We are IntechOpen, the world’s leading publisher of Open Access books
Built by scientists, for scientists

4,300 Open access books available
116,000 International authors and editors
130M Downloads

154 Countries delivered to TOP 1%
most cited scientists
12.2% Contributors from top 500 universities

WEB OF SCIENCE™
Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com
The Method of Solving for Travelling Salesman Problem Using Genetic Algorithm with Immune Adjustment Mechanism

Hirotaka Itoh
Nagoya Institute of Technology, Japan

1. Introduction

Genetic Algorithm (GA) is widely used to find solutions to optimization problems (Goldberg, 1989). One optimization problem using GA is Travelling Salesman problem (TSP) (Lawler et al., 1985). The disadvantages of using GA are premature convergence and poor local search capability. In order to overcome these disadvantages, evolutionary adaptation algorithms based on the working of the immune system have been devised. One such algorithm is Genetic Immune Recruitment Mechanism (GIRM) (Bersini & Varela, 1991) (Tazawa et al., 1995). By incorporating the immune recruitment test and concentrating the search for a solution in the vicinity of a high-fitness solution, GIRM improves local search capability. However, narrowing the search range risks conducting to local solutions. In contrast, Immune Algorithm (IA) (Mori et al., 1997) (Honna et al., 2005) (Matsui et al., 2006), using production of various antibodies by the immune system and its mechanism of their adjustment, primarily avoids convergence to local solutions. Its local search capability is not as good as that of GIRM, but it allows efficient searches for multiple local solutions. GIRM and IA incorporating the workings of the immune system take more computation time than GA. Thus, they must be performed with a smaller population size.

To that end, the author devised an algorithm to overcome these GA’s disadvantages with the small population size. The immune system has two features, the capacity to adapt to mutations in antigen and a mechanism to balance the generation of antibodies via other antibodies, and the author developed Genetic Algorithm with Immune Adjustment Mechanism (GAIAM) incorporating these features in GA. GAIAM maintains the diversity of the population as a result of the mechanism to adjust antibodies in a group of antibodies, so it avoids narrowing of the search range. In addition, its local search capability also improved as a result of the capacity to adapt to mutations in antigen. GAIAM provides effective results even with a small population size.

Using the TSP, the author compared the performance of GAIAM to that of GA, GIRM, and IA. First, an experiment was performed using eil51 from the TSPLIB. TSPLIB has benchmark data of TSP. Eil50 is one of the data in TSPLIB. Because it incorporates two features of the immune system even with a small population size, the GAIAM allows a more efficient search over the entire search range without succumbing locally. Moreover, its local search capability was found to be better than that of other techniques. Furthermore, experiments...
ware performed using data with 100 cities or more, and GAIAM was found to be effective even in large-scale problems.

This paper first offers an overview of the GAIAM. In addition, differences between it and GA, GIRM and IA are described. Next, the experiments were performed using the TSP, and GAIAM’s performance was compared to that of other algorithms. Last, GAIAM’s effectiveness is set forth.

2. GAIAM

2.1 Feature of immune system

Various antibodies are present in the human body. As antigens invade the body, antibodies for these antigens are generated to eliminate the antigens. The immune system has the following two features:

[Feature 1] Capacity to adapt to mutations in antigens

It is difficult to produce antibodies for each and every antigen beforehand. When there is no antibody adapted for an antigen, genes of the antibodies with the best specificity respond by mutating. Through repeated mutations, these genes produce antibodies that can adapt to the antigens.

[Feature 2] Mechanism to balance the generation of antibodies via other antibodies

The generation of antibodies for a given antigen is not a continuous process. Antibodies recognize each other on the basis of their structure, and when a given antibody is generated in excess, other antibodies that identify it as an antigen and are also generated to inhibit its growth, and a balance is maintained.

2.2 GAIAM algorithm

The GAIAM algorithm is shown in Fig.1. The GAIAM algorithm is following.

Step 1. Generation of an initial group of antibodies

N antibodies are generated initially. These antibodies are similar to the individuals in GA and are helpful in solving optimization problems.

Step 2. Calculation of affinities

The affinities \( ax_i \) \((i = 1, \ldots, N)\) for antigens are calculated. The \( ax_i \) is set in accordance with the problem being dealt with. The affinity for an antigen is similar to the concept of fitness in GA.

