ISSN 0146-4116, Automatic Control and Computer Sciences, 2018, Vol. 52, No. 1, pp. 67–76. © Allerton Press, Inc., 2018.

A Hybrid Genetic and Ant Colony Algorithm for Finding the Shortest Path in Dynamic Traffic Networks[1]

Shuijian Zhang* and Ying Zhang

Huzhou Vocational and Technical College, Huzhou, 313000 China

**e-mail: zsj_south@sohu.com*

Received July 27, 2017; in final form, September 29, 2017

Abstract—Solving the dynamic shortest path problem has become important in the development of intelligent transportation systems due to the increasing use of this technology in supplying accurate traffic information. This paper focuses on the problem of finding the dynamic shortest path from a single source to a destination in a given traffic network. The goal of our studies is to develop an algorithm to optimize the journey time for the traveler when traffic conditions are in a state of dynamic change. In this paper, the models of the dynamic traffic network and the dynamic shortest path were investigated. A novel dynamic shortest path algorithm based on hybridizing genetic and ant colony algorithms was developed, and some improvements in the algorithm were made according to the nature of the dynamic traffic network. The performance of the hybrid algorithm was demonstrated through an experiment on a real traffic network. The experimental results proved that the algorithm proposed in this paper could effectively find the optimum path in a dynamic traffic network. This algorithm may be useful for vehicle navigation in intelligent transportation systems.

Keywords: dynamic shortest path, ant colony algorithm, hybrid algorithm, genetic algorithm

DOI: 10.3103/S014641161801008X

1. INTRODUCTION

The shortest path problem in traffic networks is a key element in intelligent traffic systems. It is of importance because of the fact that many travelers face the practical problem of identifying the most efficient route on their daily journeys. The objective of the shortest path problem is to determine the paths for travelers, which would lead to the minimum total travel time. This is a classical combinatorial optimization problem, which has recently attracted more attention from researchers because of increasing levels of traffic congestion, particularly in urban areas.

The shortest path problem can be sub-classified into either the static or dynamic shortest path problem according to the characteristics of the network. The static shortest path problem is to find the shortest path between two points in a deterministic network, which is a P problem. There are many classical algorithms for solving the static shortest path problem. Dijkstra algorithm, Floyd algorithm and Dreyfus algorithm are well-known static shortest path algorithms [1–3]. Some variations of these algorithms are further discussed [4, 5]. Static shortest path algorithms are based on the Bellman optimization principle [6], namely, in the shortest path from the origin node to the destination, the path from the origin node to the intermediate node is also the shortest path to the intermediate node, that is, the sub-path of shortest path is also shortest path. However, although these classical algorithms are effective in static systems, they are not efficient to determine the shortest path in dynamic networks. This is because in dynamic networks the obtained sub-path is the shortest path at any one time and may not be the shortest path at another time.

In traffic networks, shortest path problems are always dynamic. The traffic conditions always change over time (e.g., some road sections may be more crowded than usual during rush hour periods), as a result of these changes, the traveler may need to change his pre-planned shortest path to his destination due to changes in real-time road conditions. An efficient approach is needed to rapidly find the shortest path when the environment changes dynamically. Therefore, the dynamic shortest path problem in real-time traffic networks is to find an optimal path for travelers according to real-time traffic conditions, which is a NP-hard problem. This problem is initially studied by Cooke and Halsey [7], who demonstrate how to

[1] The article is published in the original.

find the shortest route through a modified form of Bellman's iteration scheme. Kim et al. develop decision-making procedures for solving dynamic shortest path problems based on a Markov decision process [8]. Huang et al. propose a heuristic search approach [9]. Ardakani and Tavana introduce a decremental method to speed-up the solving process in dynamic shortest path problem [10].

The scale of urban traffic networks is becoming larger, and as a result, the shortest path problem is now facing new challenges. In large complex networks, it is not wise for the shortest path algorithm to aggressively pursue the accuracy of its solution because the higher the accuracy of an algorithm, the greater time it takes to reach a conclusion, and therefore it cannot meet real-time requirements. To overcome this issue, some scholars have studied how to use heuristic or intelligent optimization algorithms in order to solve the dynamic shortest path problem in complex networks, for example by using the genetic algorithm [11] or the ant colony algorithm [12]. Intelligent algorithm models are simple with few constraints in the objective function. The practice has proven that intelligence algorithms have excellent performance in some complex optimization problems. Various ant colony algorithms have been used for shortest path problems. Głąbowski et al. introduced an ant system called ShortestPathACO algorithm based ant colony optimization for solving the Shortest Path problem [13]. In this study, several problems about using ant colony optimization to find the shortest path were discussed. Attiratanasunthron and Fakcharoenphol studied time complexity for ant colony algorithm applied to single destination shortest path problem [14]. They proved a bound of $O\left(\frac{1}{\rho}n^2 m \log n\right)$ for the expected number of iterations required for an ant colony algorithm for solving single destination shortest path problem on a directed acyclic network with n nodes and m edges, where ρ is an evaporation rate. Angus, D. improved the ability of ant colony algorithm for shortest path problems [15]. They proved the ability of ant colony algorithm to balance multiple factors such as cost and length in solution construction process. Ok, S. Seo and Ahn J.W. proposed a preference-based shortest path ant colony algorithm [16]. In this algorithm shortest path was obtained taking into consideration the properties of the links. Faisal et al. introduced an algorithm based on Ant System called AntStar for solving single-source shortest-path problem [17]. In this case Ant System and A^* algorithm were integrated to enhance the optimization performance.

