Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.12785/amis/080235

Artificial Immune Algorithm for Solving Fixed Charge Transportation Problem

Khalid M. Altassan[1], *Mahmoud M. El-Sherbiny*[2,*] *and Ahmed D. Abid*[3]

[1] King Saud University, Faculty of Business Administration, Saudi Arabia
[2] Cairo University, Institute of Statistical Studies & Research (ISSR), Operations Research Dept. Egypt
[3] King Saud University, Faculty of Business Administration, Saudi Arabia

Received: 7 Apr. 2013, Revised: 8 Aug. 2013, Accepted: 10 Aug. 2013
Published online: 1 Mar. 2014

Abstract: Fixed Charge Transportation Problem (FCTP) is considered to be an NP-hard problem. Several genetic algorithms based on spanning tree and Prfer number were presented. Most of such methods do not guarantee the feasibility of all the generated chromosomes and need a repairing procedure for feasibility. Contrary to the findings in previous works, this paper introduces an Artificial Immune System for solving Fixed Charge Transportation Problems (AISFCTP). AISFCTP solves both balanced and unbalanced FCTP without introducing a dummy supplier or a dummy customer. In AISFCTP a coding schema is designed and algorithms are developed for decoding such schema and allocating the transported units. These are used instead of spanning tree and Prfer number. Therefore, a repairing procedure for feasibility is not needed, i.e. all the generated antibodies are feasible. Besides, some mutation functions are developed and used in AISFCTP. Due to the significant role of mutation function on the AISFCTPs 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 using the relative percentage deviation (RPD) method and paired t-tests. In addition, two problems with different sizes are solved to evaluate the performance of the AISFCTP and to compare its performance with most recent methods.

Keywords: Fixed charge transportation, Fixed charge transportation, Convergence, Genetic algorithm, Artificial immune

1 Introduction

The fixed charge transportation problem (FCTP) is considered to be an NP-hard problem[1]. Usually, anFCTP is formulated and solved as a mixed integer network programming problem. Theoretically, the FCTP can be solved by any mixed integer programming; however, these methods are not employed because of their inefficient and expensive computation. Generally, solving methods can be classified as exact or heuristic methods. Exact methods for solving the FCTP include the cutting planes method[2], the vertex ranking method [3], and the branch-and-bound method [4], amongst others. Exact methods are, however, generally not very useful when a problem reaches a certain level, because they do not make the most use of the special network structure of the FCTP. Therefore, heuristic methods have been proposed, such as the adjacent extreme point search method [5,6], and the Lagrangian relaxation method [7,8]. Although these methods are usually computationally efficient, 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 employed in the FCTP, such as the Tabu search method for FCTPs[9], genetic algorithms (GA) based on a spanning tree with Prfer numbers [10] and GAs based on a matrix permutation representation [11], have improved the effective coding of the spanning tree method based on edge sets [12,19]. The GA creates a sorted set of edges to encode the spanning tree, which is efficient compared with evolution strategies at a certain level [13]. Moreover, to improve solution quality, Hajiaghaei[14] addressed a nonlinear FCTP using a spanning tree based GA. Nevertheless, the quality of solutions attained largely depends on the randomness.

On the other hand, Immune system is made up of numerousB cells and T cells which are constantly produced in the bone marrow and thymus, respectively. The level of B cell simulation depends not only on the success of the match to the antigen, but also on how well

[*] Corresponding author e-mail: msherbiny@cu.edu.eg

it matches other B cells in the immune system. If the stimulation of B cells reaches a certain threshold, the B cell is transformed into a blast that begins to differentiate rapidly, producing clones that turn on a mutation mechanism which generates mutations in the gene coding for the antibody molecule, which is called somatic hypermutation. However, if the stimulation level falls below the threshold, the B cell will not replicate and will die timely. Some special Artificial Immune Systems (AIS) are developed to solve complex optimization problems. One of them is aiNet [15,16] that inspired by biological immune system. Opt-aiNet [15] is an application of aiNetin function optimization. Opt-aiNetconsiders the optimized objective function as antigen, and the candidate solutions as antibodies. The candidate antibodies evolve according to the matching degree between antibodies and antigen that is fitness. The better the matching between them, the less the mutation degree of candidate antibody, and vice verse.AIS has used in many applications such as job shop scheduling problems [17,18]. To improve the solution quality of the FCTP, this paper aims to introduce an Artificial Immune System for solving both balanced and unbalanced FCTP(AISFCTP) without introducing a dummy supplier or a dummy customerand 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 AISFCTP and to compare its performance with hybrid particle swarm method proposed by El-Sherbiny and Alhamali, [24], and the GA proposed in[14,19]. The rest of the paper is organized as follows: in section 2, FCTP is described. In section 3, the proposed AISFCTP is described, and in section 4 the parametric analysis is carried out. Numerical experiments with proposed AISFCTP are presented in section 5. Finally, the conclusion and future work are reported in section 6.

