Immune Genetic Algorithm for the Fixed Charge Transportation Problem

Xiaoke Ma[1] **Yan Wang**[2] **Tao Yang**[1,3] **Yuanping Zhang**[1]

[1] Department of Computer and Communication, Lanzhou University of Technology, Lanzhou 730050, P. R. China
[2] Department of Library, Northwest University for Nationalities, Lanzhou 730020, P. R. China
[3] College of Business Administration, Liaoning Technical University, Huludao 125105, P. R. China

Abstract

An immune genetic algorithm (IGA) for the fixed charge transportation problem is developed based on the immune theory in biology, which constructs an immune operator accomplished by two steps, a vaccination and an immune selection. The methods for selecting vaccines and constructing an immune operator are also proposed. The results of computation demonstrate that IGA can restrain the degenerate phenomenon and improve the search capability compared to genetic algorithm with matrix code and that with edge-set code greatly on large instances.

Keywords: Genetic algorithm, Fixed charge transportation problem, Artificial immune system, Combination optimization

1. Introduction

The fixed charge transportation problem (FCTP) has been studied extensively because of the considerable merit of practice, even though it is a NP-hard problem [1]. It is often formulated and solved as a mixed integer network programming problem. However, these methods are not employed usually because of the inefficient and expensive computation. All the algorithms for the FCTP can be divided into two categories: exact algorithms and heuristic methods.

The well known exact algorithms for FCTP include vertex ranking method [2] and branch-and-bound method [3] etc. But these methods are generally not very useful when the size of FCTP reaches a certain level. Thus, some heuristic methods have been developed. Such as Lagrangian relaxation method, adjacent extreme point search method [4] and simulation annealing algorithm (SAA) [5] and so on. Although these methods are usually computationally efficient, one of the major shortcomings with heuristics is the high possibility of terminating at a local optimum that far away from the global one. During several past decades, there has been a growing interest in algorithms that rely on analogies to natural phenomena such as evolution, heredity, immunity and so on. Classic algorithms include evolutionary algorithm (EA) [6], immune algorithm (IA) [7], ant algorithm (AA) [8] etc.

As an active branch of evolutionary algorithms, genetic algorithm (GA) is population-based stochastic optimization method with an iterative process of generation-and-test. It has been recognized that GA is promising approach for NP-hard or NP-complete problems such as graph coloring problem (GCP), maximal stable subset problem (MSSP), FCTP [9]. However, when GA used for a given problem, two main genetic operators—-crossover and mutation, not only give each individuals the evolutionary chance to obtain global optimum but also cause the degeneracy to some extent because of the random and unsupervised searching during the entire process. On the other hand, GA is lack of capability of making use of some basic and obvious characteristic or knowledge in pending problem. That is why GA opts to be trapped in local optimum with high probability. Based on the consideration above, a novel IGA is proposed with combining the concept of immunity in biology.

The paper is organized as follows:In Section 2 FCTP is described. Concrete algorithm and immune operator are proposed in Section 3 and Section 4. The outcomes of IGA is outlined in Section 5. Finally, a conclusion is presented in Section 6.

2. FCTP

In FCTP, two types of costs are considered simultaneously when the best course of action is selected, a variable cost proportional to the activity level and a fixed cost. The objective of an FCTP is to find the combination of routes that minimizes the total variable and fixed costs while satisfying the supply and demand requirements of each origin and destination. The FCTP may be stated formally as:

$$\min \sum_{i \in I} \sum_{j \in J} (c_{ij} x_{ij} + f_{ij} g(x_{ij})) \qquad (1)$$

with

$$g(x_{ij}) = \begin{cases} 1, & \text{if } x_{ij}>0 \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

$$\sum_{j \in J} x_{ij} = a_i \quad \forall i \in I \quad (3)$$

$$\sum_{i \in I} x_{ij} = b_j \quad \forall j \in J \quad (4)$$

$$x_{ij} \geq 0 \quad \forall i \in I, j \in J \quad (5)$$

with $I = \{1, 2, \cdots, m\}$ and $J = \{1, 2, \cdots, n\}$ being the sets of plants and customers, respectively. i and j are indices for plants and customers, and x_{ij} for the amount to be shipped from i to j. The capacity of i and the demand of j are denoted by a_i and b_j respectively. Equations(3) and (4) the transportation constraints must be satisfied when a schedule is made. Without loss of generality, we suppose that $\sum_{i \in I} a_i = \sum_{j \in J} b_j$.

