding algorithm of the Priority-based representation for the FCTP, developed by Lotfi and Tavakkoli-Moghaddam (2013), is used. Priority-based decoding belongs to the category of permutation decoding that does not require to corrective mechanism. So, a chromosome with a permutation of digits from 1 to m+n (total number of producer and consumer) is formed at the early stage. Then, determining the priority for nodes begins from the highest rate (m+n) and reduces when determining the priority of all nodes to be performed. Consequently, the corresponding transportation tree is then generated by sequential arcs appending between sources and depots.

4.3. Genetic algorithm

Based on the role of genetics in nature and the natural evolution of living organisms, Holland (1975) presented a special type of evolutionary algorithms, i.e. genetic algorithms in the early of 70s. Genetic algorithm is a mathematical model that turns a population of chromosomes to the new ones by using Darwin's operational patterns on replication of survival of superior generation based on the natural process of genetic. The general structure of a genetic algorithm can be assumed that first of all a mechanism to convert the answer of each problem to a chromosome should be defined. Then, a set of chromosomes, which are in fact a set of answers the problem, are considered as the initial population. After defining the initial answer, new chromosomes so called the child should be created by using the genetic operation. The operation is divided into two main types of crossover and mutation. As well as, the two concepts of crossover and mutation operators are frequently used for the selection of chromosomes which should play the role of parents, that the operator is also defined. After creating the population of children, the best of chromosomes should be selected by using the evaluation. The selection process is based on the fitness value of each string. In fact, evaluation process is the most important debate on the selection process. Accordingly, after the repetition of several generations, the best generation that is the optimal answer of the problem will be created. Four fundamental steps are mostly used in GA: reproduction, selection mechanism, crossover and mutation.
The structure of the proposed GA is given in Algorithm 1.

Algorithm 1. The proposed *pb*-GA procedure for the FCTP

Step 1: Initialize the problem and GA parameters
    Input: the data instance of the optimization problem and GA parameters;
Step 2: Initialize $P(t)$, by the spanning tree-based or priority-based.
Step 3: Evaluation $P(t)$
Step 4: While (not termination condition) do
    Crossover $P(t)$ to yield $O(t)$ by single point crossover or two-point crossover.
    Mutation $P(t)$ to yield $O(t)$ by scramble or insertion or swap mutation.
    Evaluation $O(t)$.
Step 5: Select $P(t+1)$ from $P(t)$ and $O(t)$ by rank selection mechanism.
Step 6: Check the stop criterion
    while (not termination criterion)
        Repeat step 4 and step 5;

Output: minimum total cost;

4.4. Simulated annealing algorithm

The SA is an optimization technique that has been successfully used for solving a wide range of combinatorial optimization problems. SA was introduced by Kirkpatrick *et al.* (1983). The basic idea of this algorithm was provided by Metropolis for the first time in 1953, according to cooling or refrigeration process of materials in thermodynamics. In the SA the proposed answers to the problem are at a higher temperature and are often inappropriate answers. Then, the variable which has the role of temperature are reduced over the time by increasing the repetitions, so that better answers are formed at low temperatures. Temperature reduction in SA process is like the reduction of objective function value in minimization problems that it is made by a series of improving changes. For allowing the temperature to slowly decline, uphill moves of objective function should also be accepted with a certain probability, so that the possibility is reduced by increasing the frequency (reduction of temperature). This makes the algorithm will not fall into the local optimal trap; therefore, temperature serves as a control parameter in optimization problems. For this purpose, SA is part of the neighborhood search methods which is not related to the initial answer due to the adoption of uphill move answers of objective function, unlike other methods of neighborhood search and can be largely freed from the local optimal trap. The structure of the proposed priority-basedsimulated annealing (*pb*-SA)is given in Algorithm 2.

Algorithm 2. The proposed *pb*-SA procedure for the FCTP

Step 1: Initialize the problem and SA parameters
    Input: the data instance of the optimization problem and SA parameters;
Step 2: Get an initial solution$X_m$by the priority-based.
Step 3: Set an initial temperature,$T > 0$
Step 4: While not frozen do the following:
    Step 4.1: Do the following n times:
        Step 4.1.1: Sample a neighborX'_m from, (i.e., Scramble, Insertion and Swap Mutation)
        Step 4.1.2: Let $Delta = cost\ (X'_m) - cost(X_m)$
        Step 4.1.3: If $Delta < 0$
            then set $X_m = X'_m$
            else set $X_m = X'_m$ with the probability of$\exp(-Delta/T)$
    Step 4.2: Set$T = T \times \alpha$, where (α) is the reduction factor.
Step 5: ReturnX_m
Step 6: Check the stop criterion
    While (not termination criterion)
        Repeat step 4 and step 5;
Output: minimum total cost;

4.5. Keshtel algorithm

In this section, recently developed optimization algorithm inspired by Keshtels' feeding, proposed by Hajiaghaei–Keshteli and Aminnayeri (2014), is used to solve the problem. The Keshtel is a dabbling duck in the *Anas* family. Its scientific name is *Anas Clypeata* and its common name on the north of Iran is the Keshtel.

