2012 International Conference on Innovation and Information Management (ICIIM 2012)
IPCSIT vol. 36 (2012) © (2012) IACSIT Press, Singapore

A Novel Artificial Immune Algorithm for Solving Fixed Charge Transportation Problems

Khalid M. Altassan[1,a+], Mahmoud M. El-Sherbiny[2,b], Yasser M. Ibrahim [1,c] Ahmed D. Abid[1,d]

[1]King Saud University, Faculty of Business Administration, Saudi Arabia

[2]OR Dept, ISSR, Cairo University, Egypt.

Currently, College of Business Administration, King Saud University, Saudi Arabia

Abstract. This paper introduces a novel Artificial Immune Algorithm for solving Fixed Charge Transportation Problems (AIAFCTP). AIAFCTP solves both balanced and unbalanced FCTP without introducing a dummy supplier or a dummy customer. In AIAFCTP a coding schema is designed and procedures developed for decoding such schema and allocating the transported units. These are used instead of spanning tree and Prüfer number. Therefore, a repairing procedure for feasibility is not needed. Besides, some mutation functions are developed and used in AIAFCTP. Due to the significant role of mutation function on the AIAFCTP's quality, its performances are compared to select the best one. For this purpose, various problem sizes are generated at random and then a robust calibration is applied. In addition, two problems with different sizes are solved to evaluate the performance of the AIAFCTP and to compare its performance with most recent methods.

Keywords: Fixed Charge Transportation, Artificial Immune, Convergence, Genetic algorithm

1. Introduction

The fixed charge transportation problem (FCTP) is considered to be an NP-hard problem [1]. Usually, FCTP is formulated and solved as a mixed integer network programming problem. These methods are not employed because of their inefficient and expensive computation. Generally, solving methods can be classified as exact method [2, 3] or heuristic method [5, 6]. Exact methods are, however, generally not very useful when a problem reaches a certain level [4]. Therefore, heuristic methods have been proposed, such as [7, 8]. The major disadvantage of heuristic methods is the possibility of terminating at a local optimum that is far distant from the global optimum. Recently, some meta-heuristic methods [9, 10] employed in the FCTP. Moreover, to improve solution quality, Hajiaghaei [11] addressed a nonlinear FCTP using a spanning tree based Genetic Algorithm (GA). Nevertheless, the quality of solutions attained largely depends on the randomness. On the other hand, some special Artificial Immune Algorithms (AIA) [12] are developed to solve complex optimization problems [13, 14].

To improve the solution quality of the FCTP, this paper aims to introduce an Artificial Immune Algorithm for solving both balanced and unbalanced FCTP (AIAFCTP) without introducing a dummy supplier or a dummy customer and study the effect of its factors on the performance. In addition to that two problems with different sizes have been solved to evaluate the performance of the AIAFCTP and to compare its performance with hybrid particle swarm method proposed in [16], and the GA proposed in [11].

2. Fixed Charge Transportation Problem

[+] Corresponding author
E-mail address: [a]kaltassan@ksu.edu.sa , [b]msherbiny@ksu.edu.sa , [c] ysabridarwesh@ksu.edu.sa [d]darwesh@ksu.edu.sa

FCTP can be described as a distribution problem in which there are *m* suppliers (warehouses or factories) and *n* customers (destinations or demand points). Each of the *m* suppliers can ship to any of the *n* customers at a shipping cost per unit c_{ij} (unit cost for shipping from supplier *i* to customer *j*) plus a fixed cost f_{ij}, assumed for opening this route. Each supplier i =1, 2, . . ., m has s_i units of supply and each customer j =1, 2, . . ., n demands d_j units. The objective is to determine which routes are to be opened and the size of the shipment, so that the total cost of meeting demand, given the supply constraints, is minimized. The stander mathematical model of FCTP [11] can be represented as follows:

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

$$\text{s.t.} \quad \sum_{i=1}^{m} x_{ij} \geq b_j \quad \text{for } j = 1,...,n \tag{2}$$

$$\sum_{j=1}^{n} x_{ij} \leq a_i \quad \text{for } i = 1,...,m \tag{3}$$

$$\forall i, j \quad x_{ij} \geq 0$$
$$y_{ij} = 0 \quad \text{if } x_{ij} = 0$$
$$y_{ij} = 1 \quad \text{if } x_{ij} > 0$$

Where x_{ij} is the unknown quantity to be transported on the route (*i*, *j*) that from plant *i* to customer *j*. The transportation cost for shipping per unit from plant *i* to customer *j* is $c_{ij} \times x_{ij}$.