2 Fixed charge transportation problem

Fixed Charge Transportation Problem (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 siunits of supply and each customer $j = 1, 2, ..., n$ demands dj 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 can be represented as follows:

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

$$s.t \quad \sum_{i=1}^{m} y_{ij} \geq b_j \quad forj = 1, ..., n \tag{2}$$

$$\sum_{j=1}^{n} x_{ij} \leq a_j \quad fori = 1, ..., m \tag{3}$$

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

3 The Proposed Algorithm

In this paper, a typical immune algorithm structure is utilized. The algorithm preserves the essential principles of natural immune systems including the cloning, mutation, and clonal selection. The pseudo code of the main steps for the proposed algorithm is presented as follows:

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

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 details of the main steps are adapted in the following subsections.

3.1 *Coding procedure and initialization*

One of the most important issues when designing the AIS lies on its solution (antibody) representation. In order to construct a direct relationship between the problem domain and the AISFCTP, the proposed coding scheme (antibody structure) consists of the set of all the integer numbers in the interval [1, m+n] with any sequence and without any repetition; where the length of scheme is equal to $m + n$, where m is the number of suppliers and n is the number of customers. Therefore, the length of each antibody A_i equal to the sum of the problem dimensions and the suppliers numbers represented by the integer numbers from 0 to m and the demands integer numbers from m +1 to $m + n$. Figure 2 depicts a sample antibody example, which is used to code a 4x5 FCTP such as problem in Table 1. As shown in Figure 1, the cell values are between 1 and 4+5. We can realize that any number can not be repeated.

8	3	9	5	4	7	2	1	6

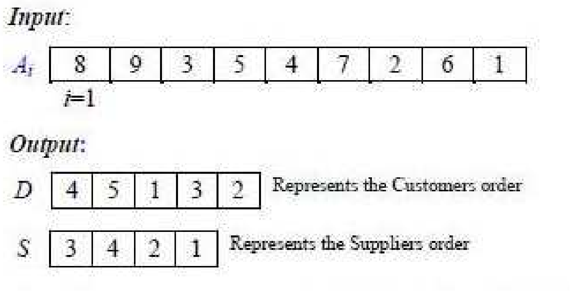
Figure 1. An Example of proposed antibody structure

The population is initialized randomly by applying the coding procedure l times to create l antibodies A_i where l represents the population size. In this algorithm, 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 pseudo code for the coding procedure of creating individual antibody is presented as follows:

```
Step 1: Create a Create collection list $Q = \{1, 2,\ldots, m + n\}$.
Step 2: Set $j = 1$.
Step 3: Generate an integer number between 1 and $m + n$ and set it to variable $c$. Take the cell $A_i(j)$
Step 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($Q$).
Step 5: Add Q[$k$] to the antibody $A_i$in the position $j$.
Step 6: Remove the item $k$ from the list $Q$
Step 7: $j = j + 1$.
Step 8: Repeat from step 3 to step 7 until $j > n + m$ .
Step 9: Return the antibody $A_i$ , where $i$ =1,..., $l$ and $l$ is the population size
```

3.2 *Decoding procedure*

The 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. Figure 2 exhibits the results of applying decoding procedure on the antibody presented in Figure 1.