3. Immune genetic algorithm

3.1. Initialization population

The fundamental design choices in an evolutionary algorithm are its representation of candidate solution and the operators that will act on that representation. There are many manners to code the feasible solution for FCTP: Gottlieb and Michalewicz (1991) analyzed some representation schemes for the linear transportation problem [10]. They claimed that bitstrings code is not suited to represent solution candidates, and instead propose GA employing permutations code or matrices code. Gottlieb and Paulmann (1998) showed that GA with matrices code superior to that with permutation code for FCTP [11]. Raidl et al (2003) developed a new edge-set(ES) representation to encode a spanning tree for GA, which is efficient on large instances compared with other representations [12]. The tabu alogrithm developed by Sun et al (1998) is the best one on the capacity of searching for FCTP at present [13].

Taken edge-set to encode antibody (feasible solution) for FCTP for its excellent performance on large instance, its initial population consists of genotypes that represent random spanning trees. Prim and Kruskal's minimum spanning tree algorithms are classic ones for spanning trees even though they do not associate uniform probabilities with spanning tree. Here we employ extending Prim algorithm to obtain an antibody by repeatedly appending the lowest cost edge that joins a new node to the growing tree. $T = (V_T, E_T)$ denotes the antibody under being constructed; $D(u)$ denotes the capacity or demand constraints of u plant or customer; P_T all the vertices in V_T but without meeting the constraint. random(1,m+n) denotes the function which produces an integer between 1 and m+n. Initialization procedure can be described as:

$u = random(1, m+n)$;
$counter = 0$;
If $T = \emptyset$ Then
  $u = random(1, m+n)$;
  $V_T = V_T \cup u$;
  $P_T = P_T \cup u$;
Else If
  Search T to obtain P_T;
End If
while counter >m+n
  If $P_T = \emptyset$ Then
    Select a node u from P_T randomly;
  Else If
    Select a node u from V_T randomly;
  End If
Select randomly an edge (u, v) from the set E where $v \in V - V_T$;
$x_{u,v}$=min (D_u, D_v);
$D_u - = x_{u,v}$;
$D_v - = x_{u,v}$;
counter+=1;
  If $D(u) = 0$ and $u \in P_T$ Then
    $P_T - = u$;
  Else If $D(v) \neq 0$ Then
    $P_T + = v$;
  End If
End while

The initialization procedure can produce a random, feasible spanning tree that the quanta on edges meet the constraints of Equations (3) and (4) with time complexity $\Theta(m+n)$.

3.2. Crossover and mutation

Supposed that two parent antibodies $T_1 = (V, E_1)$ and $T_2 = (V, E_2)$ are selected for the crossover operation. The offspring T_3 can be produced by the following algorithm called as k_1-crossover:

Select randomly k_1 independent edges from T_2;
counter=1;
While counter $>k_1$
  If the $i-th$ edge $(u, v) \notin E_1$ Then
    Insert the edge (u, v) into antibody T_1;
    $x_{u,v} = 0$;
    Make a repair operation with repair(u, v);
    counter++;
  Else
    counter++;
  End If
End while

when $k_1 = 1$, k_1-crossover operator is a single crossover operation, it is multiple points crossover

when $k_1 \geq 2$. The assignment of k plays an important role in IGA. The value of k is large at beginning of algorithm just to increase the diversity of population, and k becomes small at the end of IGA just to accelerate the convergence of search procedure. Because a new edge (u, v) is inserted into spanning tree, a repair operation must be employed to eliminate a cycle caused by (u, v). The concrete complement of repair operator is presented latter.

Supposed that the antibody $T_1 = (V, E_1)$ and $k_2(1 \leq k \leq m + n)$ are selected for mutation operation. The offspring individual T_1^* could be produced by k_2-mutation, which works as following:

counter=0;
While counter $<k_2$
Select randomly two nodes u, v where $(u, v) \notin E_{T_i}$;
Insert edge (u, v) into T_1;
Make a repair operation with repair(u, v);
counter++;
End While;

Generally speaking, $k_2 = 1$ usually is most popular choices in mutation operation. However, we can increase the probability to search global optimum through changing k_2.

