STATIC-DYNAMIC ROUTE OPTIMIZATION USING GENETIC ALGORITHM

Jocelyn L. Garrido*'[1], Paul Joseph M. Estrera[1], Ma. Esther B.Chio[1], Rhea Suzette M. Haguisan [1]
[1]Department of Information Technology, University of Science and Technology of Southern Philippines, Cagayan de Oro City, Philippines
*For Correspondence; Tel. +639656510866, Email: *jocelyn.garrido@ustp.edu.ph*

ABSTRACT: *Route optimization is prevalent in providing cost-effective, time-saving, and fuel-efficient logistic operations. In the case of a laundry shop using a Laundry Management System (LMS) which provides pickup and drop-off services to their increasing clients, the absence of route planning for drop-off and pickup led to frequent vehicle traversal increasing the operational cost of logistics. This was addressed by developing a web and mobile application that provides a static drop-off and dynamic pickup route plan using a Genetic Algorithm (GA) which was integrated into the existing LMS. Genetic Algorithm (GA) has been known as a technique useful for finding the optimal or near-optimal solutions for combinatorial optimization problems. Optimization was evident by generating a lesser number of vehicles with a shorter distance traveled for each vehicle while taking into consideration the vehicle payload. This can maximize productivity and improve work efficiency while saving operational costs. Implementation of a dynamic route plan for pickup during vehicle enroute within EDT also contributed to the efficiency to handle the increasing demand of the client. While route optimization in this study considers only distance travel, vehicle capacity, and time window, we recommend further research using artificial intelligence considering traffic situations, rerouting due to road maintenance, and other road-related situations that can affect the route plan enroute.*

Keywords: Route Optimization, Genetic Algorithm, Vehicle Routing Problem (VRP), Dynamic Route Plan

1. INTRODUCTION

Route optimization increases business efficiency for companies where logistics plays a vital role in business operations such as Dirt-bag. Dirt-bag is a laundry business operating in Cagayan de Oro City. Due to the Covid-19 pandemic which limits human mobility, the company took advantage of the technology creating a web-based laundry management system that packages an offers laundry pick-up and drop-off service to clients within the city. However, the increase in clients also demands cost-effective logistic planning to balance out customer satisfaction and business productivity. The absence of proper route planning for both laundry drop-off and pick-up on an increasing number of clients resulted in multiple vehicle traversals and eventually demand higher operational costs on logistics. This paper, therefore, presents a solution for solving the drop-off and pick-up problem (DPP) by implementing a route optimization module to manage laundry logistics.

DPP is similar to Vehicle Routing Problem with Pick-up and Delivery (VRP-PD) designed to find the optimized route considering minimum travel cost for a vehicle that departs from a laundry hub, visiting all clients for laundry drop-off only once and pick-up client's laundry while on transit back to the hub [1-3]. Travel cost includes fuel cost, distance, and travel time. However, this study extends optimization to both route and vehicle capacity during drop-off and pick-up within time-constraint to improve work efficiency and productivity and leverage customer services.

The system considered two aspects in designing an optimized route plan. A static route plan for drop-off and a dynamic route plan for pick-up.

For laundry drop-off, where clients usually have a prior drop-off schedule, a static route plan is considered. The drop-off route plan will guide the vehicle towards each drop-off client with estimated drop-off time (EDT) travel starting from the hub departure until the last mile client. Route optimization for drop-off is based on the client's distance from the hub such that the nearest client is the first drop-off stop and the furthest client is the last mile drop-off stop. The drop-off route plan can also be used as a mechanism for sorting laundry loads inside the delivery, vehicle to reduce the time of delivery and improve work efficiency.

For laundry pick-up, a dynamic route plan is considered but is constrained by EDT and vehicle capacity. The route plan is dynamic in nature such that when a new pick-up request is received while the vehicle is in its EDT range and within vehicle capacity, the system continually updates the pick-up route plan. The Pick-up route is then used by the vehicle during its transit back to the hub.

