A New Immune Genetic Algorithm for the Large Scale Ship Routing Problems

Xiaona SUN, Na LI,
Collage of Transportation Engineering and Logistics
Dalian Maritime University, China, 116026

Abstract: The ship routing problem (SRP) is a special variant of the classical vehicle routing problems (VRPs), which is different from both the VRP with backhauls (VRPB) and the VRP with pickup and delivery (VRPPD). This paper proposes a new immune genetic algorithm for the large scale SRP. The backhaul and time window constraints are also considered. The immune GA is innovated based on the classic GA and it can improve the general search ability and fight the premature convergence effectively. Numerical experiments indicate the proposed method is effective and competitive in reducing both the total shipping cost and the needed ship numbers.

Keywords: ship routing, pickup and delivery, backhaul, immune GA

1 Introduction

1.1 Problem statement

The classical vehicle routing problem (VRP) can be described briefly as follows: one set of customers requires service from the depot. There are homogeneous vehicles originally located at the depot. The vehicles will provide service (delivery or pick up) when they visit customers. The quantities to be delivered or picked up are known in advance. There is a cost associated with each pair of points from the set. A feasible solution is defined as routes that visit each point exactly once while not violating the capacity of the vehicle. The optimal solution is the feasible solution with least total cost.

The VRP with backhaul (VRPB) is an extension of the VRP involving both delivery and pickup points. The delivery points are sites that are to receive a quantity of goods from the distribution center (DC); and the pickup points are sites that send a quantity of goods back to the DC. The VRP with time windows (VRPTW) is another extension of the VRP, in which each point must be served within a specified time window. The VRP with backhaul and time windows (VRPBTW) is a combination of the VRPB and VRPTW. As to the traditional VRPB, there is a strict assumption that all the cargoes transported from the DC must be delivered before any cargoes can be picked up from pickup points on each route. (See Figure 1) Such an assumption arises from the fact that vehicles are often rear-loaded, and that it is often inconvenient or impossible to rearrange the delivery goods on board in order to accommodate new pickup loads. Another practical reason is that delivery customers have a higher service priority than pickup customers.

![Figure 1: The VRP with backhauls](image1)

![Figure 2: The container ship routing problem](image2)

The VRP with pickup and delivery (VRPPD) involves a heterogeneous vehicle fleet based at multiple
depots, which must satisfy a set of transportation requests. Each request includes a pickup customer, a corresponding delivery customer, and a demand to be transported between them. In addition to time window and vehicle capacity constraints, the VRPPD implies other constraints, which imposes coupling the pickup and corresponding delivery stops on the same vehicle routes, and imposes visit precedence between each pickup stop and its associated drop-off stop.

The ship routing problem is a special variant of the classical VRP problems. It can be described as follows. In branch-line shipping, ships have to start from the hub port and pick up cargoes from feeder ports and/or deliver cargoes to feeder ports. The cargoes can be containers, or empty containers, or general cargoes, or bulk cargoes, etc. The ship can be regarded as container ships, or general cargo ships, or bulk carrier ships, etc. The time windows and ship capacity constraints are the same as other VRP problems. The quantities to be delivered or picked up are known in advance. The objective is to decide a group of routes whose total cost is least.

The SRP is different from the VRPB, considering that the basic assumption for VRPB that on each route all deliveries have to be made before any cargo can be picked up is not true for the SRP. Unlike vehicles, ships are not rear-loaded, and it is often possible to rearrange cargoes in feeder ports. For the SRP and its difference from the VRPB, refer to Figure 2. The SRP discussed here is also different from the VRPPD. Both the coupling and precedence constraints, for the VRPPD, between the pickup and delivery stops can be relaxed for the SRP, since its pickup cargoes in all stops (feeder ports) must be taken back to the common stop (hub). Besides, in the traditional VRPPD, a vehicle is assumed to leave unloaded from its origin depot and to come back to its destination depot without any load also. For SRP, the ship loaded starts its route from the hub to feeder ports, or vice versa.

1.2 Literature review

The VRPB and its variant have attracted the attention of researchers to develop exact and approximate procedures. Toth and Vigo (1997) [1] and Mingozzi et al. (1996) [2] presented LP formulations for the VRPB and developed branch and bound algorithms. Yano et al. (1987) [3] developed a set cover based exact algorithm for a practical VRPB application. It is well known that all VRPs belong to the class of NP-hard combinatorial optimization problems. Therefore it is impossible to find exact solutions of real-life problems with a reasonable computational effort. Thus most researches in the literature have concentrated on the development of approximate algorithms. All of these contributions to VRPBTW were under the assumption that all delivery must be made on each route before any pickup (jin et al. (2000) [4]). As to ship routing problem, Jin et al. (2005) [5] presented a MIP formulation for container ship routing problem (CSRIP) and proposed variable neighbor search and tabu search heuristic algorithms.