The purpose of this paper is to further study the shortest path problem in dynamic traffic networks, and to examine how to use features of the genetic and ant colony algorithms to develop an efficient approach to solve the problem. The remainder of this paper is organized as follows: a formal description of the dynamic shortest path problem in traffic networks is provided in Section 2. Section 3 analyzes basic genetic and ant colony algorithms. Section 4 explores a newly developed algorithm based on hybridizing genetic and ant colony algorithms for solving the dynamic shortest path problem. An example of a specific real urban traffic network is then tested as the experimental object to evaluate the hybrid algorithm proposed in this paper in Section 5. Finally, the results of this study are summarized and the future applications of the algorithm are discussed in Section 6.

2. MODEL FORMULATION

2.1. Modeling the Dynamic Traffic Network

A dynamic traffic network can be modeled as a directed and connected weighted network $G:\{V,E,P,W_{(e,t)}\}$, where V is a set of the nodes of the traffic network, $V=\{(1,2,\cdots,n)|n \text{ is the node number}\}$, where a node represents an intersection of the traffic network, as shown in Fig. 1; E is a set of road arcs within the traffic network, $E\in V\times V=\{e_{(i,j)}|i\neq j;\quad i,j\in V\}$, where a road arc $e_{(i,j)}$ represents a link between node i and j; P is a set of the coordinates of the nodes of the traffic network, $P=\{(x_i,y_i)|x,y\in R^+;\quad i\in V\}$; $W_{(e,t)}$ refers to a set of weights of the road arcs at time t, where the values are measured using the travel time of the road arcs, and each road arc $e\in E$ has an associated travel time T_e at time t. The traffic conditions of a road arc might change over time, which can lead to a change in the travel time of the road arc.

2.2. Dynamic Shortest Path Model

In this paper, we focus on a single-source single-destination shortest path problem which is to find the shortest path from a given source node to a destination. A path in a traffic network from a source node to a destination is defined as an alternating sequence of different nodes and road arcs

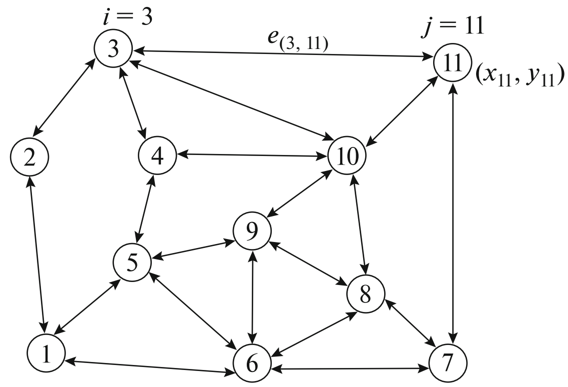

Fig. 1. Example of a traffic network.

$\left(s, e_{(s,k)}, k, e_{(k,h)}, h, \ldots, m, e_{(m,d)}, d\right)$, where the node s is the source node, node d is the destination, and there is no any circuit in the path.

In *DTN*, assuming $R_{i,j}^{t}(n)$ is the set of all paths between node i and node j at time t, the number of all paths is $N = \left|R_{i,j}^{t}(n)\right|$, $R_{i,j}^{t}(l)$ denotes one of paths in $R_{i,j}^{t}(n)$, if there is one path $R_{i,j}^{t}(k) \in R_{i,j}^{t}(n)$, and the following formula is workable for all $l \in N$:

$$F\left(R_{i,j}^{t}(k)\right) \le F\left(R_{i,j}^{t}(l)\right), \tag{1}$$

where $F(\bullet)$ is the total travel time of a path, k denotes one paths from node i to node j, $F(\bullet) = \sum\left(w_{(e,t)}\right)$, then $R_{i,j}^{t}(k)$ is defined as the shortest path between node i and node j at time t in the dynamic traffic network.

