Annals of Optimization Theory and Practice

Volume 2, Number 1, 1-9
April 2019

DOI: 10.22121/aotp.2019.172309.1018

Gravitational search algorithm for step fixed charge transportation problems

Rohollah Taghaodi

Department of Mathematics, Kashan Branch, Islamic Azad University, Kashan, Iran

✉ taghaode@yahoo.com

(Received: December 16, 2018 / Accepted: February 28, 2019)

Abstract Step fixed-charge transportation problem is an extended version of the fixed charge transportation problem, is one of the most important problems in transportation research area. To tackle such an NP-hard problem, we present Gravitational Search Algorithm (GSA). We solve the randomly generated problems by GSA and also with Genetic Algorithm (GA) to compare them. The obtained results show the proficiency of GSA comparison with GA.

Keywords Step fixed-charge transportation problem; NP-hard problem; Gravitational search algorithm; Genetic algorithm

1. Introduction

Fixed-Charge Transportation Problem (FCTP) is a special version of the transportation problem. In FCTP, each route is associated with a fixed cost and a transportation cost per unit shipped Sanei *et al.* (2014) and Baranifar (2018). Step Fixed Charge Transportation Problem (SFCTP) is an extended version of the FCTP. The SFCTP, for the first time, introduced by Kowalski and Lev (2008) and has received little attention in the transportation problem literature. In SFCTP the fixed charge is incurred for each route that is used in the solution, along with the variable cost that is proportional to the amount shipped. Also, this cost structure causes the value of the objective function to behave like a step function. In the case of the SFCTP due to the step function structure of the objective function, Kowalski and Lev (2008) were dealing with a NP- hard problem. Since there was no algorithm for the SFCTP then, they tried to suggest two heuristic methods which provide a good solution as a useful method, by extending the method proposed by Balinski (1961).

Since the problems with fixed charges are usually NP-hard, the computational time to obtain exact solutions increases in a polynomial fashion and very quickly becomes

Published by Firouzabad Institute of Higher Education, Firouzabad, Fars, Iran

extremely long as the dimensions of the problem increase Kowalski and Lev (2008) and Mosallaeipour *et al.* (2018) and Taghaodi and Kardani (2018).

Altassan *et al.* (2012), in addition to, first, being in line with them tried to suggest three other formulae, and then, tried to compare the performance of the new formulae with the earlier proposed formulae. El-Sherbiny (2012) proposed the alternate mutation based artificial immune algorithm for SFCTP. They claimed that their algorithm solve both balanced and unbalanced SFCTP without introducing a dummy supplier or a dummy customer. Mahmoodirad *et al.* (2013) focus on a technique which obtains a good solution of the SFCTP, where both the fixed cost and the unit transportation cost from each origin to each destination, have been expressed as generalized trapezoidal fuzzy numbers. To solve the problem, they convert this problem into the fuzzy transportation problem, and then, try to construct a fuzzy coefficient matrix to finding a good solution for SFCTP, by developing the earlier proposed formulae. Rajabi *et al.* (2013) formulated the SFCTP under uncertainty, particularly when variable and fixed costs are given in fuzzy forms. In order to solve the problem, they developed two meta-heuristic algorithms, namely, simulated annealing algorithm and variable neighborhood search for this NP-hard problem. Molla-Alizadeh-Zavardehi *et al.* (2014) developed Genetic Algorithm (GA) for the SFCTP and compared it with simulated annealing. Molla-Alizadeh-Zavardehi *et al.* (2014) have developed a spanning tree- based GA and a spanning tree- based Memetic Algorithm (MA) to solve the SFCTP. In order to evaluate the efficiency of developed algorithms, a new plan is extended based on previous test problems to generate random instances. The comprehensive set of computational experiments for instances with different configuration and problem sizes show that the MA provides good average relative percentage deviation results and outperforms the GA.

In this paper, we consider the SFCTP and developed the meta-heuristic algorithm, Gravitational search algorithm (GSA), for solve it and compared with GA. Up until now, no one has considered neither GSA for any kind of SFCTPs. So, we presented GSA for solving the SFCTP for the first time.

The rest of the paper is organized as follows: in Section 1, the SFCTP model is described. Then, the meta-heuristics algorithm, GSA, is developed. Later, experimental design is presented. In the next section, results and discussion is provided. Finally, conclusions are pointed out in the last section.