Socially, these dabbling ducks work together in groups while feeding, rotating like a "pin-wheel", stirring up the surface water and skimming it for food particles. As a Keshtel found a rich food source, its neighbors rush to it and they move together in a swirling way in a circle. They consume the food and swirl around it. If they find a better food source around the place, the circle moves toward the better food source and again consume the food and swirl around it. This process is done iteratively until no food remains around the place. After this co-working, the circle will be dispersed and then each Keshtel will find its way to a different direction of the lake. Again, when a Keshtel find a good food source, it swirls around it and also attracts its neighbors to do the process.

Algorithm 3shows the Keshtel algorithm. Like other metaheuristics, it starts with initial population called Keshtels. They are landed in the lake to find good solutions. Keshtels found better foods in the first time are named lucky ones. They search more for food and attract their neighbors to swirl around them. If better food is found around a lucky Keshtel, the lucky one and its neighbors swirl around the new better food source. After consuming the foods in the area, the lucky one and its neighbors are disintegrated and move in different directions to find another food source. After a while, the lucky Keshtels remain in the lake hopefully to find other food sources. Some ofthe other Keshtels move toward places where no Keshtel exists. Thothers, which have no hope to find food in the lake, fly to another lake or territory. In this time, some new Keshtels land in the lake. In fact, a Keshtel positioning in the lake is counterpart of a solution.The better food source resembles the lower (minimization) cost in the objective function.

Algorithm 3. The proposed *pb*-KA procedure for the FCTP

Step 1: Initialize the problem and KA parameters

Input: the data instance of the optimization problem and KA parameters.

Step 2: Land the Keshtels.

2.1. Generate Initial Population.

$$keshtel = [x_1, x_2, \dots, x_{(i+j)}]$$

Random Keys:	0.23	0.81	0.59	0.38	0.95	0.15	0.93	0.29	0.41
Priority:	2	7	6	4	9	1	8	3	5

2.2. Evaluate Cost of Initial Population.

$$cost = f(keshtel) = f\left(x_1, x_2, \dots, x_{(i+j)}\right)$$

Generate M Keshtels: $M_1 + M_2 + M_3$

Step 3: Find the Lucky Keshtels (LK).

Find the M_1 solutions with better (minimum) objective functions and name them Lucky Keshtels (LK).

Step 4: For each LK:

4.1: Swirl the Nearest Keshtel (NK) around the LK.

Let theM_1 lucky Keshtels to attract their neighbors and swirl around the food source.

Distance of *i*-thKeshtel from the LK is calculated as follows:

$$d_i = \sum_{k \in M} \left| keshtel_k^{Lucky} - keshtel_k^i \right|$$

4.2: If NK finds better food than LK, replace NK with LK, find new NK, go to step 4.1.

$For S < S_{max}$

Swirl the neighbor around the solution according to the S.

$$a = the\ nearest\ neighbor$$
$$b = the\ lucky\ keshtel$$
$$new\ solution_1 = b + b - a$$
$$new\ solution_2 = a + \left(\frac{b-a}{S_{max}}\right)$$
$$new\ solution_3 = (b + b - a) - \left(\frac{(b-a)}{S_{max}}\right)$$
$$new\ solution_4 = b - \left(\frac{(b-a)}{S_{max}}\right)$$
$$new\ solution_5 = b + \left(\frac{(b-a)}{S_{max}}\right)$$

Replace the solution with the better found.

Step 4.3: If the food still exists, attract the NK, go to step, 4.1. if not, go to step 5.

Step 5: Let the LKs remain in the lake.

Step 6: Startle the Keshtels which have found less food and land new ones.

6.1. Replace the worstM_2solutions with the randomly generated new ones.

Step 7: Hustle the remained Keshtels in the lake.

7.1. Move the remained M_3solutions.

Step 8: Evaluate cost of new population (Total cost)

Output: minimum total cost.

5. Proposed Heuristic Algorithm

Integration of the fixed costs with the variable costs, that we are called it the consolidated cost (*cc*), is the basis of theproposed heuristic. This means that we integrate these two types of cost by four methods to achieve the *cc*. Then, we solved the problem using the *cc* like a classical transportation problem and calculate FCTPs objective function after obtaining $x_{ij}$and replace them into the main objective function.

In a nutshell, we propose the heuristic based on the two types of costs exist in the FCTP. We consider them from different points of view in whichthey influence on the objective functions. To explain exactly about the developed heuristic and four methods to achieve the *cc*, we depict the procedure in Algorithm 4, with the following Indexesandparameters.

i	*source index*
j	*depot index*
m	*number of sources*
n	*number of depots*
a_i	*capacity of source i*
b_j	*demand of depot j*
c_{ij}	*adjusted variable transportation cost associated with route (i,j)*
f_{ij}	*adjusted fixed transportation cost associated with route (i,j)*
r_{ij}	*rating of transported product cost from source i to depot j*
tr_{ij}	*total rating of transported product cost for source i and depot j*
fr_{ij}	*final rating of transported product cost from source i to depot j*
ra_i	*supply rating of source i*
rb_j	*demand rating of depot j*
tc_{ij}	*total variable transportation cost for source i and depot j*
tf_{ij}	*total fixed transportation cost associated with route (i,j) for source i and depot j*
rc_{ij}	*rating variable transportation cost for source i and depot j*
rf_{ij}	*rating fixed transportation cost associated with route (i,j) for source i and depot j*
cc_{ij}	*consolidated cost*

Algorithm 4. Heuristic procedure based on (*cc*).

