Scientific Journal of Mechanical and Industrial Engineering

Sci. J. Mech. Ind. Eng., 3(4): 47-51 (2014)

Particle Swarm Optimization for Step Fixed Charge Transportation Problems

Leila Lagzaie

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

*Corresponding author's E-mail: leila.lagzaie@yahoo.com

Abstract

In this paper, we consider the step fixed-charge transportation problem where is one of the most important problems in transportation research area. In the step fixed-charge transportation problem due to the step function structure of the objective function, we are faced with a "NP- hard" problem. To tackle such an NP-hard problem, we present Particle swarm optimization (PSO) and also with Genetic Algorithm (GA) to compare them. The obtained results show the proficiency of GSA comparison with GA.

Original Article

PII: S238309801400007-3

Received 21 Aug. 2014
Accepted 13 Nov. 2014
Published 25 Dec. 2014

Keywords:
Transportation Problem, Step Fixed Charge Transportation Problem, Particle Swarm Optimization, Genetic Algorithm.

INTRODUCTION

Fixed charge problems arise in a large number of production and transportation systems. Such FCPs are typically modeled as 0-1 integer programming problems. A special case of the general FCP is fixed charge transportation problem [1,2]. The problem involves the distribution of a single commodity from a set of supply centers (sources) to a set of demand centers (destinations) such that the demand at each destination is satisfied without exceeding the supply at any source. The objective is to select a distribution scheme that has the least cost of transportation. Two kinds of costs are considered, a continuous cost which linearly increases with the amount transported between a source *i* and a destination j and a fixed charge which is incurred whenever a nonzero quantity is transported between source *i* and destination *j*. The fixed charge may represent toll charges on a highway; landing fees at an airport; setup costs in production systems or the cost of building roads in transportation systems. Depending on the specific applications, the importance of the fixed charge in the model will vary.

Step fixed charge transportation problem (SFCTP) is an extended version of the FCTP. The SFCTP in its representation first was founded by Kowalski and Lev [3,5]. In the SFCTP due to the step function structure of the objective function, Kowalski and Lev [3-5] were dealing with a "NP- hard" problem.

Mahmoodirad et al. [6] 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. [7] 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. [8] considered the SFCTP and proposed two meta-heuristic, a spanning tree-based genetic algorithm and a spanning tree-based memetic algorithm for this problem. Molla-Alizadeh-Zavardehi et al. [9] developed Genetic Algorithm (GA) for the SFCTP and compared it with simulated annealing.

Since the problems with fixed charges are usually NP-hard problem, the computational time to obtain exact solutions increases in a polynomial fashion and very quickly becomes extremely long as the dimensions of the problem increase [10]. In order to find the best solution, we proposed the Particle swarm optimization (PSO).

The rest of the paper is organized as follows: in Section 1, the SFCTP model is described. Then, the meta-heuristics algorithm, PSO, 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.

Step Fixed Charge Transportation Problem

The following notations are used to define the mathematical model.

Set of indices:

I set of suppliers (i=1,2,…,I)

J set of customers (j=1,2,…,J)

Parameters:

s_i capacity of supplier i

d_j capacity of customer j

c_{ij} cost of transporting one unit of product from supplier i to customer j

A_{ij} a certain amount of transporting from supplier i to customer j

$k_{ij,1}$ fixed charge of transporting one unit of product from supplier i to customer j

$k_{ij,2}$ additional fixed cost when the transported units exceeds a certain amount A_{ij}

Decision variables:

x_{ij} quantity of product shipped from plant i to customer j

$b_{ij,1}$ binary variable equal to 1 if $x_{ij} > 0$ and equal to 0 otherwise

$b_{ij,2}$ binary variable equal to 1 if $x_{ij} > A_{ij}$ and equal to 0 otherwise