Because all the VRP variants are NP-hard combinatorial optimization problems in the strong sense, the SRP can be solved exactly only for small-sized instances, while approximately for large-sized instances. In this paper, we target on branch-line shipping with time window constraint. The cargo flowing is restricted to the flow from hub to feeder ports or vice versa. We propose a new immune genetic algorithm (IGA) for solving large-scale ship routing problem instances. Numerical experiments which base on the benchmark problems show its efficiency.

2 A new Immune Genetic Algorithm for SRP

2.1 Immune genetic algorithm

Immune genetic algorithm (IGA) is a new improved genetic algorithm, which is presented based on the biological immunology. The immunity system is composed of antigen identifying system, memory system, accelerating and controlling mechanism of antibody, which produces a mass of antibodies to resist various antigens by the cell division. This immune mechanism is brought into immune genetic algorithm, which takes the target function value as antigen and its solution as antibody. During the immune operation, some antibodies are kept as the memory cell. When homologous antigens attack again, the memory cells explode quickly and produce a mass of antibodies to make it respond more quickly. Meanwhile, the immunity system has the function of self-adjustment, which can maintain the diversity of antibodies and the balance of immunity system. The mutation operation distills and injects bacteria to accelerate evolving process and avoid degradation. Comparing with standard GA, IGA can
2.2 Algorithm design based on SRP

Considering the specialties of the ship routing problem, a new immune genetic algorithm is designed as follows:

Step 1: Generating the initial population.

Experiments were carried out with the initial population being generated at random. We adopted a natural genetic chromosome representation called two-level representational scheme (Pereira et al., 2002 [6]), which is proved to be superior to one level representational scheme in common use. As in Figure 3, the natural number of the first level represents the ship route number and the natural number arranged in a row of each route presents the feeder ports.

Step 2: Calculate the fitness function value of every antibody in the current generation and select the optimal individual.

Step 3: Stop the algorithm if the evolution times exceed the maximum number of iterations, or the optimal solution of the current generation is superior to the limitative fitness value. Otherwise, go to step 4.

Step 4: Diversity estimate and adjustment.

The diversity adjustment is brought into the evolution process in order to avoid premature convergence. Thus, if the diversity is under the threshold value, we adopt the approach below to heighten the diversity of reproduction:

1. Choose 1 chromosome by a certain probability from the reproduction.
2. Select two low diversity positions at random and swaps the genes on the positions of randomly selected individual until the number of low diversity positions is less than M/n, where M is the number of genes and n is an integer part of [2,5].

In this paper, the diversity of the current population was defined by the information entropy (Zhang & Yang, 2005 [7]). Suppose there are N antibodies in each population, which contains l gene positions and the length of coding denotation is \( |S| = |S_1, S_2, \ldots, S_l| \). Since the chromosomal character is different, the gene that is carried by chromosome on each position is different too. The information entropy of the position \( j \) is denoted as:

\[
H_j(N) = -\sum_{i=1}^{l} p_{ij} \log p_{ij}.
\]

Where, \( p_{ij} \) represents the probability of denotation \( i \) presenting to position \( j \), which is given by \( p_{ij} = n_{ij} / N \), and \( n_{ij} \) is the total times of denotation \( i \) presenting to position \( j \). The diversity of population gets better when the value of \( H_j(N) \) is bigger, and it indicates deficient when the value is smaller than the threshold value. The diversity threshold value is given by:

\[
\phi(t) = A \left( \log |S| \right) e^{-\frac{Bi}{t}},
\]

where, \( A \in (0,1) \) is the coefficient of adjustment, \( B > 0 \) is the accelerating coefficient, \( t \) is the sequence of the current generation and \( G \) is the total generations of evolution. Obviously, the value of \( A \) being small indicates that the request of diversity is low, while the value of \( B \) being big indicates that the threshold value of each generation reduces fast and the evolution speeds up fast, too. However, the evolution becomes random search if \( A \) is too big or \( B \) is too small, that goes against heightening fitness function value and causes degradation of solutions; On the contrary, the evolution degenerates to be
stranded GA when \( A \) is too small or \( B \) is too big. Both \( A \) and \( B \) should be properly adjusted according to the actual instances.

Step 5: The reproductive processes: crossover and mutation.