Input: indices, decision variables and parameters
Output: the amount of transported product from source *i* to depot *j* (x_{ij}) and the lowest total cost;
Step 1:
***Method* I:**

Step 1:

$$cc_{ij}^{I} = \frac{(Min(a_i,b_j)\times c_{ij})+f_{ij}}{Min(a_i,b_j)}, \text{ for each } i,j;$$

***Method* II:**

Step 1:

$$r_{\min(i,j)} = c_{ij} + f_{ij}, \text{ for each } i,j;$$
$$r_{\text{ava}(i,j)} = \left(c_{ij} \times \min(a_i, b_j)\right) + f_{ij}, \text{ for each } i,j;$$
$$r_{\max(i,j)} = \left(c_{ij} \times \max(a_i, b_j)\right) + f_{ij}, \text{ for each } i,j;$$

Step 2:

$$tr_{\min(i,j)} = \sum_{\substack{j=1 \\ i=1,..,m}}^{n} r_{\min(i,j)} + \sum_{\substack{i=1 \\ j=1,..,n}}^{m} r_{\min(i,j)}$$

$$tr_{\text{ava}(i,j)} = \sum_{\substack{j=1 \\ i=1,..,m}}^{n} r_{\text{ava}(i,j)} + \sum_{\substack{i=1 \\ j=1,..,n}}^{m} r_{\text{ava}(i,j)}$$

$$tr_{\max(i,j)} = \sum_{\substack{j=1 \\ i=1,..,m}}^{n} r_{\max(i,j)} + \sum_{\substack{i=1 \\ j=1,..,n}}^{m} r_{\max(i,j)}$$

Step 3:

$$fr_{\min(i,j)} = \frac{tr_{\min(i,j)}}{r_{\min(i,j)}}, \text{ for each } i,j; \qquad fr_{\text{ava}(i,j)} = \frac{tr_{\text{ava}(i,j)}}{r_{\text{ava}(i,j)}}, \text{ for each } i,j; \qquad fr_{\max(i,j)} = \frac{tr_{\max(i,j)}}{r_{\max(i,j)}}, \text{ for each } i,j;$$

Step 4:

$$cc_{ij}^{II} = \frac{fr_{\min(i,j)}}{fr_{\max(i,j)}+\left(fr_{\text{ava}(i,j)}\times fr_{\min(i,j)}\right)}, \text{ for each } i,j;$$

***Method* III:**

Step 1:

$$ra_i = \frac{\sum_{i=1}^{m} a_i + \sum_{j=1}^{n} b_j}{a_i} \qquad rb_i = \frac{\sum_{i=1}^{m} a_i + \sum_{j=1}^{n} b_j}{b_j}$$

Step 2:

$$tc_{ij} = \sum_{\substack{j=1 \\ i=1,\dots,m}}^{n} C(i,j) + \sum_{\substack{i=1 \\ j=1,\dots,n}}^{m} C(i,j) \qquad tf_{ij} = \sum_{\substack{j=1 \\ i=1,\dots,m}}^{n} f(i,j) + \sum_{\substack{i=1 \\ j=1,\dots,n}}^{m} f(i,j)$$

Step 3:

$$rc_{ij} = \frac{tc_{ij}}{c_{ij}}, \text{ for each } i,j; \qquad rf_{ij} = \frac{tf_{ij}}{f_{ij}}, \text{ for each } i,j;$$

Step 4:

$$cc_{ij}^{III} = \frac{\min(ra_i, rb_i)}{(rc_{ij} \times \min(ra_i, rb_i)) + rf_{ij}}, \text{ for each } i,j;$$

***Method* IV:**

Step 1:

$$cc_{ij}^{IV} = \left(c_{ij} \times min(a_i, b_j)\right) + f_{ij}, \text{ for each } i,j;$$

Step 2:
Solving like the classical transportation problem (TP) using each consolidated costs ($cc_{ij}^{I}, cc_{ij}^{II}, cc_{ij}^{III}$ and cc_{ij}^{IV});
Step 3: obtaining amount of x_{ij} from each consolidated costs in step 3;
Step 4: calculation of the total cost using$(c_{ij} \times x_{ij}) + (f_{ij} \times y_{ij})$;
Step 5: choosing the lowest total cost from step 4;

6. Experimental Design

Taguchi (1986) presented a new approach to the design of experiments. Taguchi, as the first provider of parameter design method, proposed an engineering approach to design a product or process that aims to minimize the changes and sensitivity of disturbance factors. The first goal of an efficient parameter design is to identify and set factors that minimize the changes of answers and the next goal is to identify controllable and uncontrollable factors. The ultimate goal of this method is to find the optimal combination of controllable factors. Taguchi has created special set of overall designs for factorial experiments that cover most applications. Orthogonal arrays are part of the set designs. The use of these arrays helps us in determining the minimum number of needed experiments for a series of factors.

