Genetic Algorithm-Based Fair Order Assignment Optimization of Food Delivery Platform

Min-Yan Tsai[1], Guo-Yu Lin[2], Jiang-Yi Zeng[2], Chia-Mu Yu[1], Chi-Yuan Chen[2], and Hsin-Hung Cho[2(✉)]

[1] Department of Information Management and Finance, National Yang Ming Chiao, Tung University, Hsinchu, Taiwan
chiamuyu@nycu.edu.tw

[2] Department of Computer Science and Information Engineering, National Ilan University, Yilan, Taiwan
{n1043027,chency,hhcho}@niu.edu.tw, r1043002@ems.niu.edu.tw

Abstract. Most existing food delivery platforms lack responsibility when it comes to route planning. This often results in uneven assignment of orders or difficulty in arranging orders for delivery drivers. These issues have led to loss of consumer rights and reduced revenue for delivery platforms, as well as negative feedback and evaluations. To address this problem, it is necessary to first resolve the issue of uneven distribution of orders. In this paper, we propose using the Genetic Algorithm (GA) to solve the order assignment optimization problem. By utilizing GA's strong global search ability, we can achieve fair assignment of orders, optimize delivery routes, and balance revenue distribution. This approach creates a fair competition environment for delivery drivers and improves service quality, ultimately leading to positive feedback from consumers and creating a win-win situation.

Keywords: Artificial intelligence · genetic algorithm · traveling salesperson · delivery route planning

1 Introduction

In the era of the COVID-19 pandemic, people have increasingly been cutting back on eating out and staying at home, which has led to a rise in the use of delivery platforms for daily necessities. According to the Market Intelligence & Consulting Institute (MIC) in Taiwan, an investigation found that 72% of consumers have experience using delivery services in 2021. Among the most popular platforms, foodpanda has 78% and UberEats has 61% market share. However, the sudden influx of orders during peak hours can often lead to a misestimation of the delivery staff's ability to take and deliver orders, resulting in uneven distribution of orders and causing delays or disputes between consumers and couriers. This can ultimately harm the platform's reputation.

Y. Chen et al. (Eds.): BICT 2023, LNICST 512, pp. 122–132, 2023.
https://doi.org/10.1007/978-3-031-43135-7_12

This paper will discuss the use of metaheuristic algorithms for solving multi-objective optimization problems in delivery operations [1]. The approximate optimal solution can be effectively found by considering multiple factors such as the number of delivery personnel, the number of orders, service quality, delivery revenue, and delivery distance. The goal is to balance the rights of the delivery platform, delivery staff and consumers as much as possible. The algorithm will help assign suitable orders to delivery staff, ensuring timely delivery and reducing complaints from consumers. The multi-objective path optimization problem is an NP-hard problem [2, 3], which makes it difficult to solve using traditional algorithms. Therefore, this study proposes an architecture based on Genetic Algorithm (GA) to find a solution to this problem.

This paper is divided into five sections. Section 2 provides background information and related work on the Genetic Algorithm (GA) and Traveling Salesman Problem (TSP). Section 3 uses linear programming to define the delivery order problem. The simulation results of our proposed method are presented in Sect. 4, and the conclusion is given in the last section.

2 Related Works

2.1 Genetic Algorithm (GA)

The metaheuristic algorithm can give a combinatorial optimal solution at affordable cost through the objective observation and principle. The so call cost can be temporal, spatial or anything which can be defined [4]. Furthermore, the metaheuristic algorithm refers to the use of past experience or rules to define a random search methods in a population or individual. Compared with greedy-based algorithm, although the solution cannot be guaranteed that is definitely better, the concept of the conditional random process can avoid the local optimum and find the approximate optimal solution. Moreover, in a very hard problem, the cost will be less than the greedy approach. Therefore, these methods are widely used in much more applications which has the demand of optimization. The common and classical algorithm are Hill Climbing, (HC) [5], Simulated Annealing, (SA) [6] Ant Colony Optimization, (ACO) [7] and GA [8], etc.