3. The Proposed Algorithm

In this paper, a typical immune algorithm structure is utilized. The algorithm preserves the essential principles of artificial immune Algorithms including the cloning, mutation, and clonal selection. The implementation of the immune algorithm is often different for each problem handled. That is, the representation and hence the creation of the solutions, the mutation, and the affinity should be tailored and implemented to fit the case at hand. For the FCTP, the problem of interest in this research, the pseudocode of the main steps for the proposed algorithm is presented as follows:

1. Set $g = 1$.
2. Create initial population of l antibodies A_i with algorithm1
3. Set $i = 1$.
4. Clone i^{th} Antibody A_i in the population *CN* times.
5. Mutate each of the *CN* clones.
6. Evaluate each of the *CN* clones.
7. Apply decoding procedure1.
8. Apply allocating procedure 2.
9. Calculate the fitness of each antibody A_i.
10. Get the mutated clone with the Best Fitness *BF*.
11. If *BF* fitness better than the fitness of A_i then *BF* replaces A_i else go to step 12.
12. Set $i = i + 1$.
13. Repeat from step 4 to step 12 until $i > l$.
14. Calculate the affinity between each two antibodies in the population.
15. Select the antibodies for the new mutation based on the affinity.
16. Create new antibodies to substitute the removed antibodies.
17. $g = g + 1$.

18. Repeat step 3 to step 17 until g > number of iterations.

The details of the main steps are presented in the following subsections.

3.1. Coding Schema and Initialization

One of the most important issues when designing the AIA lies on its solution (antibody) representation. In order to construct a direct relationship between the problem domain and the AIAFCTP, the proposed coding scheme (antibody A_i structure) consists of the set of all the integer numbers in the interval [1, m+n] with any sequence and without any repetition; where its length is equal to $m + n$, where m and n are the number of suppliers and customers respectively. Therefore, the suppliers' numbers represented by the integer numbers from 1 to m and the customers' integer numbers from m+1 to $m + n$. Fig. 1 depicts a sample antibody example, which is used to code a 4x5 FCTP

8	3	9	5	4	7	2	1	6

Fig. 1 An example of proposed antibody structure

The population is initialized randomly by calling procedure1 l times to create l antibodies A_i where l represents the population size. In this procedure, the Rand(1, m+ n) is a function that returns a random integer number in the interval[1, $m + n$], Mod(x, y) is a function that returns the reminder of x when it is divided by y and Q. Remove(k) is a function that eliminate k^{th} element of queue Q. The pseudocode for procedure 1 of creating individual antibody is presented as follows:

Procedure1:

1. Create a Create collection list Q ={1, 2,…, m +n }.
2. Set j = 1.
3. Generate an integer number between 1 and m +n and set it to variable c. Take the cell $A_i(j)$
4. Set k = Mod(c, Length(Q)); where Mod(c, Length(Q)) is a function that returns the reminder of c when it is divided by length(Q).
5. Add Q[k] to the antibody A_i in the position j.
6. Remove the item k from the list Q
7. $j = j + 1$.
8. Repeat from step 3 to step 7 until $j > n + m$.
9. Return the antibody A_i , where i =1,…, l and l is the population size

3.2. Decoding Procedure

Decoding procedure is used to decode the antibody A_i into suppliers order S and customers order D. The inputs of this procedure are the generated antibody A_i , the number of suppliers m ,and the number of customers n while the results are the sequence of suppliers S and the sequence of customers D. The pseudocode for the decoding procedure is presented as follows:

1. Set j = 1.
2. Take the cell $A_i(j)$
3. If $A_i(j) \leq n$ then add $A_i(j)$ to the supplier order S.
4. If $A_i(j) > n$ then add $A_i(j)$ to the customer order D.
5. $j = j + 1$.
6. Repeat from step 2 to step 5 until $j > n + m$.
7. Return the supplier order S and the customer order D.