3.3. Repair operator

Any antibody in population for FCTP must meet the constraints from two aspects: 1) antibody must be a spanning tree; 2) the transportation amount on each edge should satisfy constraints on Equations (3) and (4). However, Both the crossover operator and mutation operation will destroy the constraints from both aspects. For the former could be removed by deleting an edge in the cycle quickly and easily. The main task in repair operation is to re-distribute the quantum on each edge to maintain the capacity or demand constraints. Supposed that u, v and operation antibody T_1 are ascertained:

Step 1 Get a path ϑ started with u, ended with v by making a deep-first search on T_1;

Step 2 $x_{rs} = \min\{x(r, s)|(r, s) \in \vartheta, Dis(u, (r, s)) \mod 2 = 0\}$. If x_{rs}=0, then $x_{rs} = \max\{c_(r, s)|Dis(u, (r, s)) \mod 2 = 0, (r, s) \in \vartheta\}$.

Step 3 For each edge $(u^{'}, v^{'}) \in \vartheta$, if $Dis(u, (r, s)) \mod 2 = 0$, then $x_{u^{'}v^{'}} - = x_{rs}$; else $x_{u^{'}v^{'}} + = x_{rs}$;

Step 4 Delete the edge (r, s). A new antibody T_1^* is constructed.

where $Dis(u, (r, s))$ denotes the minimum distance from u to edge (r, s) on path ϑ with the direction from u to v. For instance, $\vartheta = \{(u, u_1, (u_1, u_2), (u_2, u_3), (u_3, v))\}$, $Dis(u, (u_2, u_3)) = 2$. Such a repair operation can reconstruct a feasible antibody with time complexity $\Theta(m + n)$.

3.4. Vaccination and immune selection

The immune operator consists of two steps, i.e. a vaccination and an immune selection, of which the former is used for improving the fitness and the latter is for preventing degeneracy. A vaccination means the course of modifying the genes of an individual T on some edges with prior knowledge so as to gain higher fitness with greater probability. Supposed that population with size of p_s and a constant α are given, then vaccination on α means there are αp_s individual are carried out operation. The immune selection is also accomplished by two components. Firstly, an immune test is taken to determine whether a antibody can be included into next generation population according to the the fitness. If the antibody's fitness less than its parent, the individual is rejected because there is a degeneracy happened on crossover or mutation. Otherwise, it is included. Secondly, selecting an individual T_i in the present offspring to participate into the next generation with probability:

$$P(T_i) = \frac{\ln \frac{f(T_i)}{t_k}}{\sum_{i=1}^{P_s} \ln \frac{f(T_i)}{t_k}} \tag{6}$$

where $f(T_i)$ is the fitness of antibody T_i and t_k is the temperature-controlled series towards 0.

4. Immune operator

The immune operator utilizes vaccines to intervene the variation of genes in antibody. However, the performance of IGA largely depends on the validity of vaccine. The best method to abstract vaccine is to make a detailed analysis on the pending problem and find out basic characteristics as much as possible. Then, these characteristics are abstracted as schema used as the basis of the immune operator. Unfortunately, not all the problems can be distilled particular characteristics because of lacking of mature prior knowledge.

FCTP is a classic NP-hard problem without any mature prior knowledge. Thus, we can abstract information from genes of the present optimal and the worst individuals to make vaccines during evolutionary process. The construction for vaccines can be obtained by comparing the optimum antibody $T^{optimal}$ with the worst one T^{worst}, which works as: If an edge $(u, v) \in T^{optimal}$ and $(u, v) \notin T^{worst}$, then $H+ =$

(u, v, x_{uv}) as positive vaccine; If The $(u, v) \notin T^{optimal}$ and $(u, v) \in T^{worst}$, (u, v) is also included as negative vaccine. When an antibody T_i is selected for vaccination operation, select randomly a vaccine (u, v, x_{uv}). If $(u, v) \in T_i$ and the status of vaccine (u, v) is negative, then delete the edge (u, v) or decrease the value of x_{uv}; If $(u, v) \notin T_i$ and the status of vaccine is positive is positive, then add the edge (u, v) into antibody T_i. The whole algorithm with self-adaptive abstracting vaccines can be executed as:

g=0;