GA was proposed in 1975 by John Henry Holland. It is according to the Nature selects, the fittest survives theory to choose the solutions. This mechanism is to reserve the fitness value better's solutions to weed out the poor fitness values solutions. GA major focus on the optimal solution search with combinatorial objectives. It has since been used in many fields such as scheduling optimization [9], data mining [10], finical prediction [11], process management [12], etc. The basic GA is the simulation of the gene evolutionary process the process including Selection, Crossover, and Mutation. The solution of the random combination is an individual that is generally represented as a set of sequences and named a chromosome. In the initialization of GA, the initial population is created that including several chromosomes. Then GA will choose two chromosomes to operate crossover and mutation. It can be getting new solutions and comparing which solutions need to eliminate. In general, the eliminating solutions have poor fitness values. The process of selection, crossover, and mutation are continuously iterated until the termination condition is reached.

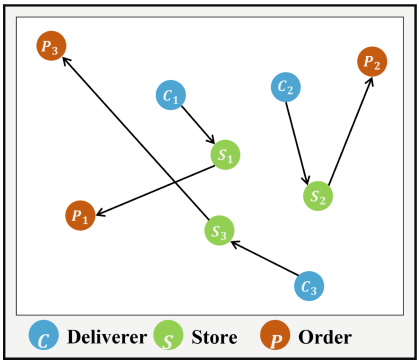

Fig. 1. Schematic diagram of the deliverer delivering the order

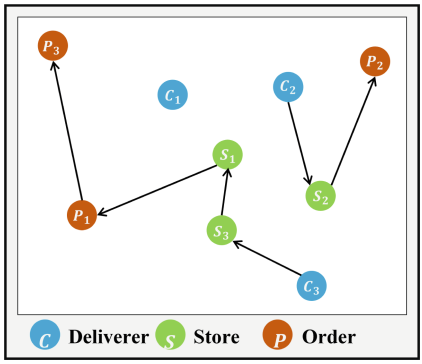

Fig. 2. Schematic diagram of delivery staff stacking orders

Some scholars have used GA to solve the Vehicle Routing Problem (VRP). This is a path planning problem for vehicles starting from warehouses and delivering goods. The similarities with this study are the characteristics of delivery from point A to point B and continuous delivery to different places and finding the shortest path [13]. Therefore, this study also GA will be taken as the main method.

2.2 Traveling Salesman Problem (TSP)

TSP is a classic NP-Hard problem proposed in the 19th century [14], which is still being studied in the field of discrete combinatorial optimization problems. The problem is defined as a set n cities where the distance between any 2 cities can be explicitly calculated. When the salesman's travel speed is at the same speed, find the shortest circuit to start from the designated city to visit each city without repeating it, and finally return to the starting city. If the exhaustive method is used to find the best solution for n cities, the time complexity is O(n!). Therefore, the more cities you visit, the less you can solve in a reasonable time. This paper wants to solve the problem of the deliverer's food delivery. However, this problem is more difficult than the original TSP that this problem considers the scenario which includes multi deliverers and the multi paths. Optimization problems with multiple variables and multiple objectives are practical problems for many complex engineering optimizations. The biggest difficulty is that when optimizing for a single goal or variable, it is necessary to exclude conflicts between other goals or variables, resulting in poor results for other goals or variables [14]. For example, the problem defines three salespeople and nine cities, to meet the conditions all 3 salesmen must travel to 3 cities, each city is visited by only 1 salesperson, and each salesperson does not repeatedly visit the cities that other people have visited at the same time.

3 Problem Definition

The delivery platform need to make order P divided the deliverer C. P and C is the sets $P = \{P_1, P_2, \ldots, P_m\}$ and $C = \{C_1, C_2, \ldots, C_n\}$. The coordination of the partner store S is the set $S = \{S_1, S_2, \ldots, S_o\}$ The delivery path assigned by the deliverer C_i is noted

