An Improved Ant Colony Algorithm for Urban Transit Network Optimization

Hong Jiang, Qingsong Yu, Yong Huang
Computer Center
East China Normal University
Shanghai, P. R. of China

***Abstract*—This paper develops an improved Ant Colony Optimization (IACO) algorithm to solve Urban Transit Network Optimization (UTNO) which is a typical nonlinear combinatorial optimization problem. An innovative concept of stagnation counter is used to determine the stages of the IACO. Extra pheromone intensity will be reinforced for the newly discovered path. To trade off between exploration and exploitation, a dynamic parameter setting method is also presented in this paper. It is verified that the solution quality and the convergence speed of our IACO have been improved significantly. A candidate node list for each city and a penalty mechanism for the dead ant are applied in UTNO. The numerical results obtained from a series of benchmark problem instances confirm that our IACO has achieved good results in direct passenger flow rate, line nonlinear factor and line overlap factor.**

Keywords-Ant Colony Algorithm; Ant Colony Optimization (ACO); Urban Transit Network Optimization (UTNO); Combinatorial Optimization; Algorithm Simulation

I. INTRODUCTION

Urban Transit Network Optimization (UTNO) provides rational urban public network and optimizes current public transport capacity by utilizing modern transportation planning theory, operations research, computing technology, artificial intelligence, etc. UTNO can not only exploit public transport's potentiality to the full, improve transit operational efficiency, and reduce urban traffic pressure, but also make full use of public transport capacity, balance traffic distribution, make transport convenient, and reduce public transport cost. UTNO plays an important role in improving the efficiency and effectiveness of public transportation.

Ant colony algorithm, inspired by the foraging behavior of real ants, was first introduced by Dorigo in 1991. It is a new meta-heuristic algorithm. Dorigo et al. [1] elaborated the basic principle of ant colony algorithm and mathematical models. Simulation results revealed that Ant Colony Optimization (ACO) had greater efficiency compared with other bionic algorithms (tabu search, simulated annealing, and genetic algorithm, etc.). Ant colony algorithm was first applied to solve the famous traveling salesman problem (TSP) [2]. Then it was improved and applied to a wide range of engineering and scientific fields. Typical ACO algorithms include EAS [3], ASrank [4], MMAS [5], and ACS [1]. Typical applications of ACO include telecommunication networks [6], combinatorial optimization [7], and data mining [8]. ACO has such advantages as positive feedback, robustness, and interoperability. ACO has good performances in solving combinatorial optimization problems such as UTNO.

II. LITERATURE REVIEW

With the development of computer technology and artificial intelligence, AI-based heuristic algorithms have been well applied in solving UTNO problems. E.g., Patnaik [9] applied genetic algorithms to design public transportation network. Baaj and Mahmassani [10] established an AI-based public transit network optimization model. Liu Hao-de [11] combined improved genetic algorithms with simulated annealing to provide transit routes optimal choice. Wu Kai-jun [12] introduced the spanning tree to represent problem solution and built a mathematical model to find the optimal path.

As ant system (AS) makes use of pheromone positive feedback mechanism, the entropy of the system gradually increases (i.e., from disorder to order). Without external influence, it will have good self-organization and solution search ability, and can obtain the preliminary optimization results for TSP and other combinatorial optimization problems. But with the continuous expansion of test cases, the performance of AS algorithm will decrease greatly. The main deficiencies of AS are as follows:

- Ant colony algorithm is easy to converge prematurely and fall into the local optimum. After a certain number of iterations, the ants may stagnate at one or some local optimal solutions, i.e., all the ants search for the same path. The solutions were exactly the same so that the solution space can not be further explored and global optimal solutions can not be found.
- For large-scale problems, time complexity of AS is O($Nc*m*n^2$, where Nc is the predefined number of iterations, m is the total number of ants in one colony, and n is the number of the cities). Compared with other optimization algorithms, AS has certain advantages in iteration times and solution quality, but the time for problem solving lags behind other algorithms.
- As the ants in AS move randomly, it is difficult for the algorithm to quickly find a better path among the

chaotic paths especially when solving large-scale problems. The algorithm converges slowly in this case.

