2016 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)

An Improved Ant Colony Optimization and Its Application on TSP Problem

Wei Luo
College of Physics and Information Engineering
Fuzhou University
China, Fuzhou 350108
Email:n141120065@fzu.edu.cn

Dong Lin
College of Physics and Information Engineering
Fuzhou University
China, Fuzhou 350108
Email:lindong@fzu.edu.cn

Xinxin Feng*
College of Physics and Information Engineering
Fuzhou University
China, Fuzhou 350108
Email:fxx1116@fzu.edu.cn

Abstract—**Ant colony optimization (ACO) is a kind of simulated evolutionary algorithm. It imitates ants' foraging process to find the shortest path, coexists with the characteristics of randomness and heuristic. It is applied successfully to solve combinatorial optimization problems, such as the TSP problem, the job-shop scheduling problem, etc. In practical application, ACO has the limitation of easily being trapped into local optimum and long time to converge. We propose an improved ant colony optimization algorithm, consisting of introducing random factor and introducing elitist ants as well as weakened strategy. Random factor provides a direction to search within the field of the optimal path. Elitist ants and weakened strategy strengthens the pheromone above the shortest path and weakens the pheromone above the suboptimal path to decrease the accumulated impact. Both of them shorten the convergence time. Simulation results show that the improved algorithm has a better performance than the traditional one. It can not only find a shorter path but also cost less convergence time, along with satisfactory time complexity. The best path length of TSPlib pr136 we get is 96910, closed to official record 96772 and relative error is 0.14%.**

1. Introduction

In the early 1990s, the Italy scholar Macro Dorigo proposed an ant colony algorithm by imitating ants' foraging process to find the shortest path[1–3]. Compared to other optimization methods, the ACO has a number of advantages which contribute to its wide use. Some of the key advantages of ACO include: (a) It does not need the calculation of derivatives. (b) The knowledge of good solutions is retained by all ants. (c) The ants in the colony share information. Moreover, ACO is less sensitive to the nature of the objective function, and hence it can be used for various objective functions[4]. Then it is applied successfully to solve combinatorial optimization problems, such as the TSP problem, the job-shop scheduling problem, etc. TSP (Traveling Salesman Problem)[5], simply speaking, is one salesman starts and ends at the same city, with all cities being visited once and only once, and the path length required as short as possible. TSP problem is a classic combinatorial optimization problem, and the study of TSP can be used for other reference. To solve TSP, the simplest method is enumerating all possible solutions, and obtain optimal path by comparing[6]. For n cities, the time complexity is $(n-1)!$, which is a large amount of engineering calculation. Because of ACO's numbers of advantages, it has played a critical role in solving TSP problem[7, 8].

In practical application, ACO has some the limitations. (a) It is easily trapped into local optimum. (b) It takes long executing time finding optimal result. These limit the use of popularization and application of the algorithm[9, 10].

In this paper, we propose an improved ant colony optimization algorithm, consisting of introducing random factor and introducing elitist ants as well as weakened strategy (EAWSA). Random factor provides a direction to search within the field of the optimal path. Elitist ants and weakened strategy strengthens the pheromone above the shortest path and weakens the pheromone above the suboptimal path to decrease the accumulated impact. Both of them overcome its long convergence time. In addition, we use common standard performance measurement TSPlib as the data source to show the good performance of our algorithm.

The remainder of this paper is organized as follows. We state the problem and review some existing work in Section II. Section III presents the research motivation and provides details of the proposed methodologies. Computational experiments and analysis of the algorithms are shown in Section IV. Section V concludes the paper and declares the future works.

2. Related Work

Martłn Pedemonte[11], introduced a new taxonomy for classifying software-based parallel ACO algorithms and also presented a systematic and comprehensive survey of the current state-of-the-art on parallel ACO implementations. Each parallel model reviewed was categorized in the new taxonomy proposed, and an insight on trends and perspectives in the field of parallel ACO implementations was provided. Barn B[12], proposed an improved ant colony optimization

978-1-5090-5880-8/16 $31.00 © 2016 IEEE
DOI 10.1109/iThings-GreenCom-CPSCom-SmartData.2016.48

to solve period vehicle routing problem with time windows (PVRPTW), in which the planning period was extended to several days and each customer must be served within a specified time window. Martin Reed[13], demonstrated the use of ACO to solve the capacitated vehicle routing problem, treated as nodes in a spatial network, the use of k-means clustering can greatly improve the efficiency of the solution. The algorithm produces high-quality solutions for two-compartment test problems. Hojjat Salehinejad[14], proposed a multi-parameter A*(Astar)-ants based algorithm in order to find the best optimized multi-parameter path between two desired points in regions. Xinyu Wang[4], presented a novel ACO algorithm to solve the VRP. The proposed algorithm allows ants to go in and out the depots more than once, which simplifies the procedure of constructing feasible solutions.