as R^{c_i} and R^{c_i} include the partner store S_k of the order P_j that will be passed through. It can be noted as $R^{c_i} = \{\{p_1, p_2, \ldots, p_x\}, \{s_1, s_2, \ldots, s_y\} | p_i \in P, s_i \in S\}$. When the platform divided the order to deliverer C_i is consider the between partner store s_j and the consumer p_k the Euclidean distance as $d^{c_i}_{s_j p_k}$ the next get order distance as $d^{c_i}_{S_x P_k}$ and find a deliverer that is closer to the partner store. The distance of that deliverer arrives at the first store is $d1^{c_i}_{p_1}$. When the deliverer finishes the job that can obtain the new order the flow is shown in Fig. 1. In addition, the platform also supplies the deliverer have much order to send at the same time its names as stacked orders that is shown in Fig. 2. The number of stacked orders varies according to the settings of each delivery platform. When the deliverer is not choose stacked orders, the deliverer unknown the next order or the store's location is near or far, which cause much more problem in delivery. If the deliverer chose stacked order method, the food delivery path will be shown after the platform accept this request. If the deliverer finds the path is not expected, the deliverer can abandon the order, but it may cause delays in overall delivery. The stacked orders have to be competed from the each deliverer, otherwise the meaning of fairness will disappear.

The delivery platform's Business Model is to build customer brand loyalty [15]. However, once the order is delayed, the consumer will lose trust in the platform. The optimization problem of the delivery path can be regarded as a TSP, and the optimal path can be obtained by a metaheuristic algorithm to shorten the delivery path of the deliverer. It's just that this only addresses a single TSP. Generally speaking, a delivery platform will have many delivery people delivering meals at the same time. It means that the Multiple Traveling Salesman Problem (MTSP) is still not solved. The deliverer can obtain enough orders and each order has a reasonable distance, Making every deliverer receive fair dispatch and remuneration, and consumers can also receive orders within a reasonable and fair time so the platform can conform to the reasonableness brought about by the principle of fair labor, and the consumer is reduced complain at the same time.

The distance of the deliverer path is $D^{c_i} = d1^{c_i}_{p_1} + \bigcup_{j=1}^{x} \sum_{k=1}^{y} d^{c_i}_{s_j p_k}$. This paper hope that each deliverer receive at least one order. Therefore we use any two D^{c_i} and D^{c_j} to find the standard deviation σ_{ij}. Our main purpose is to pursue the minimum total distance and at the same time consider the minimum difference between the individual distances of each courier, which means that each deliverer C has received orders, which means that this mechanism is fairer. The linear programming model is as follows:

$$\text{Minimize} \sum_{i=1}^{n} \rho_i$$

s.t.

$$\sigma_{ij} \sim 0, i \neq j, i, j \leq n,$$

ρ is the fitness function which means pursuit the lowest standard deviation and the shortest total distance that is shown in the following:

$$\rho = \frac{1}{\sum_{i=1}^{n} D^{c_i}} \times \frac{1}{\sum_{i=1}^{n} \sum_{j=1}^{n} \sigma_{ij}} \tag{1}$$

4 Proposed Method

This paper proposed the mechanism is use GA-based MTSP optimization. The first step is to randomly generate the orders P_i, stores S_i, and deliverers C_i to combine into a chromosome. The initialization phase defines a lot of chromosomes $Z = \{Z_1, Z_2, \ldots, Z_p\}$ and then operate the selection, crossover and mutation to create the new chromosome. Finally, the population will reserve the best chromosome via evaluation until the termination condition is reached. The process is shown in Fig. 3. The overall operation is as follows:

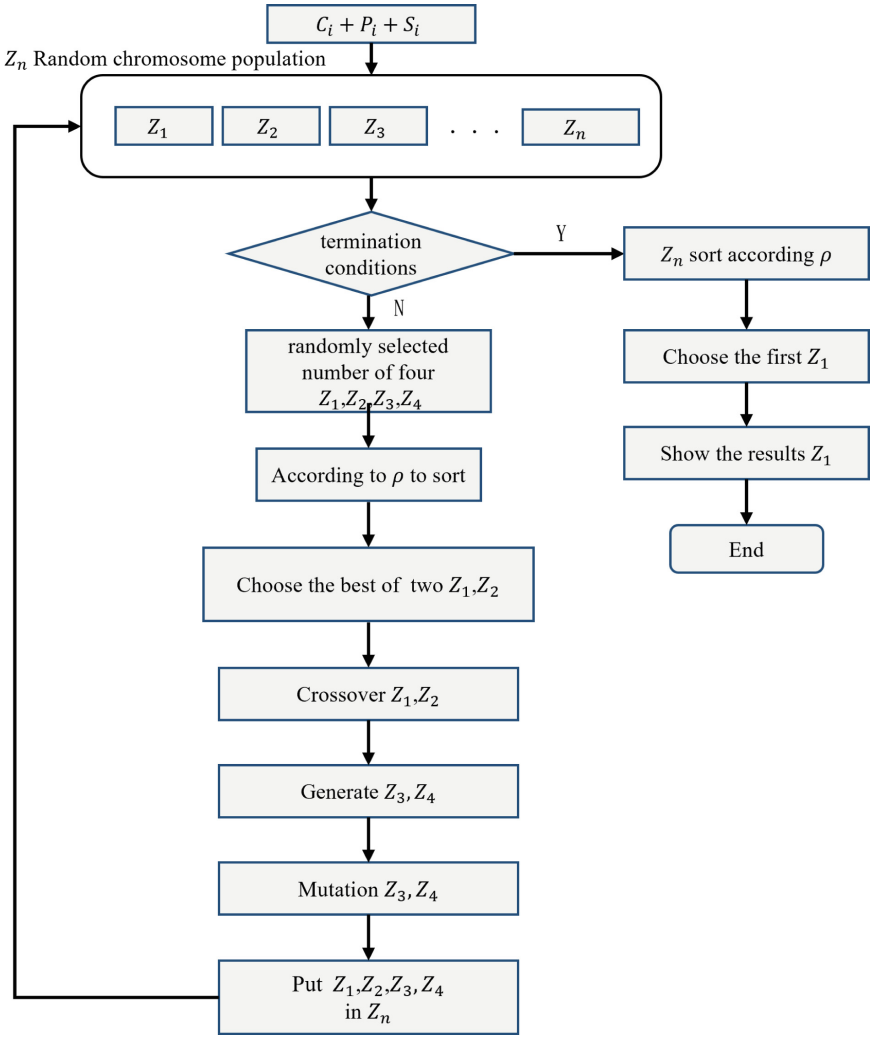

Fig. 3. GA algorithm delivery Order Assignment flowchart

1. Randomly generate P_iS_k and C_i to combine into chromosome.
2. Generate the multiple chromosomes to population Z.
3. Randomly chose 4 chromosomes from Z.
4. Define the fitness function via Eq. (1).
5. Choose the top two Z_1 and Z_2 depending on the sorting results of the fitness function.
6. Randomly invite 30% of Z_1 and Z_2 individually to exchange for crossover.
7. The new chromosomes Z_3andZ_4 will be created that is shown Fig. 4.
8. Take the random order between the non-deliverer position and the non-route last position in a chromosome for mutation such as the position of P_2S_2 and the position ofP_4S_4. Arrange and combine the store coordination (S_2, S_4) and consumer coordination (P_2,P_4) of P_2S_2 andP_4S_4, and select the shortest combination and put it back to the position of chromosome$P_2P_4S_4S_2$. The mutation rate is set 30%.
9. Put the Z_1, Z_2, Z_3, Z_4 into the Z that is shown in Fig. 5
10. Repeat the above steps until the termination condition is reached.
11. After terminating the iteration, sort Z_p according to the smallest standard deviation and the shortest total distance in ascending power, and select the first chromosome.
12. This chromosome is the best solution in the whole domain. The calculation is over.