Two individuals \( I_1 \) and \( I_2 \) are selected from the reproduction with a definite probability. Offspring are produced from the two parents using a crossover operation based on the two-level representational scheme (Wang et al., 2004 [8]). See Fig. 4 as below:

1. Select a sub route \( SR \) at random from \( I_1, SR = /v_1,v_2,..,v_n/ \).
2. Seek a pickup/delivery port \( c \), which is the nearest port to \( v_1 \), and make sure that \( c \) is not in \( SR \).
3. Seek the position of \( c \) in \( I_2 \), next to \( c \) and delete the same ports in \( I_1 \) and \( SR \), and calculate the number of needed ship again. Thus, the descendant consists of information of both parents is obtained.
4. Estimate the fitness value of the offspring and two parents, and select two better ones for the population of next generation.

\[
\text{Individual}_{1} \\
\text{Route Nc.} \\
1 \quad 2 \quad 9 \quad 3 \\
2 \quad 4 \quad 7 \quad 6 \quad 5 \\
3 \quad 10 \\
4 \\
\text{SR from} \quad 7 \quad 6 \quad 3 \\
\text{Individual}_{2} \\
\text{Route Nc.} \\
1 \quad 4 \quad 10 \\
2 \quad 7 \quad 6 \quad 3 \\
3 \\
4 \quad 6 \quad 2 \\
\text{Descendant} \\
\text{Route Nc.} \\
1 \quad 4 \quad 10 \\
2 \quad 1 \quad 7 \quad 6 \quad 3 \\
3 \quad 9 \\
4 \quad 2 \\
\text{c} \quad 1 \\
\text{Figure 4 The crossover operation}
\]

The mutation operation was carried out by swapping the positions of two genes which were selected randomly.

Step 6: distilling bacteria and vaccination operations.

The essence of vaccination in which the bacteria represents the basic character of the problem is like the heuristic algorithm. As it is difficult to obtain the bacteria of the whole problem, we took a local adjustment method (Zhang & Yang, 2005 [7]) to heighten the fitness value of individuals. Thus, we sought the nearest port of each port in each sub route as the bacteria of vaccination after the reproductive process, and vaccinate the individuals selected at random. Then the individual with improved fitness value was kept.

Step 7: Making immunity selection and updating reproduction.

Estimate the fitness value of offspring and replace the memory cell with the optimal individual if its fitness value is superior to that of parents after vaccination. If the optimal individual of offspring is worse than the memory cell, put the memory cell into the reproduction to make sure it participates in the reproductive process of the next population. Then, go back to step 3.

3 Numerical Experiments
3.1 Experimental instances

The proposed heuristic algorithms were implemented and tested on experimental instances. These experimental instances were from Solomon benchmark problems for the VRPTW (Solomon, 1991 [9]).
Each of the fifty-six instances has 100 customers. The travel time between customers is equal to the corresponding Euclidean distance. These problems vary in fleet size, vehicle capacity, travel time, spatial distribution of customers, time window density and width, and are divided into three types: R-type (uniformly distributed customers), C-type (clustered customers), and RC-type (a mix of R- and C-types). Two sets of problems are designed for each of the three types. Problem sets RC1, C1, R1 have narrow scheduling horizon, while sets RC2,C2,R2 have wide scheduling horizon. The narrow scheduling horizon problems have vehicles with small capacities and short route times, which results in only a few of customers can be served by a vehicle. By contrast, the wide scheduling horizon problems have vehicles with large capacities and long route times, so more customers may be served by a single vehicle.

We convert these VRPTW instances into SRP instances by taking some delivery customers to be pickup customers. There are two types of problems generated. In type 1, all the customers are divided into either delivery ones or pickup ones; and in type 2, all the customers are divided into delivery, pickup, and both delivery and pickup ones. In both types, 10%, 30%, and 50% delivery customers in pickup customers. There are two types of problems generated. In type 1, all the customers are divided into either delivery ones or pickup ones; and in type 2, all the customers are divided into delivery, pickup, and both delivery and pickup ones. In both types, 10%, 30%, and 50% delivery customers in original problems are designed to become pickup ones. In type 2, the rate of both delivery and pickup ones is randomly created.

### 3.2 Numerical results

The algorithms were programmed in Microsoft Visual Basic and run on a personal computer. The real distances between customers and the travel times were rounded to three decimal places. Each instance was solved by ten-runs of the algorithm (with different initial solutions) and its best solution value was obtained. Since there has not been any report about SRP, we compare our results with those obtained by the famous sweep algorithm (Gillet & Miller,1974) for VRPs and tabu search (TS) algorithm (Jin et al., 2005) for SRP. Parts of the comparison results are shown in Table 1.