Step 3. Calculation of expected values

The expected value \( e_i \) \((i = 1, \ldots, N)\) of antibodies that can survive into the next generation is calculated as

\[
e_i = \frac{ax_i}{C_i}
\]  

(1)

Where \( C_i \) is the density of antibodies of type \( i \) \((i = 1, \ldots, N)\).

\[
C_i = \frac{1}{N} \sum_{j=1}^{N} ay_{i,j}
\]  

(2)
Here, $a_{ij}$ is the similarity between antibodies of type $i$ and $j$ ($i = 1, \ldots, N, j = 1, \ldots, N$) and is set in accordance with the problem. $N/2$ antibodies with low expected values are eliminated.

**Step 4.** Antibody production

New antibodies are generated to replace the antibodies eliminated in Step 3. $N/2$ antibodies are selected from the surviving antibodies on the basis of the expected values. These selected antibodies are mutated, after which their affinities for antigens are calculated.

**Step 5.** Crossover and mutation

Antibodies are randomly selected (duplication permitted) from $N$ antibodies, they undergo crossover depending on crossover probability $P_c$, thereby generating $N/2$ antibodies. The generated antibodies undergo mutation depending on mutation probability $P_m$, after which their affinities for antigens are calculated.
Step 6. Adjustment of antibodies
With respect to each antibody \(i\) of the \(N/2\) antibodies generated in Step 5, an antibody \(j\) with the greatest affinity for \(i\) is sought from among the existing \(N\) antibodies. Among antibodies \(i\) and \(j\), the one with the higher affinity for an antigen survives into the next generation, while the other is eliminated.

Step 7. Repetition of Steps 3 to 6 for a determined number of generations.
Step 4 models [Feature 1] of the immune system and Step 6 models [Feature 2] of the immune system. In GAIAM, antibodies with low density and high affinity for an antigen tend to survive in order to maintain diversity. Moreover, such antibodies are generated in GAIAM; antibodies with a high affinity for an antigen are produced through mutation. In terms of GA, local search capability is improved. Moreover, narrowing of the search range is eliminated through Step 6. Thus, the narrowing of the search to the vicinity of a single local solution in GA is eliminated.

3. Comparison of GA, GiRM and IA to GAIAM

3.1 The GA algorithm
Here, the GA algorithm is briefly explained. The GA algorithm is shown in Fig.2.

![Fig. 2. The GA algorithm](image)

**Step 1.** Generation of an initial group
Same as Step 1 in the GAIAM

**Step 2.** Calculation of affinities
Same as Step 2 in the GAIAM

**Step 3.** Reproduction
\(N\) antibodies are selected from antibodies group to surviving next generation on the basis of the affinity.
Step 4. Crossover and mutation  
Same as Step 5 in the GAIAM

Step 5. Repetition of Steps 2 to 4 for a determined number of generations.  
The disadvantages of using GA are premature convergence and poor local search capability.

3.2 The GIRM algorithm
Here, the GIRM algorithm is briefly explained. The GIRM algorithm is shown in Fig.3.

![GIRM Algorithm Diagram](image)

**Fig. 3. The GIRM algorithm**

**Step 1.** Generation of an initial group
Same as Step 1 in the GAIAM

**Step 2.** Calculation of affinities
Same as Step 2 in the GAIAM

**Step 3.** Crossover and mutation
Same as Step 5 in the GAIAM

**Step 4.** Calculation of similarity
Same as similarity for GAIAM

**Step 5.** Immune recruitment test
An immune recruitment test is performed on antibodies obtained in Step3; antibodies passing the test are added to the group. Alternatively, antibodies with
the lowest fitness in the group are removed. For details, see references (Bersini & Varela, 1991) and (Tazawa et al., 1995).

**Step 6.** Repetition of Steps 3 to 5 for a determined number of generations.

With GIRM, highly fit antibodies in the group and similar antibodies increase as a result of the immune recruitment test, so searching in the proximity of a solution, i.e. highly fit antibodies, becomes more vigorous. That is, local searches are intensive. When, however, GIRM succumbs to a high-fitness local solution, it has difficulty escaping.

### 3.3 The IA algorithm

Here, the IA algorithm is briefly explained. The IA algorithm is shown in Fig. 4.

![The IA algorithm](image.png)

**Fig. 4. The IA algorithm**

**Step 1.** Generation of an initial group

Same as Step 1 in the GAIAM
Step 2. Calculation of affinities
   Same as Step 2 in the GAIAM

Step 3. Differentiation into memory cells
   For details, see reference (Mori et al., 1997).

Step 4. Promotion and inhibition of antibody production
   Same as Step 3 in the GAIAM

Step 5. Antibody production
   New antibodies are produced to replace $N/2$ antibodies eliminated in Step 4. New antibodies are produced by randomly determining their genes.