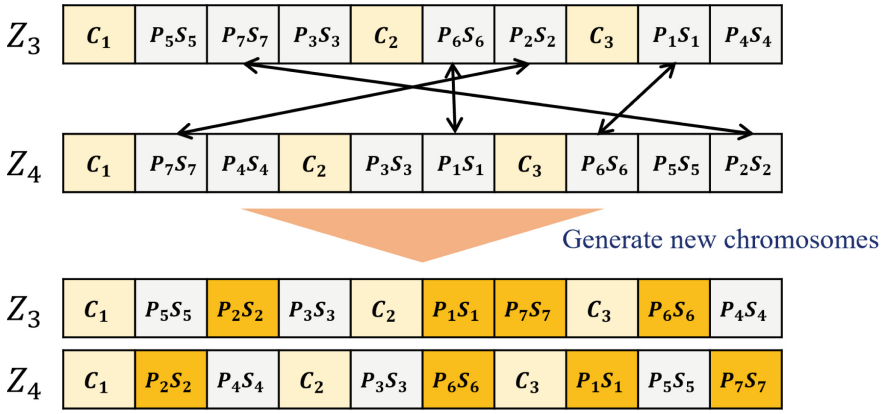

Fig. 4. Schematic diagram of GA's crossover

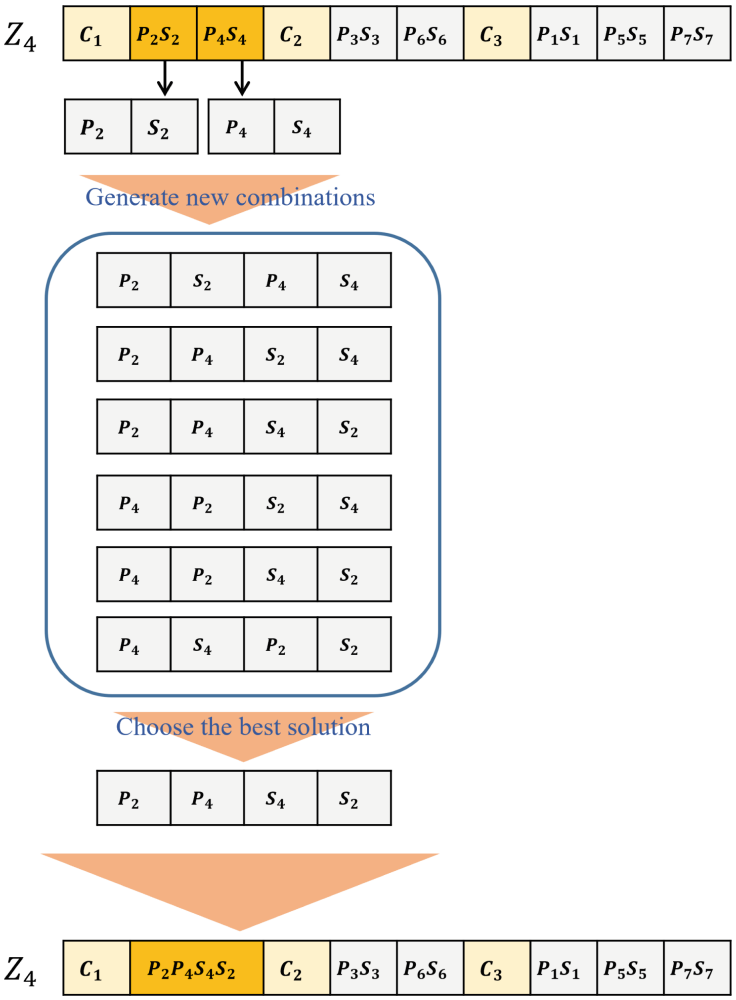

Fig. 5. Schematic diagram of GA's Mutation

The goal of this study is to make the deliverer who can use the system to obtain sufficient quantity and proper distance to reduce the occurrence of order grabbing by deliverer.

5 Simulation Results

This experiment uses C# to build and cooperate with winform's rapid development features, and then refers to Chart components for data visualization that is shown in Table 1. When the algorithm staring, an object name and random coordination should be created. The main thing is to randomize each deliverer and order and then combine them as evenly as possible that is shown in Table 2. The population initial total distance is 3157.35. The standard deviation is 431.94. We can find that the distance of SalesMan_003 is 1451.99 and the distance of SalesMan_001 is 258.66 that there is a significant difference between each other. It is not met the fairness of this paper. After many iterations, the resulting population is significantly improved that the distance of SalesMan_003 is reduced from 1451.99 to 629.3 and the distance of SalesMan_001 is increased from 258.66 to 628.80 that the difference is closer. Although the total distance raises to 3446.71 after the optimization, the standard deviation is reduced to 41.94. This result has met the fairness. This is because when each deliverer has an order and their standard deviation ate similar, it represents the delivery distance of each deliverer is also similar so that the waiting time of the consumers are also more balanced.