The mathematical model of the problem as follows [5]:

$$\text{Min} \quad Z = \sum_{i=1}^{m}\sum_{j=1}^{n}(c_{ij}x_{ij} + f_{ij}y_{ij})$$

$$\text{s.t.} \quad \sum_{j=1}^{n} x_{ij} = s_i, \qquad i = 1,2,...,m,$$

$$\sum_{i=1}^{m} x_{ij} = d_j, \qquad j = 1,2,...,n,$$

$$x_{ij} \geq 0, \qquad i = 1,2,\dots,m \ , \ j = 1,2,\dots,n,$$

$$y_{ij} = \begin{cases} 1, & x_{ij} > 0, \\ 0, & otherwise \end{cases}$$

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,

$$f_{ij} = b_{ij,1}k_{ij,1} + b_{ij,2}k_{ij,2},$$

where,

$$b_{ij,1} = \begin{cases} 1, & x_{ij} > 0, \\ 0, & otherwise \end{cases},$$

$$b_{ij,2} = \begin{cases} 1, & x_{ij} > A_{ij}, \\ 0, & otherwise \end{cases}$$

and $k_{ij,1}, k_{ij,2}, g_{ij}, A_{ij}$ are nonnegative real numbers.

Note that fixed cost associated with route *(i, j)* has two steps. It could have multiple steps, depending on the problem structure. Without loss of generality, we assume that the problem is balanced, that is:

$$\sum_{i=1}^{m} s_i = \sum_{j=1}^{n} d_j,$$

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), f_{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.

Particle Swarm Optimization

PSO is one powerful and widely used swarm intelligence paradigm introduced by Kennedy and Eberhart in 1995 for solving optimization problems [11]. This algorithm has been proposed through inspiration from social behaviors of the individuals in bird and fish swarms [4]. Individuals in the swarms are referred to as particles, and each particle consists of D-dimensional values. For a D-dimensional state, position and velocity expressions of particle i are represented as follows.

$$X_i = \{X_{i1},...,X_{iD}\} \text{ and } V_i = \{V_{i1},...,V_{iD}\}$$

Intelligent interaction among the swarm is provided with best value of each particle (*pbest*) and best value of all particles (*gbest*) until at the current iteration. For a D-dimensional search space, *pbest* of particle *i* is represented as $pbest = \{P_{i1},...,P_{iD}\}$, *gbest* is represented as $gbest = \{G_1,...,G_D\}$. Since PSO will perform update procedures according to these values, *pbest* values for each particle and the *gbest* value, which is the best value for the entire swarm, should be kept. PSO consists of two stages as beginning and calculation. In the beginning stage, all particles are distributed randomly in the search space within the determined boundaries. In calculation stage, velocities and positions of the particles are updated. Velocity of a particle is calculated as follows [11]:

$$V_i(t+1) = V_i(t) + c_1 R_1(t)(pbest_i(t) - X_i(t)) + c_2 R_2(t)(gbest(t) - X_i(t)),$$

where t is the generation number, $V_i(t)$ and $X_i(t)$ represent the velocity and position of the *i*-th particle, respectively; ω is termed inertia weight, c_1 and c_2 are the acceleration coefficients, $R_1(t)$ and $R_2(t)$ are two vectors randomly generated within $[0,1]^n$, with n being the dimension of the search space; $pbest_i(t)$ and $gbest_i(t)$ denote the personal best of the *i*-th particle and the global best of the swarm, respectively.

While the fact that acceleration coefficients take big values causes the particles to move away from each other

SJMIE To cite this paper: Lagzaie L. 2014. Particle Swarm Optimization for Step Fixed Charge Transportation Problems. *Sci. J. Mech. Ind. Eng.*, 3 (4): 47-51.

Journal homepages: http://sjmie.science-line.com/

and separate, their taking small values causes limitation of the movements of the particles, and not being able to scan the solution space adequately [12].

V_{max} and V_{min} parameters may be set for the velocity values determined for each particle to prevent occurrence of big changes on the particles or constant limit excesses. In this study, V_{max} and V_{min} was set as 20% of the upper and lower limits. Inertia weight was added to PSO by Shi and Eberhart in 1998 [13] to provide the balance between exploitation and exploration:

$$V_i(t+1) = \omega V_i(t) + c_1 R_1(t)(pbest_i(t) - X_i(t)) + c_2 R_2(t)(gbest(t) - X_i(t)),$$