With this, the system considers a hybrid static and dynamic optimized routing solution in solving VRP-PD to provide cost-effective, time-saving, and fuel-efficient logistic operations. Moreover, based on literal review [4][5][6][7][8] the system will implement the Genetic Algorithm (GA) as a solution to route optimization.

The route optimization system is then integrated into the existing system of Dirtbag to handle their laundry logistics operations in a more efficient and effective way by reducing logistic operational costs. This system will also open wider opportunities to handle increasing client demands without compromising the services offered by Dirtbag.

The main objective of this study is to develop an optimized route plan using GA for drop-off and pick-up routes and implement it using mobile and web applications which will be integrated into the existing Laundry Management System of Dirtbag.

System Framework

Figure 1 shows the conceptual framework for the integration of the existing Laundry Management System (LMS) of Dirt bags with the Route optimization system. The necessary information needed (client information, pick-up/drop-off information, etc.) to build a route plan will be based on the existing LMS. The Route Optimization System will then provide an optimized route plan using a route optimization algorithm. This uses a genetic algorithm to provide both Drop-off and Pickup route plans. The route plan will then be reflected on the assigned driver's mobile application and tagged to the assigned cluster vehicle as a guide during laundry drop-off and pick.

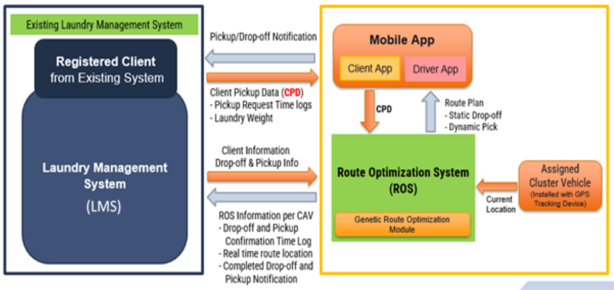

Figure 1: Integration of Route Optimization System Conceptual Framework

2. METHODOLOGY

Phase I: System Design and Development

A. Data Gathering and Requirement Analysis

Since all routes will be taken from the existing LMS, integration of the route optimization system played a vital role in data gathering. The following tasks were carried out to further analyze the integration of both systems. (1) Study the database design of the LMS to verify the data store and identify missing data that is crucial in the route optimization design. (2) Familiarized with the LMS functionality and behavior such as knowing the existing process, and the relationship among other processes to ensure smooth integration of the route optimization system. And (3) Identify LMS policies and constraints – keep in place system restrictions and adopt system policies during the route optimization design.

B. System Design

Figure 2 reflects the movement of the data and the relationship of the entities involved in the system. Route optimization in this system identified four external entities namely the Client, Laundry Management System, Route Optimization Module, and Cluster Vehicle. The figure shows data needs (arrow in) and data provision (arrow out) from each entity as it goes through the system.

