TECHMIND RESEARCH
ISSN: 2291-2118
International Journal of Research in Business and Technology
Volume 9 No. 1 December 2016

Analyzing the efficiency of meals delivery transportation

Azmin Azliza Aziz[1*], Ph.D.
Quah, Chee-Heong[2], Ph.D.
Mohammad Nazri[3], DBA.
Faculty of Business and Accountancy,
University of Malaya, Kuala Lumpur 50603,
Malaysia.
[1]aazliza@um.edu.my, [2]quahch@um.edu.my, [3]nazrry@um.edu.my
*Corresponding Author

Abstract- *This paper demonstrates a practical application of solving meal delivery problem at Coventry City Council-modeled as a multiple-Traveling Salesman Problem with Time Windows (m-TSPTW) and optimized using Trapeze®PASS application. In minimizing the total distance traveled, alternative scenarios are checked by varying the delivery time windows and the number of vehicles. Results indicate substantial saving, in contrast to manual arrangement, by employing the commercial, off-the-shelf software package in scheduling.*

Keyword- *Transportation; routing; optimization; delivery; Coventry*

Classification code- 90B06; 90B50; 90B90

1. INTRODUCTION

Vehicle Routing Problem, first introduced by Dantzig and Ramser in 1959, is defined as the problem of designing routes of minimum objective function for vehicles from a central depot to a set of geographically scattered points. The routes are designed in such a way that every point is visited only once by a fleet of capacitated vehicles; all routes originate and terminate at the depot; and the total demand of all points of any particular route must not exceed vehicle capacity (Toth and Vigo, 2002)[9].

Traveling Salesman Problem is a special case of Vehicle Routing Problem (VRP) which concerns about finding a tour from one point to another so that only a single vehicle is utilized and the vehicle capacity is unbounded. Traveling Salesman Problem is a well-known combinatorial optimization problem which has attracted enormous interest among researchers and practitioners (Lawler et al., 1985[8]; Gutin and Punnen, 2002[7]). The solutions are applied across the areas of logistics, telecommunications, military, and genetics (Applegate, 2006[1]; Bräysy et al., 2009a, b[3][4]). Whilst studies on the private sector are common, those on the public sector are comparatively scarce.

In the light of the above, this paper investigates a case of the city council of Coventry, a local district in the metropolitan of West Midlands, England. Constrained by data availability, the scope is bounded to its service of meal delivery. Research focusing on meals delivery problem is particularly limited, with the exception of two closely related studies by Bräysy et al. (2009a, b)[3][4], where the authors claimed the papers to be the first to consider the home meal delivery problem in the literature. Specifically, the case study conducted in the city of Jyvaskyla, Finland, was modelled as the multiple travelling salesman problem with time windows (m-TSPTW) and tackled using SPIDER Designer 4.0 (Bräysy et al., 2009a) [3].

This study is timely as it addresses the efficiency issue of routes that have been manually determined up till now. Utilizing an established commercial software application, the findings reveal significant gain in efficiency notwithstanding the operation’s unique structure and constraints. The findings are beneficial to the decision makers especially in the council as this assists them to improve the quality of the service. From the academic perspective, considering this real life problem sheds light on the possible optimization procedures for solving practical problems and draws insight into the literature involving real life data.

The remainder of this paper is organized as follows. In the next section, a description of the meal delivery problem is briefed. The model and method employed are detailed in Sections 3 and 4, in that order. The fifth section explores the findings. Finally, Section 6 concludes.

2. Problem Description

The Meal Delivery service, also known as Mobile Meals, provides daily delivery of meals to the disabled and elderly, including mentally-challenged adults. The service operates every day, but may be restricted during weekends and bank holidays.

Up till 2013, the routing is done manually in a traditional fashion. Authorized staff members partition the

Coventry city into eight divisions on a large map. Clients are marked and scheduled according to the corresponding divisions of their residence. Each of the eight divisions has a defined route served by a vehicle. There are eight vehicles in total. On weekends and holidays, however, only seven vehicles are used. The vehicles utilized are vans provided by the city council based at the central depot. New clients will be allocated into one of these routes.