The formal mathematic model that describes the dynamic shortest path problem is shown as follows:

$$\text{Min } T = \sum_{e_{(i,j)} \in E} W_{\left(e_{(i,j)}, t\right)} X_{ij}, \tag{2}$$

s.t.

$$X_{ij} \in \{0,1\}; \quad i, j = 1, 2, \ldots, n; \quad i \ne j, \tag{3}$$

$$\left(\sum_{i,j \in B} X_{ij}\right) < |B|; \quad 2 \le |B| \le n - 2, \quad B \subset \{1, 2, \ldots, n\}, \tag{4}$$

where formula (2) is the object function, T is the sum of travel time of all road arcs in one path from the original node to the destination; $W_{\left(e_{(i,j)},\ t\right)}$ is the travel time of road arc $e_{(i,j)}$ at time t, X_{ij} denotes whether or not the path passes the road arc $e_{(i,j)}$; $X_{ij} = 1$ means the path passes road arc $e_{(i,j)}$ at a point in time, $X_{ij} = 0$ means the path does not pass road arc $e_{(i,j)}$. The constraint condition formula (4) can avoid the loop in the path; set B is the subset of the set of all nodes that the path passes, where $|B|$ is the number of inequality nodes in set B.

3. GENETIC AND ANT COLONY ALGORITHMS

3.1. A Simple Genetic Algorithm

A genetic algorithm is a type of nature-inspired algorithm designed by simulating the process of organic evolution [18]. It includes three basic operations: selection, crossover, and mutation operations. The algorithm is designed to find a solution through a number of iterations. At each iteration, some solutions are selected in a population according to the fitness of each individual in the population, and then new solutions are created through crossover and mutation operations. In the end, an optimum solution is obtained. Genetic algorithms are used in many optimization problems [19].

3.2. Basic Ant Colony Algorithm

The ant colony algorithm is a swarm intelligence algorithm [20]. It simulates the phenomenon that a large number of ants will ultimately find the shortest path from the ant colony to the food location in the foraging process. The ant colony algorithm is often used for solving optimization problems [21].

In the foraging process, ants choose their next road arc according to the pheromones they sense with a random ratio rule; namely ant *k* that is in location *i* at moment *t* moves to *j* according to a certain transition probability. The transition probability is calculated using formula (5) as follows:

$$P_{ij}^{k}(t)=\begin{cases}\dfrac{\tau_{ij}^{\alpha}(t)\eta_{ij}^{\beta}(t)}{\sum_{s\in \mathrm{nallowed}_k}\tau_{is}^{\alpha}(t)\eta_{is}^{\beta}(t)}, & j\in \mathrm{nallowed}_k\\ 0 & \text{otherwise,}\end{cases} \tag{5}$$

where $P_{ij}^{k}(t)$ denotes the probability of ant *k* choosing road arc $e_{(i,j)}$ in location *i* at moment *t*, $\tau_{ij}(t)$ which represents the amount of pheromones present on road arc $e_{(i,j)}$ at moment *t*, and η_{ij} is the visible coefficient from the location *i* to *j*; parameter α denotes the significance of pheromones for the ant selecting the travel direction, and parameter β denotes the significance of the heuristic information; the set $nallowed_k=\{0,1,\ldots,m\}$ represents the next position, which the ant *k* is allowed to reach from position *i*, it is a subset of the nodes of the network. It can be known from formula (5) that the transition probability $P_{ij}^{k}(t)$ is directly proportional to $\tau_{ij}^{\alpha}\eta_{ij}^{\beta}$; when the amount of pheromones on the road arc is more and the heuristic information is more important, then the ant more likely chooses the road arc. After all ants finish finding paths once, the amount of pheromones on each road arc will be updated as follows:

$$\tau_{ij}(t+1)=\rho\tau_{ij}(t)+\Delta\tau_{ij}(t,t+1), \tag{6}$$

$$\Delta\tau_{ij}(t,t+1)=\sum_{k=1}^{m}\Delta\tau_{ij}^{k}(t,t+1), \tag{7}$$

where $\Delta\tau_{ij}^{k}(t,t+1)$ represents the amount of pheromones that the k-th ant left on road arc $e_{(i,j)}$; $\Delta\tau_{ij}(t,t+1)$ which represents the total amount of pheromones that the ant colony left on road arc $e_{(i,j)}$ within one loop; the pheromones on a road arc volatilize at a certain rate; $(1-\rho)$ is the volatile coefficient, where ρ, the residual coefficient is $\rho<1$.

4. HYBRID ALGORITHM

The basic hybrid strategy is to fuse a genetic algorithm into an ant colony algorithm. This is done in order to make full use of the high-efficiency performance of the ant colony algorithm in order to obtain the globally optimal solution rapidly in combination with the beneficial properties of a genetic algorithm, such as stochastic, global convergence.

In traffic networks, the traffic flow of the road arcs always changes over time, therefore, the travel time of road arcs also change over time, which are subject to the states of the road arcs. For instance, in a traffic network, one unexpected traffic accident would block the traffic flow on the road arc where the accident takes place. Then, the value of pheromones on the road arc should be assigned again.

The following is a formal description of the hybrid algorithm for finding the dynamic shortest path. When the algorithm satisfies one of the following conditions, the algorithm terminates: (1) the number of iterations reaches a given maximum number; it indicates that the ants have already done enough works; (2) the same optimal solution is repeated multiple times, it means that the algorithm has converged and no longer needs to continue.