<table>
<thead>
<tr>
<th>instances</th>
<th>p/t (%)</th>
<th>Type1</th>
<th></th>
<th></th>
<th>Type2</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Sweep</td>
<td>TS</td>
<td>IGA</td>
<td>Sweep</td>
<td>TS</td>
<td>IGA</td>
</tr>
<tr>
<td>r101</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>2259.8</td>
<td>33</td>
<td>1652.2</td>
<td>20</td>
<td>1535.9</td>
<td>19</td>
<td>2259.8</td>
</tr>
<tr>
<td>30</td>
<td>2237.2</td>
<td>33</td>
<td>1638.3</td>
<td>20</td>
<td>1566.9</td>
<td>18</td>
<td>2237.2</td>
</tr>
<tr>
<td>50</td>
<td>2237.2</td>
<td>33</td>
<td>1653.3</td>
<td>20</td>
<td>1591.8</td>
<td>18</td>
<td>2238.2</td>
</tr>
<tr>
<td>r102</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>2090.2</td>
<td>30</td>
<td>1488.0</td>
<td>19</td>
<td>1370.6</td>
<td>17</td>
<td>2097.0</td>
</tr>
<tr>
<td>30</td>
<td>2083.4</td>
<td>30</td>
<td>1479.5</td>
<td>19</td>
<td>1370.3</td>
<td>17</td>
<td>2083.4</td>
</tr>
<tr>
<td>50</td>
<td>2083.4</td>
<td>30</td>
<td>1480.7</td>
<td>19</td>
<td>1357.3</td>
<td>17</td>
<td>2077.7</td>
</tr>
<tr>
<td>r103</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>1820.8</td>
<td>22</td>
<td>1275.5</td>
<td>15</td>
<td>1176.1</td>
<td>14</td>
<td>2001.0</td>
</tr>
<tr>
<td>30</td>
<td>1809.5</td>
<td>22</td>
<td>1250.5</td>
<td>14</td>
<td>1049.8</td>
<td>13</td>
<td>1886.4</td>
</tr>
<tr>
<td>50</td>
<td>1809.5</td>
<td>22</td>
<td>1230.5</td>
<td>14</td>
<td>1140.9</td>
<td>14</td>
<td>1950.2</td>
</tr>
<tr>
<td>r104</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>1560.9</td>
<td>17</td>
<td>1089.6</td>
<td>11</td>
<td>995.8</td>
<td>11</td>
<td>1630.9</td>
</tr>
<tr>
<td>30</td>
<td>1471.8</td>
<td>16</td>
<td>1069.6</td>
<td>11</td>
<td>972.4</td>
<td>11</td>
<td>1528.0</td>
</tr>
<tr>
<td>50</td>
<td>1485.7</td>
<td>16</td>
<td>1090.2</td>
<td>12</td>
<td>955.8</td>
<td>11</td>
<td>1536.8</td>
</tr>
<tr>
<td>r105</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>1891.3</td>
<td>23</td>
<td>1391.4</td>
<td>15</td>
<td>1376.3</td>
<td>15</td>
<td>1933.6</td>
</tr>
<tr>
<td>30</td>
<td>1871.4</td>
<td>24</td>
<td>1405.7</td>
<td>15</td>
<td>1385.2</td>
<td>15</td>
<td>1870.6</td>
</tr>
<tr>
<td>50</td>
<td>1912.3</td>
<td>24</td>
<td>1419.5</td>
<td>16</td>
<td>1418.2</td>
<td>16</td>
<td>1949.7</td>
</tr>
<tr>
<td>Sum</td>
<td></td>
<td>28624.1</td>
<td>375</td>
<td>20614.5</td>
<td>240</td>
<td>19283.3</td>
<td>226</td>
</tr>
</tbody>
</table>

*p/t: pickup/total customers number RN: Route No.

Table 1 shows the computational results about type 1 instances, and Table 2 shows those of type 2 instances. The total weighted cost and the route numbers (that is, the needed vehicles or ships) are used to compare the proposed heuristics with sweep algorithm. IGA outperforms the sweep algorithm in reducing both the cumulated number of vehicles (ships) and the total travel distance. For example, comparing to sweep algorithm, the total weighted costs have decreased 33% for type 1, and 34% for type 2, respectively, the number of ships needed has decreased 40% for both two types. Comparing to TS, the total weighted costs have decreased 7% for both two types, and the number of ships needed has
also decreased 6% for type 1, and 5% for type 2, respectively. These results show the effectiveness of the proposed IGA.

As to the CPU time, IGA is equivalent to TS, and a little bit higher than sweep algorithm. Both the solution quality and the CPU time can provide a trade-off between the computational cost and the near-optimal solutions.

4 Conclusions

This paper deals with the ship routing problem (SRP). The backhaul and time window constraints are also considered. As far as we know, there have been few other reports about it. The similarities and differences between VRPs and SRP are analyzed at first. Based upon that, a new immune genetic algorithm for SRP is designed for solving large scale problems. Numerical experiments indicate the proposed method is effective and competitive in reducing both the total shipping cost and the needed ship numbers.

Acknowledgement

This research is partly supported by the National Natural Science Foundation, China (Grant No. 70572104).

References