Input:

A_i	8	9	3	5	4	7	2	6	1
	i=1								

Output:

D	4	5	1	3	2	Represents the Customers order
S	3	4	2	1		Represents the Suppliers order

Figure 2. Illustrativx example of decodirg procedure

The pseudo code for the decoding procedure is presented as follows:

```
Step 1: Set $j = 1$.
Step 2: Take the cell $A_i(j)$
Step 3: If $A_i(j)$ ? $n$ then add $A_i(j)$ to the supplier order $S$.
Step 4: If $A_i(j) > n$ then add $A_i(j)$ to the customer order $D$.
Step 5: $j = j + 1$.
Step 6: Repeat from step 2 to step 5 until $j > n + m$ .
Step 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. 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 the first (2) and the second (3) constraints. Also, this procedure can be used to solve both balanced and unbalanced transportation problems without introducing a dummy supplier or a dummy customer. The pseudo code for the allocating procedure is presented as follows:

```
Step 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 $j = D(1)$.
Step 2: If $a_i = b_j$ then {set $x_{ij}$equal to $a_i$ , remove $S(1)$, and remove $D(1)$}
Step 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 $D(1)$}
Step 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 $S(1)$}
Step 5: Repeat from step 1 to step 4 until (length of queue $S = 0$ or length of queue $D = 0$).
Step 6: Return $x_{ij}$ $\forall$ $i$ =1, 2,..., $m$ and $j$ =1, 2,..., $n$.
```

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

3.4 Evaluating the Solutions

Each antibody is evaluated to determine its fitness. As mentioned above each antibody is decoded using the decoding procedure and its result used as an input for the allocating procedure. The solution resulted from the allocating procedure 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 AISFCTP, six different mutation methods are developed and tested.

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

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

The second MF is based on generating a random number *NS* ∈ [1, *n* + *m*]. Therefore generating two random numbers (*j* and *k*) representing two different positions and swap them. This process is known as two-point-swap. This two-point-swap is performed *NS* times. The number of swaps (*NS*) for this mutation is represented in (5)

$$NS = Rand(1, n\ + m) \tag{5}$$

The third MF is based on a uniform random number located in the range of 10% to 30% of the sum of problem dimensions (*n* + *m*). The number of swaps (*NS*) for this mutation is represented by (6), where *r* is a random number in the interval [0.1, 0.3].

$$NS = Rand(1, r(m{+}n)) \tag{6}$$

The fourth MF is based on time where more is the time elapsed; less will be the number of swaps. First start with applying random number of two-points-swap till a pre-defined ratio of time is elapsed. After that the two points swap MF is applied for the remaining time. The time is represented by the ratio of current iteration to the total number of iterations.

The fifth MF is based on applying either non uniform swap times (the third *MF*) or *Inverse Swap* MFs. A random number *r* ∈[0, 1] is generated and if *r* > pre-defined value *v*, then the non uniform swap time will be applied; else *Inverse Swap MF* will be applied. The inverse swap MFs is applied through converting the order of the antibody positions.

The remaining five mutation methods are functions of two parameters. The first parameter is the non-uniform factor based on which the number of swaps is determined. The second parameter is the degree of non-uniformity (*u*). All the functions are designed to be directly related with *u* [24,25]

The sixth mutation method 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 normalized fitness closer to one, i.e. relatively bad solutions, will be subject to more number of swaps. This actually gives the chance for low affinity solutions to mutate more in order to improve their affinities. The number of swaps for this mutation function is adapted with (7) and the normalized fitness of each antibody is calculated using (8).

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