3.3. Allocating Procedure

.The allocating procedure allocates the transported units based on the order coming from decoding procedure 2. In other words, this procedure finds a feasible solution for FCTP based on the outputs of the decoding procedure. This procedure guarantees the validity of both constraint (2) and the constraint (3). Also, this procedure can be used to solve both balanced and unbalanced transportation problems without introducing a dummy supplier or a dummy customer. The pseudocode for the allocating procedure is presented as follows:

1. Set i equal to the first value in suppliers' order S and set j equal to the first value in customers' order D. i.e. $i = S(1)$ and $j = D(1)$.
2. If $a_i = b_j$ then { set x_{ij} equal to a_i , remove $S(1)$, and remove $D(1)$}
3. If $a_i > b_j$ then { set x_{ij} equal to b_j , set a_i equal to $a_i - b_j$, and remove $D(1)$}
4. If $a_i < b_j$ then { set x_{ij} equal to a_i , set b_j equal to $b_j - a_i$, and remove $S(1)$}
5. Repeat from step 1 to step 4 until (length of queue $S = 0$ or length of queue $D = 0$).
6. Return x_{ij} $\forall$ $i = 1, 2, \ldots, m$ and $j = 1, 2, \ldots, n$.

The inputs of procedure 3 are the sequence of suppliers S and the sequence of customers D (the output of algorithm 2). Based on these sequences the allocating procedure allocates units X_{ij} (feasible solution) of FCTP.

3.4. Evaluating the Solutions

Each antibody is evaluated to determine its fitness. As mentioned above each antibody is decoded using procedure 2 and its result used as an input for procedure3. The solution resulted from algorithm3 is evaluated using objective function (1). The value of objective (1) is assigned to the antibody as its fitness.

3.5. Cloning and Mutation

Each antibody is cloned (copied) number of times, determined by the number of Cloning Number *(CN*). The clones are then mutated to get new antibodies that are different from their parent. In the proposed AIAFCTP, four different Mutation Functions *(MF*s) are developed and tested.

The *first* mutation method is a uniform random where the number of swaps (*NS*) is defined by a random number in the interval [1, *MNS*] where *MNS* is a parameter representing Max Numbers of Swaps. The *NS* for this *MF* is represented in Eq. (4).

$$NS = Rand(1, MSN) \tag{4}$$

The remaining three *MF*s are functions of two parameters. The first parameter is the non-uniform factor based on which the *NS* is determined. The second parameter is the degree of non-uniformity (*u*). All the *MF*s are designed to be directly related with *u*.

The second MF is based on the fitness of the solution. As the FCTP is a minimization problem, the function is designed to be directly related with the Normalized Fitness (*NF*) of the solution. That is, solutions with *NF* closer to one, i.e. relatively bad solutions, will be subject to more *NS*. This actually gives the chance for low affinity solutions to mutate more in order to improve their affinities. The *NS* for this *MF* is adapted with Eq. (5) and the *NF* of each antibody is calculated using Eq. (6).

$$NS = MSN^{(1-(1-NF)^{u})} \tag{5}$$

$$NF = \frac{LowestFitn\ ess\text{ - }Fitn\ ess}{LowestFitn\ ess\text{ - }High\ estFitness} \tag{6}$$

In the third and the Forth MFs, we include a random factor (R) so that the NS is based on the non-uniform factor, time and fitness respectively, but with some randomization. The random factor R takes values between zero and one. The method behaves almost the same way as the original ones when Rand is close to zero. The closer the R to one is, the closer the NS to the max swaps no is. These two MFs are suggested to allow the search to escape from local optima by occasionally increasing the number of swaps. The NS for these two mutations are adapted with Eq. (7) and Eq. (8), respectively.

$$NS = MSN \times R^{(1-NF)^u} \tag{7}$$

$$NS = MSN \times R^{(T^u)} \tag{8}$$

3.6. Affinity Function

The selection of the antibodies from one generation to the next one depends on some measurement of the affinity (similarity) among all the antibodies of the current generation. The calculations of the affinity *AF* between each two antibodies are applied to prevent similar solutions with high evaluation from being copied to the next generation and hence dominating the search. The technique used to check the similarity between every two antibodies in a population counts the number of similar variables in the two solutions. The *AF* of two antibodies A_j and A_k is represented as in Eq. (9).

$$AF(A_j, A_k) = \sum_i y_i$$
$$\text{where } y_i = \begin{cases} 1 & \text{if the } i^{th} \text{ varible of } A_j = \text{the } i^{th} \text{ varible of } A_k \\ 0 & \text{Otherwise} \end{cases} \tag{9}$$

The basic idea is that the more the number of similar variables in the two antibodies is, the higher the similarity between them. Based on a specific parameter, the algorithm eliminates those solutions that have *AF* more than a specific parameter -Number of Similarities (*NS*).