6.1. Instances

To cover various types of problems, we considered several levels of influencing inputs. At first, we generated random problem instances for m = 5, 10, 15, 20, 30, and 50 suppliers and n = 10, 15, 20, 30, 50, 100, and 200 customers, respectively. We consider the instances from Hajiaghaei-Keshteli *et al.* (2010) and develop nine instances in three sizes small, medium and large sizes. After specifying the size of problems in a given instance, considering the significant influence of the fixed costs to the solution for each size, four problem types (A–D) are employed. For a given problem size, problem types differ from each other by the range of fixed costs. There are$9 \times 4 = 36$instances, in which the variable costs have discrete values from 3 to 8 and the fixed costs arise from type A to type D. The problem sizes, types, suppliers/customers, and fixed costs ranges are shown in Table 1.

Table 1
FCTPs test problems characteristics.

Problem size		Total supply	Problem type	Range of variable costs		Range of fixed costs	
				Lower limit	Upper limit	Lower limit	Upper limit
Small	5×10	5000	A	3	8	50	200
	10×10	8000	B	3	8	100	400
	10×20	10000	C	3	8	200	800
Medium	15×15	15000	D	3	8	400	1600
	10×30	15000					
	20×30	20000					
Large	50×50	50000					
	30×100	30000					
	50×200	50000					

6.2. Parameter setting

In each algorithm we face several parameters and each of them should be assigned by a discrete or continuous value. In the other hand, we know that the correct choice of the parameters strongly affect on the performance of an algorithm. So, in this section, we study the performance of the algorithm in dealing with different parameters. In order to examine the performance of the presented algorithms thirty-six test problems with various sizes are solved. We implement the algorithms in Matlab 14a, and run on a PC with 2.5 GHz Intel Core i5-3210M CPU and 4GB of RAM memory. The experiments on the SA and KA were based on the L_9 orthogonal array. Also in order to achieve the more reliable results five replications were done for each trial. In addition, since we deal various scales of objective functions in each instance, we utilize the relative percentage deviation (RPD) according to formula (6) for each instance.Using the average of RPD measures of trials, the parameters and operators that have minimum RPD average are selected as the best ones.

$$RPD=\frac{Alg_{sol}\text{-}Min_{sol}}{Min_{sol}}\times 100 \qquad (6)$$

Where Alg_{sol} and Min_{sol} are the obtained 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. We also transform the mean RPDs into the S/N ratios. The S/N ratios of trials are averaged in each level. In the Taguchi method, the S/N ratio of the minimization objectives is as such formula (7).The aim is tomaximize the signal-to-noise ratio.

$$S/N\ ratio = -10\log_{10}(objective\ function)^2 \qquad (7)$$

In accordance with the Table 2, 3 and 4 all investigated and effective parameters in algorithms and all its modes are listed.

Table 2
Factors and their levels in GA algorithm.

Factors	GA symbols	GA Levels
Type of crossover	A	A (1) – one-point crossover A (2) – two-point crossover
Type of mutation	B	B (1) – Swap B (2) – Scramble B (3) – Insertion
Population size	C	C (1) – 20 C (2) – 30 C (3) – 40
Crossover percentage	D	D (1) – 60% D (2) – 70% D (3) – 80%
Mutation probability	E	E (1) – 0.05 E (2) – 0.1 E (3) – 0.15

Table 3
Factors and their levels in SA algorithm.

Factors	SA symbols	SA Levels
Initial temperature	A	A (1) – 4000 A (2) – 4500 A (3) – 5000
Alpha	B	B (1) – 0.93 B (2) – 0.95 B (3) – 0.97
Type of mutation	C	C (1) – Swap C (2) – Scramble C (3) – Insertion

Table 4
Factors and their levels in KA algorithm.

Factors	KA symbols	KA Levels
Neighbor Keshtel	A	A (1) – 10 A (2) – 20 A (3) – 30
M_1	B	B (1) – 0.4 B (2) – 0.5 B (3) – 0.6
M_2	C	C (1) – 0.2 C (2) – 0.3 C (3) – 0.4
S_{max}	D	D (1) – 10 D (2) – 15 D (3) – 20

Using an $L18$ orthogonal array there are totally 18 experiments to be conducted. Therefore, the parameters of GA were set as follows: $Type\ of\ crossover = two-point\ crossover, Type\ of\ mutation = Scramble, Population\ size = 30, Crossover\ percentage = 70\%\ and\ Mutation\ probability = 0.1$.Using an $L9$ orthogonal arraythere are totally 9 experiments to be conducted. Therefore, the parameters of SA were set as follows:$Initial\ temperature = 4500, Alpha = 0.95$, Type of $Mutation = Scramble$ and also the parameters of KA were set as follows: $Neighbor\ Keshtel = 20$, $M_1 = 0.5$, $M_2 = 0.3$, $S_{max} = 15$.

6.3. Data generation

To evaluate the performance of our proposed heuristic and metaheuristics for solving the given problem, a plan is utilized to generate the test data. The data required for the problem include the number of plants and customers. As mentioned earlier, not only the test data are used from Hajiaghaei-Keshteli *et al.*(2010) but also we added two new different problem sizes to seven previous problems. Thus, nine different problem sizes in three levels small, medium and large are considered for experimental study, which present different levels of difficulty for alternative solution approaches as shown in Table 1.