$$NF = \frac{LowestFitness - Fitness}{LowestFitness - HighestFitness} \tag{8}$$

The seventh mutation method is designed to be inversely related with the ratio (*T*) of the current iteration number (*CIN*) and the total number of iterations (*TNI*). That is, the more the search goes, the less the number of swaps is. This is really intuitive as in contrast to the first stages of the search where a real exploration of the search space through significant changes in the solutions are required, at the last stages of the search fine tuning with little changes of the supposed-to-be near-optimal solutions is more reasonable. The number of swaps (*NS*) for this mutation is represented in (9) where *u* is the degree of non-uniformity.

$$NS = MNS^{(1-(T^{u})}\ ,\ \text{where } T = \frac{CIN}{TNI} \tag{9}$$

The eighth mutation method is based on both the time and the normalized fitness of the solution. It basically uses the average of these two factors to decide the number of swaps. Basically, the function is designed to be directly related with the fitness but inversely related with the time. The average of Time and normalized Fitness (*TF*) is calculated as represented in (10) and the number of swaps for this mutation is adapted with (11).

$$TF = \tfrac{1}{2}(NF + (1 - T) \tag{10}$$

$$NS = MNS^{(1-(1-TF)^{u})} \tag{11}$$

In the *ninth* and the *tenth mutation methods*, we include a random factor (*R*) so that the number of swaps 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 functions behave almost the same way as the original ones when

Rand is close to zero. The closer the R to one is, the closer the number of swaps to the max swaps no is. These two mutation functions are suggested to allow the search to escape from local optima by occasionally increasing the number of swaps. The number of swaps for these two mutations is adapted with (12) and (13), respectively.

$$NS = MSN \times R^{(1-NF)^{u}} \tag{12}$$

$$NS = MSN \times R^{(T^{u})} \tag{13}$$

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. This is technically applied to reduce the chance of a premature convergence to local optima.

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 affinity function of two antibodies A_j and A_k is represented as in (14).

$$AF(A_j,\ A_k) = \sum_i y_i$$
$$\text{where } y_i = \begin{cases} 1 \text{ if the } i^{th} \text{ varible of } A_j & = \text{the } i^{th} \text{ varible of } A_k \\ 0 & \text{Otherwise} \end{cases} \tag{14}$$

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 proposed algorithm eliminates those solutions that have *AF* more than a specific parameter -Number of Similarities (*NS*).

4 Parametric analysis

In this section, two techniques are used in order to discover the best mutation function from the tenth implemented. Which are the Relative Percentage Deviation (*RPD*) and the paired sample t-test. Because the scale of the objective functions in each problem is different, they could not be used directly. Therefore, the *RPD* is used for each combinatioin [21]. *RPD* is calculated by using (15).

$$RPD = \frac{A\lg_{sol} - Min_{sol}}{Min_{sol}} \times 100 \tag{15}$$

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 mutation function from the implemented seven. The characteristics of these problems are presented in Table 1.

Table 1: Characteristics of FCT test problems

Problem size	Total supply	Rang of variable costs		Rang of fixed costs	
		Lower Limit	Upper Limit	Lower Limit	Upper Limit
14×18	310	1	10	90	250
5×10	1,500	1	8	100	400
10×10	3,000	1	8	150	400
10×20	4,000	1	10	100	500
30×30	7,000	1	10	10	100

Table 2: 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	11.9	3.1	8.7	3.7	5.8
2	2	3.4	7.9	10.4	17.1	8.2
3	1.7	4.7	5.5	8	10.4	6.1
4	1.7	4.9	7.2	8.7	16.1	7.7
5	2.8	7	8.5	13.1	18.1	9.9
6	0.3	4.5	1.6	2.9	2.9	2.4
7	0.2	0.0	0.0	0.0	0.0	0.0
8	0.1	5.2	1.4	1.9	1.2	2.0
9	1	8.1	2.7	6	3	4.2
10	0	2.6	0.1	2.2	0.8	1.1

As illustrated in Table 2, the quality of the results of using the seventh mutation method is very close to the tenth mutation method and both are superior to the others. But the seventh mutation method is most superior. Therefore, in the next section, the seventh mutation method will be used in our comparison with the most recent algorithms in the literatures.