Step 6. Crossover and mutation
   Same as Step 5 in the GAIAM

Step 7. Repetition of Steps 3 to 6 for a determined number of generation.
Because of Step 5, the IA avoided narrowing the search to local solutions. However, its local search capability was not as good as that of the GIRM, because new antibodies were randomly produced in Step 5.

3.4 Comparison of GA, GIRM and IA to GAIAM
The disadvantages of using GA are premature convergence and poor local search capability. GIRM has enhanced local search capability but easily conducts to local solutions. IA allows efficient searches for multiple local solutions, but its search capability in the vicinity of individual local solutions is poor. GAIAM has improved local search capability because of its capacity to adapt to mutations and avoids narrowing of the search range because of its mechanism to adjust antibodies via antibodies; in short, it allows efficient searches.

4. Use in the TSP
4.1 Path representation
Path representation is used for the coding of antibodies. Path representation is shown in Fig.5. Path representation orders cities by number in the order they are visited. The method in which initial antibodies are generated will now be explained. First, city numbers are randomly ordered. Next, antibodies are generated using the 2-opt method (Johnson, 1991).

Fig. 5. Path representation
4.2 Affinity of antibody and antigen
The affinity $a_{x_i}$ of antibody $i$ and the antigen is the inverse of the tour length $d$.

$$a_{x_i} = \frac{1}{d}$$ (3)

4.3 Crossover and mutation
Various studies have been conducted on crossover methods used in the TSP (Nagata et al., 1999). The current work, however, sought to assess the performance of an algorithm, so OX crossover was used instead of a well-performing crossover method (Davis, 1985). With regard to mutation, antibodies first undergo substitution or inversion once. Next, the 2-opt method was used. Substitution is shown in Fig.6 and inversion in Fig.7. For substitution, the positions of 2 random cities in the antibodies are switched and for inversion, the order of 2 random cities is reversed.

Fig. 6. Substitution

![Substitution Diagram]

Fig. 7. Inversion

![Inversion Diagram]

4.4 Similarity for antibodies
Similarity $a_{y_{i,j}}$ for antibodies $i$ and $j$ is represented by the following equation.

$$a_{y_{i,j}} = 1.0 - \frac{\sum_{a=1}^{L} \sum_{b=1}^{L} (-p_{a,b}) \log(p_{a,b})}{-L \cdot 2.0 \cdot \log(1.0 / 2.0)}$$ (4)

$$p_{a,b} = \frac{N_{ab}}{2}$$ (5)

$L$ is the number of cities.
\( N_{ab} \) is, for antibodies \( i \) and \( j \), the number of branches connecting city \( a \) and city \( b \) on the tour. Similarity \( ay_{ij} \) was normalized; as they approached 0, \( i \) and \( j \) were dissimilar while they were similar as they approached 1.

5. Experiments

5.1 Experiment using eil51 from the TSPLIB
An experiment was performed using eil51 from the TSPLIB to assess the performance of GAIAM. A Linux machine was used. The shortest tour for eil51 is shown in Fig. 8. The shortest tour length is 426. Parameters used in the experiment were generations of \( G=500 \), a crossover probability of \( P_c=0.8 \), and a mutation probability of \( P_m=0.2 \). There were 4 population sizes of 20, 30, 50, and 100. The experiment was performed 30 times each.

![Fig. 8. The shortest tour length of eil51 (length=426)](image-url)
Results are shown in Table.1. In Table.1, $d_m$ is the shortest route length found as a result of 30 individual attempts. $\text{avg}$ is the average of $d_m$ for 30 iterations, $\text{min}$ is the minimum of $d_m$ for 30 iterations, $\text{max}$ is the maximum of $d_m$ for those iterations, and $\text{std}$ is the standard deviation. $N_o$ is the number of times the shortest tour length of 426 was found out of 30 attempts.