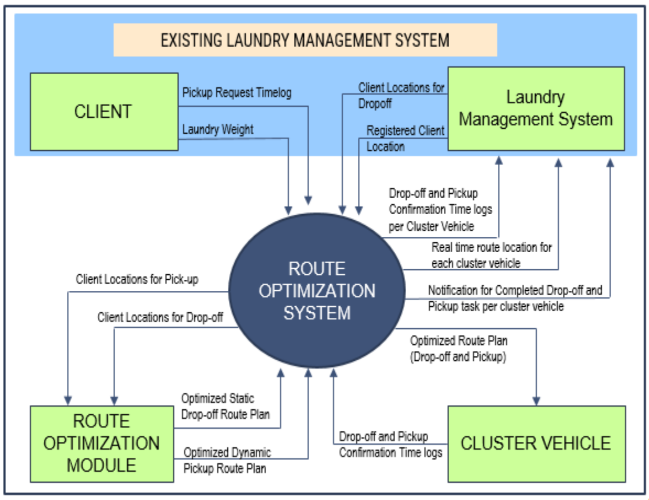

Figure 2: Context Diagram

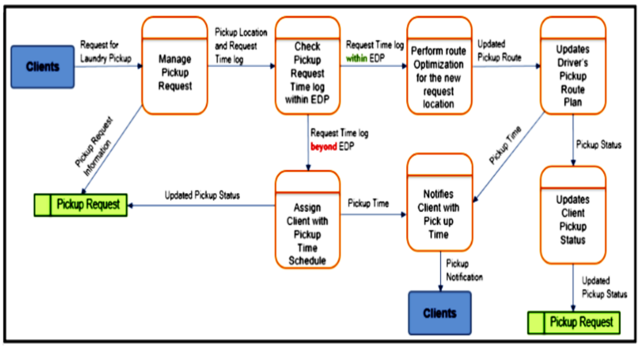

Figure 3: Pickup Request Dataflow Diagram

C. Data flow Diagram

Figure 3 illustrates the detailed data flow of a client request for laundry pickup reflecting the data inputs and data outputs as well as the data store and the process itself to perform the tasks indicated.

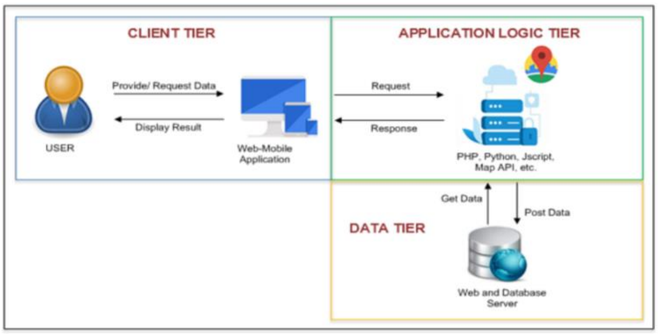

Figure 4: System Architecture Design

D. System Architecture

Figure 4 shows the system's architectural design. The client tier specifies the application that was used by the system stakeholders namely Dirtbag and its Clients. Dirtbag will use both a web application for route management and a mobile application to view routes for assigned drivers. Clients will use mobile applications for laundry appointments. The application tier reflects the technology that was used to implement the route optimization system. It integrates available technology applications such as cloud-based APIs for tracking location, MAP APIs to create route plans, Genetic Algorithms using Python to provide optimized route plans, and other APIs needed for data request and processing. A data management server was used to store all system data.

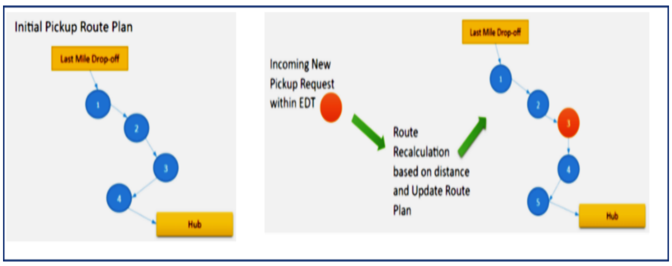

Figure 5: Dynamic Pickup Route Process

Phase II: Route Optimization

The data used in this study came from Dirtbag Laundry Company in Cagayan de Oro City and location and distance data were measured by using the Google Maps tool.

A. Defining Constraints: Routes plans are created on a one-directional route wherein a straight route for drop-off clients (from hub to last mile drop-off) is created. A separate straight route for pickup clients is also created (backhaul) wherein the route will start at the last mile drop-off client back to the hub. Furthermore, for dynamic pickup route implementation (Figure 5), it is a constraint within the Estimated Drop-off Time (EDT). Calculation of EDT is from the start time based on vehicle

