Comparison of Three Search Algorithms for Mobile Trip Planner for Eskisehir City

Ahmet Aydin, Sedat Telceken
Department of Computer Engineering
Anadolu University
Eskisehir, Turkey
{ahmetaydin, stelceken}@anadolu.edu.tr

*Abstract***— The comparison of three searching algorithms; A*, Ant Colony Optimization and Genetic Algorithms to solve the Traveler Salesman Problem for a mobile trip planning application for Eskisehir City, Turkey, is presented in this paper. The algorithms work on more than 30 point-of-interests and 150 sub-point-of-interests. The algorithms are compared with respect to their running times for scenarios with different number of point-of-interests. Experimental results show that the A* algorithm is 400-600% faster than the other algorithms. The mobile application calculates the best route trip planned according to the traveler's preferences on categorized points-of-interests. The mobile application also recommends alternative route plans during the trip when the traveler is ahead or behind the schedule.**

Keywords—A*; Ant Colony Optimization; Genetic Algorithms; Traveler Salesman Problem; Mobile Application; Trip Planning; Route Planning

I. Introduction

The travelling salesman problem (TSP) is an NP-hard problem in theoretical computer science and this problem has mostly been studied in Artificial Intelligence (AI) methods. The time complexity of TSP is $O(n!)$ when exhaustive search is used and $O(n^2 2^2)$ when dynamic programming is used. Both of the search methods have combinatorial explosion.

In the last decades, mobile devices has become very popular and this popularity has made these devices indispensable for human life. Moreover, mobile devices present lots of considerable solutions which make daily life easier. Trip planning and routing in driving or walking is one of these solutions which can be run on the mobile phones people always carry.

People generally expect from mobile trip applications to be dynamic, user-friendly, responsive, and well-designed. For a trip planning application, routes must be calculated quickly and places must be recommended according to the traveler's preferences and the ratings of other travelers. A dynamic application that recommends alternative routes or places worth-visiting during a trip, or recommends to by-pass some previously planned places that can be ignored can be appealing and useful for travelers.

In the literature, route planning algorithms have been used to compute the optimum route to collect municipal solid waste combined with geographic information systems [1], and to plan the transportation routes for hazardous materials [2]. Another study was proposed to plan time-dependent routes for Milano City which is a real world problem [3].

This paper presents a mobile application which makes dynamic recommendations using the A* search method, the Ant Colony Optimization (ACO) method, and the Genetic Algorithm (GA) method. These search methods are tested and compared on different trip scenarios that have different cases. The effect of number of point-of-interests, recommending alternative sub point-of-interests and alternative route planning on the performance of the algorithms are analyzed.

The route-planning problem and its details have been described in Section II. The search methods and the constraints are briefly explained in Section III. Experimental results are presented in Section IV and the results are discussed and future works are mentioned in Section V.

II. Problem Definition

A. Problem Definition

Detailed touristic information and fine route recommendations for travelers can easily be accessed on the Internet for popular touristic cities, like Madrid, New York, Tokyo, etc. [4]. Mobile applications also exist in app stores, especially, the *tripadvisor* mobile applications specified for popular cities. However, less popular places, like Eskisehir in Turkey, don't have such detailed guides. Recently, Eskisehir has become one of the favorite touristic cities for native tourists but touristic information is not well-organized and is not published and processed digitally. In order to fill this gap, a mobile application has been developed in this study which can be customized for individual user experiences, and the application is specialized for Eskisehir city.

B. Data to be used in the application

The data to be processed in the application has been collected from Eskisehir Provincial Directorate of Culture&Tourism and Eskisehir Metropolitan Municipality. The data contain information about worth-visiting places, high resolution photographs, short videos, GPS coordinates, and average time duration required to visit a place [5, 6]. The ideal average visiting times and ratings for each place have been estimated initially by the authors. More than 30 point-of-interests (POIs),

This study was supported by Anadolu University Scientific Research Projects Commission under the grant no: 1402F047

978-1-4673-7751-5/15/$31.00 ©2015 IEEE

like museums, parks, etc., and more than 150 sub-POIs, that are the places like cafeterias or monuments located inside POIs, has been written into a database and they have been grouped under the categories like Architecture, Art, Entertainment, Museum, Nature, Popular, etc.