Inertia weight controls effect of previous velocity increases of the particles on the velocity value, and takes part in providing the balance between global search and local search. When the inertia weight takes large values, global search is more suitable and a small inertia weight facilitates local search. Shi and Eberhart [13] proposed a linearly decreasing inertia weight over the course of search. Usually, max and min values are determined for inertia, too. In this study, inertia update is made as follows:

$$\omega = \frac{Max_{iter} - iter}{Max_{iter}}$$

Where Max_{iter} refers to maximum number of iteration and *iter* is the current iteration. New position values are obtained by adding the velocity updates determined by the formula given in the following to the particles:

$$X_i(t+1) = X_i(t) + V_i(t+1),$$

After making the velocity updates, new fitness values of the particles are calculated, and if necessary, *pbest* and *gbest* updates are performed, and the same procedure is continued until the stop criteria are provided.

Experimental Design Instances

Molla-Alizadeh-Zavardehi et al. [8] 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 and 2.

Table 1. Test problems characteristics.

			Range of variable costs			
Problem Size	Total Demand	Problem Type	A_{ij}	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)
10×20	15,000	B	400	U(3, 7)	U(0, 1)	U(0.25, 1)
15×15	15,000	C	400	U(3, 7)	U(0, 1)	U(0.25, 1)
10×30	15,000	D	400	U(3, 7)	U(0, 1)	U(0.25, 1)
50×50	50,000					
30×100	30,000					
50×200	50,000					

Table 2. Test problems characteristics.

Range of first fixed costs			Range of second fixed costs		
a^l	a^l-b^l	α and β	a^l	a^l-b^l	α and β
U(50, 200)	U(0, 25)	U(5, 25)	U(50, 200)	U(0, 25)	U(5, 25)
U(100, 400)	U(0, 50)	U(10, 50)	U(100, 400)	U(0, 50)	U(10, 50)
U(200, 800)	U(0, 100)	U(20, 100)	U(200, 800)	U(0, 100)	U(20, 100)
U(400, 1,600)	U(0, 200)	U(40, 200)	U(400, 1,600)	U(0, 200)	U(40, 200)

Parameter Setting

The performance of the GA 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 objective functions in each instance the relative percentage deviation (RPD) is used for each instance. The RPD is obtained by the following formula:

$$RPD = \frac{Algsol - Minsol}{Minsol} \times 100$$

Where Alg_{sol} and Min_{sol} are the obtained objective value and minimum objective value found from both proposed algorithms for each instance, respectively. After

SJMIE To cite this paper: Lagzaie L. 2014. Particle Swarm Optimization for Step Fixed Charge Transportation Problems. *Sci. J. Mech. Ind. Eng.*, 3 (4): 47-51.

Journal homepages: http://sjmie.science-line.com/

obtaining the results of the test problems in different trial, results of each trial are transformed into *RPD* measure.

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. The more the number of suppliers and customers, the more the rise of searching time increases. 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. 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 Fig. 1.

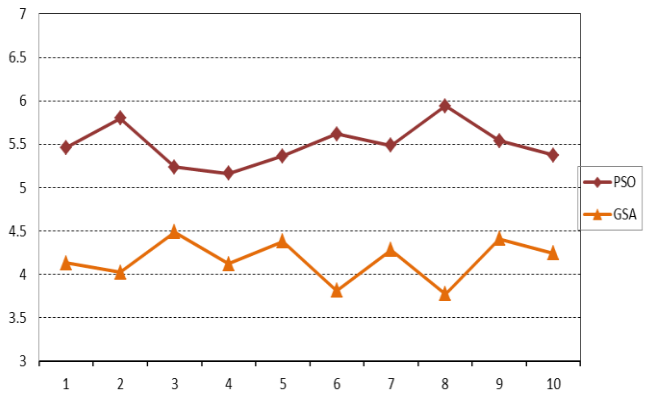

Fig 1. Means plot for the interaction between each algorithm and problem size.

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 Fig. 2.

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 Fig. 1.