- The performance of AS is closely related with the setting of algorithm parameters. There is no good reference for parameters setting when algorithms are improved or applied to solve a new problem.

An improved ACO (IACO) algorithm combined with MMAS and ACS is proposed in this paper to solve UTNO problems. The remainder of this paper is structured as follows. Section 3 presents the improved ACO and performance analysis. In section 4, the proposed IACO is applied to UTNO problems. Section 5 provides a conclusion.

III. IMPROVED ACO AND PERFORMANCE ANALYSIS

A. Algorithm description

The improvements of our IACO are as follows:

1) A stagnation counter is proposed to determine whether the algorithm has fallen into stagnation. When there is no better solution found after an iteration, the stagnation counter will increase by 1. When the stagnation counter accumulates to a certain threshold (e.g. 1000), it indicates that the algorithm has fallen into stagnation. Then the amount of pheromone associated with edge (i, j) is reinitialized by

$$\tau_{ij} = \tau_{ij} + \delta(\tau_{\max} - \tau_{ij}), \tag{1}$$

where δ determines the amount of retained pheromone.

After pheromones updated, the order of the pheromone on each path is the same as before but the discrepancy is relatively less than before. This method can facilitate the algorithm's exploration and help to find a new path. The stagnation counter is reset to 0 after pheromone is reinitialized. After the stagnation status is determined, pseudo-random probability will not be used in order to increase the access probability to the path with weaker pheromone concentration. When the ant colony finds a new better solution, pseudo-random probability rule will be reused to enhance the access probability to the path with stronger pheromone concentration.

2) After stagnation, pheromone intensity on the newly discovered optimal path will be reinforced and incented to provide more opportunities to explore the optima. With the stagnation counter, whenever the ant finds a better solution, pheromone intensity is reinforced according to the current value of stagnation counter. Reinforced pheromone intensity can increase the selection probability of the path and help to reserve this better solution. The pheromone intensity associated with edge (i, j) is reinforced by

$$\tau_{ij} = \tau_{ij} + Q\tau_{\max}(i, j \in L^{newbs}), \tag{2}$$

where Q is a constant, τ_{max} is the upper bound of pheromone intensity.

3) Set parameters dynamically. At the beginning of the algorithm, the pheromone concentration on each path is equal that it can not give any guidance to ants. Only heuristic information can provide useful local information. After further iterations, the deposition of pheromone on each path becomes inconsistent so the ants can get more and more guidance from the deposited pheromone on each path whereas the guidance of heuristic information should be weakened gradually. A relatively small α and a relatively large β should be set at the beginning of the algorithm. α should be increased and β should be decreased accordingly in further iterations. In this paper, α and β are set dynamically according to the total number of iterations n:

$$\begin{cases} \alpha = 1, \beta = 5 & 1 \le n < 200 \\ \alpha = 2, \beta = 4 & 200 \le n < 500 \\ \alpha = 2, \beta = 3 & n \ge 500 \end{cases}. \tag{3}$$

At the same time, q_0 ($0 \le q_0 \le 1$) is set dynamically. When q_0 is larger, the algorithm makes full use of the exploited knowledge and is inclined to choose the path with stronger pheromone concentration (exploitation); when q_0 is smaller, the algorithm is inclined to randomly select a path which may be a better solution (exploration). In order to obtain the global solution, q_0 should be set larger at first. With the accumulation of the stagnation counter n', q_0 should be decreased slowly. This can help the ant lay particular stress on the search of the random path. When a better solution is found, q_0 is enlarged for the ants to search the neighboring area of the optimal solution. q_0 is set by the following rules:

$$q_0 = \begin{cases} 0.3 & 0 \le n' \le 200 \\ (500 - n')/1000 & 200 \le n' \le 500 \end{cases}. \tag{4}$$

B. Parameters setting

Initially, let the total number of ants in one colony be equal to the number of the cities, α=1, β=5, ρ=0.02. The value of α and β will be adjusted dynamically during the running of the algorithm. The maximum pheromone τ_{max} is firstly determined by the evaluated optimal path or the optimum obtained by the greedy algorithm, then it is updated when a new path is discovered during iterations. The setting of τ_{min} is associated with the scale of the city, i.e., $\tau_{min}=\tau_{min}/m\_city$. At the same time, τ_{max} and τ_{min} should be updated by the distance of the better solution.