III. Methods and Constraints

The mobile application's life cycle can be explained with the following scenario:

A traveler comes to Eskisehir for one day or more. The purpose of the visit can be for business or touristic. Anyway, the traveler has a free time to tour the city without a guide and wants to see the most important historical or natural places in the city during his or her free time. The traveler needs a route recommendation optimized for the time period, and user ratings of the places of his/her preferred category.

At this point of the scenario, the traveler enters his/her free time and preferred categories into the mobile application and the application recommends at least one route according to the specified constraints by taking into consideration of open-hours of places, ideal visiting hours, average visit duration, and transportation times. The application should find and sort the places as fast as possible.

The traveler starts to follow one of the routes recommended by the application and the application starts tracking him/her and continues to make suggestions like adding a new POI to the route or removing a POI that becomes ignorable when time passes. A POI becomes ignorable when visiting a prior place longer than the recommended duration and the POI is less important according to the posterior places. In other words, recommendation can be dynamically updated according to the traveler being ahead or behind of the schedule. The application also suggests some coffee houses, souvenir shops, cousins, etc. at any time of the trip. This artificial intelligence suggestions depend on trip time or current location. The application shows routes and displays information about places like a tourist guide. One of the strong aspects of the application is that it supports the traveler by recommending sub-routes for optimum visiting time containing sub-POIs. The traveler can rate places and write comments them after the tour.

A. Nomenclature

A directed graph $G = (P, A)$ representing the route, with POIs set P and set of arcs A, is given. Two criteria are defined on the arcs: travel time and travel type. Each POI includes visiting time (v), visit duration (d) and count of POIs (c). This identifications are used in a triplet (v, d, c). For each arc (i, j), $ttime_{ij}^{v}$ indicates the travel time and $ttype_{ij}^{c}$ the travel type. These arc values can change during the time, we consider a discretization of the day in 5 time slots. Therefore each criterion can be considered constant on each time slot. On the graph G, a set $\Psi \subset P \times P \times P$ of route is also defined, where each (p, q, r) in Ψ denotes that the arc (q, r) cannot be traveled after arc (p, q).

A traveler (u) who uses mobile application identified by the triplet (v, d, c), given his current GPS position o' at the instant time τ and its route ϱ. The search problem in the application has been solved by the methods A*, ACO, and GA, whose details are explained below.

B. A* Algorithm

A* searching algorithm is developed in 1968 by Hart, Nilsson and Raphael [7]. This algorithm tries to find optimum route from start point to target point if there is route which is usually used for network routing, finding shortest path for games etc. [8-10]. A* search algorithm chooses the next point by current cost from starting point and heuristic distance to target point.

$$f(n) = g(n) + h(n) \tag{1}$$

$f(n)$ denotes heuristic function, $g(n)$ denotes lowest cost value from starting point to n^{th} point, $h(n)$ denotes the estimated of cost of cheapest(distance) from the n^{th} point to a target point. According to calculated $f(n)$ values, points are stored in an ordered stack. The algorithm calculates the route by selecting points from stack by order and checks the route if it is optimum. It stops when the next point in the stack is end point. A* search is both complete and optimal

C. Ant Colony Optimization

ACO was presented as a PhD Thesis by Marco Dorigo in 1992 [11]. ACO is an optimized search technique that was proven to work reliably. ACO search concept requires an idea of how ants function in nature. Ants wander randomly in nature. If an ant is to discover food it begins emitting pheromones and starts to wander back to its nest. Likely other ants discover food and back to the nest. That behavior happens many times in a loop. These pheromone are volatile. Thus, it is more intense on shorter paths. Other ants prefers the road with more pheromones but not always. Some ants prefers new roads to seek shorter paths. By time the shortest paths will be preferred by ants and total shortest path reveals [12].
An arbitrary value is chosen in [0, 1] initially. A random probabilistic value is assigned for the preference of ant when it reaches to first crossroad. This probabilistic value compared with the initial value. After that step path is decided by using the pheromone track. That decision calculation for other u points for i^{th} point and for k^{th} ant is formulized below.