While condition=true

Select the optimum $T_g^{optimal}$ and the worst T_g^{worst} antibodies according to fitness;

For each edge $(u, v) \in T_g^{optimal}$ and T_g^{worst}

If $(u, v) \in T_g^{optimal}, (u, v) \notin T_g^{worst}$ Then

H+=((u, v),x_{uv},positive);

Else If $(u, v) \notin T_g^{optimal}, (u, v) \in T_g^{worst}$

H+=((u, v),x_{uv},negative);

End If

For i to P_s

If $P_v = true$

R=random(1, length of H);

Vaccination:

If $(u, v) \in H_R$ and $H_R.status = positive$ and $(u, v) \notin T_i$ Then

$T_i^* = T_i + (u, v)$;

Make a repair on T_i^* with repair(u, v);

Else If $(u, v) \in H_R$ and $H_R.status = negative$ and $(u, v) \in T_i$ Then

T_i^* constructed by decreasing the the value of x_{uv};

Make a repair to keep the feasibility of T_i^*;

End If

Immune test:

If $T_i^* < T_i$ Then

T_i^* is included for next generation;

Else

T_i^* is rejected;

Annealing selection;

g++;

End while

5. Computational results

A computational experiment is designed to test the performance of the IGA for the FCTP. In this experimental design, the performance of the procedure is evaluated by the solution quality. The genetic algorithm with matrix code (GA-M) developed by Vignaux et al [10] and the genetic algorithm with edge-set code (GA-ES) proposed by Raidl et al [12] are selected as references against which to gauge the performance of IGA. GA-M is selected because GA-M is the high efficient GA for FCTP. And GA-ES is selected because of the excellent performance on large instances. Other GAs are not included because them are proved to be inefficient, i.e. Vignaux et al [11] proved that GA with bitstrings code is inferior to GA-M and Raidl et al [12]showed that GA-ES is superior to GA with Blob Code and so on. Only small and easy test problems were solved with exact algorithm because of the excessive computational time that is why exact algorithm is not employed.

In the computational tests, seven problem sizes, measured by $n \times m$, are used: 10×10, 20×20, 30×30, 40×40, 50×50. All the problems are fully connected and fully fixed cost dense. The total supply(demand) for each problem size are given in Table 1. Within each problem size, five problem types(A-E) are employed. The differences among these problem types are tin their ranges of fixed costs, which are listed in Table 2. For each problem type, 15 test problems are randomly generated and solved.

Problem size	Total supply
10×10	2500–5000
20×20	5000–10000
30×30	10000–15000
40×40	15000–20000
50×50	20000–30000

Table 1: Total supply(demand) for problem sizes.

Problem type	Lower limit	Upper limit
A	0	400
B	200	800
C	400	1600
D	800	3200
E	1600	6400

Table 2: Ranges of fixed cost for instances types.

The quality of solution obtained by three algorithms

$$\eta = (\frac{z^*}{z^{optimal}} - 1) \times 100 \quad (7)$$

where z^* is the objective function value of the best solution obtained by the GA-M, GA-ES and LS, and $z^{optimal}$ is the optimum function values obtained by three algorithms. Objective function values of these problems obtained with GA-M, GA-ES are given in Table 3 as a percentage of those of the optimal solutions, in terms of worst and best for each problem size and type. If the solution get an optimum solution, this percentage is 0, otherwise, greater than 0.