At the *Kitchen*, meals are packed in small-size boxes and then loaded into containers. Each container can capacitate up to 100 boxes. Despite that, in practice, as each van only serves one route, packed meals are loaded into the containers at the *Kitchen* according to the number of clients served on a particular route.

Daily operation begins at 10:45 am where all vans depart from the depot to the *Kitchen* which is about 1 mile away before containers are loaded into the vans. The loading job for each vehicle normally takes about 10 minutes and all vans can be loaded simultaneously. By 11:00 am, drivers depart from the *Kitchen* and deliver the meals to the clients.

Weekday service is operated by two personnel, a driver and a co-driver. In cases where the driver might not find a space to park, he has to wait on the van while the co-driver delivers the meals. On weekends, though, the service is only operated by seven drivers instead of fourteen, that is, one worker one van.

Observation during site visits indicates that, on average, one minute is taken for a single service time, namely the time needed to get down the van, walk to the client's door and hand over the meal.

As soon as all clients are served, the van returns to the *Kitchen* to unload the empty container. This takes about two minutes before it travels back to the depot. In general, all drivers complete their duties at approximately 1:00 pm. Typically, the daily mileage traveled for weekday and weekend are 210.6 miles and 163.5 miles, respectively. Average speed of 24.85 miles per hour is deployed throughout the operation. Figure 1 depicts the daily routing cycle of the delivery service.

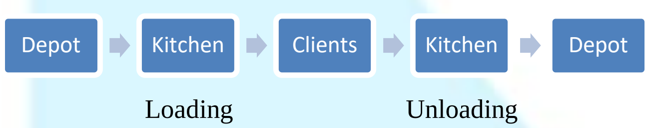

Figure 1 Daily routing cycle of meal delivery service

3. Model

The delivery service can be regarded as a Vehicle Routing Problem with Time Windows (VRPTW). The objective is to minimize total distance, measured by the sum of distances traveled by all the vans used.

Problems of practical interest often deal with real life constraints that add greater complexities in finding the solutions. The essential constraint for delivery service is time window. There are two time windows involved.

The first is that which is identical in all routes, denoted by [E, L] where E and L are the earliest departure time from the depot and the latest arrival time at the depot. The service operates from 10:45 am to 14:45 pm, a period of four hours, which is also the maximum working period for the drivers. Since the drivers are paid on hourly basis, restricting the maximal travel duration is crucial. Therefore, allocating a hard time window of [10:45, 14:45] is appropriate.

The second time window concerns about the clients. Meals should be delivered to clients within 3.5 hours, between 11:00 am and 14:30 pm. As delivery time may vary within the interval of [11:00, 14:30], this constraint is considered as soft time window. Even though the vans are limited on the capacity at a time, the capacity restriction is not practically significant as the demand for a particular route is always below the maximum capacity.

Along these lines, the meal delivery service can be modeled as a multiple-Traveling Salesman Problem with Time Windows (m-TSPTW). The m-TSPTW is regarded as a generalization of TSP and a relaxation of VRPTW, that is, a model without restriction on the capacity of vehicle (Bektas, 2006)[2].

The problem can be defined on a directed graph $G = (V, A)$ where $V = \{v_0, v_1, \ldots, v_n\}$ is the vertex set and $A = \{(v_i, v_j): v_i, v_j \in V, i \neq j\}$ is the arc set. Vertex v_0 corresponds to the depot at which the fleet of vehicles is based, v_1 represents the *Kitchen* and the clients are represented by vertices v_2 to v_n. Each vertex $v_i \in V, i = 0$ to n is related with the service time t_i, namely the time spent at a particular vertex, and the time window $[e_i, l_i]$ as defined by the earliest e_i and the latest l_i possible time for the vertex to be visited. While service time is not applicable to the depot ($t_0 = 0$), the service times at the *Kitchen* and the client locations represent respectively the time taken for loading-unloading the meal containers and that for delivering the meals.

The drivers operating the vehicles shall not exceed the maximum working hours which is linked with the total duration of the whole service, denoted by [E, L]. To reiterate, this is the time window when all vehicles leave the depot to the *Kitchen*, traveling from one client to another while delivering the meals, returning to the *Kitchen* and finally to the depot again. Each arc (v_i, v_j) is given a non-negative length, $d_{i,j}$ which is the distance traveled between v_i and v_j. A fleet of m homogeneous vehicles is globally assigned to the depot. Since each van only serves one route, the problem consists of finding m vehicle routes with minimum total distance traveled, each route starting and terminating at the depot, each client $v_i, i = 2$ to n is visited exactly once within the associated time window $[e_i, l_i]$.