In this work, we propose an improved ant colony optimization algorithm. Inevitably, we manipulate the transition probability. On one hand, introducing random factor provides a direction to search within the field of the optimal path. On the other hand, EAWSA overcomes its long convergence time by strengthening the pheromone above the shortest path and weakening the pheromone above the suboptimal path it found last to decrease the accumulated impact.

3. Improved Ants Colony Algorithm

3.1. Ants Colony Algorithm Model

For the sake of convenience, we list all of the mathematical symbols used in this article in TABLE I.

According to different pheromone update rule, there are three different models of $\Delta\tau_{ij}^{k}(t+1)$ formulated by Dorigo, called ant-quantity system, ant-density system and ant-cycle system, separately. Large number of experiments show that ant-cycle system performs better than the others, for it considers the global feature [15]. In ant-cycle system,

$$\Delta\tau_{ij}^{k}(t+1) = \begin{cases} \frac{Q}{L_k}, & k\mathrm{th}\ ant\ passes\ route\ (i,j) \\ 0, & otherwise \end{cases}, \quad (1)$$

in which, Q is the amount of pheromone ants released, L_k is the length of path the kth ant visited. We choose ant-cycle system in this paper.

At time t, kth ant follows the transition probability $p_{ij}^{k}(t)$ to choose next city j to visit,

$$p_{ij}^{k}(t) = \begin{cases} \frac{\tau_{ij}^{\alpha}(t)\eta_{ij}^{\beta}(t)}{\sum_{j\in allowed_k}\tau_{ij}^{\alpha}(t)\eta_{ij}^{\beta}(t)}, & j \in allowed_k \\ 0, & otherwise \end{cases}, \quad (2)$$

$allowed_k = \{1, 2, 3, ..., n\}$ is the list of the cities the kth hasn't been visited yet. Weight α is pheromone factor, which denoted as the influence of pheromone concentration to the path choosing. When it equals to 0, the ant currently selects completely according to greedy rule for path planning. Weight β is the heuristic factor, denoted as the influence of distance of two cities to the path choosing. When it equals to 0, the path choosing depends entirely on the pheromone concentration.

TABLE 1: MATHEMATICAL MODEL OF THE ACO

Symbols	Meaning
m	the number of ants in algorithm
n	the number of cities to be visited
d_{ij}	distance between city i and j
η_{ij}	expectation from city i to j, defined as: $\frac{1}{d_{ij}}$
p	pheromone evaporation rate above the path;
$\tau_{ij}^{k}(t+1)$	pheromone concentration from city i to j at time $(t+1)$
$\Delta\tau_{ij}^{k}(t+1)$	pheromone concentration increment from city i to j
tabu	tabu list

The main idea of improving the ACO is to manipulate the transition probability between cities[16], which refers to the pheromone updating. Transition probability involves the pheromone concentration and expectation(parameters of the distance between cities) etc, it is the key point of the ACO.

3.2. Introducing The Random Factor

We set a threshold: $randomSet \in [0,1]$, and reset the corresponding transition probability to next city j:

$$j = \begin{cases} \arg max\{\tau_{ij}^{\alpha}(t)\eta_{ij}^{\beta}(t)\}, & q < randomSet \\ S, & otherwise \end{cases}, \quad (3)$$

S denoted as equation (2), where the city number j derived.

We use computer to generate a random number $q \in [0,1]$. if $q < randomSet$, we choose the city j, where the $\tau_{ij}^{\alpha}(t)\eta_{ij}^{\beta}(t)$ is the maximum, and return city serial number. For example, when we set $randomSet = 0.3$, there will be 30% of the ants obey this transition rule when choose next city otherwise the transition rule follows the equation (2).

By using random factor, the ant not only can be guided to search within the field of the optimal path but also can use the accumulated knowledge to find the better route following roulette algorithm . Therefore it is essential to find the optimal value of the $randomSet$.