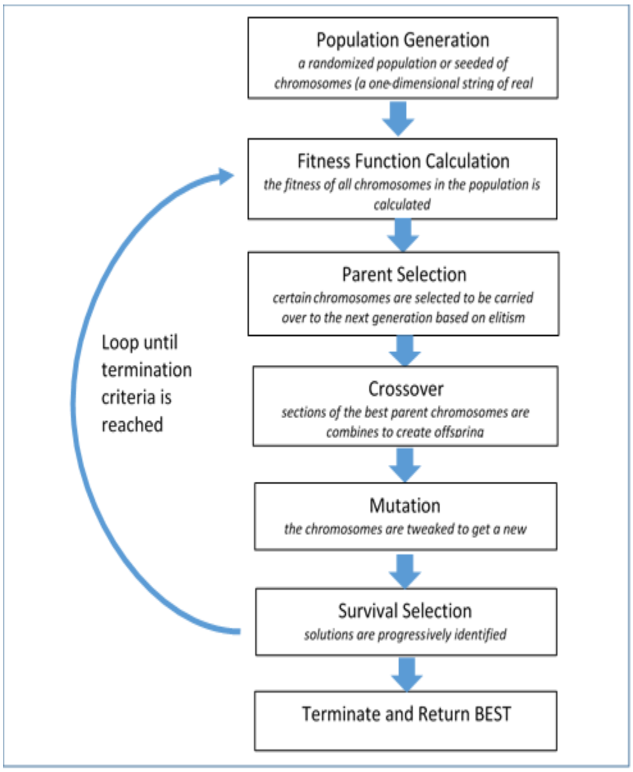

Figure 6: GA Structure

departure from the hub and end time based on the estimated calculation time travel from departure time to the last mile drop-off location. The route plan also presumes that all Drop-off goods come from Hub and all Pickup goods are taken back to Hub.

B. Optimizing Route using Genetic Algorithm

Genetic Algorithm (GA) is a search-based optimization technique based on the principles of Genetics and Natural Selection[9, 10]. This technique is useful for finding the optimal or near-optimal solutions for combinatorial optimization problems that traditional methods fail to solve efficiently [11-13]. This algorithm reflects the process of natural selection where the fittest individuals are selected for reproduction in order to produce offspring of the next generation [14]. Each solution is represented by a chromosome, which consists of a sequence of genes that represent a solution to the problem. GAs operate upon a population of solutions, which are manipulated over several iterations, which are called generations. Better solutions are progressively identified, as the algorithm pairs parent chromosomes to produce offspring, or applies random mutations on previously generated chromosomes. The structure of GA (Figure 6) starts with an initial population (which may be generated at random or seeded by other heuristics), and selects parents from this population for mating. Apply crossover and mutation operators on the parents to generate new off-springs. And finally, these off-springs replace the existing individuals in the population, and the process repeats. In this way, genetic algorithms actually try to mimic human evolution to some extent.

(1) *Initialization:* First we import the pickup or drop-off dataset from the LMS that contains the client locations (location coordinates using Google Map API), the distance of each location from the hub and the estimated time travel for each client location. Using the dataset we generate the distance matrix between all clients and the hub. Then *vehicle_payload* was defined to ensure that the vehicle can carry within its capacity. Using the Distributed Evolutionary Algorithms in Python (DEAP) library, we "create" the chromosomes *'Individual'* and the fitness objective, set as a minimization by weights=(-1.0,).

(2) *Chromosome Presentation*. The *'individuals'* of a population in the GA can be seen as an ordered list of artificial chromosomes where every chromosome represents a route a vehicle is going to take [9]. Each chromosome will consist of two lists: The first list consists of the name of the clients to visit, and indicates the order at which each client is visited. The second list consists of the vehicle that visits the corresponding client of the first index. For example, the following chromosome [1,9,6,8,4,5,2,3,7,10], [3,0,3,3,1,1,0,3,1,0] corresponds to the following route:

R1: [2 4 9 10]

R2: [4 5 7]

R3: [ ]

R4: [3 1 6 8]

R1 is served by vehicel1 that visits the ordered list from left, starting with client2, to the right ending at client 10. A vehicle that isn't needed in the solution has an empty list.