4. Parametric Analysis

In this section we try to discover the best *MF* from the implemented four. Because the scale of the objective functions in each problem is different, they could not be used directly. Therefore, the Relative Percentage Deviation (*RPD*) is used for each combinatioin [15]. *RPD* is calculated by using Eq. (10).

$$RPD = \frac{A\lg_{sol} - Min_{sol}}{Min_{sol}} \times 100 \tag{10}$$

Where Alg_{sol} and Min_{sol} are the obtained objective values for each replication of trial in a given combination and the obtained best solution, respectively. After converting the objective values to *RPD*s, the mean *RPD* is calculated for each trial. Five problems with different size are generated, and used to discover the best *MF* from the implemented fourth. Each one of these problems is solved tine times. As illustrated in Table 1, the quality of the results of using the second *MF* is close to the fourth *MF* and both are superior to the others. But the fourth *MF* is most superior. Therefore, in the next section, the fourth *MF* will be used in our comparison with the most recent algorithms in the literatures.

Table 1. The Comparitive results of the *RPD* for the *MF*

MF	*RPD* of the test problems					Mean *RPD*
	14x18	5x10	10x10	10x20	30x30	
1	1.5%	9.3%	3.0%	6.5%	3.0%	4.2%
2	0.3%	1.9%	1.5%	0.7%	2.1%	1.4%
3	1.0%	5.5%	2.6%	3.7%	2.2%	2.8%
4	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%

5. Numerical Experiments

To evaluate the performance of the proposed AIAFCTP two problems with different sizes, previously addressed in [11] and [16] are solved, comparing with the solution presented by them The sizes of the problems are 4×5 and 5×10, respectively. The shipping costs, and the fixed costs of the first and the second problems is given in Tables 2 and 3 respectively. The parameters used for the proposed method in these problems are optimally tuned parameters and operators from experimental results.

Table 2. Unit variable cost in 4 x 5 problem.

Plants	Costumers									
	Shipping costs c_{ij}					Fixed costs f_{ij}				
	1	2	3	4	5	1	2	3	4	5
1	8	4	3	5	8	60	88	95	76	97
2	3	6	4	8	5	51	72	65	87	76
3	8	4	5	3	4	67	89	99	89	100
4	4	6	8	3	3	86	84	70	92	88

Table 3. Unit variable cost in 5 x 10 problem

Plants	Costumers																			
	Shipping costs c_{ij}										Fixed costs f_{ij}									
	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10
1	8	4	3	5	2	1	3	5	2	6	160	488	295	376	297	360	199	292	481	162
2	3	3	4	8	5	3	5	1	4	5	451	172	265	487	176	260	280	300	354	201
3	7	4	5	3	4	2	4	3	7	3	167	250	499	189	340	216	177	495	170	414
4	1	2	8	1	3	1	4	6	8	2	386	184	370	292	188	206	340	205	465	273
5	4	5	6	3	3	4	2	1	2	1	156	244	460	382	270	180	235	355	276	190

Concerning the first problem, the supplies and demands from each plant 1 to 4 for the each customer 1 to 5 are as follows: b_1 = 88, b_2 = 57, b_3 = 24, b_4 = 73, b_5 = 33, a_1 = 57, a_2 = 93, a_3 = 50, a_4 = 75. The obtained local optimal solution from our algorithm is the same as the solution found by [16], and [11] and is equal to 1484. The transportation allocation matrixes for each method are shown in Tables 4.

Table 4.Transportation allocation matrix found in [11, 16] ,and the proposed AIAFCTP for 4 x 5 problem.

	D1	D2	D3	D4	D5
S1		57	24		
S2	69				
S3				50	
S4	19			23	33

Concerning the second problem, the supplies and demands from the each plant 1 to 5 for each customer 1–10 are as follows: b_1 = 225, b_2 = 150, b_3 = 90, b_4 = 215, b_5 = 130, b_6 = 88, b_7 = 57, b_8 = 124, b_9 = 273, b_{10} = 133, and a_1 = 157, a_2 = 293, a_3 = 150, a_4 = 575, a_5 = 310.

The obtained local optimal solution for this problem from our algorithm is 6255, the solution found by [16] is 6296, while the solution found by [11] is 6305. The transportation allocation matrixes for each method are shown in Tables 5–7.

Table 5. Transportation allocation matrix found in [11] for 5 x 10 problem.