3.3. Introducing The Elitist Ants And Weaken Strategy (EAWSA)

For the sake of leading the algorithm to converge quickly, we used EAWSA. We define the ant as the Elitist Ant which travelled in the way of shortest path in each round of iteration, that is m ants complete a circle visit of the cities, and record the optimal path of the current round and the total length L_k. Then we enforce the pheromone above the way elitist ant passed.

We first take the ants' local pheromone updating of each iteration, then strengthen the pheromone concentration above the shortest path to make sure the algorithm converge quickly. The strengthened rule is as follows.

$$\tau_{ij}^{k}(t+1) = (1-p)\tau_{ij}^{k}(t) + \Delta\tau_{ij}^{k}(t+1)\,. \quad (4)$$

Then, we weaken the suboptimal path it found last, for excessive amounts of iteration will accumulate much pheromone on the suboptimal path, affected probability the optimal path being chosen. The pheromone concentration decreasing rule is as follows.

$$\tau_{ij}^{k}(t+1) = \begin{cases} \tau_{ij}^{k}(t+1)(1-p)^{n}, & if\,shorter \\ & path\,found \\ \tau_{ij}^{k}(t+1), & otherwise \end{cases}, \quad (5)$$

where n denoted as the accumulated pheromone weakened times as well as the suboptimal path found last strengthened times, p is pheromone evaporation rate above the path.

We record the strengthened times of suboptimal path last found, and decrease its pheromone in one time effort whenever it finds the better path. In this way, we can alleviate the accumulated impact. And the value of p should be fully considered.

3.4. Improved Version And Its Time Complexity

The pseudocode of the improved version is shown in Algorithm 1.

Algorithm 1 Algorithm With Introducing Random Factor, Elitist Ants And Weakened Strategy

```
Input:city info,antsNumber,randomSet
Initialization: city info, pheromone concentration matrix
while i ≤ executedTimes do
  while j ≤ iterationTimes do
    while k ≤ antsNumber do
      if q ≤ randomSet then
        choose city where the transition probability is
        maximum ;
      else
        choose city ruled by equation (2);
      end if
      add city number to tabu;
    end while
    update the local pheromone and strengthen elitist
    ant's path;
    count the strengthened times of elitist ant's path as
    n;
    record best path;
    if shorterpathfound then
      weaken the suboptimal path last record to the
      power of n;
    end if
    update pheromone concentration above the best path;
  end while
  average executedTimes result taken as output;
end while
```

We combine introducing random factor and EAWSA as the improved version, the time complexity analysis is as follows.

There is no much calculation difference between improved version and traditional algorithm. About introducing random factor, we only change the transition probability slightly, it is an if-else selection, no more redundant operation. As for EAWSA, we change its pheromone updating rule by repeating pheromone strengthened process, we don't change the algorithm itself. The same time complexity, but the better performance.

4. Simulation Experiment And Analysis

In theory, the theoretical definition of the convergence of the ACO is that, the route ants choose is not change or most of ants choose the same one during one iteration[17]. In practical applications, due to the introduction of many random factors, it is difficult to achieve that state in the case of reasonable design. So in practice, as long as the proportion of the pheromone on a certain path is relatively large, or we don't find a better solution during a certain iterations, the algorithm can be considered as reach to a convergence state.

In solving TSP problem, the implementation of the international TSP sample collection case library TSPlib is a common standard to measure the performance of the algorithm[10]. In this paper, we choose pr136.tsp as the data source. Without loss of generality, the verification process takes 500 iteration times, executes algorithm 100 times, and takes the average value as the output.

4.1. Find The Optimal $randomSet$ Value

We set evaporation rate $p = 0.1$. It doesn't matter the value of p at first because all the algorithms are in the same conditions. As shown in the Fig.1, there are six lines in the figure. Both of them collapse at the beginning and the end. With the increase of the $randomSet$, the algorithm drops faster. But when $randomSet = 0.95$, it begins to rebound especially when it takes value 1.00, it presents vertical dropping state.

The reason can be brief analyzed as, if $randomSet$ value is too small, the effect of introducing random factor is not obvious, which means ant can't be guided within the field of the optimal path effectively. If $randomSet$ value closes to 1.00, the algorithm will take a great proportion selecting the city where the transition probability is the maximum every time, that may ignore the other optimal path and will easily trapped into the local optimum.

From the Fig.1, we can tell the differences. In order to guide the ant visit effectively and not be trapped in local optimum,it is advisable to choose $randomSet = 0.90$.