6.4. Numerical experiments

To perform the computational experiment nine test problems in three levels small, medium and large with different values of m and n are randomly generated for every four problem types (A–D). All the test problems are solved by the proposed heuristic and metaheuristics and LINGO software. We implement the proposed heuristic and metaheuristics in Matlab 14a, and run on a PC with 2.5 GHz Intel Core i5-3210M CPU and 4GB of RAM memory. The LINGO software is run to obtain Local or Global optimal solution for the small, medium and large problems in 600, 1200 and 1800 seconds, respectively. We set searching time to be identical for both metaheuristics, which is equal to 0.001×m×n seconds. Hence, this criterion is affected by both m and n. using this stopping criterion, searching time increases according to the rise of either number of plants or number of customers. Due to having stochastic nature of proposed metaheuristics, 30 replications were performed for each trial to achieve the more reliable results. In order to compare the related results, the average cost in terms of the solution quality is considered.
Table 5 show computational results (quality and time) obtained from the suggested solution methods for test sample. In each of these Tables, 36 problems in four different types of fixed charge values have been classified. Column of objective function (OF) in metaheuristic algorithms due to random nature of these methods represents the average value of objective function from 30 times of the algorithm implementation; and in heuristic algorithm due to the certainty of solutions, it represents the value of objective function

from one time of the implementation of these methods. Gap column also represents the percentage of deviations from the best solution. The best solution for each size of the test problem is the best objective function that it has been achieved by the all proposed solution methods.

Table 5
Computational results of the proposed metaheuristics and heuristic for the FCTP
OF= Objective function; L= Local; G= Global; Gap=Percentage deviation from best solutions $(z - z^*/z^*) \times 100$

Type	Size	Metaheuristics Priority based									Heuristic				
		GA			SA			KA			Based on consolidated cost			Lingo Optimal	
		OF	Gap	Time (s)	OF	Gap	Time (s)	OF	Gap	Time (s)	OF	Gap	Time (s)	OF	Gap
A	5×10	21935.8	0.58	0.05	22299.1	2.24	0.05	21873.1	0.29	0.05	21850	0.18	0.060	21810 G	0.00
	10×10	28813.5	1.45	0.10	29525.35	3.96	0.10	28908.6	1.79	0.10	28435	0.12	0.066	28401 G	0.00
	10×20	36951.9	4.47	0.20	37800.75	6.87	0.20	37079.9	4.83	0.20	35558	0.53	0.068	35372 G	0.00
	15×15	52376.3	4.85	0.23	53709.6	7.52	0.23	52133.1	4.36	0.23	50030	0.15	0.075	49955 G	0.00
	10×30	52594.5	3.47	0.30	53229.95	4.72	0.30	52323	2.94	0.30	50956	0.25	0.077	50830 G	0.00
	20×30	67897.9	4.03	0.60	68695.4	5.25	0.60	68887.6	5.54	0.60	65676	0.62	0.086	65270 L	0.00
	50×50	162212.6	2.22	2.50	164213.3	3.48	2.50	163422.1	2.99	2.50	158684	0.00	0.149	158856 L	0.11
	30×100	104571.4	2.31	3.00	105763.1	3.48	3.00	105496.1	3.22	3.00	102260	0.05	0.183	102207 L	0.00
	50×200	171823	1.97	10.00	172999.5	2.67	10.00	172543.3	2.40	10.00	168496	0.00	0.457	173151 L	2.76
B	5×10	24434.1	0.35	0.05	24990.25	2.64	0.05	24476.15	0.53	0.05	24725	1.55	0.041	24348 G	0.00
	10×10	31304	0.93	0.10	31940.55	2.98	0.10	31302.85	0.92	0.10	31333	1.02	0.058	31017 G	0.00
	10×20	40644.5	1.97	0.20	40926.8	2.68	0.20	40547.95	1.73	0.20	40252	0.99	0.066	39858 L	0.00
	15×15	60148	2.35	0.23	60689.8	3.27	0.23	60330.05	2.66	0.23	58932	0.28	0.073	58766 L	0.00
	10×30	62441.5	5.19	0.30	63150.85	6.38	0.30	62835.2	5.85	0.30	59362	0.00	0.081	60445 L	1.82
	20×30	72331.4	2.25	0.60	73034.1	3.24	0.60	72478.05	2.46	0.60	70759	0.03	0.098	70740 L	0.00
	50×50	172111.3	2.90	2.50	173574.8	3.78	2.50	172359.3	3.05	2.50	167260	0.00	0.141	167981 L	0.43
	30×100	115098.5	2.65	3.00	116308.6	3.73	3.00	115591.3	3.09	3.00	112289	0.15	0.178	112122 L	0.00
	50×200	192000.9	3.01	10.00	194021	4.09	10.00	192974.2	3.53	10.00	186393	0.00	0.497	192867 L	3.47
C	5×10	25401	0.42	0.05	26025.05	2.88	0.05	25418.35	0.48	0.05	25338	0.17	0.051	25296 G	0.00
	10×10	37544	1.82	0.10	38093.65	3.31	0.10	37584.2	1.93	0.10	37844	2.63	0.060	36873 G	0.00
	10×20	45883.7	2.77	0.20	46570.35	4.31	0.20	45874.25	2.75	0.20	45443	1.79	0.067	44645 L	0.00
	15×15	60685	3.34	0.23	61285.05	4.36	0.23	60333.05	2.74	0.23	59319	1.01	0.073	58725 L	0.00
	10×30	64674.7	2.24	0.30	65375.85	3.35	0.30	64917.25	2.62	0.30	64159	1.42	0.079	63258 L	0.00
	20×30	81147.1	3.72	0.60	82099.25	4.94	0.60	81213.15	3.80	0.60	79835	2.04	0.086	78237 L	0.00
	50×50	187934.3	2.09	2.50	189080.1	2.71	2.50	188735.7	2.53	2.50	184083	0.00	0.168	185074 L	0.54
	30×100	135643.6	2.92	3.00	136937.2	3.90	3.00	136038.5	3.22	3.00	131800	0.00	0.193	132153 L	0.27
	50×200	227418.2	2.86	10.00	229242.5	3.68	10.00	228110.2	3.17	10.00	221096	0.00	0.422	233710 L	5.71
D	5×10	30313.8	0.69	0.05	31436.3	4.42	0.05	30392.7	0.95	0.05	30836	2.42	0.052	30107 G	0.00
	10×10	40150.4	0.98	0.10	41224.75	3.68	0.10	40128.25	0.93	0.10	41013	3.15	0.060	39760 G	0.00
	10×20	62000.2	2.88	0.20	62720.35	4.07	0.20	62066.75	2.99	0.20	63133	4.76	0.066	60267 L	0.00
	15×15	76503.5	3.50	0.23	77622.4	5.02	0.23	76764.7	3.86	0.23	75829	2.59	0.082	73913 L	0.00
	10×30	82872	2.35	0.30	84249.8	4.05	0.30	83175	2.72	0.30	82705	2.14	0.083	80971 L	0.00
	20×30	100622.2	1.65	0.60	102059.3	3.10	0.60	101144.6	2.18	0.60	102203	3.25	0.092	99204 L	0.22
	50×50	218694.7	1.34	2.50	220083	1.99	2.50	219398.2	1.67	2.50	215793	0.00	0.173	218209 L	1.12
	30×100	174298.8	2.21	3.00	177568.4	4.13	3.00	176040.5	3.23	3.00	170533	0.00	0.222	171059 L	0.31
	50×200	298771.8	4.05	10.00	303272.1	5.62	10.00	301846.3	5.12	10.00	287137	0.00	0.472	308963 L	7.60