Table 1. Experimental environment

Type	Tool
Development	Visual Studio 2019
Platform	Winform
Language	C#
Framework	.Net Framework 4.8
Visualization	Chart
Data source	Randomly generated coordinates

This paper still has unresolved issues, such as the stack orders situation. Therefore how to use GA to optimize this problem is a major point in the future.Since the order of the store to the consumer cannot be reversed so that the path looks complex and messy that is shown in Fig. 6. For example, the coordination of SalesMan_002 is [28, 27] the former coordination represents the store coordination. The latter one represents the consumer coordination. The last index is the order code. We can see that the SalesMan_002 got three orders that the orders index are 002,008 and 011. This deliverer starts from [28, 27]. Firstly, this deliverer should go to coordination's [115, 184] of store of StoreCost 002 to take an order and then deliver this meal to the consumer [92, 193]. Then this deliverer continue to go to the coordination [26, 144] of StoreCost 008 to take the second order and go to the consumer [192, 44] to deliver the order. Then return to the coordination [16, 140] of StoreCost 011 to get the third order and finally go to deliver the meal to consumer [153, 68] to finish the task.

Table 2. Simulation results

Population initial value			Optimal distance		Distance difference
Delivery men coordinates	Distance	Path coordinates	Optimal distance	Optimal coordinates	Difference
SalesMan_000 [119,175]	340.26	SalesMan_000:[119,175]→ StoreCost_009:[146,122][122,4]→ StoreCost_010:[75,60][166,106]→ StoreCost_002:[115,184][92,193]	646.10	SalesMan_000:[119,175]→ StoreCost_017:[63,40][60,83]→ StoreCost_005:[53,107][100,54]→ StoreCost_014:[76,113][186,121]→ StoreCost_013:[176,114][19,12]→ StoreCost_001:[28,27][188,32]	+305.84
SalesMan_001 [166,30]	258.66	SalesMan_001:[166,30]→ StoreCost_000:[119,175][166,30]→ StoreCost_018:[136,74][32,191]	628.80	SalesMan_001:[166,30]→ StoreCost_004:[7,59][77,107]→ StoreCost_010:[75,60][166,106]→ StoreCost_015:[121,4][21,120]→ StoreCost_003:[51,56][83,7]	+370.14
SalesMan_002 [28,27]	444.64	SalesMan_002:[28,27]→ StoreCost_005:[53,107][100,54]→ StoreCost_011:[16,140][153,68]→ StoreCost_001:[28,27][188,32]	749.15	SalesMan_002:[28,27]→ StoreCost_002:[115,184][92,193]→ StoreCost_008:[26,144][192,44]→ StoreCost_011:[16,140][153,68]	+304.51
SalesMan_003 [188,32]	1451.99	SalesMan_003:[188,32]→ StoreCost_008:[26,144][192,44]→ StoreCost_017:[63,40][60,83]→ StoreCost_003:[51,56][83,7]→ StoreCost_012:[49,141][180,61]→ StoreCost_015:[121,4][21,120]→ StoreCost_006:[42,106][104,184]→ StoreCost_013:[176,114][19,12]→ StoreCost_004:[7,59][77,107]→ StoreCost_014:[76,113][186,121]	629.30	SalesMan_003:[188,32]→ StoreCost_016:[14,194][167,105]→ StoreCost_018:[136,74][32,191]→ StoreCost_012:[49,141][180,61]	-822.69
SalesMan_004 [115,184]	661.79	SalesMan_004:[115,184]→ StoreCost_019:[139,192][39,101]→ StoreCost_016:[14,194][167,105]→ StoreCost_007:[11,193][170,96]	793.33	SalesMan_004:[115,184]→ StoreCost_000:[119,175][166,30]→ StoreCost_006:[42,106][104,184]→ StoreCost_009:[146,122][122,4]→ StoreCost_019:[139,192][39,101]→ StoreCost_007:[11,193][170,96]	+131.54
Total distance: 3157.35			Total distance: 3446.71		+289.39
Standard deviation: 431.94			Standard deviatio: 68.59		-363.35