C. Computational experiment and performance analysis

The performance of our IACO was benchmarked using the thirty-one cities of the standard database instances in TSPLIB [13]. The maximum number of iterations is 2000 for each test. The ant algorithm is run for 100 times in each test. Five experiments are tested in total; τ_{max}=0.003125; τ_{min}=0.0001. The experimental results of performance contrast between MMAS and IACO are shown in Table I and Table II.

TABLE I. THE OPTIMAL PATH AND THE EXPERIMENTAL DATA FOUND BY MMAS

No.	Avg. path distance	Optimal path distance	Avg. number of iterations
1	15560.2	15380.5	987.43
2	15561.2	15380.5	1103.11
3	15549.8	15380.5	1118.88
4	15549.8	15380.5	1014.69
5	15552.7	15380.5	1076.8
Average	15554.74		1060.18
The optimal path:			
26 27 25 24 23 19 20 21 17 2 16 18 22 10 5 4 15 3 1 7 8 9 6 12 11 13 14 0 28 30 29			

TABLE II. THE OPTIMAL PATH AND THE EXPERIMENTAL DATA FOUND BY IACO

No.	Avg. path distance	Optimal path distance	Avg. number of iterations
1	15507.7	15377.7	519.88
2	15503.9	15377.7	581.62
3	15514	15377.7	517.7
4	15511.7	15377.7	545.02
5	15507.8	15377.7	578.16
Average	15509.02		548.476
The optimal path:			
26 27 25 24 19 20 21 17 2 16 18 23 10 22 15 3 7 8 9 1 4 5 6 12 11 13 14 0 28 30 29			

The experimental results show that the convergence speed of our IACO has been improved significantly. The average number of iterations to find the optimal solution was reduced from 1060 down to 548. The performance of searching for the optimum is improved substantially. The average path distance was shortened from 15554.74 down to 15509.02. The solution quality and the convergence speed have been improved in Our IACO.

IV. IMPROVED ACO FOR UTNO PROBLEMS

A. Mathematical model for UTNO

Not like in TSP, each ant in UTNO doesn't need to traverse all the nodes. The ant departs from the origin stop, selects the route which satisfies the constraints and the objective function, extends to neighboring transportation districts and bus stops. The extension will continue till the ant reaches the destination stop or dies because of not satisfying the constraints. Therefore the number the ant moves at each iteration is uncertain. Three further improvements for IACO applying to UTNO are proposed as follows:

1) A candidate list $allow_k$ for each bus stop is introduced. In UTNO, not like in TSP, each bus at any stop has at most four optional routes. In order to maximize passenger capacity and serve more passengers, the bus in each route should pause at each stop. In UTNO, the ant will waste a large amount of time to calculate the heuristic information and update pheromone intensity for those nodes it can not reach. So in this paper, a candidate list $allow_k$ is introduced to record the target nodes which the ant can move to. The candidate list can effectively indicate the ant's search preference, reduce the ant's searching complexity, improve the solution quality, and increase the problem solving speed. The probability foumula for the ant to select the next node is updated by:

$$P_{ij}^{k} = \begin{cases} \dfrac{\tau_{ij}^{\alpha}(t)\eta_{ij}^{\beta}}{\sum \tau_{is}^{\alpha}(t)\eta_{is}^{\beta}} & s, j \in allow_i \;\&\notin tabu_k \\ 0 & otherwise \end{cases} . \quad (5)$$

2) Dynamic heuristic information is introduced. According to the definition of the objective function, heuristic information is defined as $\Delta Q_{ij}/T_{ij}$, where T_{ij} represents the travel time from node i to node j, and ΔQ_{ij} represents the newly increased passenger flow from node i to node j. Here double passenger flow is considered:

$$\Delta Q_{ij} = \sum_{s=1}^{i} q_{sj} + \sum_{s=1}^{i} q_{js} , \quad (6)$$

where q_{sj} represents the direct passenger flow from node s to node j. Not like in TSP, the heuristic information can not be obtained in advance but calculated dynamically.