In addition to the above, in order to statistically test the significance of effectiveness of the results using different mutation methods, the paired sample t-tests were used to determine the significant differences in the *RPD* values obtained using the ten MFs, for each of the pairs. For the purpose of comparisons the *RPD* values obtained using all the 50 problems were used. The results of the tests are summarized in Table 3.

Inputs:

D	4	5	1	3	2	Represents the Customers order

S	3	4	2	1	Represents the Suppliers order

Processing:

	D1	D2	D3	D4	D5	a_i	Ta_i
S1						57	57
S2						93	93
S3				50		50	
S4						75	75
b_j	88	57	24	73	33	275	
Tb_j	88	57	24	23	33	225	

	D1	D2	D3	D4	D5	a_i	Ta_i
S1						57	57
S2						93	93
S3				50		50	
S4				*23*		75	52
b_j	88	57	24	73	33	275	
Tb_j	88	57	24		33	202	

	D1	D2	D3	D4	D5	a_i	Ta_i
S1						57	57
S2						93	93
S3				50		50	
S4				23	*33*	75	19
b_j	88	57	24	73	33	275	
Tb_j	88	57	24			169	

	D1	D2	D3	D4	D5	a_i	Ta_i
S1						57	57
S2						93	93
S3				50		50	
S4	*19*			23	33	75	
b_j	88	57	24	73	33	275	
Tb_j	69	57	24			150	

	D1	D2	D3	D4	D5	a_i	Ta_i
S1						57	57
S2	*69*					93	24
S3				50		50	
S4	19			23	33	75	
b_j	88	57	24	73	33	275	
Tb_j		57	24			81	

	D1	D2	D3	D4	D5	a_i	Ta_i
S1						57	57
S2	69		*24*			93	
S3				50		50	
S4	19			23	33	75	
b_j	88	57	24	73	33	275	
Tb_j		57				57	

	D1	D2	D3	D4	D5	a_i	Ta_i
S1		*57*				57	
S2	69		24			93	
S3				50		50	
S4	19			23	33	75	
b_j	88	57	24	73	33	275	
Tb_j							

Output:

The final allocation is:

	D1	D2	D3	D4	D5	a_i
S1		57				**57**
S2	69		24			**93**
S3				50		**50**
S4	19			23	33	**75**
b_j	**88**	**57**	**24**	**73**	**33**	**275**

Figure 3. Illustrative example of applying allocation algorithm

Table 3: The p-values of paired sample t-tests of the mutation functions (Based *values*)

MF	10	9	8	7	6	5	4	3	2
1	0.004	0.018	0.003	0.000	0.000	0.012	0.003	0.006	0.003
2	0.280	0.057	0.170	0.000	0.315	0.023	0.372	0.055	
3	0.049	0.016	0.166	0.002	0.615	0.016	0.071		
4	0.739	0.029	0.292	0.000	0.513	0.011			
5	0.020	0.714	0.016	0.000	0.002				
6	0.639	0.011	0.882	0.000					
7	0.000	0.000	0.000						
8	0.348	0.020							
9	0.029								

Based on the data presented in Table 3 and Figure 4, it can be concluded at 0.01 level of significance the quality of the results using the seventh MF is very close to the tenth one and both are superior to the others. But the seventh MF is most superior. This corroborates the results obtained based on the *RPD* analysis. Therefore, in the next section, the seventh MF will be used in comparing with the by El-Sherbiny and Alhamali [23], M. Hajiaghaei et al. [14] methods for solving FCTPs.

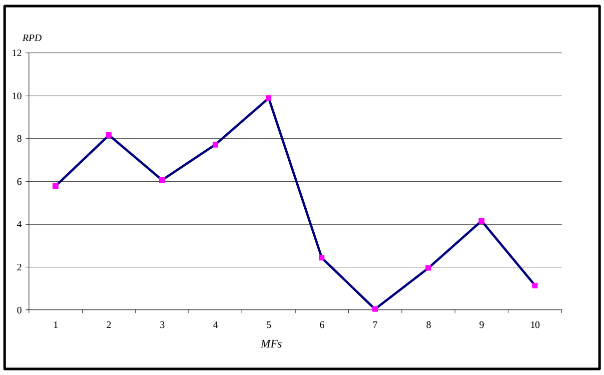


Figure 4. Fitted mean Plot for *RPD* at each MF

5 Numerical experiments

To evaluate the performance of the proposed AISFCTP algorithm two problems with different sizes, previously addressed by M. Hajiaghaei et al. [14] and El-Sherbiny and Alhamali,[23] are solved, comparing with the solution presented by them. The sizes of the problems are 4×5 and 5×10, respectively. The variable costs, and the Fixed costs for first problem is given in Table 4. The parameters used for the proposed method in these problems are optimally tuned parameters and operators from experimental results.

Concerning the first problem, the obtained local optimal solution from proposed algorithm is the same as the solution found by M. Hajiaghaei et al. [14], and El-Sherbiny & Alhamali [23], is equal to 1484 for this problem. The transportation allocation matrix for this solution is shown in Table 5.

Table 4: Data of the first problem [14,23].

Plants	a_i	Costumers									
		Shipping costs c_{ij}					Fixed costs f_{ij}				
		1	2	3	4	5	1	2	3	4	5
1	57	8	4	3	5	8	60	88	95	76	97
2	93	3	6	4	8	5	51	72	65	87	76
3	50	8	4	5	3	4	67	89	99	89	100
4	75	4	6	8	3	3	86	84	70	92	88
						b_j	88	57	24	73	33

Table 5: Transportation allocation matrix found in [14,23] and the proposed AISFCTP for the first problem.

	D1	D2	D3	D4	D5	a_i
S1		57				57