2. Problem Formulation

Consider a transportation problem with m sources and n destinations. Each of the source i=1,2,…,m has units of supply, and each destination j=1,2,…,n has a demand of units and also, each of the m source can ship to any of the n destinations at a shipping cost per unit plus a fixed cost assumed for opening this route (i,j). Let denote the number of units to be shipped from source i to destination j. We need to determine which routes are to be opened and the size of the shipment on those routes, so that the total cost of meeting demand, given the supply constraints, is minimized. Then, the mixed integer programming formulation for the SFCTP is well known Kowalski and Lev (2008):

$$\begin{aligned}
& MinZ=\sum_{i=1}^{m}\sum_{j=1}^{n}(c_{ij}x_{ij}+g_{ij}\,y_{ij}) \\
& \text{subject to} \\
& \sum_{j=1}^{n}x_{ij}=S_i && i=1,2,\ldots,m \\
& \sum_{i=1}^{m}x_{ij}=D_j && j=1,2,\ldots,n \qquad (1)\\
& x_{ij}\geq 0, && i=1,2,\ldots,m \;\; j=1,2,\ldots,n \\
& y_{ij}=\begin{cases}1 & x_{ij}>0\\ 0 & otherwise\end{cases} && i=1,2,\ldots,m \;\; j=1,2,\ldots,n
\end{aligned}$$

The fixed cost for route (i, j) is proportional to the transported amount through its route. This consists of a fixed cost $k_{ij,1}$ for opening the route (i,j) and an additional cost $k_{ij,2}$ when the transported units exceeds a certain amount A_{ij}. Thus,

$$g_{ij}=b_{ij,1}k_{ij,1}+b_{ij,2}k_{ij,2} \qquad i=1,2,\ldots,m \;\; j=1,2,\ldots,n \qquad (2)$$

where

$$b_{ij,1}=\begin{cases}1 & x_{ij}>0\\ 0 & otherwise\end{cases} \qquad i=1,2,\ldots,m \;\; j=1,2,\ldots,n \qquad (3)$$

$$b_{ij,2}=\begin{cases}1 & x_{ij}>A_{ij}\\ 0 & otherwise\end{cases} \qquad i=1,2,\ldots,m \;\; j=1,2,\ldots,n \qquad (4)$$

and $k_{ij,1}, k_{ij,2}, g_{ij}, A_{ij}$ are nonnegative real numbers. Also, we assume that $\sum_{i=1}^{m} S_i = \sum_{j=1}^{n} D_j$ that is, it be balanced, if it be unbalanced, can by introducing a dummy source or a dummy destination be converted to a balanced transportation problem. Note that, if all $A_{ij} > \min\{S_i, D_j\}$, then the SFCTP becomes a FCTP with a single fixed cost $k_{ij,1}$ Also, in the model (1), g_{ij} have two steps. It could have multiple steps, depending on the problem structure. Despite its similarity to a standard transportation problem, the SFCTP is significantly harder to solve because of the discontinuity in the objective function Z introduced by the fixed costs Kowalski and Lev (2008).

3. Meta-heuristic Algorithm

The use of conventional tools for solving mathematical programming models is limited due to the complexity of the problem and the large number of variables and constraints, particularly for realistically sized problems. Regarded as the time complexity function and a class of combinational optimization problems known as nondeterministic polynomial-time hard (NP-hard), we propose met-heuristic algorithm to solve the SFCTP.

3.1 Initialization

Most of the meta-heuristics use a random procedure to make an initial set of solutions. Each generated solution is considered as an individual solution to the problem. In the first-generation solution are generated as many as population size. The random method is applied for generating the initial population.

3.2 Gravitational search algorithm

The GSA is a newly developed stochastic search algorithm based on the law of gravity and mass interactions Rashedi *et al.* (2009) and Mahmoodirad and Sanei (2016). In this approach, the search agents are a collection of masses which interact with each other based on the Newtonian gravity and the laws of motion, in which the method is completely different from other well-known population-based optimization method inspired by the swarm behaviors. In GSA, agents are considered as objects and their performance are measured by their masses. All of the objects attract each other by the gravity force, while this force causes a global movement of all objects towards the objects with heavier masses Rashedi *et al.* (2009). The heavy masses correspond to good solutions of the problem. In other words, each mass presents a solution, and the algorithm is navigated by properly adjusting the gravitational and inertia masses. By lapse of time, the masses will be attracted by the heaviest mass which it presents an optimum solution in the search space.

