Proceedings of the 6th International Conference on Computing and Informatics, ICOCI 2017
25-27April, 2017 Kuala Lumpur. Universiti Utara Malaysia (http://www.uum.edu.my)

Paper No. 155

How to cite this paper:

Mohammad Fariduddin Jalaluddin, Ezzatul Akmal Kamaru-Zaman, Shuzlina Abdul-Rahman, & Sofianita Mutalib. (2017). Courier delivery services visualisor (CDSV) with an integration of genetic algorithm and A* engine in Zulikha, J. & N. H. Zakaria (Eds.), Proceedings of the 6th International Conference of Computing & Informatics (pp 126-131). Sintok: School of Computing.

COURIER DELIVERY SERVICES VISUALISOR (CDSV) WITH AN INTEGRATION OF GENETIC ALGORITHM AND A* ENGINE

Mohammad Fariduddin Jalaluddin[1], Ezzatul Akmal Kamaru-Zaman[1], and Shuzlina Abdul-Rahman[1], Sofianita Mutalib[1]

[1]Faculty of Computer and Mathematical Sciences,
Universiti Teknologi MARA. Shah Alam, Malaysia,
mohammadfariduddinjalaluddin@gmail.com.my
{ezzatul,shuzlina,sofi}@tmsk.uitm.edu.my

ABSTRACT. Online shopping has become one of the popular mediums for people to use online transactions due to its economical and easiness. It is more convenient to those who simply do not have time to shop physically and prefer delivery service. However, the courier services nowadays are unable to keep up with the increasing consumer demand. The problem is caused by the delivery process that is not synchronized due to the problem of finding the best route of distribution. Distributors are unable to plan their distribution path with the minimal distance. Furthermore distributors are only able to reach each district distribution centre once a day and revisit the distribution centre will increase the time spent and operation cost. This study developed Courier Delivery Services Visualisor (CDSV) that is able to visualize the best route to be taken by distributor, so that the courier service can arrive on time. CDSV employed Genetic Algorithm (GA) and A-star Algorithm (A*) that integrates with Geographical Information System (GIS) data. A graphical user interface in the form of simulation map that suggests the best route and the optimal distance are displayed for easier courier service distribution references.

Keywords: A-star algorithm, courier services, distribution system, genetic algorithm, geographical information system (GIS), openstreetmap

INTRODUCTION

Delivery services play an important role for cooperate and societies. Every day there are inward and outward goods occur in online shop. The consumers of these services have been increased annually. Since the 20th century, the online shopping has become the medium for people to do their online transactions. This causes the postal and parcel delivery services hardly to make it on time due to the increase of consumer demand. In Malaysia, the delivering process is not synchronized because the distributor has problems in identifying the best route from state distribution center to district distribution center. The distributors could have limited information to plan their journey in distributing postal and parcels with minimal distance. Furthermore distributor only able to reach each district distribution center once per day and this has caused difficulties to come back from the starting point with the minimum distance.

Hence, this study developed a Courier Delivery Services Visualisor (CDSV) that is able to visualize the best route to be taken by distributor with the total distance so that the courier service can arrive on time. In most studies of Geographical Information System (GIS), Djikstra algorithm has been used (Goel, 2017). However, the drawback of this technique is more time is required in producing the final result. In order to optimize the time taken in path finding, we introduced the integration of Genetic Algorithm (GA) with A* algorithm. Solving travelling salesman problem (TSP) using GA only considers a straight line between each city meanwhile a real map usually does not have straight route. Hence, A* algorithm was incorporated with GA to tackle the real map problem in finding the best route in the shortest time and making use the Geographical Information System (GIS) data.

LITERATURE REVIEW

Traveling Salesman Problem (TSP) is one of the problems that had been studied by many researchers from various domains. This problem is a classic algorithm problem and NP-Complete in the field of computer science. NP refers to non-deterministic polynomial time and NP-Complete means when problem is in both NP and NP-Hard. In NP-complete, there is no fast solution to solve the problem. This is common due to the complexity of multiple variables or cities available in finding the best solution and normally researcher only able to get optimal solution. In past studies, TSP can be solved by many kind techniques such as using polynomial time algorithm (Çela et al., 2012), neighbourhood search technique (Mladenović et al., 2012), ant colony optimization (Aziz, 2015) and dynamic programming techniques (Roberti & Wen, 2016). The difficulty of this problem is to find the best solution in short time. The researcher needs to get best combination of sequences of cities and to find the best combination that they need to explore all possible alternatives. This method will consume a lot of time, depending on the number of cities they need to visit. However, heuristics can overcome the time consumption problem. Heuristics apply some rules to select the next city such as GA.