In order to compare the results of suggested solution methods, the average gap for each test sample sizes obtained has been shown in Figure2. As seen in the figure, our proposed heuristic gives the best results.

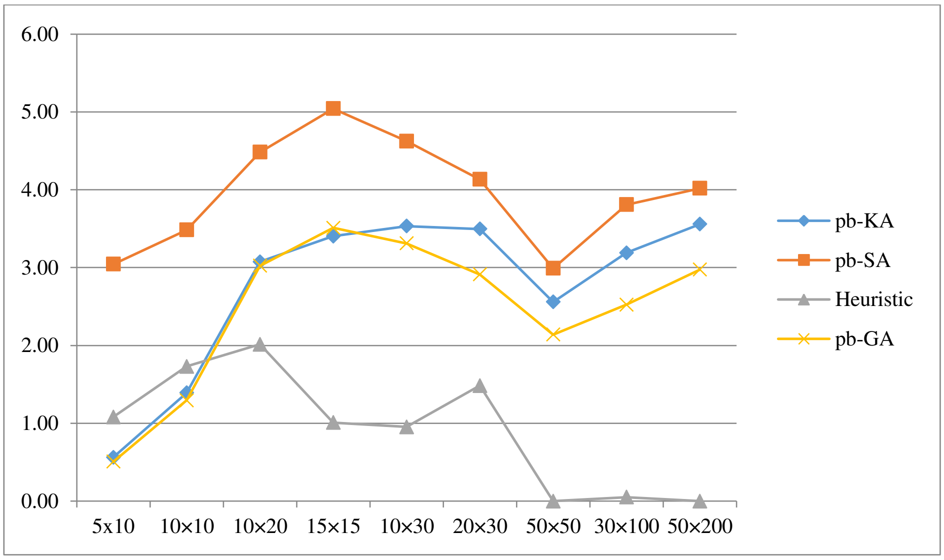

Fig. 2. Comparison of the mean of optimal solution gap for different problem sizes.

In order to compare the overall performance of the suggested solutions, we obtain the average gap among 9 different sizes of the problem for each of the four types of samples tested. In this way, we will have 4 average gaps according to the Tables 6 for each of the methods of solving. Finally, the overall performance of each solution methods is also determined by calculating the average values. From this Table, we can conclude that the proposed heuristic based on consolidated cost has better performance than the other solution methods.

Table 6
Average gap among for each of the four types of test problem.

Type	*Pb-GA*	*pb*-KA	*pb*-SA	Heuristic
A	2.82	3.15	4.47	0.21
B	2.40	2.65	3.64	0.45
C	2.46	2.58	3.72	1.01
D	2.18	2.63	4.01	2.03
ave	2.47	2.75	3.96	0.92

In order to more study about the results obtained, analysis of variance (*ANOVA*) was used. The statistical results show that there is a significant difference in the performance of different solution methods. Figure 3shows the average chart and least significant difference (*LSD*) for the proposed solution methods at 95% confidence level. As can be seen from this Figure, it can be concluded the proposed heuristic based on consolidated cost has better performance than the other solution methods.