4. Data and Method

Data of customers used comprise of details about meal preference, address, status of disability, dietary requirement, and the choice of weekday and/or weekend service. Altogether, there are 237 and 174 active clients on weekdays and weekends, respectively.

The routing and scheduling exercise is performed using Trapeze® PASS software (or simply Trapeze), which is a software package currently utilized to plan and manage transportation system. Produced by Trapeze Group, a Canada-based company, the application is well-known for its scheduling efficiency. Integrated with geographical information system (GIS) functionality and map data obtained from Ordnance Survey, Esri, or MapInfo, the Trapeze system supports a number of transport schemes.

Along the objective of minimization of total distance, Trapeze defines this objective in terms of costing weight criteria. Each criterion is associated with a scale of which its weight can be adjusted by sliding the indicator to the right or left. The settings of the costing weight criteria determine the relative priority of each criterion when the algorithm looks for solution. **Table 1** lists down the costing weight criteria and definitions.

Costing Weight Criteria	Description
Minimize Distance	To minimize the total distance for all the vehicles used in the service
Minimize "Out of the Way"	To ensure that any trip occurs within certain predefined area, rather than having trips across the area
Minimize "Total Out of the Way"	Similar to the above, where the whole trips are taken into consideration, therefore assist with keeping the routes well clustered and all vehicles working within the predefined area
Minimize Backtrack	To reduce the times a vehicle returns to a point at or near where it has recently been
Minimize Violation	To reduce any inconvenience to the passenger or administrators/ drivers that may cause as a result of violations so that efficient schedules can be produced

Table 1 Costing weight criteria and description (Source: Trapeze manual)

Figure 2 depicts a costing weight scale as appeared in the program. For the purpose of this study, five scale intervals have been specified, which are 0, 0.25, 0.5, 0.75 and 1, with 0 and 1 indicating least important and most important, respectively.

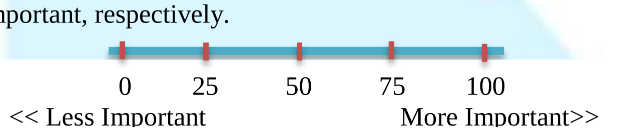

Figure 2 Costing weight scale (Source: Trapeze manual)

The costing weights are initially set to default at the midpoint of 0.5 with the assumption that all five objectives have equal weights. Meanwhile, the time window constraint is set according to the earliest and the latest time, [10:45, 14:45]. The number of vehicles is set to 8. Data are that of weekday clients. Each output provides total distance, computational elapsed time, and vehicle routing plan of which total distance is of interest here. Costing weight scales are adjusted, left or right, depending on their marginal effects on total distance until a consistent smallest total distance is obtained.

The routing plan in the output of Trapeze is given as follows:

Depot→ Clients→Depot

Nevertheless, the meal delivery service has a slightly different routing form, which is:

Depot→Kitchen→Clients→Kitchen→Depot

Therefore, in order to make the schedule practicable, slight manual modification is made. A single trip to the *Kitchen* is manually slotted when the vehicle leaves the depot and when it returns to it.

Apart from using present parameters, the analysis also seeks to optimize resources by inserting into the iterations fewer vehicles and smaller windows of delivery time. Correspondingly, alternative findings are checked by varying the delivery time window interval and number of vehicles.

The initially set delivery time window is [10:45, 14:45]. Nonetheless, once the optimal costing weights are identified, delivery time window, and number of vehicles are altered to check if smaller sum of distance can be obtained. The five scenarios proposed using weekday data are as shown in **Table 1**.