$$j = \max_{u \epsilon J_k(i)} \{[\tau(i,u)]^{\alpha} \times [\eta(i,u)]^{\beta}\} \qquad if\ q \le q_0 \tag{2}$$

$$P_k(i,j) = \begin{cases} \frac{[\tau(i,j)]^{\propto} \times [\eta(i,j)]^{\beta}}{\sum_{u \in j_k(i)} [\tau(i,u)]^{\propto} \times [\eta(i,u)]^{\beta}} & if\ j \in j_k(i) \\ 0 & otherwise \end{cases} \tag{3}$$

$\tau(i, u)$ denotes the pheromene track, $\tau(i, u)$ denotes distance from i to u point in (2,3). Probability of the i and j points in (3) is updated by using pheromone amount in (2). Finally the appropriate path among travel points is determined by using ACO algorithm which is given in [13,14].

D. Genetic Algorithm

Another popular optimum search algorithm is genetic algorithm that was proposed in 1986 [15]. First step of algorithm is allocating a sufficient number of bits to each search dimension to obtain the desired path. Consider a trip path representation which has some list of POIs: (POI$_1$ POI$_2$ POI$_3$ POI$_4$ POI$_5$). The first problem is that order is not unique, because of each trip path has $N!$ order. This problem can be solved by fixing the initial POI likely current location of the traveler. Another problem is that crossover operator does not generally give child node which are acceptable trip path. Heuristic crossovers are used to solve that problem. The heuristic crossover operator can be applied to a linear function of the length of the POIs data structure. That solution incorporate that, if D is the number of trip plan and N is the number of POIs, GA's for the TSP run with asymptotic complexity $O(DN)$.

IV. Experimental results

Each of the search algorithms compared in three different scenarios related two cases which are detailed below:

A. Case1: Start-up Scenario

A regular mobile application calculates a route by traveler's preferences and finishes its job. But practically, traveler couldn't be able to follow the trip plan strictly. Extra ordinary cases like: visiting durations, transportation problems or veering of the route may happen during the trip. In that scenario with traveler chooses 5 POIs which is given in Fig. 1.

Fig. 1. Start-up Scenario

In the middle of the trip (i.e. between POI$_2$ and POI$_3$), the application could recommend a café or a restaurant to traveler. Several places can be recommended by mobile application. Traveler chooses a local cuisine called "Papagan Restaurant". The new trip route then calculated like in Fig. 2.

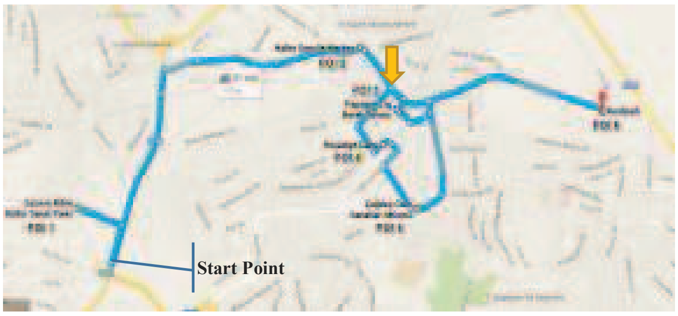

Fig. 2. Scenario after chosen recommendation

After that calculation new POI$_3$ is inserted into the trip plan. Total visiting time will not be enough for the first schedule at that time. Then the mobile application checks POIs from POI$_4$ to POI$_6$ for transportation and visiting times if total trip time is enough after removing one or more POI. Then it suggests several alternative schedules and user chooses one. The new route is calculated and figured in Fig. 3. This recommendation trip mode gives the traveler re-plan his/her route dynamically.

Fig. 3. Recalculated route after delaying

B. Case2: Sub-POIs planning

Another strong aspects of the mobile application is planning a sub-route for a complex POIs. These complex POI may be a big park which has several interesting places. These places are denoted as sub-POIs. Calculation of optimum visiting route for that POIs will help the traveler to visit all interesting places. For example, POI$_1$ (Sazova Park) has 5 sub-places (Tale Castle, Pirate Ship, Aquarium, Noah's Ark, Science Center) to visit. Traveler must be assisted for planning these sub-POIs to save visiting time. While making the plan under the Case1, traveler chooses the POI$_1$, but when s/he visits, sub-POIs need to be introduced to traveler. Then informed traveler can make sub-plan as distinct from the main trip plan.