(3) *Fitness Calculation*. After the generation of the value chromosome, we then calculate the fitness value. The fitness value is used to rank each individual in the population. This calculation shows the ability of individuals to survive and continue the next process [15]. The fitness value can be derived using the equation:

$$fitness = (1/ (1 + tt)) + (tp * (-1)), \quad (3)$$

where it is the total travel time, and tp is the total number of penalties/violations. The penalty is calculated from the sum of waiting time and the overtime to drop products based on the time window.

(4) *Parent Selection*. There are a few options for how to select the parents that will be used to create the next generation. In this study, we consider the use of elitism. With elitism, the best-performing individuals (based on the calculated fitness value) from the population will automatically carry over to the next generation, ensuring that the most successful individuals persist [16].

(5) *Crossover*. The crossover process is one method of reproduction to produce new individuals referred to as offspring [17]. The crossover process is vital in generating new chromosomes by combing two or more parent chromosomes with the hope that they create new and efficient chromosomes. Crossover occurs after the selection of pairs of parent chromosomes and helps in exchanging information between parents to create children. During crossover, the parent chromosomes are taken in pairs and their genes are swapped in

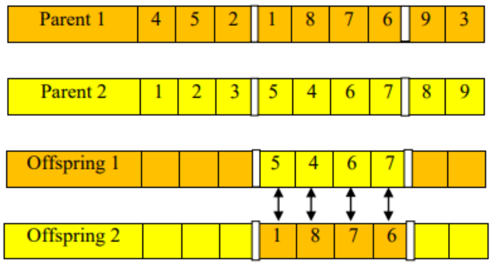

Figure 7a: Preliminary Stage of PMX

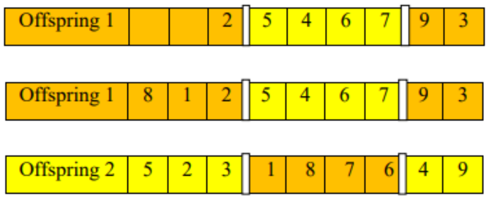

Figure 7b: Final Stage of PMX

a certain order to obtain offspring. These offspring become next-generation parent chromosomes. It is performed by swapping alleles between two selected parent chromosomes in order to explore new solution space [18]. Here we used a partially mapped crossover (PMX) technique (Figures 7a and 7b by Varun Kumar S G and Dr. R. Panneerselvam) to generate offspring. In this method, two crossover points are chosen among two parents and the genes between two crossover points are exchanged while remaining genes are filled by partial mapping [19].

(6) *Mutation*. The mutation process in the genetic algorithm is optional, depending on the problem resolved. In this study, we add some randomization to the results so we don't lose genetic diversity. It works of selecting 2 genes randomly to exchange each selected gene. The parent used in the mutation process is only one parent and is randomly selected.

(7) *Selection*. The idea of the selection phase is to select the fittest individuals and let them pass their genes to the next generation. Two pairs of individuals (parents) are selected based on their fitness scores. Individuals with high fitness have more chances to be selected for reproduction. Since a vehicle can only carry a maximum payload defined in *vehicle_payload*, we need to check that all solutions are feasible.

(8) *Termination*. GA terminates if the population has converged (does not produce offspring which are significantly different from the previous generation). Then it is said that the genetic algorithm has provided a set of solutions to our problem.

Phase III: Implementation and Testing

The GA was run in Python using DAEP with 20 locations within the city, 4 vehicles and each vehicle has a *vehicle_payload* of 8. Testing of population size of 200 with *cr*=0.4 and *mr*=0.6 [20] where *cr* is the probability of crossover and *mr* is the probability of mutation.

5. RESULTS AND DISCUSSIONS

Fitness value was tested with 1000 generations. The result is shown in Figure 8. The performance of GA is graphically presented in the convergence plot (Figure 9).

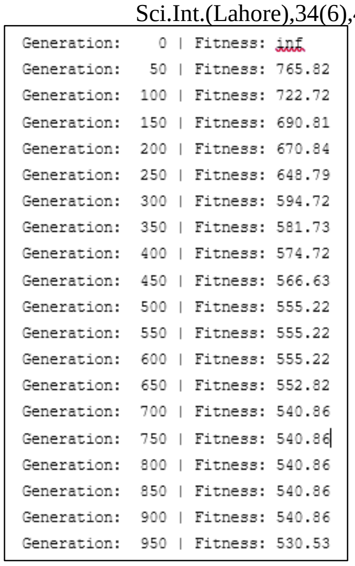
```
Generation:    0 | Fitness: inf
Generation:   50 | Fitness: 765.82
Generation:  100 | Fitness: 722.72
Generation:  150 | Fitness: 690.81
Generation:  200 | Fitness: 670.84
Generation:  250 | Fitness: 648.79
Generation:  300 | Fitness: 594.72
Generation:  350 | Fitness: 581.73
Generation:  400 | Fitness: 574.72
Generation:  450 | Fitness: 566.63
Generation:  500 | Fitness: 555.22
Generation:  550 | Fitness: 555.22
Generation:  600 | Fitness: 555.22
Generation:  650 | Fitness: 552.82
Generation:  700 | Fitness: 540.86
Generation:  750 | Fitness: 540.86
Generation:  800 | Fitness: 540.86
Generation:  850 | Fitness: 540.86
Generation:  900 | Fitness: 540.86
Generation:  950 | Fitness: 530.53
```

Figure 8: Fitness in each chromosome

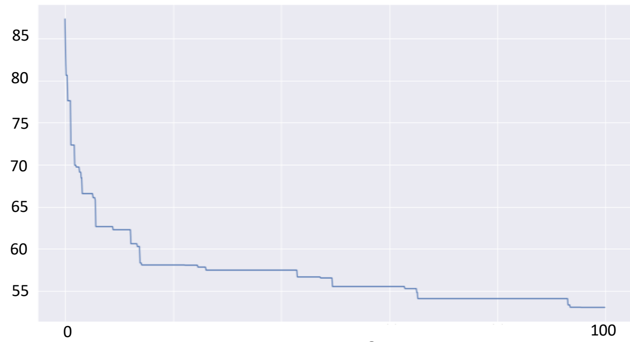

Figure 9: GA Performance

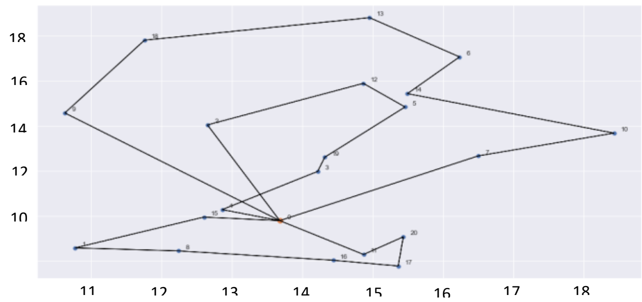

Figure 10: Plotted Routes produced by GA

The final routes are produced and plotted (Figure 10).

```
[[], [4, 3, 19, 5, 12, 2], [15, 1, 8, 16, 17, 20, 11],
[7, 10, 14, 6, 13, 18, 9]].
```

The empty set [] represents the vehicle that will not be needed based on the solution.

GA was used to calculate both the dynamic pickup route and static drop-off route. The GA is then used as a route optimization module during the development of the web and mobile application for the pickup and drop-off delivery system which was integrated into the Laundry System as shown in Figures 11, 12, and 13. Route clustering was manually done by clicking the points in the map to indicate boundaries/clusters (Figure 11).

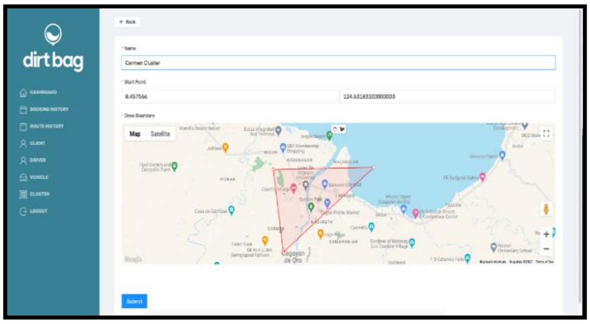

Figure 11: Route Clustering

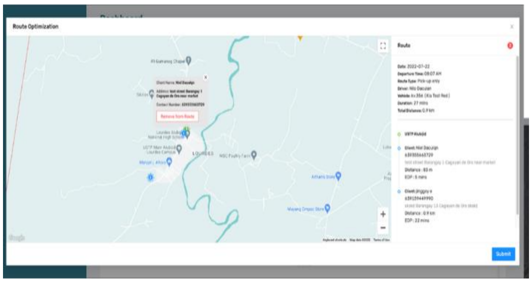

Figure 12: Delete route note during

When the vehicle is enroute, a dynamic route is initiated. The dynamic route includes deleting and adding a node in the route. Deleting usually happens when the client informs cancellation or rescheduling of a pickup appointment. Adding nodes happens when a new pickup request is detected within

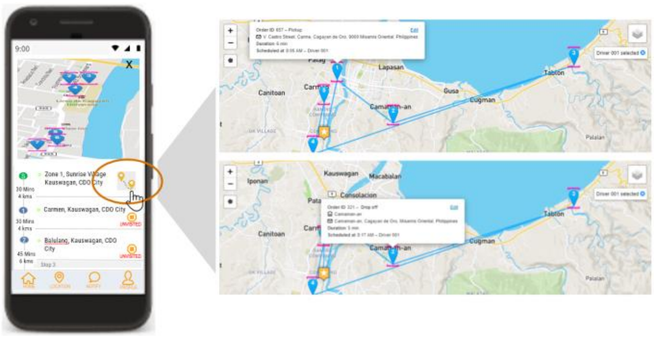

Figure 13: Driver's Mobile App Route Plan

EDT period (Figure 12).

Route plan is also reflected in the mobile application used by vehicle drivers as a guide route (Figure 13). The location of the driver can also be monitored in the web application of the admin.

6. CONCLUSION AND RECOMMENDATION

The development of an optimized route plan for pickup and drop-off deliveries using GA was able to provide an efficient and favorable solution to minimize travel costs while accommodating increasing clients. Notably given 4 vehicles to carry the route, GA was able to curtail it to 3 vehicles which can have a great impact on the operational cost of the company. Dynamic route plan was also successfully implemented and was evident when a real-time update on the web and mobile route plan is reflected when adding or deleting a new node (new pickup request) on the route during vehicle en route within the EDT period. New pick-up request that is made beyond the EDT period is queued on the next pick-up schedule. Since this study only considers route and vehicle capacity within time constraints, we suggest that further study using artificial intelligence will be considered to address traffic situations, rerouting due to road repairs, and other road-related situations that can alter the route plan en route.

Acknowledgment

This research project is funded by the Department of Science and Technology, Region 10 (DOST-R10) and Northern Mindanao Consortium for Industry, Energy and Emerging Technology Research and Development (NORMINCIEERD), Philippines.

REFERENCES

[1] C.Y. Liu, J.J. Yu, (2012). "Multiple depots vehicle routing based on the ant colony with the genetic algorithm". Journal of Industrial Engineering and Management, 6 (4) , pp. 1013-1026.

[2] C.C. Lu, V.F. Yu, (2013). "Data envelopment analysis for evaluating the efficiency of genetic algorithms on solving the vehicle routing problem with soft time windows". Computers & Industrial Engineering, 63 , pp. 520-529.