GA is a style of problem solving by mimicking a set of genes in deoxyribonucleic acid (DNA) representation in computer. It is also known as genetic programming. GA process does not explore all possible solutions. The main purpose of GA is to find the optimal solution and to get optimization strategies successfully. GA can solve different kind of problem such as scheduling, optimization, and other related domain problem in engineering problem. According to Chen et al. (2015), GA involved five steps which is coding, generate initial population, selection, mutation and crossover. In the previous study (Chen et al., 2015), GA is able to solve supermarket shopping route problem successfully and it also can find optimal solution in one iteration (Hardi, 2015) due to near-optimal solution set. According to Sridhar and Balasubramaniam (2011), GIS integrated with DNA Computing or can be defined as GA can solve NP-Complete problem in TSP. Hence, this shows that using GA is a relevant technique to be incorporated.

Apart from GA, A* is a searching algorithm that mostly used to find shorter path or route between two points. A* is no breadth first search, nor is it deep first search. Most of robotic and game applications apply the A* searching to find the shortest path. A* is an extension of Dijkstra's algorithm. According to Jia et al. (2015), A* algorithm improves significant performance in most maps. Eq. (1) show the A-star algorithm formula and further explanation of every notation can be found in the architecture design section.

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

METHODOLOGY

There are three core steps involved in order to develop CDSV: the design of variables representation and operational processes for each algorithm, the development of the codes and the evaluation to the paths found by the algorithm. .

Architecture Design

In this phase, three general activities were performed, namely the design of chromosome representation, fitness function and genetic operator (selection, crossover and mutation) to be used in GA engine, while A* was integrated by an external library. Figure 1 shows CDSV architecture by using six locations.

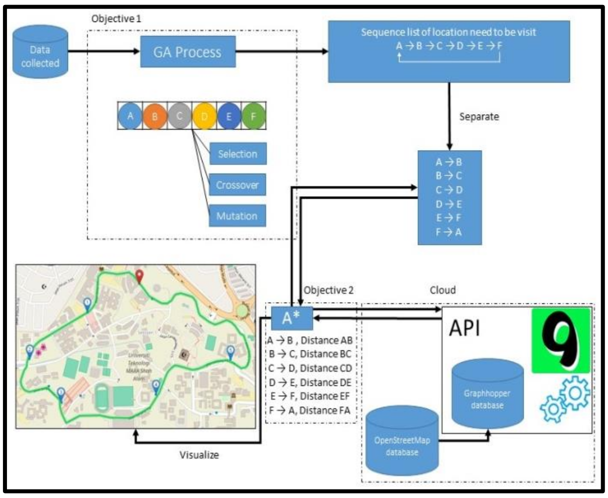

Figure 1. CDSV Architecture

The location data is represented as city name, *c* and geographical position, *g*. An array list was initialized at the beginning of the process and each of the objects that represent location/city was added later in the array list. In this situation, we can assume that objects act as genes and array list act as the chromosome as shown in Figure 2. This chromosome simulates the six locations.

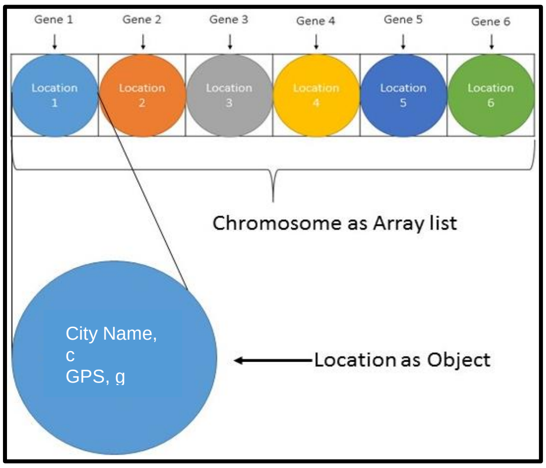

Figure 2. Chromosome Representation

Next, in order to determine the best set of siblings or children, the fitness function was identified and this fitness value plays an important role to solve the optimization problem. A bad design of fitness function may lead to failure of the system in producing the desired result. Hence, the best chromosome would be selected for the next iteration and this step will conserve the optimal solution to be reached. Based on our studies, many researchers used nearest neighbour which originated pythagorean theorem in geometry to find the distance between two points. Eq. (2) shows a pythagorean theorem formula, Eq. (3) shows the nearest neighbour formula between two points while Eq. (4) shows the fitness function based on three points.