<table>
<thead>
<tr>
<th></th>
<th>$N$</th>
<th>$d_m$</th>
<th>$N_o$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>avg</td>
<td>min</td>
</tr>
<tr>
<td>GAIAM</td>
<td>20</td>
<td>426.83</td>
<td>426</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>426.70</td>
<td>426</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>426.67</td>
<td>426</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>426.60</td>
<td>426</td>
</tr>
<tr>
<td>GA</td>
<td>20</td>
<td>428.87</td>
<td>426</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>428.77</td>
<td>426</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>428.57</td>
<td>426</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>428.47</td>
<td>427</td>
</tr>
<tr>
<td>GIRM</td>
<td>20</td>
<td>429.73</td>
<td>426</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>428.77</td>
<td>426</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>428.03</td>
<td>426</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>426.83</td>
<td>426</td>
</tr>
<tr>
<td>IA</td>
<td>20</td>
<td>429.90</td>
<td>426</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>428.53</td>
<td>426</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>428.27</td>
<td>426</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>427.60</td>
<td>426</td>
</tr>
</tbody>
</table>

Table 1. The result of the experiment

Generational changes in entropy when $N = 100$ are shown in Fig.9. Entropy is the average similarity of individual antibodies and other antibodies; the average for 30 attempts is shown in Fig.9. It indicates that there are more similar antibodies in the group as entropy approaches 1.

In Table 1, GAIAM yielded the best results. GAIAM yielded consistent results regardless of the population size. Even when $N=20$, $\text{avg}$ is markedly better than that of other techniques. Both the $\text{max}$ and $\text{std}$ were smaller than those with other techniques, and the shortest tour length was found numerous times. With a smaller population size, results for GIRM and IA were poorer than those for GA. With a larger population size, results were better. When $N=100$, GIRM yielded results similar to those from GAIAM, but comparison of results for GAIAM when $N=20$ indicate that GAIAM yielded better results overall.

GIRM and IA are modifications of GA, but somewhat larger population size is required for them to perform well.

In Fig.9, maximum entropy recorded with the GIRM. In Table.1, the $\text{avg}$ for GIRM was better than that for GA and IA, but the $\text{max}$ and $\text{std}$ were poor. As is apparent from Fig.9 and Table.1, GIRM has enhanced local search capability but once it conducts to a search in the
vicinity of high-fitness local solutions it cannot escape, and the search range narrows. Thus, if the vicinity of optimal values is searched in the search process, optimal values will be found, but when the search shifts to the vicinity of local solutions away from local values escape is difficult, and optimal values will not be reached. In Fig.9, the entropy of IA is the lowest. In Table.1, for IA, the \textit{avg} was poorer than that for GIRM but better than that for GA. The \textit{max} and \textit{std} were better than with GIRM. For IA, the local search capability was not as good as with GIRM, but it allowed an efficient search for multiple local solutions. In Fig.9, entropy for GAIAM remained about 0.67. This is almost midway between entropies for GIRM and IA. Similarly, GAIAM is superior to other techniques in Table.1 as well. GAIAM did not conduct to local values and allowed the efficient search over the entire search range, and its local search capability was also enhanced. In addition, GAIAM provided viable results even with the small population size.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{entropy.png}
\caption{Generation transition of entropy}
\end{figure}

\subsection*{5.2 Experiments using ch150, a280, and pcb442}
Experiments were performed using ch150, a280, and pcb442 from the TSPLIB to assess the performance of GAIAM in large city problems. Outlines of ch150, a280, and pcb442 are shown in Table.2. Parameters are shown in Table.3. The shortest routes for ch150, a280, and pcb442 are shown in Fig.10, Fig.11 and Fig.12. Results of the experiments are shown in Table.4, Table.5 and Table.6. As expected, GAIAM offered the best results. Trends in results were the same as when eil51 was used. GAIAM's \textit{max} was better than the \textit{min} for GA and IA. In addition, GAIAM found the shortest route for ch150 in 7 iterations, which is much better than other techniques. It found a unique shortest route for a280. Following GAIAM, GIRM had the best of \textit{avg}, but its \textit{max} was roughly worse than that of IA and it had the worst \textit{std}. As expected, it had enhanced local search capability, but when it conducted to
local values it lacked the ability to escape them. The above results indicate that GAIAM is able to provide viable results with the small-size population size even in large-scale problems. GAIAM allows efficient searches without conducting to local solutions and also provides enhanced local search capability.

6. Conclusion

GAIAM has faithfully incorporated the two features of the immune system, i.e. the capacity to adapt to mutations in antigens and the mechanism to adjust antibodies in a group of antibodies, into GA, and its effectiveness increased as the optimization technique. Results of experiments with GA, GIRM, and IA were compared to those with GAIAM and GAIAM’s performance was assessed via use of GAIAM in the TSP. According to the results, GAIAM was found to offer more efficient and balanced searches in the population than IA and GIRM. Consequently, GAIAM sought optimum values from the entire search range and also displayed enhancement in local search capability. It also provided viable results with the small population size in large-scale problems as well. GAIAM is superior to GA, GIRM, and IA in finding solutions for TSP. GAIAM is efficient as solving method of TSP.