To describe the GSA, consider a system with N agents (masses), the position of the agent i is defined by:

$$X_i = (x_i^1, \ldots, x_i^d, \ldots, x_i^n) \qquad i = 1, 2, \ldots, N \tag{5}$$

where x_i^d presents the position of the agent i in the dimension d and n is the search space dimension. After evaluating the current population fitness, the mass of each agent is calculated as follows:

$$\mathrm{M}_i(t) = \frac{m_i(t)}{\sum_{j=1}^{N} m_j(t)} \tag{6}$$

where

$$m_i(t) = \frac{fit_i(t) - worst(t)}{best(t) - worst(t)} \tag{7}$$

where $fit_i(t)$ represent the fitness value of the agent i at time t. best (t) and worst (t) are the best and worst fitness of all agents, respectively and defined as follows:

$$best(t) = \min_{j \in 1,2,\ldots,N} fit_j(t) \tag{8}$$

$$worst(t) = \max_{j \in 1,2,\ldots,N} fit_j(t) \tag{9}$$

Published by Firouzabad Institute of Higher Education, Firouzabad, Fars, Iran

To evaluate the acceleration of an agent, total forces from a set of heavier masses applied on it should be considered based on a combination of the law of gravity according to:

$$F_{ij}^{d}(t) = G(t) \times \frac{M_i(t) \times M_j(t)}{R_{ij}(t) + \mathcal{E}} \times \left(x_j^d(\mathrm{t}) - x_i^d(\mathrm{t})\right) \tag{10}$$

$$F_i^d(t) = \sum_{j \in Kbest,\, j \neq i}^{N} rand_j \times F_{ij}^d(t) \tag{11}$$

where $Rand_j$ is a random number in the interval [0, 1], G(t) is the gravitational constant at time t, M_i and M_j are masses of agents i and j, ε is a small value and $R_{ij}(t)$ is the Euclidean distance between two agents, i and j. kbest is the set of first K agents with the best fitness value and biggest mass, which is a function of time, initialized to K_0 at the beginning and decreased with time. Here K_0 is set to N (total number of agents) and is decreased linearly to 1.

By the law of motion, the acceleration of the agent i at time t, and in direction d, $a_i^d(t)$ is given as follows:

$$a_i^d(t) = \frac{F_i^d(t)}{M_{ii}(t)} \tag{12}$$

Finally, the searching strategy on this concept can be described by following equations:

$$v_i^d(t+1) = rand_i \times v_i^d(t) + a_i^d(t) \tag{13}$$

$$x_i^d(t+1) = x_i^d(t) + v_i^d(t+1) \tag{14}$$

where x_i^d, v_i^d and a_i^d represents the position, velocity and acceleration of ith agent in dth dimension, respectively. randi is a uniform random variable in the interval [0,1]. This random number is applied to give a randomized characteristic to the search. It must be pointed out that the gravitational constant G(t) is important in determining the performance of GSA and is defined as a function of time t:

$$G(t) = G_0 \exp(-\alpha \frac{t}{T}) \tag{15}$$

where G_0 is the initial value, α is a constant, t is the current iterations, T is the maximum number of iterations. The parameters of maximum iteration T, population size N, initial gravitational constant G_0 and constant α control the performance of GSA (N, G_0, α and T).

4. Experimental design

4.1 Data generation

Molla-Alizadeh-Zavardehi *et al.* (2011) generated random instances to verify the effectiveness of their GA approach. We use the same datasets except step cost in this paper. To cover various types of problems, we considered several levels of influencing inputs. First, we generated random problem instances for m = 10, 15, 30, and 50 suppliers and n = 10, 15, 20, 30, 50, 100, and 200 customers, respectively. We considered both small-sized and large-sized problem instances, which was presented by the number of suppliers and customers. Seven different problem sizes, 10 ×10, 10 ×20, 15 ×15, 10 ×30, 50 ×50, 30 ×100 and 50 ×200 are considered for experimental study, which present different levels of difficulty for alternative solution methods. 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, which increases upon progressing from problem type A through problem type D. The variable costs range over the discrete values from 3 to 8. The problem sizes, types, suppliers/customers, and fixed costs ranges are shown in Table 1.

Table 1. Test problems characteristics.