Begin

Step 1: $t=0$.

Step 2: Initialize the parameters of the algorithm.

Step 3: Assign the value of pheromones on the road according to road conditions.

Step 4: Search for the shortest path with the ant colony algorithm; if the solution meets the termination conditions, then output the shortest path; or execute the genetic operations, and go into the next iteration computation.

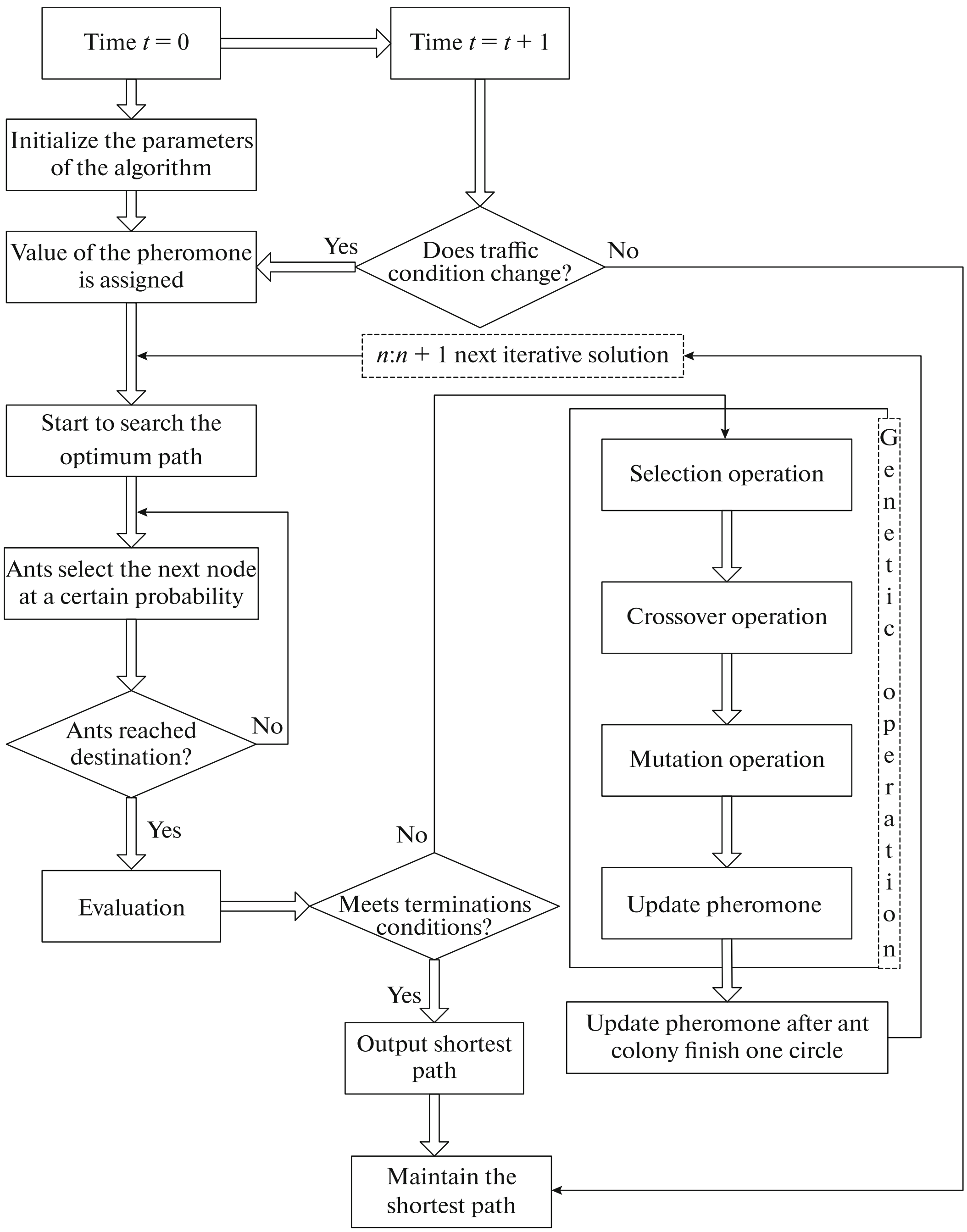

Fig. 2. Flowchart of the hybrid algorithm scheme.

Do

Step 5: $t = t + 1$, if the travel time of road arcs change and affects the planned shortest path at time $t = t + 1$, then update the pheromones on the roads whose conditions have changed, carry out the operations of step 4, or maintain the shortest path.

While (the traveler reaches the destination)

End

The flow of the hybrid algorithm is shown below in Fig. 2.

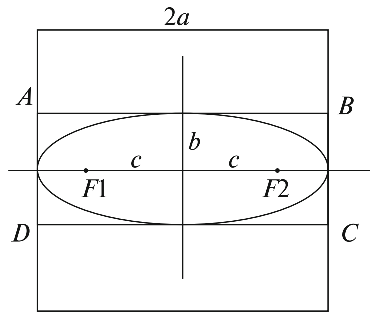

Fig. 3. Calculation of the search scope in the algorithm.

4.1. Improvements on the Search Method

In a spatial network, it is very time consuming to find the shortest paths if the network condition is updated frequently. The basic ant colony algorithm searches the whole network, which is so time consuming that it cannot identify the shortest paths efficiently. To overcome this drawback, we propose a novel method to prune the algorithm's search scope.

As described above, the ant colony algorithm simulates the principle that ants search for the shortest path in the foraging process. In the foraging process, ants search for the shortest path within their reachable area. A traffic network is a geographical network with a certain spatial distribution of each of its intersections and road arcs. The shortest path between two nodes always distributes along the straight line between the two nodes and will not deviate too far from the line; namely, the shortest path will be restricted within a certain spatial scope, so the search operation of an algorithm can be limited to this scope. Through this method, the algorithm can avoid searching the invalid space, and further improve its efficiency.

Search scope can be determined according to the following method, as shown in Fig. 3. Given that $F1$ and $F2$ represent the origin node and the destination, we assume $F1$ and $F2$ as elliptic foci and obtain the linear distance between $F1$ and $F2$ as the elliptic focal length $2C$; one-third of the area of a square whose side length is the length of the long axis $2a$ is the elliptical area; then we can get the following formula:

$$\frac{1}{3}(2a \times 2a) = \pi ab, \tag{8}$$

where b is the minor semi-axis of the ellipse; according to elliptical characteristics we know $b = \sqrt{a^2 - c^2}$ and can substitute it into formula (8):

$$\frac{4}{3}a = \pi\sqrt{a^2 - c^2}. \tag{9}$$

Then the following formula can be obtained from formula (9):

$$a = \sqrt{\pi^2 c^2 / \left(\pi^2 - 16/9\right)} \approx 1.1c. \tag{10}$$

Therefore, the elliptical area can be determined. For convenience, we take the exterior contact rectangle $ABCD$ as our search scope for calculations (shown in Fig. 3).

In the foraging process, ants search for their next walking direction without guidance. The basic ant algorithm simulates this process to select the next location in the process of marching forward, which also occurs according to the random principle, and this will affect the algorithm's search efficiency. In order to solve this problem, we improved the transition rule. As analyzed above, in a traffic network, the shortest path between the origin node and the destination node distributes along the line between the origin node and the destination, therefore we can use the following approach to improve the transition rule. As shown in Fig. 4, given A is the origin node, B is the destination, and an ant has moved to node i from A, and given that there are nodes of j and m to choose to walk to, it can be seen from the figure that the value of included angle θ between line $\overline{jB}$ and $\overline{AB}$ is less than the included angle φ between line $\overline{mB}$ and $\overline{AB}$; it can also be seen that line $\overline{ij}$ inclines to the line $\overline{AB}$ more. If the amount of pheromones on road arc $e_{(i,j)}$ are approximately equal to that of road arc $e_{(i,m)}$ and the road conditions of the two road arcs are similar, then $e_{(i,j)}$ will be selected to walk along, which is in accord with the psychology of people in the selection of their travel path.

In addition to selecting the path with the shortest travel time, people will usually choose main roads or wide roads with good road conditions. Therefore, such factors shall be considered when calculating the transition probability.

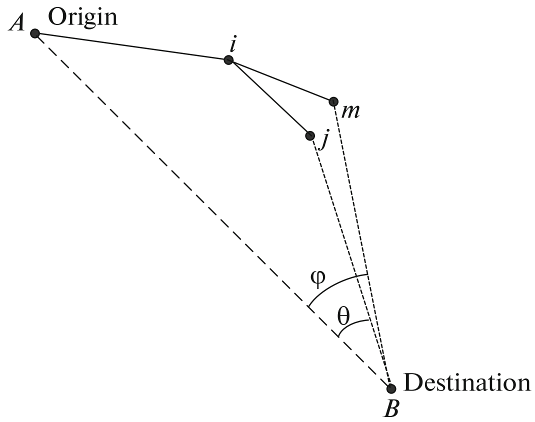

Fig. 4. Example of search direction in the hybrid algorithm.

In the improved ant colony algorithm, the coefficient η_{ij} in formula (1) is modified when calculating the transition probability of ants selecting their next position in the process of marching forward; the heuristic information of the road is considered when calculating η_{ij} as shown in the following formula:

$$\eta_{ij} = 1/\mu\theta, \tag{11}$$

where μ denotes the coefficient of the road condition, μ will be smaller as the road is better for traveling. Coefficient θ is described above. It can be known from formula (11) that η_{ij} will be larger with improving road conditions and as the road inclines towards the line between the origin node and destination, it is more likely the road will be selected.

4.2. Improvements in Genetic Operations

The Genetic algorithm generates new individuals in a population through the crossover and mutation operations while the optimum solution is found through the selection operation. The genetic operation can produce a large number of new individuals when the crossover and mutation rates are relatively large, which ensures the global optimization capability of the algorithm. In this paper, the proposed algorithm fuses the crossover operation of the genetic algorithm with the ant colony algorithm to enable to application of the crossover operation to the paths found by the ant colony in order to generate new solutions. This endows the algorithm with a stronger ability to find new solutions. The traffic network is a sparse network, which is unlike the complete graph in the TSP problem; therefore, unlawful individuals will be generated in the crossover operation. In order to avoid modification of the crossover operation, the specific procedures are performed as follows:

Step 1: Select chromosomes for the crossover operation according to a certain crossover rate P_c; and randomly choose two chromosomes for mating.

Step 2: Scan the parent chromosomes; find two nearest nodes in the two chromosomes according to the coordinates of each node, and then carry out the crossover operation between the nodes.

In order to elaborate on the crossover operation described above, the traffic network shown in Fig. 1 is used as an example (as shown in Fig. 5). Given there are two effective paths from node 1 to node 11: $1 \to 5 \to 4 \to 10 \to 11$ and $1 \to 5 \to 6 \to 8 \to 7 \to 11$; the two paths are taken as parent chromosomes; then we find the two nearest nodes on the two chromosomes according to the coordinates of each node; through calculation the two nodes found are node 10 and node 8. Then, the crossover operation is carried out between the two nodes by swapping sub-path $10 \to 11$ with sub-path $8 \to 7 \to 11$ between the parent chromosomes; and then by finding a connected path between node 4 and node 8 with the way-finding rule of the ant colony algorithm, and by finding a connected path between node 6 and node 10 through the operation in the same way, two lawful offspring are generated.

In the genetic algorithm, the chromosomes mutate at a certain probability, and a large number of new individuals are generated by the mutation operation, which is equivalent to making new searches in an unsearched solution space. This process can improve the global searching ability of the algorithm where the best individuals are retained through the selection operation so that the solution quality is improved.

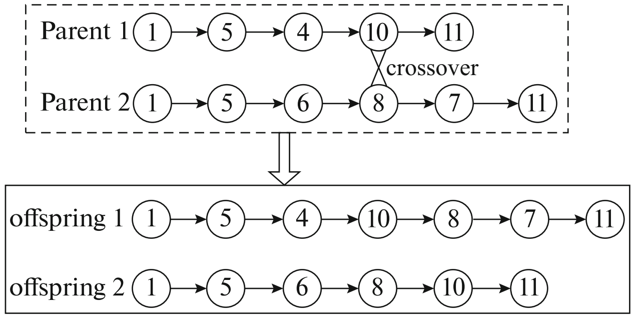

Fig. 5. Diagrammatic representation of the crossover operation from the genetic algorithm.

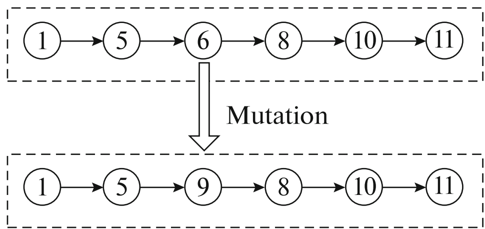

Fig. 6. Example of a mutation operation in a genetic algorithm.

Better solutions can be obtained through less iterative calculations, hence, the convergence rate of the algorithm is accelerated. Fusing the mutation operation of the genetic algorithm into the ant colony algorithm is a good way to improve the solution quality of each generation of the ant colony algorithm and to speed up the convergence of the algorithms.

As mentioned above, the traffic network is a sparse graph; a disconnected path would be generated through the mutation operation of a simple genetic algorithm, due to the creation of unlawful individuals. Therefore, the following measures are made in order to improve the mutation operation for avoiding the generation of unlawful individuals:

Step 1: Select gene location for mutation in all paths at a certain probability.

Step 2: Set the gene (network node) as disconnected and then find a connected path to connect before and after the gene according to the way-finding rule of the ant colony algorithm.

In order to explain the mutation operation described above, still using the traffic network described in Fig. 1 as our example (as shown in Fig. 6), given one gene (node 6) of offspring 2 generated by the crossover operation (the above example) is selected as the mutated gene, set the position of node 6 as disconnected, then find a connected path between node 5 and node 8, thus a new path is found. That is, a new individual is generated.

5. EXPERIMENT

In this section, an experiment was conducted to evaluate the dynamic shortest path algorithm proposed above. A traffic network, which is composed of 1017 nodes and 1007 road arcs (Fig. 7) was taken as the experimental object. The rush hour (7:30 am to 8:30 am) was selected as the time period for research. The selected time extent is divided into 12 sections, each composed of a 5-min interval. The experiment simulated the traffic information issuing platform to update traffic information in real-time. For example, for traffic congestion, which occurred in a certain road arc the estimated congestion duration time was provided, and if the planned path passed the congested road arc, then the pheromone level of the road arc was updated and the algorithm was triggered to re-compute the shortest path. In the experiment we compared the performance of the hybrid algorithm with the basic ant colony algorithm.

The algorithm was implemented using the programming language C#, based on a digital map, which is constructed on the platform of the geographic information system software ArcGIS10.2. The algorithm parameters were set as follows: the population size of the ant colony was *pop_size* = 30; the maximum iteration generations *maxgen* = 200; the amount of initial pheromone was $\tau_c = 10$; the amount of pheromones that the ants leave in the path after each iteration was $Q = 100$; the residual coefficient of pheromone was

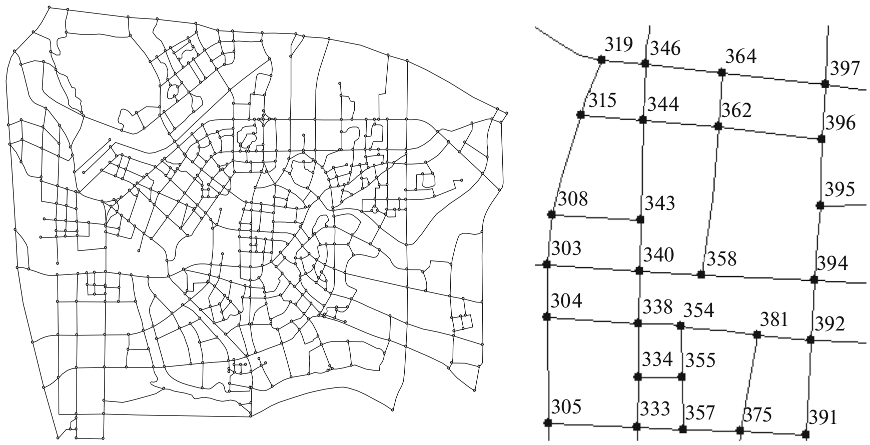

Fig. 7. A single urban traffic network that was used in the experimental setting (the figure on the right is a partially enlarged drawing of the network on the left; the numbers are serial numbers of nodes).

$\rho = 0.8$, where parameter $\alpha = 1$ and $\beta = 2$; the crossover probability was $P_c = 0.85$ and mutation probability was $p_m = 0.02$. Six origin-destinations pairs had been selected from the experimental traffic network to conduct the experiment. The shortest paths of each origin-destination pair were calculated each time the traffic conditions changed. The experimental results are shown below in Table 1 (the solution was the shortest time to arrive at the destination). The results showed that the algorithm proposed in this paper had improved astringency and high efficiency. This is shown in Table 1, which reveals that the hybrid algorithm is several times as fast as the basic ant algorithm. There was a clear improvement in the performance of the hybrid algorithm compared with the basic ant colony algorithm.

It can be seen from the experimental results that the hybrid algorithm can avoid falling into the local solutions through integrating genetic algorithm into the ant colony algorithm; and then can find better solutions. The efficiency of the algorithm has been greatly improved by the improvement on the algorithm according to the spatial characteristics of traffic networks.

6. DISCUSSION AND CONCLUSION

Under conditions when congestion in traffic networks is a common occurrence, there is significant practical meaning in studying the dynamic shortest path problem. In this paper, we analyzed the spatial distribution characteristics and dynamics of a traffic network; we analyzed the characteristics of a genetic algorithm and an ant colony algorithm; discussed the basic method and strategy to integrate the genetic algorithm into an ant colony algorithm; and based on that we developed a novel dynamic shortest path algorithm. Our goal was to find effective methods to identify the optimal path in a dynamic traffic network. In order to study the performance of the new algorithm, an experiment was designed to test its performance. The experimental results showed that the proposed algorithm had a much-improved performance compared with the unmodified ant colony algorithm; therefore, the novel algorithm is practicable.

Table 1. Experimental results

Origin–destination	Basic ant colony algorithm		Hybrid algorithm	
	shortest path, min	CPU time, s	shortest path, min	CPU time, s
1–616	27.4	19.7	24.3	3.6
20–568	28.3	24.3	25.6	3.9
77–288	20.6	16.2	19.8	2.7
506–60	22.1	17.5	20.3	3.4
256–450	17.5	15.7	15.4	2.5
158–497	11.9	14.4	10.5	2.2
10–583	26.4	23.3	24.7	3.8

From the above experimental results, it can be known that it is very effective that integrating ant colony algorithm and genetic algorithm to find the shortest path between two nodes in the traffic network. In the hybrid algorithm, the excellent performance of ant colony algorithm and genetic algorithm are fully exhibited. In this paper, the characteristics of the traffic network are taken into account during the process of searching the shortest path, therefore, the intelligent algorithms make full use of their excellent performance and the efficiency of the algorithm is greatly improved. Genetic algorithms and ant colony algorithms are intelligent optimization algorithms with fine properties. The main purpose of fusing the two intelligence algorithms was to enhance the problem solving ability of the algorithm in order to obtain better quality solutions than the traditional algorithms. The dynamic shortest path algorithm proposed in this paper made full use of the advantages of the ant colony and the genetic algorithms, and overcame their weaknesses. We proved that the hybrid algorithm proposed in this paper had good stringency through the experimental testing, and this can therefore solve dynamic shortest path problems in traffic networks effectively.

In the future study, we will carry out the study on the theoretical analysis of complexity of the algorithm, and study how to apply it to guide travelers along the optimum path to travel.

ACKNOWLEDGMENTS

The authors acknowledge comments from anonymous referees. This work has been supported in part by Natural Science Research Project of Education Department of Zhejiang Province (Y201432450), and Huzhou Municipal Science and Technology project (2014YZ09).

REFERENCES

1. Dijkstra, E.W., A note on two problems in connection with graphs, *Numer. Math.,* 1959, vol. 1, no. 1, pp. 269–271.
2. Floyd, R.W., Algorithm 97: Shortest path, *Commun. ACM.*, 1962, vol. 5, no. 6, p. 345.
3. Dreyfus, S., An appraisal of some shortest path algorithms, *Oper. Res.,* 1969, vol. 17, no. 3, pp. 395–412.
4. Ramalingam, G. and Reps, T., An incremental algorithm for a generalization of the shortest-path problem, *J. Algorithms,* 1996, vol. 21, no. 2, pp. 267–305.
5. Santos, L., Coutinho, J., and Current, J.R., An improved solution algorithm for the constrained shortest path problem, *Transp. Res. A: Methodol.,* 2007, vol. 41, no. 7, pp. 756–771.
6. Bellman, E., On a routing problem, *Q. Appl. Math.,* 1958, vol. 16, no. 1, pp. 87–90.
7. Cooke, L. and Halsey, E., The shortest route through a network with time-dependent internodal transit times, *J. Math. Anal. Appl.,* 1966, vol. 14, no. 3, pp. 492–498.
8. Kim, S., Lewis, M., and White, C., Optimal vehicle routing with real-time traffic information, *IEEE Trans. Intell. Transp. Syst.,* 2005, vol. 6, no. 2, pp. 178–188.
9. Huang, B., Wu, Q., and Zhan, F.B., A shortest path algorithm with novel heuristics for dynamic transportation networks, *Int. J. Geogr. Inf. Sci.,* 2007, vol. 21, no. 6, pp. 625–644.
10. Ardakani, M.K. and Tavana, M., A decremental approach with the A* algorithm for speeding up the optimization process in dynamic shortest path problems, *Measurement,* 2015, vol. 60, pp. 299–307.
11. Hanshar, F.T. and Ombuki-Berman, B.M., Dynamic vehicle routing using genetic algorithms, *Appl. Intell.,* 2007, vol. 27, no. 1, pp. 89–99.
12. Lissovoi, A. and Witt, C., Runtime analysis of ant colony optimization on dynamic shortest path problems, *Theor. Comput. Sci.,* 2015, vol. 561, pp. 73–85.
13. Głąbowski, M., Musznicki, B., Nowak, P., and Zwierzykowski, P., Shortest path problem solving based on ant colony optimization metaheuristic, *Image Process. Commun.,* 2012, vol. 17, pp. 7–17.
14. Attiratanasunthron, N. and Fakcharoenphol, J., A running time analysis of ant colony optimization algorithm for shortest paths in directed acyclic graphs, *Inf. Process. Lett.,* 2008, vol. 105, pp. 8–92.
15. Angus, D., Solving a unique Shortest Path problem using ant colony optimization, Communicated by Baeck, T., 2005.
16. Seung-Ho Ok, Woo-Jin Seo, Jin-Ho Ahn, Sungho Kang, and Byungin Moon, An ant colony optimization approach for the preference-based shortest path search, *J. Chin. Inst. Eng.,* 2011, vol. 34, no. 2, pp. 181–196.
17. Faisal, M., Mathkour, H., and Alsulaiman, M., AntStar: Enhancing optimization problems by integrating an Ant System and A* algorithm, *Sci. Program.,* 2016, vol. 2016, pp. 1–12.
18. Holland, J.H., Genetic algorithms, *Sci. Am.,* 1992, vol. 267, no. 1, pp. 66–72.
19. Li, S.G., Genetic algorithm for solving dynamic simultaneous route and departure time equilibrium problem, *Transport,* 2008, vol. 23, no. 1, pp. 73–77.
20. Dorigo, M., Birattari, M., and Stutzle, T., Ant colony optimization, *IEEE. Comput. Intell. Mag.,* 2006, vol. 1, no. 4, pp. 28–39.
21. Fouad, A., Boukhetala, D., Boudjema, F., Zenger, K., and Gao, X.Z., A novel global harmony search method based on ant colony optimisation algorithm, *J. Exp. Theor. Artif. Intell.,* 2016, vol. 28, nos. 1–2, pp. 215–238.