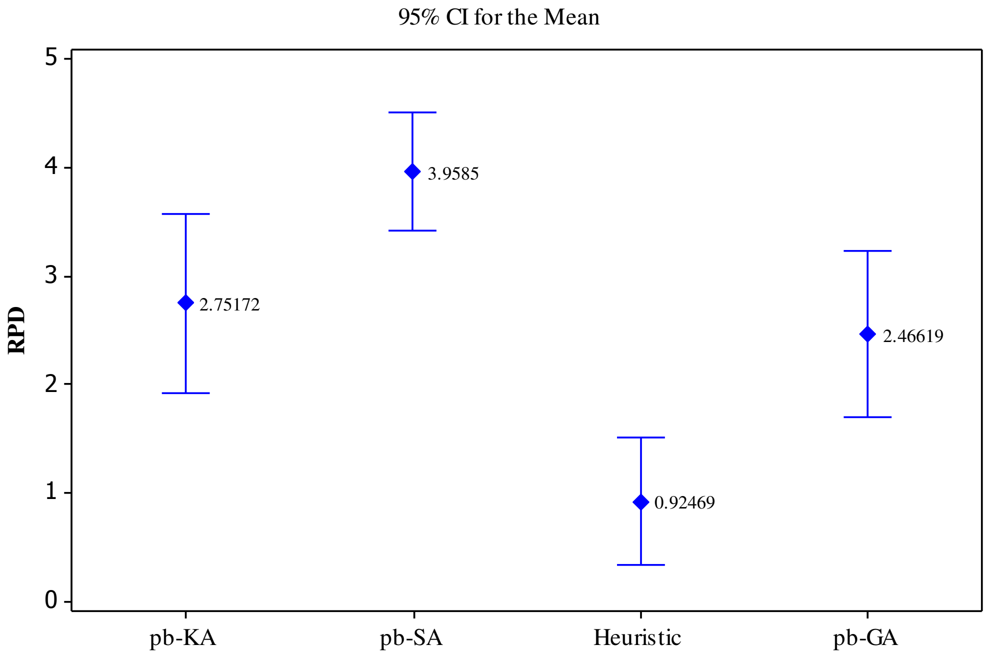

Fig. 3. Means plot and LSD intervals for the proposed solution methods

7. Conclusion and Future Works

Fixed charge transportation problem is one of the non-polynomial difficult problems that its solution by old methods is difficult. A classification for fixed charge transportation problems and its solution methods in the literature is provided. This study not only considered GA, SA and KA as the tools to evolve optimal or near-optimal solution but also proposed novel heuristic based on consolidated cost to solve the problem. In the following, priority-based representation and solution mechanism of GA, SA and KA as well as heuristic algorithm were examined. Besides, Taguchi experimental design was used to adjust the parameters of the proposed metaheuristic algorithms and better performance of them. Finally, by providing several experimental tries, performance of solution methods in different models was reviewed and evaluated.

According to the achieved results, we can say that the *pb*-GA generates better solutions than the *pb*-KA and *pb*-SA. Also numerical experiments showed that our proposed heuristic base on consolidated cost gives better results than the metaheuristics; both in terms of the solution quality and computation time, and also especially for medium and large sized problems.

To propose research directions for future works, at first, we recommend using our novel heuristic in other types of TP and FCTP. Developing the heuristic and utilizing recent and strong metaheuristics are also recommended in this research area.

References

Adlakha, V., & Kowalski, K. (2003). A simple heuristic for solving small fixed-charge transportation problems. *Omega*, *31*(3), 205-211.

Aguado, J. S. (2009). Fixed Charge Transportation Problems: a new heuristic approach based on Lagrangean relaxation and the solving of core problems. *Annals of Operations Research*, *172*(1), 45.

Balinski, M. L. (1961). Fixed‐cost transportation problems. *Naval Research Logistics (NRL)*, *8*(1), 41-54.

Buson, E., Roberti, R., & Toth, P. (2014). A reduced-cost iterated local search heuristic for the fixed-charge transportation problem. *Operations Research*, *62*(5), 1095-1106.

Dwyer, P. S. (1966). Use of completely reduced matrices in solving transportation problems with fixed charges. *Naval Research Logistics Quarterly*, *13*(3), 289-313.

El-Sherbiny, M. M., & Alhamali, R. M. (2013). A hybrid particle swarm algorithm with artificial immune learning for solving the fixed charge transportation problem. *Computers & Industrial Engineering*, *64*(2), 610-620.

Gen, M., & Cheng, R. (1997). Foundations of genetic algorithms. *Genetic Algorithms and Engineering Design*, 1-41.

Gen, M., & Cheng, R. (2000). *Genetic algorithms and engineering optimization* (Vol. 7). John Wiley & Sons.

Gen, M., & Li, Y. (1999). Spanning tree-based genetic algorithm for the bicriteria fixed charge transportation problem. In *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on* (Vol. 3, pp. 2265-2271). IEEE.

Gen, M., Altiparmak, F., & Lin, L. (2006). A genetic algorithm for two-stage transportation problem using priority-based encoding. *OR spectrum*, *28*(3), 337-354.

Glover, F. (2005). Parametric ghost image processes for fixed-charge problems: A study of transportation networks. *Journal of Heuristics*, *11*(4), 307-336.

Guignard, M. (1988). A Lagrangean dual ascent algorithm for simple plant location problems. *European Journal of Operational Research*, *35*(2), 193-200.