S2	69		24			93
S3				50		50
S4	19			23	33	75
b_j	88	57	24	73	33	

Table 6: Data of the second problem [14][23]

Plant		**Costumers**									
		Shipping costs c_{ij}									
	a_i	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
1	157	8	4	3	5	2	1	3	5	2	6
2	293	3	3	4	8	5	3	5	1	4	5
3	150	7	4	5	3	4	2	4	3	7	3
4	575	1	2	8	1	3	1	4	6	8	2
5	310	4	5	6	3	3	4	2	1	2	1
	bj	225	150	90	215	130	88	57	124	273	133
		Fixed costs f_{ij}									
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
1		160	488	295	376	297	360	199	292	481	162
2		451	172	265	487	176	260	280	300	354	201
3		167	250	499	189	340	216	177	495	170	414
4		386	184	370	292	188	206	340	205	465	273
5		156	244	460	382	270	180	235	355	276	190

Concerning the second problem, the variable costs, the fixed costs and the supplies and demands from each of the plant 1 to 5 and for each customer 1to10 are given in Table 6.

The obtained local optimal solution for this problem by the proposed algorithm is 6255**,** the solution found by El-Sherbiny & Alhamali [23] is 6296, while the solution found by M. Hajiaghaei et al. [14] is 6305. The transportation allocation matrixes for Hajiaghaei et al. [14] method, El-Sherbiny & Alhamali (2012) [23] method and the proposed AISFCTP algorithm for the second problem are shown in Tables 7,8,9 respectively.

Table 7: Transportation allocation matrix found by M. Hajiaghaei et al. [14] for the 2^{nd} problem [14].

	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	a_i
S1			90		67						157
S2		150						124		19	293
S3				5		88	57				150
S4	225			210	63					77	575
S5									273	37	310
b_j	225	150	90	215	130	88	57	124	273	133	

Table 8: Transportation allocation matrix for the 2^{nd} problem found in [23].

	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	a_i
S1					130		27				157
S2		15	90					124		64	293
S3						88	30		32		150
S4	225	135		215							575
S5									241	69	310
b_j	225	150	90	215	130	88	57	124	273	133	

Table 9: Transportation allocation matrix found by the proposed AISFCTP for the 2^{nd} problem.

	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	a_i
S1			27		130						157
S2		106	63					124			293
S3						88	57		5		150
S4	225	44		215						91	575
S5									268	42	310
b_j	225	150	90	215	130	88	57	124	273	133	

6 Conclusion

This paper has proposed an artificial immune algorithm for solving fixed charge transportation problem (AISFCTP). AISFCTP solves both balanced and unbalanced FCTP without introducing a dummy supplier or a dummy customer. In AISFCTP the presented schema, decoding and allocation procedures are used instead of spanning tree used with genetic algorithm. While using the spanning tree and Prfer number with the GA may result in non-feasible solutions, the AISFCTP guarantees the feasibility of all the generated solutions. In addition, ten mutation functions are developed and used in AISFCTP. Due to the significant role of mutation function on the AISFCTP's quality, its performances are compared to select the best one. Also, the comparative study of the AISFCTP with the hybrid particle swarm method presented by El-Sherbiny & Alhamali [23], and M. Hajiaghaei et al. [14] showed that the proposed algorithm (AISFCT) is superior to the others. The performance of AISFCTP and the solution quality prove that AISFCTP is highly competitive and can be considered as a viable alternative to solve FCTPs.

Future work includes further experimentation with parameters of AISFCT, testing the proposed AISFCTP on other real life problems, and investigating using other metaheuristic techniques combined with the proposed decoding and allocation procedures for solving problems.

Acknowledgement

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

References

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

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

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

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

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

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

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