[3] Y.Zhang X.D.Chen. (2014). "An Optimization Model for the Vehicle Routing Problem in Multi-product Frozen Food Delivery" .Journal of Applied Research and Technology Volume 12, Issue 2, April 2014, Pages 239-250.

[4] Gunawan,Stacia ; Susyanto, Nanang ; Bahar, Syafri, (2019). "Vehicle routing problem with pick-up and deliveries using genetic algorithm in express delivery services". AIP Conference Proceedings 2192, 060009.

[5] Hommadi, Abbas, (2018). "Multi-Stop Routing Optimization: A Genetic Algorithm Approach" (2018). All Graduate Theses and Dissertations. 7048.

[6] Chew, Joanne Suk Chun, Lee, Lai Soon, (2012). " A Genetic Algorithm for Urban Transit Routing Problem". International Conference Mathematical and Computational Biology. Vol. 9 (2012) 411–421.

[7] Kurnia, H., Wahyuni, EG, Pembrani, EC, Gardini ST, Aditya, SK. (2018). " Vehicle Routing Problem Using Genetic Algorithm with Multi Compartment on Vegetable Distribution. IOP Conf. Series: Materials Science and Engineering 325 (2018) 012012.

[8] Olowadare, S., Oguntuyi, I., Nwaiwu, J. (2018). "Solving School Bus Routing Problem using Genetic Algorithm-based Model". I.J. Intelligent Systems and Applications, 2018, 3, 50-58.

[9] Masun, AKM et.al.(2011). "Solving the Vehicle Routing Problem using Genetic Algorithm". International Journal of Advanced Computer Science and Applications · August 2011

[10] Sbai, I and S. Krichen,(2020). "An Adaptive Genetic Algorithm for Dynamic Vehicle Routing Problem with Backhaul and Two-dimensional loading constraints," 2020 International Multi-Conference on: "Organization of Knowledge and Advanced Technologies" (OCTA), 2020, pp. 1-7, doi: 10.1109/OCTA49274.2020.9151691.

[11] Park,Hyungbin;Son,Dongmin; Koo,Bonwoo and Jeong,Bongju, (2021). "Waiting strategy for the vehicle routing problem with simultaneous pickup and delivery using genetic algorithm". Expert Systems with Applications Volume 165, 1 March 2021, 113959.

[12] Ifwah, Nurfahizul, M.Shaiful1, M.Z Shamsunarnie1, Z.M Zainuddin2 ,M.Fuad2, "Genetic Algorithm for Vehicle Routing Problem with Backhauls". Journal of Science and Technology, undated.

[13] Gorunescu, F., Belciug, S.,(2019). “Genetic Algorithm”. Encyclopedia of Biomedical Engineering.

[14] Mallawaarachchi, V., (2017). “Introduction to Genetic Algorithms — Including Example Code”. Towards Data Science.

[15] Malhotra, R., Singh, N., and Y. Singh, (2011). “Genetic Algorithms : Concepts , Design for Optimization of Process Controllers,” Computer. Inf. Sci., vol. 4, no. 2, p. 39–54.

[16] Stoltz, E. (2018). “Evolution of a salesman: A complete genetic algorithm tutorial for Python”. Towards Data Science.

[17] Alfiyatin, AN., Mahmudy, WF., Anggodo YP. (2018). “K-Means Clustering and Genetic Algorithm to Solve Vehicle Routing Problem with Time Windows Problem”. Indonesian Journal of Electrical Engineering and Computer Science. Vol. 11, No. 2, August 2018, pp. 462~468.

[18] Kumar, VSG., Panneerselvam, R. (2017). “A Study of Crossover Operators for Genetic Algorithms to Solve VRP and its Variants and New Sinusoidal Motion Crossover Operator”. International Journal of Computational Intelligence Research. ISSN 0973-1873 Volume 13, Number 7 (2017), pp. 1717-1733.