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., & 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., 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.

Hirsch, Warren M., and George B. Dantzig.(1968). "The fixed charge problem. *Naval Research Logistics Quarterly, 15*(3), 413-424**.**

Holland, J. H. (1975). Adaptation in natural and artificial systems. An introductory analysis with application to biology, control, and artificial intelligence. Ann Arbor, MI: University of Michigan Press.

Hwang, R. K., Katayama, H., & Gen, M. (2008). U-shaped assembly line balancing problem with genetic algorithm. *International Journal of Production Research*, *46*(16), 4637-4649.

Jawahar, N., Gunasekaran, A., & Balaji, N. (2012). A simulated annealing algorithm to the multi-period fixed charge distribution problem associated with backorder and inventory. *International Journal of Production Research*, *50*(9), 2533-2554.

Jo, J. B., Li, Y., & Gen, M. (2007). Nonlinear fixed charge transportation problem by spanning tree-based genetic algorithm. *Computers & Industrial Engineering*, *53*(2), 290-298.

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

Klose, A. (2008). Algorithms for solving the single-sink fixed-charge transportation problem. *Computers & Operations Research*, *35*(6), 2079-2092.

Lee, J. E., Gen, M., & Rhee, K. G. (2009). Hybrid priority-based genetic algorithm for multi-stage reverse logistics network. *Industrial Engineering and Management Systems*, *8*(1), 14-21.

Loch, G. V., & da Silva, A. C. L. (2014). A computational experiment in a heuristic for the Fixed Charge Transportation Problem. *International Refereed Journal of Engineering and Science*, *3*(4).

Loch, G. V., & da Silva, A. C. L. (2014). A computational experiment in a heuristic for the Fixed Charge Transportation Problem. *International Refereed Journal of Engineering and Science*, *3*(4).

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.

Mahmoodi-Rad, A., Molla-Alizadeh-Zavardehi, S., Dehghan, R., Sanei, M., & Niroomand, S. (2014). Genetic and differential evolution algorithms for the allocation of customers to potential distribution centers in a fuzzy environment. *The International Journal of Advanced Manufacturing Technology*, *70*(9-12), 1939-1954.

Molla-Alizadeh-Zavardehi, S., Hajiaghaei-Keshteli, M., & Tavakkoli-Moghaddam, R. (2011). Solving a capacitated fixed-charge transportation problem by artificial immune and genetic algorithms with a Prüfer number representation. *Expert Systems with Applications*, *38*(8), 10462-10474.

Shetty, B. (1990). A relaxation/decomposition algorithm for the fixed charged network problem. *Naval Research Logistics (NRL)*, *37*(2), 327-340.

Steinberg, D. I. (1970). The fixed charge problem. *Naval Research Logistics Quarterly, 17*(2), 217-235.

Sun, M., & McKeown, P. G. (1993). Tabu search applied to the general fixed charge problem. *Annals of Operations Research, 41*(4), 405-420.

Sun, M., Aronson, J. E., McKeown, P. G., & Drinka, D. (1998). A tabu search heuristic procedure for the fixed charge transportation problem.*European Journal of Operational Research, 106*(2), 441-456.

Sun, M., Aronson, J. E., McKeown, P. G., & Drinka, D. (1998). A tabu search heuristic procedure for the fixed charge transportation problem. *European Journal of Operational Research*, *106*(2-3), 441-456.

Taguchi, G. (1986). *Introduction to quality engineering: designing quality into products and processes*.

Tari, F. G., & Hashemi, Z. (2016). A priority based genetic algorithm for nonlinear transportation costs problems. *Computers & Industrial Engineering*, *96*, 86-95.

Walker, W. E. (1976). A heuristic adjacent extreme point algorithm for the fixed charge problem. *Management Science, 22*(5), 587-596.

Wright DD, Haehling von Lanzenauer C (1989) Solving the fixed charge problem with Lagrangian relaxation and cost allocation heuristics. *Eur. J. Oper. Res. 42*(3):305–312.

Wright, D., & von Lanzenauer, C. H. (1991). COAL: A new heuristic approach for solving the fixed charge problem—computational results. *European Journal of Operational Research, 52*(2), 235-246.

Xie, Fanrong, and Renan Jia. "Nonlinear fixed charge transportation problem by minimum cost flow-based genetic algorithm." Computers & Industrial Engineering 63, no. 4 (2012): 763-778.

Yadegari, E., Zandieh, M., & Najmi, H. (2015). A hybrid spanning tree-based genetic/simulated annealing algorithm for a closed-loop logistics network design problem. *International Journal of Applied Decision Sciences*, *8*(4), 400-426.

Yaghini, M., Momeni, M., & Sarmadi, M. (2012). A simplex-based simulated annealing algorithm for node-arc capacitated multicommodity network design. *Applied Soft Computing*, *12*(9), 2997-3003.

This article can be cited: Yousefi K Afshari, A.J. & Hajiaghaei-Keshteli, M. (2019). Solving the Fixed Charge Transportation Problem by New Heuristic Approach. *Journal of Optimization in Industrial Engineering*. 12 (1), *41- 52*.

http://www.qjie.ir/article_538020.html
DOI: 10.22094/JOIE.2017.738.1469