3) Penalty mechanism is introduced. UTNO should meet certain constraints such as suitable route length and travel time, no loop route (bus loop is not considered here), etc. But in the implementation of the algorithm, when all the nodes in the candidate list have been visited by the ant, it will inevitably lead to a loop which results in an unreasonable solution. In such cases, the ant is declared dead, and is imposed penalty on its search path L^{die} to reduce its pheromone intensity associated with edge(i, j), i.e.:

$$\tau_{ij} = (1-\rho^{die})\tau_{ij}\,(i, j \in L^{die}) , \quad (7)$$

where ρ^{die} represents the extra pheromone evaporation parameter on the path of L^{die}.

The objective of UTNO based on IACO is to maximize direct passenger flow and minimize passenger travel time under the condition of satisfying the constraints for transit network design. The mathematical model is built by

$$MaxD_{od} = \sum_{i=0}^{n}\sum_{j=0}^{n} q_{ij}x_{ij} / (Z \times \sum_{i\in F} T_{i,i+1}) , \quad (8)$$

where D_{od} represents direct passenger flow density; q_{ij} is the direct passenger flow from node i to node j; F is the set of transit network nodes, i.e., the set of nodes on the optimal path the ants discovered; T_s represents the shortest travel time between origin node and destination node; x_{ij} is defined by

TABLE III. A REGIONAL TRANSIT PASSENGER O-D MATRIX

O \ D	A	B	C	D	E	F	G	H	I	J	K	L	Σ
A	81	347	501	763	347	201	39	112	31	142	23	59	2646
B	357	32	491	801	377	108	51	67	39	80	28	77	2508
C	511	501	81	691	401	111	62	73	62	96	33	41	2663
D	758	799	701	34	701	231	128	141	72	121	39	103	3828
E	350	381	421	684	38	111	38	47	62	58	41	80	2311
F	202	108	123	241	128	21	41	32	16	81	34	56	1083
G	41	50	52	132	40	39	18	28	13	16	21	60	510
H	108	70	67	150	41	36	27	16	14	18	28	30	605
I	38	40	61	61	58	18	14	15	16	21	28	29	399
J	150	79	101	128	73	79	15	19	20	17	29	41	751
K	28	30	23	42	39	41	20	28	25	28	19	21	344
L	61	80	39	113	79	50	51	25	27	39	24	19	607
Σ	26 85	25 17	26 61	38 40	23 22	10 46	50 4	60 3	39 7	71 7	34 7	61 6	182 55

$$x_{ij} = \begin{cases} 1 & edge\ (i,j)\ on\ the\ planned\ \ line \\ 0 & otherwise \end{cases}, \tag{9}$$

and line nonlinear factor Z is defined as follows:

$$Z = T_k / T_s. \tag{10}$$

B. Numerical analysis for UTNO

A regional transit passenger origin–destination (O-D) matrix is showed in Table III and its corresponding transportation districts and road network are showed in Fig. 1. With the stops of 1, 6, 19 and 24 within the districts of A, B, C, and D as the origin and destination stops, our IACO is applied to optimize the four bus lines within these districts.