				Range of variable costs			Range of first fixed costs			Range of second fixed costs		
Problem size	Total Demand	Problem type	A_{ij}	a^l	a^l-b^l	α and β	a^l	a^l-b^l	α and β	a^l	a^l-b^l	α and β
10×10	10,000	A	400	U(3, 7)	U(0, 1)	U(0.25, 1)	U(50, 200)	U(0, 25)	U(5, 25)	U(50, 200)	U(0, 25)	U(5, 25)
10×20	15,000	B	400	U(3, 7)	U(0, 1)	U(0.25, 1)	U(100, 400)	U(0, 50)	U(10, 50)	U(100, 400)	U(0, 50)	U(10, 50)
15×15	15,000	C	400	U(3, 7)	U(0, 1)	U(0.25, 1)	U(200, 800)	U(0, 100)	U(20, 100)	U(200, 800)	U(0, 100)	U(20, 100)
10×30	15,000	D	400	U(3, 7)	U(0, 1)	U(0.25, 1)	U(400, 1,600)	U(0, 200)	U(40, 200)	U(400, 1,600)	U(0, 200)	U(40, 200)
50×50	50,000											
30×100	30,000											
50×200	50,000											

4.2 Parameter setting

The performance of the GSA is generally sensitive to the parameter setting which influences the search efficiency and the convergence quality. Twenty-eight test problems, with different sizes and specifications, are generated and solved to evaluate the performance of the presented algorithms.

The instances are implemented using MATLAB on a PC with dual core Duo 2 2.8 GHz and 4 GB of RAM. All algorithms ran 3 times and Due to having different scale of

Published by Firouzabad Institute of Higher Education, Firouzabad, Fars, Iran

objective functions in each instance the Relative Percentage Deviation (RPD) is used for each instance. The RPD is obtained by the following formula:

$$RPD = \frac{Alg_{sol} - Min_{sol}}{Min_{sol}} \times 100 \qquad (16)$$

where Algsol and Minsol are the obtained objective value and minimum objective value found from both proposed algorithms for each instance, respectively. After obtaining the results of the test problems in different trial, results of each trial are transformed into RPD measure.

Using the average of RPD measures of trials, the parameters and operators that have minimum RPD average are selected as the best ones. Therefore, the parameters of GSA were set as follows: population size =100, $G_0 = 80, \alpha = 10$ and T=1000.

4.3 Experimental results

We set searching time to be identical for both algorithms which is equal to 1.5× (n + m) milliseconds. Hence, this criterion is affected by both n and m. We generated 20 instances for each twenty-eight-problem type, summing to $28 \times 20 = 560$ instances which are different from the ones used for parameter setting to avoid bias in the results. For further comparison, the maximum generations are set to 1000.

Considering 20 instances for each of the 28-problem type, or 80 instances for each of the 7 problem sizes, for both algorithms, the instances have been run 5 times and hence, by using the RPD we deal with 400 data for each algorithm. The averages of these data for each algorithm and each instance are shown in Figure 1.

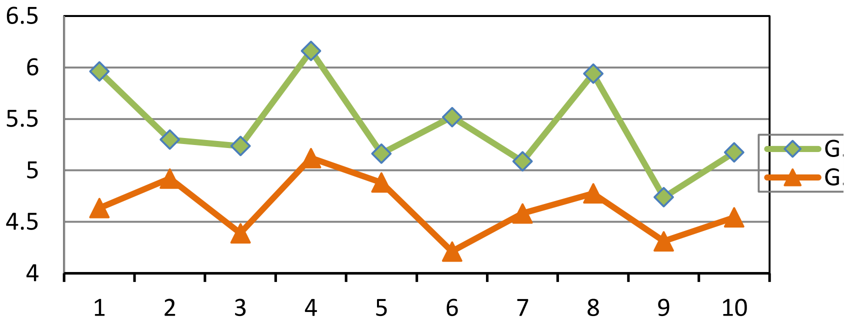

Figure1. Means plot for the interaction between each algorithm and problem size.

In order to verify the statistical validity of the results, we have performed an analysis of variance (ANOVA) to accurately analyze the results. The point that can be concluded from the results is that there is a clear statistically meaningful difference between performances of the algorithms. The means plot and LSD intervals (at the 95% confidence level) for two algorithms are shown in Figure 2.

Published by Firouzabad Institute of Higher Education, Firouzabad, Fars, Iran

Since, we are to appraise the robustness of the algorithms in different circumstances, the effects of the problem sizes on the performance of both algorithms are analyzed. The reciprocal between the capability of the algorithms and the size of problems is illustrated in Figure 1. As can be seen from the result figure, not only is the overall performance of GSA better than SA, but GSA is more robust. Thus, GSA has the capability to reduce the search space significantly and to obtain better solutions with less computational time than GA.