C. Results of Different Scenarios

Scenario 1) Traveler plans a trip with first 5 POIs. Between POI$_2$ and POI$_3$, the mobile application offers a recommended place POI$_{Rec1}$. If traveler accept this recommendation, the application produces two alternative trip plans for rest of the trip. Alternative 1: POI$_1$, POI$_2$, POI$_{Rec1}$, POI$_5$, Alternative 2: POI$_1$, POI$_2$, POI$_{Rec1}$, POI$_3$, POI$_4$.

Scenario 2) Traveler chooses a trip plan with 8 POIs: POI$_2$, POI$_6$, POI$_3$, POI$_{10}$, POI$_7$, POI$_8$, POI$_5$, POI$_9$. Between POI$_3$ and POI$_{10}$, the mobile application offers POI$_{Rec2}$.Between POI$_8$ and POI$_5$, POI$_{Rec3}$ is offered. Alternative 3: POI$_2$, POI$_6$, POI$_3$, POI$_{Rec2}$, POI$_7$, POI$_8$, POI$_5$, POI$_9$ and Alternative 4: POI$_2$, POI$_6$, POI$_3$, POI$_{10}$, POI$_7$, POI$_8$, POI$_{Rec3}$.

Scenario 3) Traveler chooses a trip plan with 10 POIs which is ordered: POI$_1$, POI$_2$, POI$_6$, POI$_3$, POI$_{10}$, POI$_7$, POI$_4$, POI$_8$, POI$_5$, POI$_9$. If the mobile application offers POI$_{Rec4}$ between POI$_6$ and POI$_3$, and POI$_{Rec5}$ between POI$_8$ and POI$_5$. According to traveler's choices, two alternative plan is produced. Alternative 5: POI$_1$, POI$_2$, POI$_6$, POI$_{Rec4}$, POI$_3$, POI$_4$, POI$_8$, POI$_5$, POI$_9$, Alternative 6: POI$_1$, POI$_2$, POI$_6$, POI$_3$, POI$_{10}$, POI$_7$, POI$_4$, POI$_8$, POI$_{Rec5}$, POI$_5$.

TABLE I. RESULTS OF SCENARIOS WITH ALTERNATIVE PLAN

Scenario No / POI_{start}	Recommendation	POI_{Rec}	Total Travel Time (min.)	Total Dist. (Km.)	Total Visit Duration	Alg. Running Time (ms.)		
						A*	ACO	GA
1/5	No	5	26′ 12″	7.140	6h 30m	10	45	48
1/5	Yes (Alt. 1)	4	31′ 45″	6.525	5h 50m	10	42	42
1/5	Yes (Alt. 2)	5	23′ 02″	6.996	6h	11	44	44
2/8	No	8	37′ 12″	7.744	8h 50m	13	44	66
2/8	Yes (Alt. 3)	8	36′ 01″	6.412	8h 40m	12	44	67
2/8	Yes (Alt. 4)	7	35′ 25″	7.500	8h 20m	12	43	63
3/10	No	10	47′ 54″	10.234	11h 20m	14	49	78
3/10	Yes (Alt. 5)	9	48′ 31″	9.421	10h 10m	14	47	74
3/10	Yes (Alt. 6)	10	50′ 23″	10.030	10h 30m	13	45	78

In Table I, experimental results according to different scenarios have been presented. *Total Travel Time* is calculated by $\sum_{\substack{1 \le i \le POI_{n-1} \\ 2<j<POI_n}} ttime_{i,j}^{v}$ summation. If there is no recommendation normal trip plan will overlap startup first trip plan for traveler. POI_{Rec} shows the POI numbers are changed after recommendations in two different alternative plans which are explained briefly above. For these alternative plans, new trip routes are recalculated and *Total Travel Time*, *Total Distance*, *Total Visit Duration* are affected. Comparison of three methods reveals that A* algorithm runs (400-600)% faster than other two methods according to number of POIs and type of scenario.