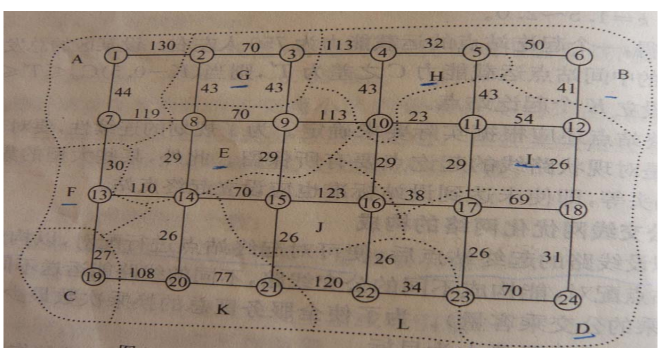

Figure 1. A regional transportation districts and road network

1) Single transit line optimization results

We take the public transport network distribution from district B to district C, i.e., from stop 6 to stop 19 as an example to illustrate the single bus line optimization. The original direction of this line is: 6 5 4 3 2 8 14 20 19; travel time = 471; passenger flow = 4275; direct passenger flow = 9.07643. With the objective of maximum direct passenger flow per unit time and heuristic information defined as q_{ij}/T_{ij}, we run the program for several times and get the same results: direct passenger flow per unit time = 31.7474; time = 978; passenger flow = 31049; the direction of this line is 6 12 18 24 23 22 16 17 11 10 4 3 9 15 21 20 14 8 2 1 7 13 19. Although the direct passenger flow per unit time of this line is actually very large, but it does not accord with the reality. It is too meandrous with large line nonlinear factor and too many bus stops. Considering the influence of line nonlinear factor on passenger flow, we get the results showed in Table IV. The related sketch map is showed in Fig. 2.

TABLE IV. COMPARISON OF DIFFERENT LINE NONLINEAR FACTORS

	Z_{max}=2	Z_{max}=1.5	Z_{max}=1.2
direct passenger flow density	28.7317	18.6135	9.62023
total travel time	794	709	524
direct passenger flow	22813	13197	5041
Line direction	6 12 18 24 23 22 16 17 11 10 9 15 21 20 14 8 7 13 19	6 12 18 24 23 17 11 10 16 22 21 15 9 8 14 20 19	6 5 11 10 9 15 21 20 14 13 19

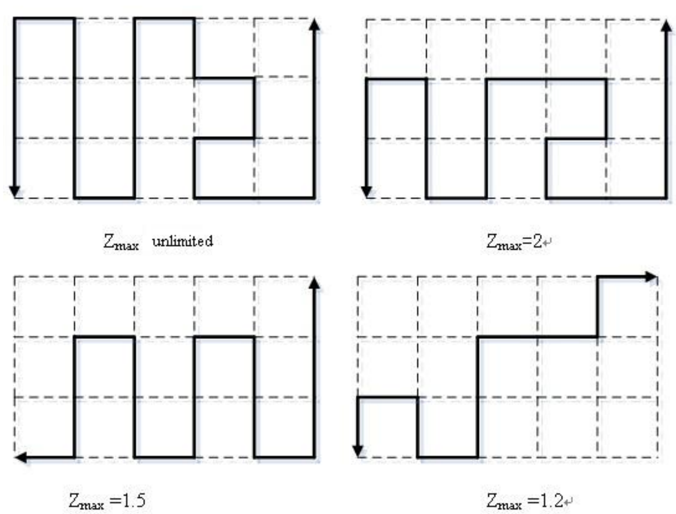

Figure 2. Sketch maps of UTNO with different line nonlinear factors (with maximum direct passenger flow density)

Then with the objective of maximum dynamic direct passenger flow per unit time (Eq. (8)), considering the influence of line length and line nonlinear factor on passenger flow, we solve the UTNO problems with Z_{max}=1.5 and Z_{max}=1.2

respectively and get the same results: dynamic direct passenger flow density = 9.4335; total travel time = 462; direct passenger flow = 4275; the direction of this line is 6 5 11 10 9 8 14 20 19. The related sketch map is showed in Fig. 3.

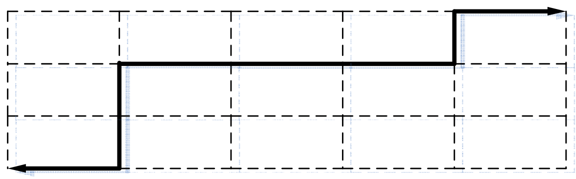

Figure 3. Sketch maps of UTNO (with maximum dynamic direct passenger flow per unit time)

The mathematical model presented in this paper considered the direct impact of line nonlinear factor on the passenger flow. It accords with the passenger mentality, and provides more reasonable solutions for UTNO.

2) Regional transit network optimization results

Optimizing the regional public transportation network for the 4 bus lines of 1→24, 6→19, 19→24, and 1→6, we get the following results: for line 1→24, dynamic direct passenger flow density = 9.41227, the direction of this line is 1 7 13 19 20 14 8 9 10 11 12 18 24; for line 6→19, dynamic direct passenger flow density = 9.32033, the direction of this line is 6 5 4 3 9 15 14 13 19; for line 19→24, dynamic direct passenger flow density = 5.26008, the direction of this line is 19 20 21 15 16 22 23 24; for line 1→6, dynamic direct passenger flow density = 4.75928, the direction of this line is 1 2 3 4 10 11 5 6. The related sketch map is showed in Fig. 4.