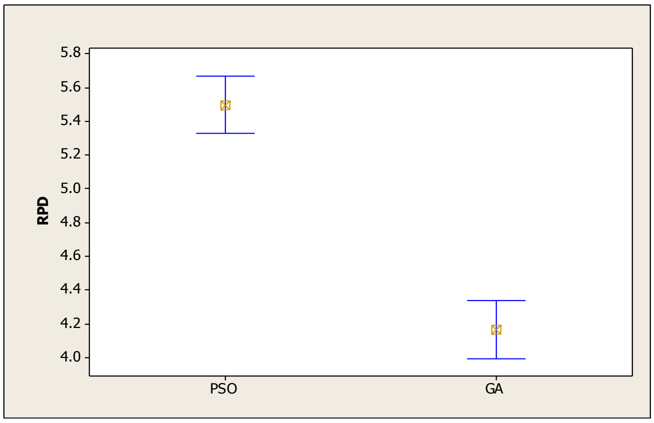

Fig 2. Means plot and LSD intervals for the PSO and GA algorithms.

CONCLUSION

In this paper, a real-world modeling of transportation problem, namely, step fixed charge transportation problem has been investigated. We have proposed a PSO 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 PSO and also with GA to compare them. The obtained results show the proficiency of PSO comparison with GA. Also, considering other well-known meta-heuristics such as simulated Annealing and variable neighborhood search or new ones such as imperialist competitive algorithm is encouraged.

Acknowledgements

This study was partially supported by Islamic Azad University, Zabol Branch. The author is grateful for this financial support.

REFERENCES

1. Balinski, M. L., Fixed cost transportation problems. Naval Research Logistic Quarterly, 1961. 8(1): p. 41–54.
2. Sanei, M., Hassasi. H., Mahmoodirad, A., Rahimian. M., Fixed-Charge Transportation Problem with Fuzzy Costs, Journal of Applied Science and Agriculture, 2014. 9(9): p. 1-8.
3. Altassan, Kh, El-Sherbiny M., Bokkasam. S., In Handling the step fixed charge transportation problem (ICDeM2012), Kedah, Malaysia, March 15–18, 2012.
4. El-Sherbiny, M. M., Alternate mutation based artificial immune algorithm for step fixed charge transportation problem, Egyptian Informatics Journal, 2012. 13 (2): 123–134.
5. Kowalski, K Lev. B., On step fixed-charge transportation problem, OMEGA, The International Journal of Management Science, 2008. 36(5): p. 913-917.
6. Mahmoodirad, A., Hassasi. H., Tohidi. Gh., Sanei. M., Molla-Alizadeh-Zavardehi. S., Step Fixed Charge Transportation Problems with Fuzzy Numbers, Scientific Journal of Mechanical and Industrial Engineering, 2013. 2(3): p. 50-56.
7. Rajabi, F., Najafi. S., Hajiaghaei-Keshteli. M., Molla-Alizadeh-Zavardehi. S., Solving fuzzy step fixed charge transportation problems via metaheuristics, International Journal of Research in Industrial Engineering, 2013. 2: 24-34.
8. Molla-Alizadeh-Zavardehi, S., Sanei. M., Soltani. R., Mahmoodirad. A., Solving a step fixed charge transporation problem by a spanning tree-based memetic algorithm, International Journal of Mathematical Modelling & Computations, 2014. 04(2): p. 181-191.
9. Molla-Alizadeh-Zavardehi, S., Mahmoodirad. A., Rahimian. M., Step Fixed Charge Transportation

Problems via Genetic Algorithm, Indian Journal of Science and Technology, 2014. 7 (7): p. 949-954.
10. Molla-Alizadeh-Zavardehi, S., Hajiaghaei-Keshteli. M., Tavakkoli-moghaddam. R., Solving a capacitated fixed-charge transportation problem by artificial immune and genetic algorithms with a Prüfer number representation, Expert Systems with Applications, 2011. 38: p. 10462–10474.
11. Kennedy, J., Eberhart, R., Particle swarm optimization, in: Proceedings of the Sixth International Symposium on Micro Machine and Human Science, Nagoya, Japan, 1995. pp. 39-43.
12. Ortakc, Y., Comparison of Particle Swarm Optimization Methods in Applications, Karabuk University, Graduate School of Natural and Applied Sciences, 2011.
13. Shi, Y., Eberhart, R., A modified particle swarm optimizer, in: The 1998 IEEE International Conference on Evolutionary Computation Proceedings, Anchorage, AK, 1998, pp. 69–73.