THE FITNESS LANDSCAPE ANALYSIS OF THE ANT COLONY SYSTEM ALGORITHM IN SOLVING A VEHICLE ROUTING PROBLEM

Anita AGÁRDI1,*, László KOVÁCS1 and Tamás BÁNYAI2

1 University of Miskolc, Institute of Informatics, Egyetemváros, 3515, Miskolc, Hungary
2 University of Miskolc, Institute of Logistics, Egyetemváros, 3515, Miskolc, Hungary

Corresponding author, Anita Agárdi, e-mail: agardianita@iit.uni-miskolc.hu

ABSTRACT: In this article, we examine the effectiveness of the Ant Colony System (ACS) algorithm for a Vehicle Routing Problem (VRP). Fitness landscape analysis determines the complexity of the optimization search space. By analyzing the search space, we can conclude the complexity of the task, whether a given algorithm (its operators) is effective for a given type of task. When most researchers develop an algorithm, they test it on benchmark data. If this achieves the best result known so far for the benchmark data (or are close to the big ones), the results will be published. However, in addition to existing tests, various analyzes can also be performed.

KEYWORDS: fitness landscape analysis, Ant Colony System, optimization, Vehicle Routing Problem

1 INTRODUCTION

The task of logistics is to deliver the right goods to the right place at the right time. The Vehicle Routing Problem (VRP) models just that. During the VRP, the positions of the customers, the number and capacity limit of the vehicles are given. The objective is the minimization of the route length.

The first VRP article was published in 1959, by Datzig and Ramster [1]. Since then, several versions of the task have emerged. The number of depots in the system can be one or more [2]. Vehicles usually have a capacity limit for the goods to be transported [3]. The type of goods can be one or more [4]. It can be a time window for each customer. The time window can be soft [5] or hard [6], and each customer can have only one [6] or more time windows [7].

The fitness landscape analysis helps to solve the optimization problem. It shows the complexity of the task, the efficiency of the optimization algorithms and their operators. Analyzing the search space for discrete optimization tasks is not a very researched area yet. In the following, we introduce some of them.

In [8], the fitness landscape analysis of the Quadratic Assignment Problem and the Methetic Algorithm is performed. The following analytical techniques were used by the authors: distance to optimum, cost difference, random walk correlation function, autocorrelation coefficient, fitness distance correlation coefficient (FDC), flow dominance, epistasis.

Authors [9] made a comprehensive survey of fitness landscape analysis. The basic concepts such as fitness value, fitness function, fitness mapping, solution candidate, etc. are followed by the local optima and plateaus: modality section, where the importance of avoiding sticking to the local optimum and getting out of it is important. It may therefore be important to know the number of local optima and their distribution in space. The next topic is the basins of attraction. The basins of attraction areas are around the optimum, they are such attractive areas during optimization. During landscape walks, you can crawl the search space. The following walk techniques can be distinguished: random walk, adaptive walk, reverse adaptive walk, neutral walk, reverse neutral walk, uphill-downhill walk. Another topic is evolvability. The evolvability is the chance to evolve along with the iterations.
Article [10] focuses on the analysis of Multidimensional Knapsack Problem. It analyzes the space for different representations. The following representations are reported: Binary Representation, Ordinal Representation, Permutation Representation, Random-Key Representation, Weight-Coding Representation, Related Work. The authors use the following fitness landscape analysis techniques: Fitness Distance Correlation, Autocorrelation Function, Correlation Length, Crossover Measures.

2 ANTI COLONY SYSTEM ALGORITHM

The algorithm models the behaviour of ants. Ants leave a pheromone on the traversed route on the way to food. The pheromone attracts other ants. The higher the pheromone on the road section, the more likely the ants to pass. Thus, during the algorithm, those road sections that will be short have a high pheromone content. The pheromone also evaporates on each road segment, and the algorithm must take this into account. However, if more ants cross the route, they will place another dose of pheromone on the road section.

Figure 2. illustrates the pseudo-code of the Ant Colony System Algorithm (ACS).

**ALGORITHM: ANT SYSTEM**

**BEGIN PROCEDURE**

1. Pheromone value initialization.

   **WHILE** (termination criteria is not met) **DO**
   **WHILE** (for all ants) **DO**
   2. Path construction
   **END WHILE**
   3. Updating the pheromone).
   **END WHILE**

**END**

**Figure 2.** The Ant Colony System Algorithm (ACS) [14]

Route construction [14]:

\[
\pi_{ij}^k = \frac{[\tau_{ij}(t)]^{\alpha} * [\eta_{ij}]^{\beta}}{\sum_{l \in N_i^k} [\tau_{il}(t)]^{\alpha} * [\eta_{il}]^{\beta}} \quad ha \ j \in N_i^k \quad (1)
\]

where \( k \) means the ant, \( i \) and \( j \) the node, \( t \) the iteration, \( \eta_{ij} = \frac{1}{d_{ij}} \cdot \tau_{ij}(t) \) is the pheromone content.