[10] Gen, M., K. Ida, and Y. Z. Li, Bicriteria transportation problem by hybrid genetic algorithm, Computers and Industrial Engineering, **35**, 363-366 (1998).

[11] Gottlieb, J., B. A. Julstrom, F.Rothlauf, and Raidl, G.R.: Prfer 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).

[12] Raidl, B. A., Julstrom, G. R., Edge-sets: An effective evolutionary coding of spanning trees, IEEE Transactions on Evolutionary Computation, **7**, 225-239 (2003).

[13] Su, S., Zhan, D. C., New genetic algorithm for the fixed charge transportation problem, Proceedings of the 6th World Congress on Intelligent Control and Automation, 7039-7043 (2006).

[14] Hajiaghaei, M. Keshteli, S. Molla-Alizadeh-Zavardehi, R. Tavakkoli-Moghaddam, Addressing a nonlinear fixed-charge transportation problem using a spanning tree-based genetic

algorithm, Computers & Industrial Engineering, **59**, 259-271 (2010).

[15] Timmis, J., 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).

[16] De Castro, L. N. and J. Timmis, An artificial immune network for multimodal function optimization in Proc. of IEEE congress on evolutionary computation, **1**, 699-674 (2002).

[17] OrhanEngin, AlperDyen, A new approach to solve hybrid flow shop scheduling problems by artificial immune system, Future Generation Computer Systems, **20**, 10831095 (2004).

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

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

[20] Kannan, G., Sasi Kumar, P., & Vinay, V. P., Comments on Nonlinear fixed charge transportation problem by spanning tree-based genetic algorithm by Jung-Bok Jo, Yinzhen Li, Mitsuo Gen, Computers & Industrial Engineering (2007).Computers & Industrial Engineering, **55**, 533-534 (2008).

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

[22] Zalinda Othman, Mohammad-Reza RostamianDelavar, Sarah Behnam, SinaLessanibahri, Adaptive genetic algorithm for fixed-charge transportation problem, International Multi Conference of Engineers & Computer Sciences (IMECS) March, 16-18 (2011).

[23] El-Sherbiny, M. M. , Alhamali, R. M, A Hybrid Particle Swarm Method with Artificial Immune Learning for Solving the Fixed Charge Transportation Problem,Computers & Industrial Engineering, **64**, 610-620 (2013)

[24] El-Sherbiny, M. M., Ibrahim, Y. M. (2012). An artificial immune algorithm with alternative mutation methods: applied to the student project assignment problem, International Conference on Innovation and Information Management (ICIIM2012), Chengdu, China, January, 7-8 (2012).

[25] Altassan, Khaled. M., El-Sherbiny, Mahmoud. M., Ibrahim, Yasser. M., Abid, Ahmed. D. (2012). A novel artificial immune algorithm for solving fixed charge transportation problems, International Conference on Innovation and Information Management (ICIIM2012), Chengdu, China, January, 7-8 (2012).

Khalid M. Altassan received the MS degree in Accounting from Oklahoma State University in 1988, other M.S. in Quantitative Business Analysis from Louisiana State University in 1990 and received the PhD degree in Engineering -Industrial and Management from Engineering- Rensselaer Polytechnic Institute in 1997. His research interests are in the areas of Automated storage and Retrieval Systems, Mathematical Application in Scheduling, Quantitative analysis in Business and Economics, Operations Research Applications in Managerial Accounting.

Mahmoud M. El-Sherbiny received the MS & PhD degrees in Operations Research form Institute of Statistical Studies & Research (ISSR), Cairo University, Egypt in 1996 and 2002 respectively. He is a professor in ISSR, Cairo University, Egypt. His research interests are in the areas of Computational Intelligence, Particle Swarm, Immune Systems, Quantum computing, Multiobjectives, Fuzzy Systems, Transportation and Scheduling.

Ahmed Darwesh Omar Abid received the MS degree in Applied Statistics from Purdue University, USA in 1980 and received the Ph.D. in Statistics from Oklahoma State University, USA in 1985. He is currently a Professor at King Saud University, Riyadh, Saudi Arabia. His research interests are in the areas of Sequential Analysis, Demography and Actuarial Science.