Scenario	Delivery time window		Number of vehicles
1	11:00-14:30	(3.5 hrs)	8
2	11:00-14:30	(3.5 hrs)	X
3	11:00-14:30	(3.5 hrs)	X*
4	11:00-13:00	(2 hrs)	Y
5	11:00-13:00	(2 hrs)	Y*

Table 1 Five scenarios using weekday data

Scenario 1 follows the current practice with the delivery time window of 3.5 hours and eight vehicles. Scenarios 2 and 4 aim to minimize the number of vehicles within 3.5 and 2 hours time window, respectively. In Scenario 2, three vehicles are initially set since it is the lower bound of the number of vehicles taking into account the maximum capacity of the container and number of clients. Meantime, Scenarios 3 and 5 check if total distance in Scenarios 2 and 4 can be decreased by adding one van at a time.

For weekend, due to significantly fewer clients, only two scenarios are proposed as shown in **Table 3**. The exercise aims to minimize the number of vehicles constrained by time windows of 3.5 and 2 hours.

Scenario	Delivery window	time	Number of vehicles
1	11:00-14:30	(3.5 hrs)	X
2	11:00-13:00	(2 hrs)	Y

Table 3 Two scenarios using weekend data

5. Results

Analysis is run using Trapeze® PASS with the objective of minimizing total travel distance. [Figure 3](#) depicts 16 trial sets of costing weights, including the optimal set that minimizes the total distance, namely set 13. [Table 2](#) details the optimal weights.

[Table 3](#) collects the results of the five scenarios using weekday data. For Scenario 1, a minimum total distance of 193.88 miles with 3.5 hours time window and 8 vehicles is obtained. As results of Scenario 2 indicate, by just adding one van to the lower bound of 3 vans, total distance is reduced to 145.99 miles. Quite the opposite, using 5 vans increases total distance to 162.03 miles, as output of Scenario 3 shows. For a smaller time window of 2 hours, a total of 7 and 8 vans are needed to produce total distances of 182.83 and 193.88 miles, respectively, as Scenarios 4 and 5 reveal.

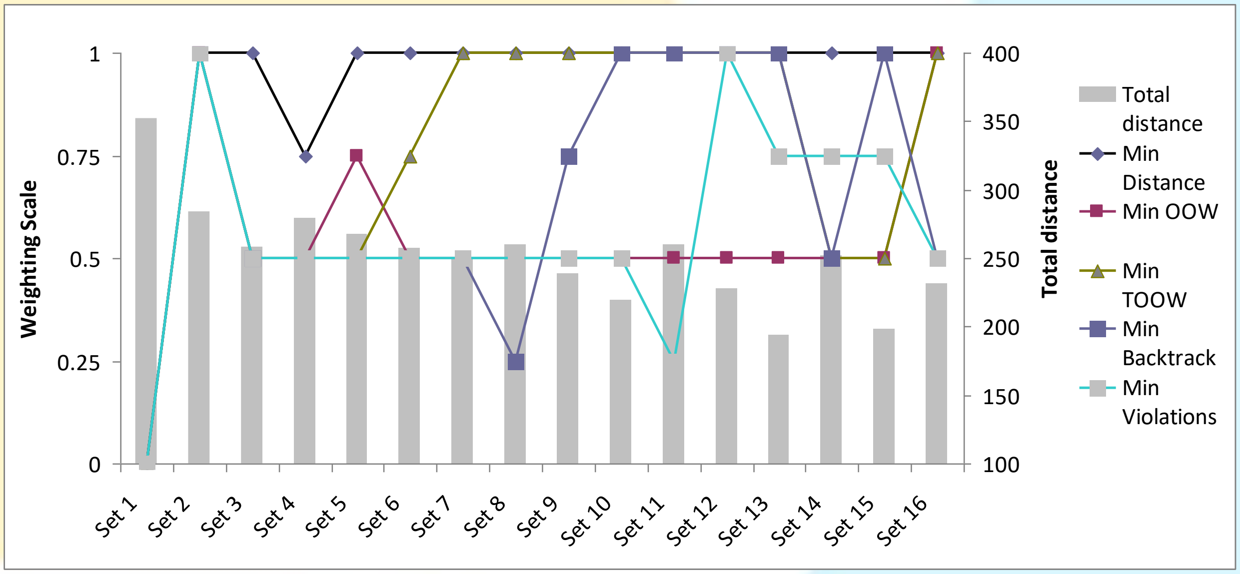

Figure 3Trial sets of costing weights including the optimal set (Source: Findings using Trapeze)