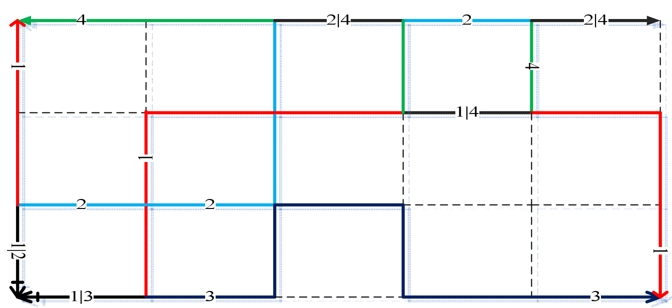

Figure 4. Sketch map of a regional UTNO

V. Conclusion and Future Work

As classical ACOs have the shortcomings of easily falling into stagnation and therefore obtaining local optimal solution and the parameters being difficult to determine when applied to the problems with different scales and different kinds, in this paper, a stagnation counter is proposed to determine the different algorithm stages; the optimal solution discovered by chance is enhanced to reinforce algorithm's learning ability to the optimal path; the parameters are set dynamically to trade off between exploration and exploitation. Then an optimized UTNO model with maximum dynamic direct passenger flow per unit time is presented. The candidate list for each city node and the penalty mechanism for the dead ant are introduced. The experimental study demonstrates that our IACO has improved the solution quality and the convergence speed. It can shorten total travel time, improve direct passenger flow, increase transit network coverage, decrease line overlap factor, and provide more scientific transit network layout.

Future work will consider making the algorithm parallelization to reduce problem solving time and building the solutions with different heuristic information. Furthermore, how to establish a comprehensive transit network optimization system needs further study.

References

[1] Dorigo M, Maniezzo V, Colorni A, "The ant system: optimization by a colony of cooperating agents", IEEE Trans on SMC, 1996, 26(1): 28-41.

[2] Dorigo M, Gambardella L M, "Ant colony system: a cooperative learning approach to the traveling salesman problem", IEEE Trans on Evolutionary Computing, 1997, 1(1): 53–56.

[3] Dorigo M. Optimization, learning and natural algorithms, Milan: Dipartimento di Elettronica, Politecnioco di Milano, 1992.

[4] Bullnheimer B, Hartl R F, Strauss C, "A new rank-based version of the ant system: a computational study", Central European Journal for Operations Research and Economics, 1999: 7(1), 25-38.

[5] Stutzle T, Hoos H, "MAX-MIN ant system", Future Generation Computer System, 2000(16): 237-255.

[6] G. Di Caro, M. Dorigo, "AntNet: distributed stigmergetic control for communications networks", J. Artif. Intell. Res. 9 (1998) 317–365.

[7] Maniezzo V, "Exact and approximate nondeterministic tree search procedures for the quadratic assignment problem", Informs J of Computer, 1999, (11): 358-369.

[8] Rafael S. Parpinelli, Heitor S.Lopes, and Alex A.Freitas, "Data mining with an ant colony optimization algorithm", IEEE Transactions on Evolutionary Cmputing, Vol.6, 2002: 321-332.

[9] Pattnaik SB, Mohan S, Tom VM, "Urban bus transit route network design using genetic algorithm", Journal of Transportation Engineering, 1998, 123(4): 368-375.

[10] Baaj M H, Mahmassani H S. "TRUST: a LISP program for the analysis of transit route configurations", Transportation Research Record, 1990, 1283:125-135.

[11] LIU Hao-de, YANG Xiao-guang, "Research on transit routes network design based on improved genetic algorithm", Computer Engineering and Application, 2007, 43(8): 10-14.

[12] Wu Kai-jun, Zheng Li-ying, "Study on the optimization method of public transit network based on ant colony algorithm", Journal of Lanzhou Jiaotong University, Vol. 26(4): 4-6.

[13] Reinelt G, "TSPLIB - a traveling salesman problem library", ORSA J Comput 1991; 3: 376-384.