It can be found from the delivery route of SalesMan_002 that this research can improve the method of increasing stacking orders and optimize the route to be shorter. If StoreCost 008 and 011 are stacked for delivery, a new route can be generated:

StoreCost_008Join011
[26, 144][16, 140][153, 68][192, 44]

The SalesMan 002' starting coordination is [28, 27]. Firstly, the deliverer came to coordination [115, 184] of StoreCost 002 to obtain the meal and then deliver it to consumer [92, 193]. Second, the deliverer came to coordination [26, 144] of StoreCost 008 to get the meal for [16, 140]. Finally, the deliverer will go to coordination [153, 68] of StoreCost 01 to obtain the meal then deliver to coordination [192, 44] to consumer. The optimization paths between three consecutive order can be found.

In the future, in addition to adding the method of stacking orders, this research will continue to strengthen the data visualization part, so that the coordination of merchants and consumers are more clearly marked, and directional optimization reading is added to the behavior route.

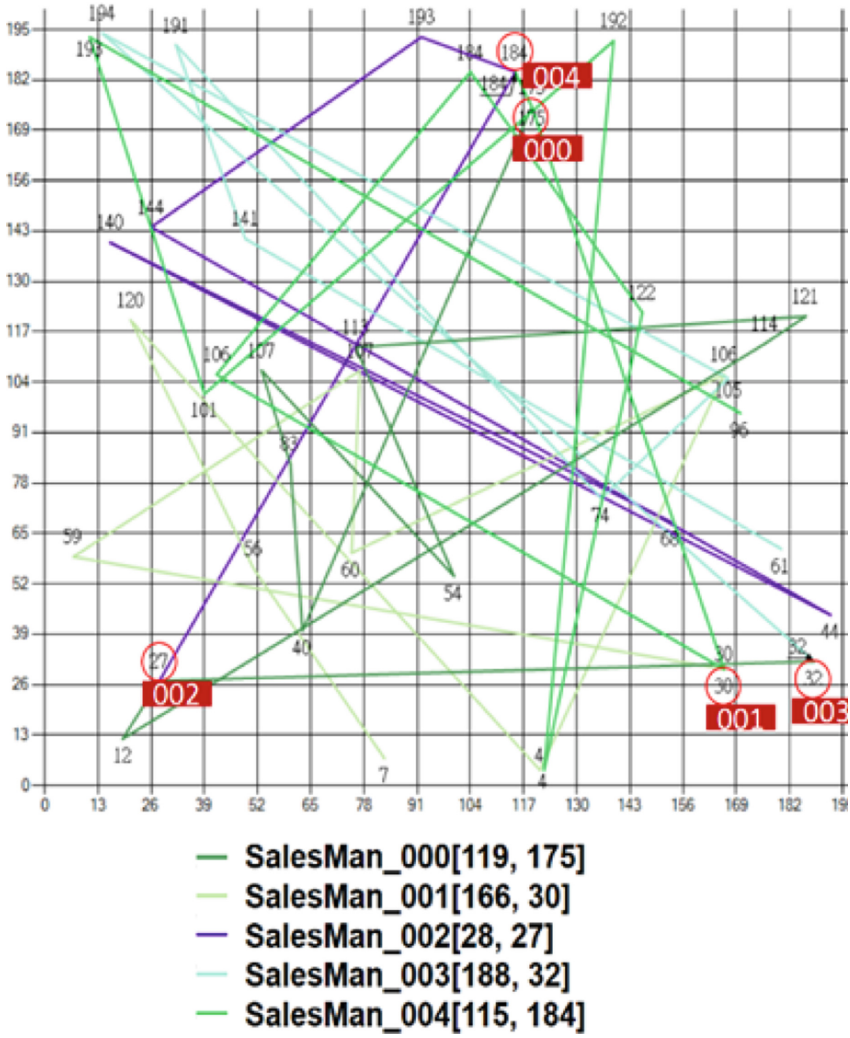

Fig. 6. Simulation topology

6 Conclusion

Food delivery platforms have become inseparable from human life. But the current popular platform just allows the deliveryman to grab the order by himself. This is not fair. Therefore, the main purpose of this study is to propose a fair delivery platform, which takes into account the ability of each delivery person and the time it takes for the meal to reach the consumer. It turns out that we will be able to set the delivery path and it can have fair and effective characteristics to achieve the highest net value for all. In the future, we will consider the situation of stacking orders to make the process closer to reality. Besides, In the future, we hope the proposed method can extend to annoying

work in life such as operating room surgery scheduling, production line automation scheduling, etc., making work arrangements easier and more reasonable.