Pheromone update [14]:

\[
\tau_{ij}(t + 1) = (1 - \rho) * \tau_{ij}(t) + \sum_{k=1}^{m} \Delta \tau_{ij}^k(t) \quad (2)
\]

\( 0 < \rho \leq 1 \) is the evaporation rate.

3 FITNESS LANDSCAPE ANALYSIS OF ANTI COLONY SYSTEM ALGORITHM

During the analysis of the algorithm, we analyzed the solutions given by each iteration. The following analysis techniques were taken into consideration: distance of solutions (the average distance of each solution to other solutions), distance of best solution, cost density (how different the fitness values of each solution are), distance of best solution with filtered global optima (distances between the solutions given by each iteration and the sampled optimum).

The following three distances were calculated for the solutions: fitness, Hamming, and basic swap sequence. Fitness distance is the absolute value of the difference in fitness value between the two solutions. The Hamming [12] distance is the number of different elements of the two solutions for each position. And the basic swap sequence [13] is the minimum number of swaps, so the minimum number of edge swaps from one solution must be done to get the other solution.
According to Figure 3, the higher the fitness value of one solution, the higher the average of the fitness distances taken from the other solutions. This shows a linear function. The averages of the Hamming and basic swap sequence distances are not overly influenced by the fitness value (Figure 4.-5.).

The fitness distance from the best solution naturally increases linearly as a function of the fitness value of the solutions (Figure 6.). The distances from the best solution during Hamming and basic swap sequence distances are not affected by the fitness values of the solutions (Figure 7.-8.).

Cost density values (Figure 9.) range from 1 to 5, which means that some solutions are the same.
Fitness distances from the filtered global optima result (Figure 10) decrease with increasing fitness values, which means that the filtered global optima is better than ant colony system solutions. The Hamming and basic swap sequence distances (Figure 11-12) are condensed into one node.

The summary table is illustrated in Table 1.

Table 1. Analysis of the iteration of the Ant Colony System

<table>
<thead>
<tr>
<th>Type</th>
<th>Distance</th>
<th>Lower bound</th>
<th>Upper bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fitness values</td>
<td>Fitness</td>
<td>110,000</td>
<td>130,000</td>
</tr>
<tr>
<td>Average of fitness distances</td>
<td>Fitness</td>
<td>3,500</td>
<td>9,500</td>
</tr>
<tr>
<td>Average of hamming distances</td>
<td>Hamming</td>
<td>10</td>
<td>26</td>
</tr>
<tr>
<td>Average of basic swap sequence distances</td>
<td>Basic swap sequence</td>
<td>8</td>
<td>21</td>
</tr>
<tr>
<td>Fitness distances of the best solution</td>
<td>Fitness</td>
<td>2,000</td>
<td>15,000</td>
</tr>
<tr>
<td>Hamming distances of the best solution</td>
<td>Hamming</td>
<td>24</td>
<td>36</td>
</tr>
<tr>
<td>Basic swap sequence distances of the best solution</td>
<td>Basic swap sequence</td>
<td>20</td>
<td>30</td>
</tr>
<tr>
<td>Cost density</td>
<td></td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>Fitness distance of filtered global optima</td>
<td>Fitness</td>
<td>500</td>
<td>10,500</td>
</tr>
<tr>
<td>Hamming distance of filtered global optima</td>
<td>Hamming</td>
<td>34</td>
<td>36</td>
</tr>
<tr>
<td>Basic swap sequence distance of filtered global optima</td>
<td>Basic swap sequence</td>
<td>26</td>
<td>28</td>
</tr>
</tbody>
</table>
The results of the iteration of the algorithm are illustrated in Figure 13.

4 CONCLUSION

In conclusion, the fitness value of the iterations of the ant colony system algorithm for the examined Vehicle Routing Problem moves on a larger scale. The average distances between Hamming and basic swap sequence are also large. Thus, the distances from the best solution (best iteration solution) are also large, and also from the filtered global optima solution. The cost density values were also 5, which means that 5 iterations resulted in the same fitness value. Based on the test results, the algorithm maps the search space well because there are large average distances between each iteration, however, during the last few iterations the algorithm did not find a better solution, this is also indicated by the maximum value of cost density 5, so the algorithm is small. After the number of iterations, it reaches the optimum (local or global).

5 REFERENCES