	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10
S1			90		67					
S2		150						124		19
S3				5		88	57			
S4	225			210	63					77
S5									273	37

Table 6. Transportation allocation matrix found in [16] for 5 x 10 problem

	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10
S1					130		27			
S2		15	90					124		64
S3						88	30		32	
S4	225	135		215						
S5									241	69

Table 7. Transportation allocation matrix found by the AIAFCTP for 5 x 10 problem.

	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10
S1			27		130					
S2		106	63					124		
S3						88	57		5	
S4	225	44		215						91
S5									268	42

6. Summary

This paper has proposed a novel artificial immune algorithm for solving fixed charge transportation problem (AIAFCTP). While using the spanning tree and Prüfer number with the GA may result in non-feasible solutions and need repairing procedure, the AIAFCTP guarantees the feasibility of all the generated solutions. Hence, in AIAFCTP the decoding and the allocation procedures are used instead of spanning tree used with GA. Further, four mutation functions are developed and used in AIAFCTP. Due to the significant role of mutation function on the quality of AIAFCTP, its performances are compared to select the best one. The comparative study of the AIAFCTP with the hybrid particle swarm method presented in [16, 11] showed that the AIAFCTP is superior to them. The performance of AIAFCTP and the quality of the solution prove that AIAFCTP is highly competitive and can be considered as a novel alternative to solve FCTPs. Future work includes experimentation with parameters of AIAFCTP, such as different mutation functions and parameter settings.

7. Acknowledgement

This paper is supported by the Research Center at the College of Business Administration and the Deanship of Scientific Research at King Suad University, Riyadh.

8. References

[1] K.G. Murty, Solving the fixed charge problem by ranking the extreme points, *Operations Research*, 268-279 (1968).

[2] J. M. Rousseau, A cutting plane method for the fixed cost problem, Doctoral dissertation, *Massachusetts Institute of Technology. Cambridge, MA* (1973).

[3] P. G. McKeown, A vertex ranking procedure for solving the linear fixed charge problem, *Operations Research*, 1183-1191, (1975).

[4] U. S. Palekar, M. K. Karwan and S. Zionts, A branch-and bound method for the fixed charge transportation problem, *Management Science*, 36(9): 1092-1105(1990).

[5] M. L.Balinski, Fixed cost transportation problem, *Naval Research Logistics Quarterly*, 8, 41-54 (1961).

[6] M. Sun and P. G. McKeown, Tabu search applied to the general fixed charge problem, *Annals of Operations Research*, 41(14):405-420 (1993).

[7] D. Wright, C. Haehling Von Lanzenauer, Solving the fixed charge problem with lagrangian relaxation and cost allocation heuristics, *European Journal of Operational Research*, 42, 304-312 (1989).

[8] D. Wright, C. Haehling von Lanzenauer, COLE: A new heuristic approach for solving the fixed charge problem Computational results. *European Journal of Operational Research,* 52, 235-246 (1991).

[9] M. Sun, J. E. Aronson and P. G. Mckeown, A tabu search heuristic procedure for the fixed charge transportation problem, *European Journal of Operational Research*,106, 441-456 (1998).

[10] J. Gottlieb, B. A. Julstrom, F. Rothlauf and G. R. Raidl, Prüfer numbers: A poor representation of spanning trees for evolutionary search, *In: Spector L, et al., eds. Proc. of the 2001 Genetic and Evolutionary Computation Conf. San Francisco: Morgan Kaufmann Publishers*, 343-350 (2001).

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

[12] J. Timmis,T. Knight, L. N. Catro and Hart, An overview of artificial immune systems, Computation in Cells and Tissues: Perspectives and Tools Thought. Natural Computation Series, *Springer-Verlag*, 51-86 (2004).

[13] O. Engi and A. Doyen, A new approach to solve hybrid flow shop scheduling problems by artificial immune system, Future Generation Computer Systems, 20, 1083–1095 (2004).

[14] J. T.Tsai, W. H. Ho, T. K. Liu and J. H. Chou, Improved immune algorithm for global numerical optimization and job shop scheduling problems, Applied Mathematics and Computation, 194, 406–424 (2007).

[15] G.Taguchi, Introduction to quality engineering, White Plains: Asian Productivity Organization/ UNIPUB (1986).

[16] M. M. El-Sherbiny and R. M. Alhamali, A Hybrid Particle Swarm Method with Artificial Immune Learning for Solving the Fixed Charge Transportation Problem, submitted to Computers & Industrial Engineering (2011).