Acknowledgments. This work was financially supported from the MOST in Taiwan, under grant MOST 111-2221-E-197-017.

References

1. Silva, A.D., Neves, R.F., Horta, N.: Multi-objective optimization. In: Portfolio Optimization Using Fundamental Indicators Based on Multi-Objective EA. SAST, pp. 57–72. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-29392-9_4
2. Cho, H.H., Wu, H.T., Lai, C.F., Shih, T.K., Tseng, F.H.: Intelligent charging path planning for IoT network over Blockchain-based edge architecture. IEEE Internet Things J. **8**(4), 2379–2394 (2020)
3. O'Rourke, J., Supowit, K.: Some NP-hard polygon decomposition problems. IEEE Trans. Inf. Theory **29**(2), 181–190 (1983)
4. Chong, K.L., et al.: Review on dam and reservoir optimal operation for irrigation and hydropower energy generation utilizing meta-heuristic algorithms. IEEE Access **9**, 19488–19505 (2021)
5. Ohashi, T., Aghbari, Z., Makinouchi, A.: Hill-climbing algorithm for efficient color-based image segmentation. In: IASTED International Conference on Signal Processing, Pattern Recognition, and Applications, pp. 17–22 (2003)
6. Cho, H. H., Tseng, F. H., Shih, T. K., Chou, L. D., Chao, H. C.: SA-based multimedia conversion system for multi-users environment. In: Sun, X., Liu, A., Chao, HC., Bertino, E. (eds.) Cloud Computing and Security. ICCCS 2016. LNCS, vol. 10040. Springer, Cham. https://doi.org/10.1007/978-3-319-48674-1_26
7. Tseng, F.-H., Cho, H.-H., Lai, C.-F.: Mobile charger planning for wireless rechargeable sensor network based on ant colony optimization. In: Park, J.J., Fong, S.J., Pan, Y., Sung, Y. (eds.) Advances in Computer Science and Ubiquitous Computing. LNEE, vol. 715, pp. 387–394. Springer, Singapore (2021). https://doi.org/10.1007/978-981-15-9343-7_53
8. Deng, W., et al.: An enhanced fast non-dominated solution sorting genetic algorithm for multi-objective problems. Inf. Sci. **585**, 441–453 (2022)
9. Akarsu, C.H., Küçükdeniz, T.: Job shop scheduling with genetic algorithm-based hyperheuristic approach. Int. Adv. Res. Eng. J. **6**(1), 16–25 (2022)
10. Dogan, A., Birant, D.: Machine learning and data mining in manufacturing. Expert Syst. Appl. **166**, 114060 (2021)
11. Elazouni, A.M., Metwally, F.G.: Finance-based scheduling: tool to maximize project profit using improved genetic algorithms. J. Constr. Eng. Manag. **131**(4), 400–412 (2005)
12. Lee, C.K.H.: A review of applications of genetic algorithms in operations management. Eng. Appl. Artif. Intell. **76**, 1–12 (2018)
13. Liu, N., Pan, J.S., Chu, S.C.: A competitive learning quasi affine transformation evolutionary for global optimization and its application in CVRP. J. Internet Technol. **21**(7), 1863–1883 (2020)
14. Konak, A., Coit, D.W., Smith, A.E.: Multi-objective optimization using genetic algorithms: a tutorial. Reliab. Eng. Syst. Saf. **91**(9), 992–1007 (2006)
15. Liberti, L., Lavor, C., Maculan, N., Mucherino, A.: Euclidean distance geometry and applications. SIAM Rev. **56**(1), 3–69 (2014)