$$c^2 = a^2 + b^2 \tag{2}$$

$$d_{[1,2]} = \sqrt{(a_2 - a_1)^2 + (b_2 - b_1)^2} \tag{3}$$

$$f(x) = d_{[i,j]} + d_{[i,j]} + d_{[i,j]} \tag{4}$$

$$f^{-1}(x) = \frac{1}{f(x)} \tag{5}$$

Where,

a_2, a_1 is latitude of point one and two,

b_2, b_1 is longitude of point one and two,

d, is distance between two point, and

$f(x)$ is fitness function.

This study used minimize fitness function, *f(x)* in equation but it will appear nicely if the optimal graph is concave down, hence we inversed the equation to become maximize problem as shown in Eq. (5). The last activity is to determine genetic operators for selection, crossover and mutation. In order to generate the optimal solution, stopping condition is defined by using average fitness value and if the fitness value is similar for three consecutive iterations in the same generation, we concluded that, there is no more improvement found.

The GA engine and A* engine were developed using JAVA with GraphHopper library. In order to collect the information on cities, Openstreetmap API was used. CDSV helps in finding the best route by GA engine and optimizing the routes to be taken by A* calculator. User may first input the location to be used for the GA and the sequences of location given by GA as result. The sequences of location were divided to represent the sources and destinations of each tour.

Each tour to be taken was optimized using A* Engine. A* uses its heuristic function that add up *g(n)* and *h(n)* to get *f(n)*. *g(n)* is defined as measures of actual length from any state to start state meanwhile *h(n)* is the estimated distance from a state to goal. Hence, *f(n)* is calculated to get the cheapest solution of the route taken based on *h(n)* and *g(n)*. Once CDSV has run its course, an output is generated to the user containing information of *g(n)* defined as distance of each tour, *f(n)*, total distance taken to complete tour and sequence of road to be taken; visualized in a map.

EXPERIMENTS AND RESULTS

In this experiment, we benchmarked De Jong and Spears (1989) guideline which is to start GA with a moderately sized population, relatively low probability of mutation (0.02, 0.04, 0.06 and 0.08), and a relatively high probability of crossover (0.6, 0.7, 0.8 and 0.9). The implemented termination criteria of GA iterations is either 250 as maximum generation or same fitness value in three consecutive iterations in the same generation, when 13 locations were used. Every combination of parameter was iterated 10 times and the maximum value of score was recorded as best result and bolded.

Table 1 is the experiment being done on GA and we found a variety of result. The best result is based on the optimal score **($f^{-1}(x)$)** that acts as the fitness value with score 0.518. These results were obtained from 60 population size with optimal generation of 41 populations. From the evaluation, we also found out that the best parameter configuration were probability of crossover with 0.9, probability mutation with 0.08. This parameter values resulted into the highest optimal score for every experiment. Hence, with this combination, we stored this parameter values in the GA engine to produce the optimal sequence of locations to be visited.

Table 1. Result for experiment on GA with different parameters.

Crossover				Mutation				Population Size			
Probability	No. of generation	Opt. generation	Optimal score ($f^{1}(x)$)	Probability	No. of generation	Opt. generation	Optimal score ($f^{1}(x)$)	Size	Number of maximum	Opt. generation	Optimal score ($f^{1}(x)$)
0.6	37	29	0.462	0.02	25	18	0.478	20	34	25	0.484
0.7	14	7	0.482	0.04	28	15	0.498	40	56	30	0.518
0.8	42	32	0.467	0.06	45	27	0.498	**60**	**250**	**41**	**0.518**
0.9	**29**	**11**	**0.489**	**0.08**	**67**	**43**	**0.509**	80	250	70	0.518

Next, the result obtained from the GA engine which is sequence of cities to be visited are used in A* API. By integrating this API, *f(n)* is then calculated in the API. CDSV would show the map of sequence of best route to travel and also the total distance cost, *f(n)* with value of 300.468 as shown in Figure 5.