4.2. Find The Optimal Evaporation Rate p Value

We can see in the Fig.2, the smaller value of the p, the better state the algorithm can reach to. When $p = 0.025$, showed by black solid line, it reaches to the bottom and performed outstanding than any others. The line $p = 0.02$, $p = 0.005$, $p = 0.03$ and $p = 0.5$, all located above line $p = 0.025$.

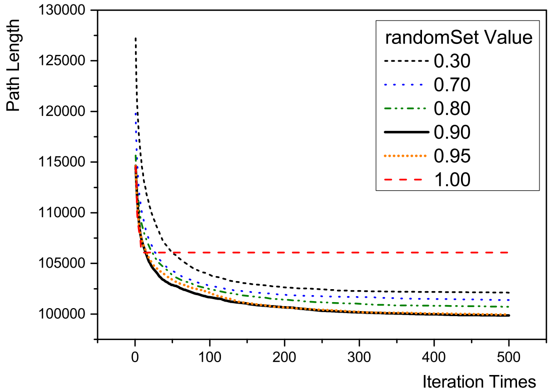

Figure 1: Find Optimal $randomSet$ Value(pr136.tsp)

When p is small, the pheromone updating process is slowed down, for it cannot be substantially increase or decrease, it is a subtle change iteration by iteration. Therefore, the smaller p it is, the shorter path we get. On the other hand, smaller p will slow down the converge rate, for example, when $p = 0.001$, it seems to have the much tendency to decrease, but iteration times has been 500 already.

When p is higher, for example, $p = 0.5$, it converges too fast to have the ability for other optimal path. Higher p value manipulates pheromone substantially. It will get into the local optimum state quickly, which is not what we expect.

In this work, for its excellent ability finding optimal solution is obvious and its satisfactory convergence rate, we set $p = 0.025$.

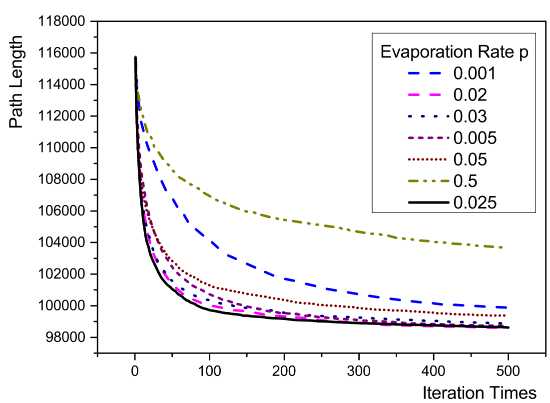

Figure 2: FIND OPTIMAL p VALUE

4.3. Traditional Algorithm And Improved Version

We set the $randomSet = 0.90$, $p = 0.025$, and execute the four algorithms, namely, traditional algorithm, introducing random factor, EAWSA and improved version.

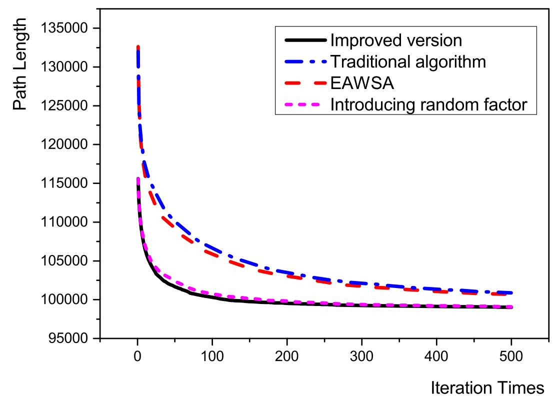

Figure 3: Different Algorithm Results

As shown in the Fig.3, there are four lines in the picture. All of them collapse at the very beginning, different in the middle parts. When algorithm starts, they have no direction to search. But after few iterations, there are some gaps between four lines. And at the end of the algorithm, or the iteration times reach to a threshold, the algorithm got to an end, both of them come to a convergence state.

Firstly, compare with traditional algorithm and EAWSA. Concluded from the picture, latter drops-off a little bit faster than Normal one, although they collapse at the beginning. But after $200th$ iteration, latter one has a tendency to converge while the traditional algorithm still need some iteration time. The reason can be explained as, due to the introduction of the elitist ants strategy, when the shortest path in one iteration constantly strengthened, the pheromone above is higher than the other path, affecting the transition probability, which makes the algorithm converge faster. And whenever it found the better path, the the pheromone above the suboptimal path found last will be weakened, which alleviates accumulated pheromone in formal iteration. Adopting this strategy will surely fasten algorithm's convergence.