Specifically, running time ratio is increased between A* and ACO when the number of POI is increased. Same situation is admissible for comparison of A* and GA.

GA is faster than ACO for the number of POI: 5 and 8. When POI number reaches 10 and more, ACO is faster than GA.

A* algorithm provides less time consuming and reasonable accuracy and these are essential for mobile application.

In this study, recommendation based mobile trip planner application which assists travelers and would be travelers in finding routes that fit their personal preferences is proposed. Android operating system and Google API are user-friendly and that helps travelers to choose categorized places. Mobile application calculates best route in specified time interval for traveler after that preference. A*, ACO and GA are chosen for computation of the best trip routes. All of the methods are examined for three different scenarios divided into normal trip route selection, recommendation based sub-planning. Their test results presented that A* method for less than or equal to 10 POIs gives the best results rather than ACO and GA in terms of running time. Tracking the traveler during the trip and giving some suggestions to him/her increase the total trip experience and frustrate the overflow of trip plan.

Sub-POIs planning for each methods minimize the visiting time for complexity POIs. This feature also help the traveler not to skip interesting popular places inside of current POIs.

Finally, all of three searching methods are complete and optimal. But A* search method calculates the trip route faster than Ant Colony Optimization and Genetic Algorithm Methods for the experimental data set which is limited 30 POIs.

REFERENCES

[1] MZ. Siddiqui, and JW. Everett, “Landfill siting using geographic information systems: a demonstration,” Journal of environmental engineering, 122(6), 515-523, 1996.

[2] A. Lue, and A. Colorni, “A multicriteria spatial decision support system for hazardous material transport,” R. Bisdorff, R. Dias, L. Meyer, P. Mousseau, 1000.

[3] M. Bruglieri, A. Colorni, F. Lia, and A. Lue, “A multi-objective time dependent route planner: areal world application to Milano city,” in 17th Meeting of the EURO Working Group on Transportation, EWGT2014, Sevilla-Spain, 2-4 July 2014.

[4] www.tripomatic.com, Last visited: 05/14/2015.

[5] www.eskisehirkulturturizm.gov.tr/, Last visited: 05/14/2015.

[6] www.eskisehir-bld.gov.tr/ebb.php, Last visited: 05/14/2015.

[7] PE. Hart, NJ: Nilsson, and B. Raphael, “A formal basis for the heuristic determination of minimum cost paths,” Systems Science and Cybernetics, IEEE Transactions on 4.2, pp. 100-107, 1968.

[8] X. Cui, and H. Shi, “A*-based pathfinding in modern computer games,” International Journal of Computer Science and Network Security, 11(1), 125-130, 2011.

[9] JY. Wang, and YB. Lin, “Game AI: Simulating Car Racing Game by Applying Pathfinding Algorithms,” International Journal of Machine Learning and Computing, 2(1).

[10] S. Russel, and P. Norvig, “Artificial Intelligence: A Modern Approach,” 2nd Ed., Prentice Hall., 97-104.

[11] M. Dorigo, “Optimization, Learning and Natural Algorithms,” PhD thesis, Politecnico di Milano, Italy, 1992.

[12] http://en.wikibooks.org/wiki/Robotics/Print_version, Last visited: 05/17/2015.

[13] A. Aydin, and S. Telceken, “Mobil Gezgin Uygulaması için A* Algoritması ve Karınca Kolonisi Optimizasyonu Karşılaştırılması,” 23th

Conference of Signal Processing and Applications (SIU2015), 16-19 May 2015, Malatya, Turkey.

[14] A. Aydin, and S. Telceken, “Artificial Intelligence Aided Recommendation Based Mobile Trip Planner for Eskisehir City,” IEEE Conference on Industrial Electronics and Applications, 15-17 June 2015, Auckland, New Zealand.

[15] J. Grefenstette, R. Gopal, B. Rosmaita, and D. Van Gucht, "Genetic algorithms for the traveling salesman problem," Proceedings of the first International Conference on Genetic Algorithms and their Applications. Lawrence Erlbaum, New Jersey (160-168), 1985.