<table>
<thead>
<tr>
<th>Number of city</th>
<th>The shortest length</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ch150</td>
<td>150</td>
</tr>
<tr>
<td>a280</td>
<td>280</td>
</tr>
<tr>
<td>Pcb442</td>
<td>442</td>
</tr>
</tbody>
</table>

Table 2. The outlines of ch150, a280 and pcb442

<table>
<thead>
<tr>
<th>Number of trials</th>
<th>Generation number</th>
<th>Antibody size</th>
</tr>
</thead>
<tbody>
<tr>
<td>ch150</td>
<td>20</td>
<td>1000</td>
</tr>
<tr>
<td>a280</td>
<td>20</td>
<td>2000</td>
</tr>
<tr>
<td>pcb442</td>
<td>10</td>
<td>2000</td>
</tr>
</tbody>
</table>

Table 3. The parameters of the experiments

<table>
<thead>
<tr>
<th></th>
<th>N</th>
<th>d_w</th>
<th>N_o</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>avg</td>
<td>min</td>
<td>max</td>
</tr>
<tr>
<td>GAIAM</td>
<td>30</td>
<td>6548.85</td>
<td>6528</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>6544.50</td>
<td>6528</td>
</tr>
<tr>
<td>GA</td>
<td>30</td>
<td>6668.35</td>
<td>6604</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>6691.60</td>
<td>6568</td>
</tr>
<tr>
<td>GIRM</td>
<td>30</td>
<td>6589.40</td>
<td>6528</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>6572.65</td>
<td>6528</td>
</tr>
<tr>
<td>IA</td>
<td>30</td>
<td>6659.80</td>
<td>6625</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>6639.50</td>
<td>6571</td>
</tr>
</tbody>
</table>

Table 4. The result of ch150
The Method of Solving for Travelling Salesman Problem Using Genetic Algorithm with Immune Adjustment Mechanism

Fig. 10. The shortest route of ch150

Fig. 11. The shortest route of a280
Fig. 12. The shortest route of pcb442
The Method of Solving for Travelling Salesman Problem Using Genetic Algorithm with Immune Adjustment Mechanism

<table>
<thead>
<tr>
<th></th>
<th>N</th>
<th>$d_m$</th>
<th>$N_o$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>avg</td>
<td>min</td>
</tr>
<tr>
<td>GAIAM</td>
<td>30</td>
<td>2591.3</td>
<td>2579</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>2588.5</td>
<td>2579</td>
</tr>
<tr>
<td>GA</td>
<td>30</td>
<td>2670.5</td>
<td>2637</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>2678.4</td>
<td>2651</td>
</tr>
<tr>
<td>GIRM</td>
<td>30</td>
<td>2640.8</td>
<td>2592</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>2621.3</td>
<td>2583</td>
</tr>
<tr>
<td>IA</td>
<td>30</td>
<td>2655.6</td>
<td>2633</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>2655.7</td>
<td>2634</td>
</tr>
</tbody>
</table>

Table 5. The result of a280

<table>
<thead>
<tr>
<th></th>
<th>N</th>
<th>$d_m$</th>
<th>$N_o$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>avg</td>
<td>min</td>
</tr>
<tr>
<td>GAIAM</td>
<td>30</td>
<td>51406.3</td>
<td>51141</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>51353.6</td>
<td>51069</td>
</tr>
<tr>
<td>GA</td>
<td>30</td>
<td>53351.3</td>
<td>53133</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>53274.1</td>
<td>52972</td>
</tr>
<tr>
<td>GIRM</td>
<td>30</td>
<td>52878.0</td>
<td>52048</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>51856.7</td>
<td>51426</td>
</tr>
<tr>
<td>IA</td>
<td>30</td>
<td>52886.5</td>
<td>52466</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>52692.7</td>
<td>52470</td>
</tr>
</tbody>
</table>

Table 6. The result of pcb442

7. References


www.intechopen.com


This book is a collection of current research in the application of evolutionary algorithms and other optimal algorithms to solving the TSP problem. It brings together researchers with applications in Artificial Immune Systems, Genetic Algorithms, Neural Networks and Differential Evolution Algorithm. Hybrid systems, like Fuzzy Maps, Chaotic Maps and Parallelized TSP are also presented. Most importantly, this book presents both theoretical as well as practical applications of TSP, which will be a vital tool for researchers and graduate entry students in the field of applied Mathematics, Computing Science and Engineering.

How to reference
In order to correctly reference this scholarly work, feel free to copy and paste the following:
