International Journal of Hybrid Information Technology
Vol.8, No.7 (2015), pp.189-198
http://dx.doi.org/10.14257/ijhit.2015.8.7.17

Incorporating Bidirectional Heuristic Search and Improved ACO in Route Planning

Guangming Xiong*, Xiaoyun Li, Shuai Zhou, Hao Lu, Yan Jiang and Jianwei Gong

Intelligent Vehicle Research Center, Beijing Institute of Technology, Beijing,China
xiongguangming@bit.edu.cn

Abstract

A multi-objective multi-node dynamic route planning system for a vehicle is presented in this paper. In this system, a bidirectional heuristic search algorithm is designed to perform path planning between two nodes in a topological map created by OpenStreetMap for urban scenarios. And then an improved multi-objective Ant Colony Optimization (ACO) algorithm considering the timeliness of goal nodes is proposed to generate the node sequence. Experimental results validated the proposed approach.

Keywords: *bidirectional heuristic search; improved ACO; route planning*

1. Introduction

In recent years, with the rapid increase in traffic volume and the complexity of transportation networks, congestion and the risk of accidents are becoming increasingly critical issues worldwide [1]. For example, in China, the average time of road network congestion is about one hour and forty minutes on weekdays in 2013. Meanwhile, there were over 55 000 fatalities in that year. Intelligent Transportation System (ITS) is widely considered as an effective solution to this challenge [2].

ITS covers all modes of transportation and considers all elements of the transportation system – the vehicle, the infrastructure, and the driver or user, interacting together dynamically. The overall function of ITS is to improve decision making, often in real time, by transportation network controllers and other users, thereby building up a real-time, accurate, and efficient transportation system. Undoubtedly, ITS will play an important role in people’s everyday life.

In an urban scenario, a vehicle may need to visit several nodes for a certain mission. This is similar to the traveling salesman problem (TSP). In the traveling salesman problem, given a set of cities (nodes), the task is to determine the shortest route to visit each city (node) only once and return to the starting city. If the distance between any two nodes is determined by Euclidean distance, it is called the Euclidean Traveling Salesman Problem (ETSP). However, in real urban scenarios, it is not feasible to use Euclidean distance instead of the real distance. Yu X. attempted to solve the traveling salesman problem for car-like robots while considering only a minimum turning radius [3]. However, he did not consider real traffic rules and scenarios.

In this paper, we propose a two-stage planning method to address the challenges of the multi-node route planning problem of a vehicle driving in urban scenarios. The first stage is to plan out the path between any two nodes which is used to navigate the vehicle in the real urban scenarios while obeying the traffic rules. The second stage is to generate a node sequence by which the vehicle will visit each node only once and then return to the starting point.

For the traditional multi-node route planning problem, tasks with multi-objective optimization often need to be taken into consideration. In these tasks, objectives other

ISSN: 1738-9968 IJHIT
Copyright ⓒ 2015 SERSC

than whole route distance, such as traffic situations, traveling orders, and resource utilization [4-6] should also be considered in order to reach suitable solutions. Larsen et al. presented the dynamic traveling salesman problem with Time Windows [7], in which each node of a random subset must be visited within a certain time period. However, in some situations, the timeliness of visiting the nodes must be considered. Timeliness refers to the importance of the nodes, or the priority of the nodes that determines which nodes need to be visited first. In some applications, for example, city rescue, it is critical to consider the priority of nodes. Minimizing the global travel cost and satisfying different timeliness requirements are the two objectives to be optimized. The process of the second stage of our approach becomes a multi-objective optimization problem. Ant Colony Optimization (ACO) is used extensively in many research areas because of its simple operations and efficiency [8]. In this paper, an improved Multi-Objective Ant Colony Optimization (MOACO) algorithm considering the timeliness of each goal node is proposed to generate the node sequence.

Traffic situations often change dynamically due to the complexity of real urban scenarios. Some research has focused on Dynamic TSP in which additional nodes may be added to the graph during a traversal [9-10]. Jeffrey Miller et al. presented the Intelligent Transportation Systems Traveling Salesman Problem (ITS-TSP) [11]. Therefore, in the second stage, we plan out the node sequence considering the node timeliness and changing situation.

The remainder of this paper is organized as follows. In Section 2, we describe path planning in the topological map created by OpenStreetMap for urban scenarios and dynamic planning between two nodes. In Section 3, we describe the multi-node multi-objective route planning considering node timeliness using an improved multi-objective ant colony optimization algorithm. Our experimental results and conclusions are presented in Section 4 and 5, respectively.

2. Dynamic Planning Between Two Nodes

The first stage of our approach is to perform path planning between two nodes. The aim is to determine an optimized accessible path and the path cost between two selected nodes. A topological map in urban scenarios that include real traffic information such as traffic lights and intersections is created using OpenStreetMap in order to search for the path that obeys the traffic rules.

2.1. Topological Map in Urban Scenario

Topological map uses waypoints to represent specific positions of the road and use the relationship of the waypoints to represent the attribute of the road. This kind of map has simple structure, easy storage, good coherence, high efficiency and robustness. A topological map must be created first to experiment path planning between two nodes in urban scenario.

OpenStreetMap (OSM) [12] is used to extract the urban topological structure for our experiments. The OSM is a well-known project that provides user-generated street maps. Data downloaded from the OSM website is in eXtensible Markup Language (XML) form, organized in a treelike structure. In the XML file structure, there are four elements under OSM root, including bound, node, way and relation. Attributes from a node provide latitude, longitude and the ID of the node. Some nodes may have sub-elements that provide additional information about the node. This information indicates whether it is an intersection or belongs to a building, bus stop, restaurant, etc. If a node does not have any sub-element, then it means that it belongs to a road or highway instead of an intersection. The attribute of way element indicates its identification and the sub-elements of way include its category, traffic rules and member nodes information.

 Copyright ⓒ 2015 SERSC

2.2. Dynamic Path Planning Between Two Nodes

Based on the topological map, we can plan out an accessible path from the starting node to the ending node. A* algorithm is a heuristic search method which is widely used for path search between two points on the map.

Due to the complex traffic environment and dynamic variable tasks, some new points between the original two nodes need to be visited. In this section, a bidirectional heuristic search algorithm is proposed to deal with this issue.

In the bidirectional heuristic search algorithm, the new point is considered as an initial node; meanwhile the route spreads from the initial node to the target node. Comparing with the A* search algorithm in which each state node has one evaluation function of the target state node, the bidirectional heuristic search algorithm associates each state node with two evaluation functions of the initial state node and the target state node. Equations (1) and (2) express their relationships.

$$f_{\text{start}}(s) = g(s) + h_{\text{start}}(s) \tag{1}$$

$$f_{\text{goal}}(s) = g(s) + h_{\text{goal}}(s) \tag{2}$$

wherein, $f_{\text{start}}(s)$ and $f_{\text{goal}}(s)$ represent the evaluation functions whose range is from the current node to the initial state node and to the target state node, respectively. $h_{\text{start}}(s)$ and $h_{\text{goal}}(s)$ are the heuristic functions whose range is from the current state node to the initial state node and to the target state node, respectively. And $g(s)$ means the actual length from the guidance node to the current state node.

The pseudo code of the bidirectional heuristic search algorithm is shown in Table 1. All state nodes are set to infinitive initially. There are only new points in the OPEN table whose $g(s_{need})$ is set 0. Values of new points $f_{start}(s_{need})$ and $f_{goal}(s_{need})$ (from line 1 to line 4) are calculated simultaneously.

Table 1. Pseudo Code Of The Bidirectional Heuristic Search Algorithm

1 **Initialize()**
2 for all $s \in S$, $g(s) \leftarrow \infty$;
3 $OPEN \leftarrow \{g(s_{need}) \leftarrow 0\}$;
4 $f_{start}(s_{need}) \leftarrow g(s_{need}) + h(s_{start})$;
5 $f_{goal}(s_{need}) \leftarrow g(s_{need}) + h(s_{goal})$;

6**Search(*s*)**
7 while(*s* is not expanded)
8 remove *s* with the smallest *f*(*s*) from *OPEN*;
9 for each success *s'* of *s*
10 if $g(s') > g(s) + c(s, s')$
11 $g(s') \leftarrow g(s) + c(s,s')$;
12 insert or update *s'* in *OPEN* with $f(s') \leftarrow g(s') + h(s')$;

13**main()**
14 Initialize()
15while($s_{start}$and s_{goal} are not expanded)
16 remove s with the smallest $f_{start}(s)$ or $f_{goal}(s)$ from *OPEN*;
17 if $g(s') > g(s) + c(s, s')$
18 $g(s') \leftarrow g(s) + c(s, s')$;
19 insert or update s' in *OPEN* with $f_{\text{start}}(s') \leftarrow g(s') + h(s'_{start})$ and $f_{goal}(s') \leftarrow g(s') + h(s'_{goal})$;
20 if(s_{start} is expanded)
21 save solution from s_{need} to s_{start};

Copyright ⓒ 2015 SERSC

Search(s_{goal});
else
save solution from s_{need} to $s_{goal;}$
Search(s_{start});
get the final solution with combining solution s_{goal} and solution $s_{start;}$

2.3. Output of the First Stage

2.3.1. Node Pairs Selecting: Considering that there are n nodes need to be visited, in order to get the accessible path between any node pairs and its path cost value, C_n^2 times path planning between two nodes is needed.

2.3.2. Planning Output: After the planning of all the selected node pairs, we get all the paths and path cost which are useful for the next stage planning. Matrix $Path$ saves all the paths for the node pairs. P_{mn} is the path points from node m to node n.

$$Path = \begin{bmatrix} P_{11} & P_{12} & \cdots & P_{1n} \\ P_{21} & P_{22} & \cdots & P_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ P_{n1} & P_{n2} & \cdots & P_{nn} \end{bmatrix} \tag{3}$$

Matrix Dis saves all the path cost of the node pairs. D_{mn} is the path cost from node m to n.

$$Dis = \begin{bmatrix} D_{11} & D_{12} & \cdots & D_{1n} \\ D_{21} & D_{22} & \cdots & D_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ D_{n1} & D_{n2} & \cdots & D_{nn} \end{bmatrix} \tag{4}$$

3. Multi-objective Route Planning with Node Timeliness

In the Section 2, we solved the feasible path between two nodes and obtained a series of waypoints to navigate the vehicles driving in the real urban scenarios while obeying the traffic rules. Also we got the path cost of the selected nodes. In this section, we propose an improved Multi-Objective Ant Colony Optimization (MOACO) algorithm to solve the node sequence planning problem with the node timeliness in urban scenarios.

3.1. Mathematic Model

The node timeliness is denoted by a weight value which refers to the urgency of the node to be visited. Larger weight value means the timeliness priority of the node is higher.

Traditional TSP can be described as follows: Given a list of cities (nodes) $\mathrm{N} = \{1, 2, \cdots, n\}$ and the distance between city i and city j $D(n_i, n_j)$. The task is to find an order in which the salesman visits each city once and returns to the origin city at minimum route cost.

In this paper, we extend the visiting task into a Multi-Objective TSP (MOTSP). In the solution process, we consider two optimization objectives proposed above. Like traditional TSP, the first optimization objective is to minimize the route cost. In our problem, the path cost between two stochastic nodes $D(n_i, n_j)$ which is the output of the first stage planning, is shown as D_{ij} in equation (4). Thus, the first optimization objective can be expressed as follows:

 Copyright ⓒ 2015 SERSC

$$D_R = D(n_n, n_1) + \sum_{i=1}^{n-1} D(n_i, n_{i+1}) \tag{5}$$

The second optimization objective is to meet the node timeliness requirement which means important nodes need to be visited earlier. The node timeliness priority are quantified by the node weight value. The timeliness cost function is expressed as follows:

$$T_j = \varepsilon_j \left(\sum_{i=1}^{j-1} D(n_i, n_{i+1}) \right) / v \tag{6}$$

where ε_j is the weight value of the node j. $D(n_i, n_{i+1})$ represents the path cost from node i to node $(i+1)$. v is the speed of the vehicle in urban scenarios. $\sum_{i=1}^{j-1} D(n_i, n_{i+1})$ represents the route cost before the vehicle visiting the node j.

In this way, we quantify the timeliness requirement of the visiting task. Thus, the second planning objective is to minimize the total timeliness cost. The objective function is expressed as follows:

$$T_R = \sum_{i=1}^{n} T_i \tag{7}$$

3.2. Planning Base on MOACO

ACO algorithm is used extensively in many regions for its simple operations, good parallelism, strong global searching capacity and high efficiency. Compared with other optimization methods, for example, genetic algorithm (GA), ACO algorithm is more efficient and has showed a better global optimal solution [13]. Thereby, ACO algorithm is chosen in this paper. Then, an improved Multi-Objective Ant Colony Optimization (MOACO) algorithm is proposed which is based on two optimization objective functions mentioned above.

Traditional ACO algorithm only takes the route cost as a pheromone factor when dealing with TSP, but for the nodes with timeliness search problem studied in this paper, we take timeliness cost into the procedure as another pheromone factor. Based on new pheromone factors, a MOACO algorithm is proposed to solve this problem.

In the MOACO algorithm, ants choose a node from the node set consist of nodes unsearched by the probabilities repeatedly until all nodes get searched. Probabilities of the unsearched nodes calculated in equation (8) are decided by the heuristic value and pheromone value corresponding to the node.

$$p_{ij}^m(t) = \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}(t)]^\beta}{\sum_{l \in N_i^m(t)} [\tau_{il}(t)]^\alpha [\eta_{il}(t)]^\beta} \quad j \in N_i^m(t) \tag{8}$$

where $p_{ij}^m(t)$ is the probability of ant m moves from node i to node j in the t times iteration process. Node j is from $N_i^m(t)$ which is the unsearched node set of ant m. $\eta_{ij}(t)$ is the heuristic value for ant moving from node i to node j, set as $1/\mathrm{D}(n_i, n_j)$, the inverse of the distance between the two nodes. $\tau_{ij}(t)$ is the pheromone value for ant moving from node i to node j, related to the whole solution. α and β are constants, which represent the importance of the two values each.

After one iteration process, each ant's traversal route is a solution for the nodes with timeliness search problem studied. For Ant-cycle Model adopted in the MOACO proposed, after all ants have traversed all nodes (an iterative process), pheromone value of each path needs to be updated. The ants with excellent solutions spray more pheromone on the path result in more ants choosing this path later which forms a positive feedback effect. The pheromone value updated through equation (9-11). In equation (9), $\Delta\tau_{ij}^m(t)$ is the pheromone value sprayed on the path from node i to node j by ant m after the t

Copyright ⓒ 2015 SERSC

times iteration process. D_m is the total path cost of the search route planned by ant m calculated in equation (5) and T_m is the total time cost calculated in equation (7). Q is a constant, represents quality of the pheromone value; γ is the weight coefficient used to adjust the two factors' (path cost and time cost) influence on the final solution.

$$\Delta\tau_{ij}^{m}(t) = (1-\gamma)\bullet\frac{Q}{L_m} + \gamma\bullet\frac{Q}{T_m} \quad (0 \le \gamma \le 1) \tag{9}$$

$$\Delta\tau_{ij}(t) = \sum_{m=1}^{n}\Delta\tau_{ij}^{m}(t) \tag{10}$$

$$\tau_{ij}(t+1) = (1-\rho)\bullet\tau_{ij}(t) + \Delta\tau_{ij}(t) \quad (0 < \rho \le 1) \tag{11}$$

The total pheromone increment on the path from node i to node j produced by all the n ants after the iteration process calculated by equation (10). Before the next iteration process, pheromone values for all paths on the route get updated by equation (11), wherein ρ is the pheromone evaporation parameter. Repeat this iteration and update process before getting the optimal solution.

4. Experiments

The dynamic multi-objective route planning system in urban scenarios proposed in this paper can be applied to solve some practical problems. Assume that a vehicle with some visitors has to conduct a mission to visit many nodes in the area of education in Beijing. These nodes include 8 universities, such as Beijing Institute of Technology (BIT), Peking University (PKU), Tsinghua University (THU), USTB, BUAA, CUFE, BNU, BJTU which are numbered from 1 to 8, shown in Table 2. The locations of these 8 nodes are illustrated in Figure1.

In this mission, the vehicle has to start from BIT and visit each node only once then returns to BIT. Moreover, due to the preference of some visitors, some universities such as PKU, THU and BUAA have higher timeliness priority, which means they need to be visited earlier than other universities. The weights of nodes are given in Table 2. For this practical problem, we employ the dynamic multi-node multi-objective route planning system to plan the route for the vehicle.

Table 2. Weight of All Nodes

Name	BIT	PKU	THU	USTB	BUAA	CUFE	BNU	BJTU
No.	1	2	3	4	5	6	7	8
Weight	1	5	4	1	3	1	1	1

 Copyright ⓒ 2015 SERSC

Figure 1. Location of 8 Nodes

4.1. Initial Planning

In the proposed planning system, an optimal route should be planned out to navigate the vehicle to meet the requirements of two objectives. Based on the node information of the task shown in Table 2, we can execute the two-stage planning process.

4.1.1. First Stage: The first stage is to plan out the path between any two nodes that form a node pair, and the number of the node pairs is 28(C_8^2) for the total 8 nodes in this case.

Next, feasible paths between the selected node pairs can be planned out by the method presented in Section 2. The results including the feasible paths between node pairs and the path costs are saved in the matrix $Path$ and Dis (shown in equation (3) and (4)). Element P_{mn} in $Path$ represents the feasible path from node m to node n. In the case of P_{26}, it means the feasible path from node 2 (PKU) to node 6 (CUFE). All feasible paths contain a series of consecutive way points. Element D_{mn} in Dis represents the path cost from node m to node n. For example, D_{12} represents the path cost of the feasible path from node 1 (BIT) to node 2 (PKU). Figure 2 shows the real path between BIT and PKU planned by the proposed method in Section 2.

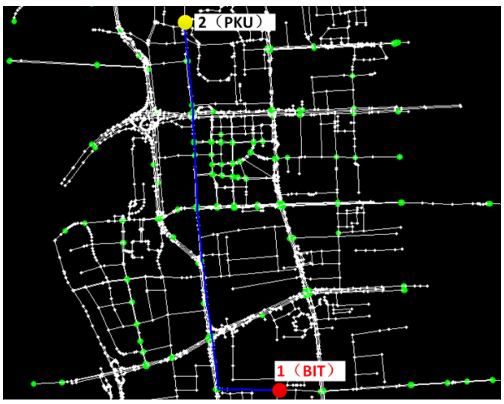

Figure 2. Real Path From BIT To PKU

4.1.2. Second Stage: With the results of the first stage planning, we carry out the second stage planning (global route planning) aiming at generating the visiting order of all goal nodes in the task. The improved MOCOA mentioned in Section 3 is used to solve this problem. In detail, the input values include the weight vector ε which is shown in Table 2 and the route cost matrix Dis. The output of this algorithm is a sequence which represents the visiting order of all goal nodes (universities). The planning result is a sequence of integers (2 3 5 4 7 8 6). Thus the visiting order is 1->2->3->5->4->7->8->6->1. Combined with the matrix $Path$ generated in the first stage, we can get the global driving route as follows,

$$R_0\ (P_{12}, P23, P35, P54, P47, P78, P86, P61).$$

Using this output, the vehicle can finish the mission while obeying the traffic rules and meeting the requirement the timeliness of nodes.

4.2. Dynamic Re-Planning

During the execution of the task, for example, when the vehicle is on the way to node 2 (PKU), the visiting task changes. Two new goal nodes of Beijing University of Posts and Telecommunications (BUPT) and Central University for Nationalities (CUN) are added from the locations of the goal nodes, we can see that BUPT is very close to node 7 (BNU), thus the method of dynamic planning between two nodes introduced in Section 2 can be used to cope with this situation. Treat BUPT as the guidance point while re-planning the path from node 4 to node 7, as shown in Figure 3. Thus the new path from node 4 to node 7 by way of the guidance point (BUPT) is obtained while the corresponding path cost increases slightly.

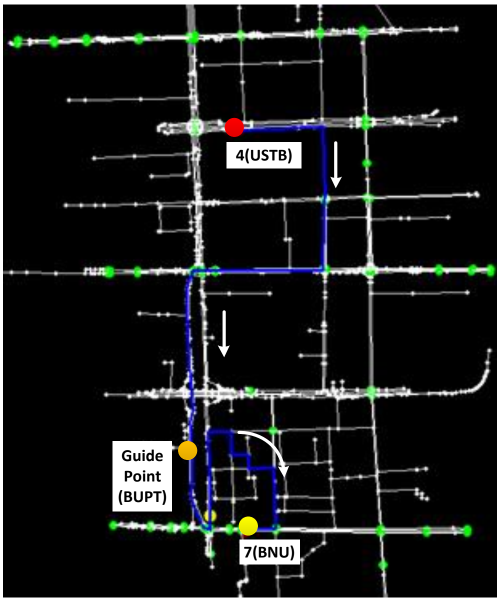

Figure 3. Adding a New Goal Node

Compared with BUPT, CUN (node 9) cannot be treated as a guidance point, because no node is close to it. In this case, we use the method of dynamic planning between two nodes mentioned in Section 2 to add a new goal node (CUN) and re-plan to get a new route. The new rest route is $R_1(P_{23},P_{35},P_{54},P_{47},P_{78},P_{86},P_{69},P_{91})$.