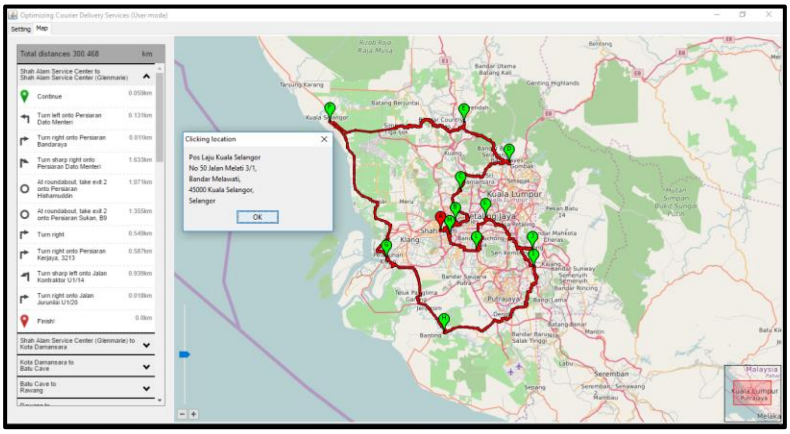

Figure 5. Map of Optimized Route using A* API.

CONCLUSION

The main purpose of CDSV is assist the distributor in courier services -in identifying and optimizing the best route in delivering the parcels and postal. Integration of GA algorithm and A* algorithm in CDSV can visualize the best route to facilitate the distributor in delivering the courier. CDSV also can be used as an evaluation model by other researchers in order to solve distribution problem. Future enhancement may be implemented by using multi-objective optimization to identify the best route using GA, including avoiding the traffic congestion, road speed limit and real time speed and later to develop mobile based application to make it easier for user to use.

REFERENCES

Aksoy, D., Altinel, M., Bose, R., Cetintemel, U., Franklin, M., Wang, J., & Zdonik, S. (1999). Research in data broadcast and dissemination *Advanced Multimedia Content Processing* (pp. 194-207): Springer.

Aziz, Z. A. (2015). Ant Colony Hyper-heuristics for Travelling Salesman Problem. *Procedia Computer Science, 76*, 534-538.

Çela, E., Deineko, V., & Woeginger, G. J. (2012). The x-and-y-axes travelling salesman problem. *European Journal of Operational Research, 223*(2), 333-345.

Chen, X., Li, Y., & Hu, T. (2015). *Solving the supermarket shopping route planning problem based on genetic algorithm.* Paper presented at the Computer and Information Science (ICIS), 2015 IEEE/ACIS 14th International Conference on.

De Jong, K. A., & Spears, W. M. (1989). *Using Genetic Algorithms to Solve NP-Complete Problems.* Paper presented at the ICGA.

Hardi, R. (2015). *Genetic algorithm in solving the TSP on these mineral water.* Paper presented at the Intelligent Technology and Its Applications (ISITIA), 2015 International Seminar on.

Jia, J., Pan, J.-S., Xu, H.-R., Wang, C., & Meng, Z.-Y. (2015). An Improved JPS Algorithm in Symmetric Graph. Paper presented at the *2015 Third International Conference on Robot, Vision and Signal Processing (RVSP)*.

Goel, N., Velaga, N. R., Vedagiri, P., & Mathew, T. V. (2017). Optimal routing algorithm for large scale multi-modal transit networks.

Mladenović, N., Urošević, D., & Ilić, A. (2012). A general variable neighborhood search for the one-commodity pickup-and-delivery travelling salesman problem. *European Journal of Operational Research, 220*(1), 270-285.

Nor Aziran, Y. (2016, March 28). Peniaga online daftar SSM meningkat. *Utusan Online.* Retrieved October 13, 2016, from http://www.utusan.com.my/berita/nasional/peniaga-online-daftar-ssm-meningkat-1.205568

Roberti, R., & Wen, M. (2016). The Electric Traveling Salesman Problem with Time Windows. *Transportation Research Part E: Logistics and Transportation Review, 89*, 32-52.

Sridhar, R., & Balasubramaniam, S. (2011). GIS integrated DNA computing for solving travelling salesman problem. *Paper presented at the Computers & Informatics (ISCI), 2011* IEEE Symposium on.

United States Postal Service. (2016). *Just one day in the life of the U.S. Postal Service*... by the numbers. Retrieved October 3, 2016, from https://about.usps.com/who-we-are/postal-facts/one-day-by-the-numbers.htm