Problems		GA-M		GA-ES		IEA	
Size	Type	Worst	Best	Worst	Best	Worst	Best
10×10	A	.047	▲	.011	.011	.032	.010
	B	.050	.005	.053	.053	.045	▲
	C	.062	▲	.101	.092	.070	.002
	D	.120	▲	.246	.246	.089	.039
	E	.077	.026	.119	.119	.066	▲
20×20	A	.029	.017	.000	▲	.032	.019
	B	.045	.023	.000	▲	.019	▲
	C	.046	.023	.000	▲	.049	.023
	D	.020	▲	.040	.040	.018	.004
	E	.055	.016	.080	.080	.058	▲
30×30	A	.008	.000	.007	.007	.012	▲
	B	.046	.030	.000	▲	.048	.032
	C	.038	.007	.010	.010	.031	▲
	D	.013	.002	.013	.013	.006	▲
	E	.108	.049	.065	.065	.070	▲
40×40	A	.079	.067	.000	▲	.076	.070
	B	.031	.022	.000	▲	.024	.010
	C	.032	.019	.000	▲	.024	.017
	D	.021	▲	.063	.063	.023	.015
	E	.026	.001	.018	.018	.018	▲
50×50	A	.090	.083	.002	▲	.087	.073
	B	.006	.003	.001	▲	.005	.001
	C	.031	.016	.050	▲	.030	.013
	D	.049	.038	.001	▲	.004	▲
	E	.027	.007	.025	.004	.002	▲

Table 3: The comparison on the capacity of searching among GA-M, GA-ES and IGA, where ▲ represents the optimal solution, and .* represents the value 0.*.

In simulation, the single-point crossover is adapted, whose positions are selected randomly. 1-mutation is accepted for mutation operation. The size of population P_s=100. The probability for crossover P_c=0.85. That for mutation P_m=0.15. The proportion for vaccination operation α =0.1. The maximal iteration number is 1000.

In the immune operator, we abstract the vaccines with the comparison between the optimum and the worst antibodies. After every genetic operation, we select some antibodies for vaccination according to the immune probability P_i. During the procedure, the annealing temperature t_k in the immune selection is calculated as follows

$$t_{g+1} = \beta t_g, t_0 = 100 \tag{8}$$

where g is the generation and $\beta = 0.95$.

The comparison on the capacity of searching among three algorithms is presented in Table 3 indicates

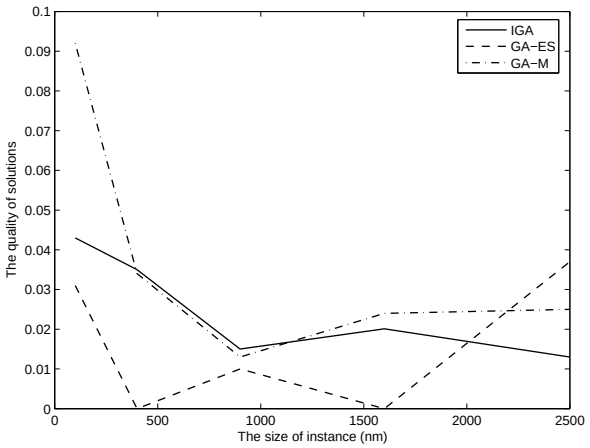

Fig. 1: The comparison on the capacity of searching in term of average case with type C.

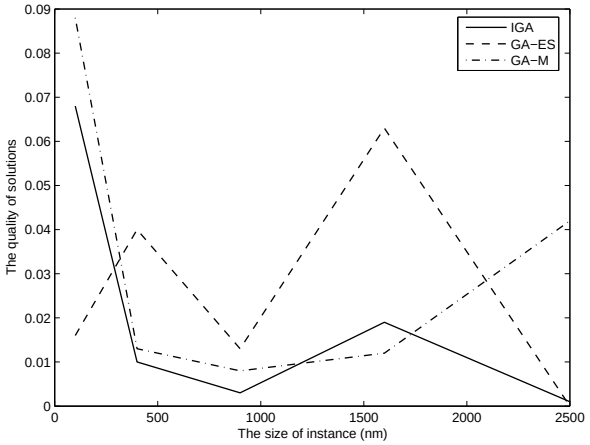

Fig. 2: The comparison on the capacity of searching in term of average case with type D.

that: GA-M algorithm is more efficient than GA-ES and IGA for the small instance such as 10×10. The superiority of GA-ES is demonstrated at the moderate instances for example, in 20×20, but it is not effective to handle with the types D and E. That also tallies with Raidl's experiments [12]. We can also concluded that IGA algorithm is most efficient one with large instance and types D and E.