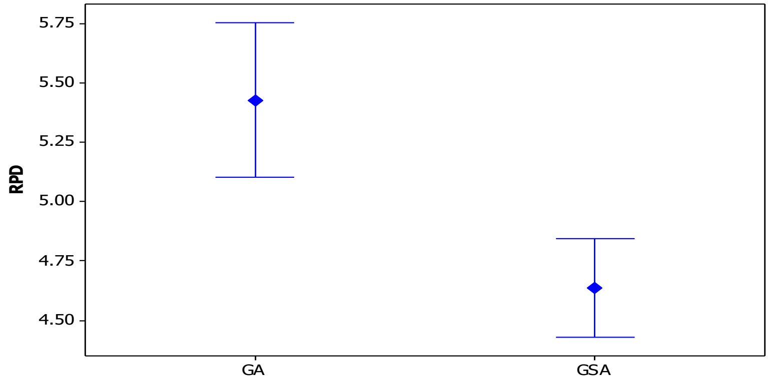

Figure 2. Means plot and LSD intervals for the GSA and GA algorithms

5. Conclusions

In this paper, a real-world modeling of transportation problem, namely, step fixed charge transportation problem has been investigated. We have proposed a gravitational search algorithm to solve this NP-hard problem. In order to evaluate the efficiency of proposed algorithm for solving the problem, a plan is extended based on previous test problems to generate random instances. We solved the randomly generated problems by GSA and also with GA to compare them. The obtained results show the proficiency of GSA comparison with GA. Results showed that the GSA proposed was capable of obtaining better solutions with a more reasonable computational time compared to the GA, for all sizes.

References

1. Altassan, Kh, El-Sherbiny, M.M., Bokkasam, S. (2012). In handling the step fixed charge transportation problem (ICDeM2012), Kedah, Malaysia, March 15–18.
2. Balinski, M. L. (1961). Fixed cost transportation problems. *Naval Research Logistic Quarterly*, 8(1): 41–54.
3. Baranifar, S. (2018). A credibility-constrained programming for closed-loop supply chain network design problem under uncertainty. *Annals of Optimization Theory and Practice*, 1(1), 69-83.

Published by Firouzabad Institute of Higher Education, Firouzabad, Fars, Iran

4. El-Sherbiny, M. M. (2012). Alternate mutation based artificial immune algorithm for step fixed charge transportation problem, *Egyptian Informatics Journal*, 13 (2), 123–134.
5. Kowalski, K., Lev, B. (2008). On step fixed-charge transportation problem, *OMEGA*, 36(5):913-917.
6. Mahmoodirad, A., Hassasi, H., Tohidi, Gh, Sanei, M., Molla-Alizadeh-Zavardehi, S. (2013). Step Fixed Charge Transportation Problems with Fuzzy Numbers, *Scientific Journal of Mechanical and Industrial Engineering*, 2(3), 50-56.
7. Mahmoodirad, A., Sanei, M. (2016). Solving a multi-stage multi-product solid supply chain network design problem by meta-heuristics. *Scientia Iranica*, 23 (3), 1429-1440.
8. 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, 10462–10474.
9. Molla-Alizadeh-Zavardehi, S., Sanei, M., Soltani, R., Mahmoodirad., A. (2014). Solving a step fixed charge transporation problem by a spanning tree-based memetic algorithm, *International Journal of Mathematical Modelling & Computations*, 4 (2), 181-191.
10. Molla-Alizadeh-Zavardehi, S., Mahmoodirad, A., Rahimian, M. (2014). Step Fixed Charge Transportation Problems via Genetic Algorithm, *Indian Journal of Science and Technology*, 7 (7), 949-954.
11. Mosallaeipour, S., Mahmoodirad, A., Niroomand,S., Vizvari, B. (2018). Simultaneous selection of material and supplier under uncertainty in carton box industries: a fuzzy possibilistic multi-criteria approach. *Soft computing*, 22 (9), 2891–2905.
12. Rajabi, F., Najafi, S., Hajiaghaei-Keshteli, M., Molla-Alizadeh-Zavardehi, S. (2013). Solving fuzzy step fixed charge transportation problems via metaheuristics, *International Journal of Research in Industrial Engineering*, 2, 24-34.
13. Rashedi, E., Nezamabadi-pour, H., Saryazdi, S. (2009). GSA: a gravitational search algorithm, *Information Sciences*, 179 (13), 2232-2248.
14. Sanei, M., Hassasi, H., Mahmoodirad, A., Rahimian, M. (2014). Fixed-Charge Transportation Problem with Fuzzy Costs, *Journal of Applied Science and Agriculture*, 9 (9), 1-8.
15. Taghaodi, R., Kardani, F. (2018). Linear programming problem with generalized interval-valued fuzzy numbers. *Annals of Optimization Theory and Practice*, 1(2), 1-9.