Secondly, compare with traditional algorithm and introducing random factor. The latter drops off faster. Obviously, when we execute same iteration times, for example, in $50th$ iteration,traditional algorithm best path length is 11000, longer than introducing random factor, 10300. It also can be expressed as finding the same path, introducing random factor executes less iteration times. And when introducing random factor converges, its best path length can reach to 100000 while the traditional algorithm only can reach to 103000. Introducing random factor provides a direction to search within the field of the optimal path, which shortens the convergence time substantially.

Thirdly, compare with traditional algorithm and improved version. Improved version drops-off substantially, at $150th$ iteration, the algorithm gets to average convergence, faster than any other. Combined with the advantages of the two algorithms above, the improved version performs better is not surprising. It not only can find better path but also can

TABLE 2: Result Comparison among Different Algorithm

Algorithm	item	Pr136	KroA100	Ch130
Traditional	optimal value	97922	21333	6175
algorithm	average value	100867	22026	6219
Introducing	optimal value	97050	21335	6138
random factor	average value	99083	21650	6224
EAWSA	optimal value	97058	21307	6160
	average value	100705	21743	6221
Improved version	optimal value	96910	21285	6110
	average value	99017	21564	6219
Official records	optimal value	96772	21282	6110

converge fast. The simulation result is showed in TABLE II.

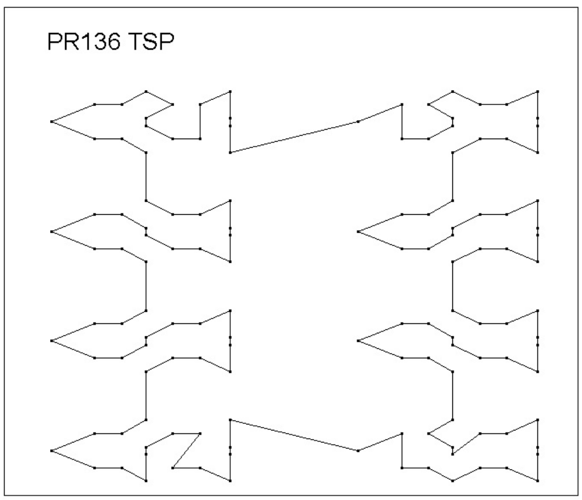

Figure 4: The Specific Route of pr136.tsp Use Improved Version

From TABLE II, we can conclude that, the algorithm introducing random factor and EAWSA both advance than the traditional algorithm, regardless of the data source. The combined improved version, performs better than the others, the advantages is obvious. Compared to the TSPlib official record, the gap is quite small. Pr136's shortest path we get is 96910, close to official record is 96772 and the relative error is 0.14%. Ch130 reaches its official record 6110. KroA100's best path is 21285, relative error is 0.01%.

Fig.4 is the best route map of pr136.tsp. Specific route is as follows.

8,25,24,20,13,9,2,10,11,3,12,21,33,22,23,32,35,43,42,48, 59,58,54,47,44,45,34,46,55,67,56,57,66,69,78,79,68,80,89, 101,90,91,88,81,77,76,87,93,92,100,103,111,115,122,112, 113,102,114,123,132,124,125,133,126,127,110,121,104,116, 120,134,128,129,109,108,117,119,135,130,131,118,106,107, 105,98,95,94,99,86,82,70,75,74,83,85,96,97,84,72,73,71,64, 62,63,51,38,39,50,52,61,60,65,53,49,36,41,40,37,30,28,29, 17,4,5,0,6,16,18,27,26,7,1,15,19,31,14,8.

5. Conclusion And Future Work

To overcome the problem of long time to converge in ACO, we propose an improved version, consisting of introducing random factor and EAWSA. Introducing random factor provides a direction within the field of the optimal path. EAWSA overcomes its long convergence time. Simulation testifies the algorithm introducing random factor and EAWSA both advance than the traditional algorithm, regardless of the data source. The combined improved version, has obvious advantages and performs better than the others. Compared to the official record, the result gap is quite small. Pr136's shortest path we get is 96910, close to official record is 96772 and the relative error is 0.14%. Ch130 reaches its official record 6110. KroA100's best path is 21285 and relative error is 0.01%. Improved algorithm has a better performance than the traditional one. It can not only find a shorter path but also cost less convergence time, along with satisfactory time complexity. ACO has a bright future to further potential applications. After many years of development, it has been gradually infiltrated into other fields, especially in the application of combinatorial optimization problems, such as the vehicle routing problem (VRP), Graph Coloring Problem (GCP), quality of service (QoS), etc. VRP is mainly about selecting path. But it is not often a unilateral parameter constrained problem, but a multiple objective constrained problem recently. It takes the length of the road, safety factor and road conditions into account, which is a multi-objective combinatorial optimization problem. This work can be applied to solve the VRP, which is an important scenario for future application.