5. Conclusion

In this paper, we have proposed a dynamic multi-node multi-objective route planning system for a vehicle in urban scenarios. The whole system involves a two-stage process. In the first stage, the topological map in urban scenarios was firstly created using OSM

 Copyright ⓒ 2015 SERSC

which include real traffic information such as traffic lights and intersections, then the optimized accessible path was planned using a bidirectional heuristic search algorithm , and the path points and path cost between two selected nodes were gotten. After that, an improved multi-objective Ant Colony Optimization algorithm considering the timeliness of each goal node was proposed to generate the node sequence in the second stage. Moreover, considering complex traffic environment and dynamic variable tasks, the changing situation and corresponding planning schedule were discussed. Experimental results showed that the proposed approach can be used for multi-node visiting tasks with timeliness for a vehicle in urban scenarios.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (Grant No. 61304194 & 91120015).

References

[1] C. E. Turcu, V. G. Gaitan and C. O. Turcu, “An internet of things-based distributed intelligent system with self-optimization for controlling traffic-light intersections [C]”, Applied and Theoretical Electricity (ICATE), 2012 International Conference on, (**2012**) October 25-27; Craiova, Romania.

[2] W. Shi, W. Jian, S. G. Deng, L. Kuang, H. J. Chen, D. C. Shi, Z. Xu, Z. Y. Ye, Z. H. Wu, X. H. Qian, X. H. Chen, Q. M. Xu and F. H. Chen, “ITS Grid: A Novel Integrated Intelligent Transportation Information and Service Platform [C]”, Services Computing, 2006, SCC'06. IEEE International Conference on, (**2006**) September 18-22; Chicago, America.

[3] X. Yu, “A genetic algorithm for the Dubins Traveling Salesman Problem [C]”, Industrial Electronics (ISIE), 2012 IEEE International Symposium on, (**2012**) May 28-31; Hangzhou, China.

[4] P. S. Krishnan, J. K. S. Paw and T. S. Kiong, “Cognitive Map approach for mobility path optimization using multiple objectives genetic algorithm [C]”, Autonomous Robots and Agents, 2009, ICARA 2009, 4th International Conference on, (**2009**) February 10-12; Wellington, New Zealand.

[5] A. Haghani and S. Jung, “A dynamic vehicle routing problem with time-dependent travel times [J]”, Computers & operations research, vol. 32, no. 11, (**2005**).

[6] F. Altiparmak, M. Gen, Lin L and T. Paksoyc, “A genetic algorithm approach for multi-objective optimization of supply chain networks [J]”, Computers & Industrial Engineering, vol. 51, no. 1, (**2006**).

[7] A. Larsen, O. B. G. Madsen and M. M. Solomon, “The a priori dynamic traveling salesman problem with time windows [J]”, Transportation Science, vol. 38, no. 4, (**2004**).

[8] I. Alaya, C. Solnon and K. Ghedira, “Ant colony optimization for multi-objective optimization problems [C]”, Tools with Artificial Intelligence, 2007, ICTAI 2007, 19th IEEE International Conference on, (**2007**) October 29-31; Patras, Greece.

[9] X. W. Lu, A. C. Regan and S. Iraní, “The Dynamic Traveling Salesman Problem: An Examination of Alternative Heuristics [J]”, Transportation Science, (**2000**).

[10] A. Zhou, L. Kang and Z. Yan, “Solving dynamic TSP with evolutionary approach in real time [C]”, Evolutionary Computation, 2003, CEC '03, The 2003 Congress on, (**2003**) December 8-12; Canberra, Australia.

[11] J. Miller, S. Kim and T. Menard, “Intelligent Transportation Systems Traveling Salesman Problem (ITS-TSP)-a specialized tsp with dynamic edge weights and intermediate cities [C]”, Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on, (**2010**) September 19-22; Madeira Island, Portugal.

[12] M. Haklay and P. Weber, “Openstreetmap: User-generated street maps [J]’< Pervasive Computing, IEEE, vol. 7, no. 4, (**2008**).

[13] Z. J. Lee, S. F. Su, C. C. Chuang and K. H. Liu, Genetic algorithm with ant colony optimization (GA-ACO) for multiple sequence alignment [J]. Applied Soft Computing, vol. 8, no. 1, (**2008**).

Copyright ⓒ 2015 SERSC

 Copyright ⓒ 2015 SERSC