On the average capacity of searching: From Fig.1 we can obtain that the GA-M is best algorithm with small size of instance at types C, but it is inefficient with types D and E from Fig.2 and Fig.3. The gap between GA-ES and IGA is not obvious on types C and D from Fig.1 and Fig.2. However, it can be asserted that IGA is superior to GA-ES from Fig.3. The performance of

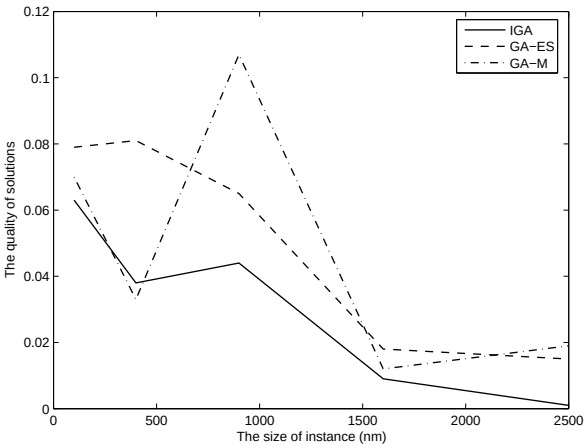

Fig. 3: The comparison on the capacity of searching in term of average case with type E.

GA-ES is inferior to GA-M on small size with type C. However, GA-ES predominates GA-M on types D and E.

6. Conclusion

An IGA for the FCTP is developed, which get the immune mechanism and the evolutionary mechanism together. The methods of selecting vaccines and constructing immune operator are also proposed. The results of computation demonstrate that IGA is effective and superior to GA-M and GA-ES on large instances.

Further research may be performed to test different strategies to improve the performance of IGA. Some of the areas of interest may be: 1) different strategies to selecting vaccine; 2) new method to construct immune operator; 3) new immune mechanism.

7. Acknowledgement

This work is supported by Nature Science Foundation of Gansu Province (Grant No. 3ZS051-A25-037) and Youth Foundation of Liaoning Technical University (Grant No. 07A205).

References

[1] G.M. Guisewite and P.M. Pardalos, Minimum concavercost network flow problems: Applications, complexity, and algorithms. *Annals of Operations Research*, 25:75-100, 1990.

[2] P.G. McKeown, A vertex ranking procedure for solving the linear fixed charge problem. *Operations Research*, 23:1183-1191, 1975.

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

[4] W.E. Walker, A heuristic adjacent extreme point for the fixed charge problem. *Management Science*, 22:587-596, 1976.

[5] N.H. Tran and K. Tran, Combination of fuzzy ranking and simulated annealing to improve discrete fracture inversion. *Mathematical and Computer Modelling*, 45:1010-1020, 2007.

[6] H.G. Santos, L.S. Ochi, E.H. Marinho and L.M.A. Drummond, Combining an evolutionary algorithm with data mining to solve a single-vehicle routing problem. *Neurocomputing*, 70:70-77, 2006.

[7] P. Musilek, A. Lau, M. Reformat and L. Wyard-Scott, Immune programming. *Information Sciences*, 176:972-1002, 2006.

[8] M. Solimanpur, P. Vrat and R. Shankar, An ant algorithm for the single row layout problem in flexible manufacturing systems. *Computers and Operations Research*, 32:583-598, 2005.

[9] D.E. Goldberg, Editor. *Genetic algorithms in search, Optimization and Machine Learning*. Addison-Weslery, Massachusetts, 1989.

[10] G.A. Vignaux and Z. Michalewicz, A genetic algorithm for the linear transportation problem. *IEEE Transcations on Systems, Man and Cybernentics*, 21:445-452, 1991.

[11] J. Gottlieb and L. Paulmann, Genetic alorithms for the fixed charge transportation problem. *Proceeding of the 1998 IEEE International Conference on Evolutionary Computation*, pp.330-335, 1998.

[12] G.R. Raidl and B.A. Julstrom, Edges sets: an effective evolutionary coding of spanning tree. *IEEE Transactions on evoluationary computation*, 7:225-239, 2003.

[13] M. Sun, J.E. Aroson, P.G. Mckeown and D. Drinka, A tabu search heuristic procedure for the fixed charge transportation problem. *European Journal of Operation Research*, 106:441-456, 1998.