Acknowledgments

This work is supported by NSF China (No. U1405251, 61571129, 61601126), Foundation of Fujian Province (Nos. 2015J01250, 2016J01299), Foundation of Fujian Educational Committee (No. JA15089), and Research Fund of Fuzhou University (Nos.2014- XQ-37, XRC-1460, 022572).

References

[1] A. Colorni, M. Dorigo, V. Maniezzo *et al.*, "Distributed optimization by ant colonies," in *Proceedings of the first European conference on artificial life*, vol. 142. Paris, France, 1991, pp. 134–142.

[2] A. Colorni, M. Dorigo *et al.*, "An investigation of some properties of an"ant algorithm"," in *PPSN*, vol. 92, 1992, pp. 509–520.

[3] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: optimization by a colony of cooperating agents," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 26, no. 1, pp. 29–41, 1996.

[4] X. Wang, T.-M. Choi, H. Liu, and X. Yue, "Novel ant colony optimization methods for simplifying solution construction in vehicle routing problems."

[5] J. B. Kruskal, "On the shortest spanning subtree of a graph and the traveling salesman problem," *Proceedings of the American Mathematical society*, vol. 7, no. 1, pp. 48–50, 1956.
[6] I. Chaari, A. Koubaa, H. Bennaceur, S. Trigui, and K. Al-Shalfan, "smartpath: A hybrid aco-ga algorithm for robot path planning," in *2012 IEEE Congress on Evolutionary Computation*. IEEE, 2012, pp. 1–8.
[7] B. Shuang, J. Chen, and Z. Li, "Study on hybrid ps-aco algorithm," *Applied Intelligence*, vol. 34, no. 1, pp. 64–73, 2011.
[8] B. Zhang, X. Sun, L. Gao, and L. Yang, "Endmember extraction of hyperspectral remote sensing images based on the ant colony optimization (aco) algorithm," *IEEE transactions on geoscience and remote sensing*, vol. 49, no. 7, pp. 2635–2646, 2011.
[9] X. Zhang, X. Chen, and Z. He, "An aco-based algorithm for parameter optimization of support vector machines," *Expert Systems with Applications*, vol. 37, no. 9, pp. 6618–6628, 2010.
[10] M. Tuba and R. Jovanovic, "Improved aco algorithm with pheromone correction strategy for the traveling salesman problem," *International Journal of Computers Communications & Control*, vol. 8, no. 3, pp. 477–485, 2013.
[11] M. Pedemonte, S. Nesmachnow, and H. Cancela, "A survey on parallel ant colony optimization," *Applied Soft Computing*, vol. 11, no. 8, pp. 5181–5197, 2011.
[12] B. Barán and M. Schaerer, "A multiobjective ant colony system for vehicle routing problem with time windows." in *Applied Informatics*, 2003, pp. 97–102.
[13] M. Reed, A. Yiannakou, and R. Evering, "An ant colony algorithm for the multi-compartment vehicle routing problem," *Applied Soft Computing*, vol. 15, pp. 169–176, 2014.
[14] H. Salehinejad, H. Nezamabadi-pour, S. Saryazdi, and F. Farrahi-Moghaddam, "Combined a*-ants algorithm: a new multi-parameter vehicle navigation scheme," *arXiv preprint arXiv:1504.07329*, 2015.
[15] H. Y. Shi, "The application of the improved ant colony algorithm on actual vehicle routing problem," Ph.D. dissertation, Shandong University, 2012.
[16] M. Mavrovouniotis and S. Yang, "Ant algorithms with immigrants schemes for the dynamic vehicle routing problem," *Information Sciences*, vol. 294, pp. 456–477, 2015.
[17] W. J. Gutjahr, "A generalized convergence result for the graph-based ant system metaheuristic," *Probability in the Engineering and Informational Sciences*, vol. 17, no. 04, pp. 545–569, 2003.