Costing Weight Criteria	***Weights***
Minimize Distance	1
Minimize “Out of the Way”	0.5
Minimize “Total Out of the Way”	1
Minimize Backtrack	1
Minimize Violation	0.75

Table 2 Optimal costing weights used for all scenarios for weekday and weekend (Source: Findings using Trapeze)

Scenario	**1**	**2**	**3**	**4**	**5**
Delivery time window (hrs)	3.5	3.5	3.5	2	2
Number of vehicles	8	4	5	7	8
Total distance (miles)	193.88	145.99	162.03	182.83	193.88

Table 3 Results of five scenarios for weekday (Source: Findings using Trapeze)

Two observations on total distance can be drawn from [Table 3](#). First, total distance increases when the number of vehicles increases. Second, when van number is doubled from 4 to 8, Scenarios 1 and 5 produce the same total distance of 193.88 miles despite different time windows. As a subsidiary finding, the computational elapsed time is found to span from one to six minutes. This indicates the practicality of using the approach in real-life setting. The savings in percent against the present practice are presented in [Table 4](#)

Current	***Saving (%)Scenario***					
Practice		**1**	**2**	**3**	**4**	**5**
210.6 miles	Total distance	7.94	30.68	23.06	13.19	7.94
8	Number of vehicles	0	50.0	37.5	12.5	0
3.5 hours	Delivery time window	0	0	0	42.86	42.86

Table 4 Savings for weekday (Source: Author’s calculation)

The total distance of 193.88 miles of Scenarios 1 and 5 yields a saving of 7.94% in contrast to current practice. Reducing the number of vehicles by one (Scenario 4) produces a saving in total distance of 13.19%. Using just 4 vehicles in Scenario 2 offers the greatest saving of 30.68% in distance in comparison to present manual arrangement. Scenario 2 yields the highest savings both in terms of total distance and number of vehicles at about 50%. For shorter time window, Scenario 4 saves 12.5% of vehicles compared to present setting, with a moderate saving in total distance.

When manual adjustments are made to account for the visit to the *Kitchen* before and after serving the clients, the increment in total distance is qualitatively negligible because the *Kitchen* is only about 1.16 miles from the depot and 1.49 miles vice versa.

For weekend, the numbers of clients and vehicles are smaller, namely 174 clients and 7 vehicles. Two scenarios with 3.5 hours and 2 hours time windows and 3 and 6 vans respectively are checked for their total distances, as shown in **Table 7**. The minimum total distances are 126.50 and 160.93 miles, correspondingly. Against present manual arrangement, the savings, as shown in **Table 5**, are gained.

If smaller total distance is priority, Scenario 1 offers better solution with greatest saving on both total distance and number of vehicles, that is, 22.63% and 57.1%, respectively. Nevertheless, if smaller delivery time window is preferred, Scenario 2 is better as it reduces time window by 42.86% compared to current practice. Incorporating a trip to the *Kitchen* before and after serving the clients only marginally increases the total distance by less than 2.3 miles.

Scenario	**1**	**2**
Delivery time window (hrs)	3.5	2
Number of vehicles	3	6
Total distance (miles)	126.50	160.93

Table 7 Results of two scenarios for weekend (Source: Findings using Trapeze)

Current	**Savings (%)**	**Scenario**	
Practice		**1**	**2**
163.5 miles	Total distance	22.63	1.57
7	Number of vehicles	57.1	14.3
3.5 hours	Delivery time window	0	42.86

Table 5 Savings for weekend (Source: Author's calculation)

6. Conclusion

This paper has examined a case of meal delivery service in the city of Coventry. The service is modeled as a multiple-traveling salesman problem with time windows, optimized using Trapeze® PASS with the objective of minimizing the total distance traveled by the vehicles. Several alternative scenarios with different delivery time windows and numbers of vehicles are compared against that of current practice. These scenarios serve as a reference in the selection of outcome.

Results reveal that significant savings can be obtained by employing the commercial, off-the-shelf software package in the scheduling exercise. For weekday delivery, up to 30% and 13% distance savings can be achieved with 3.5 and 2 hours time windows, respectively. In the former case, vehicle utilization can be reduced by half. For weekend delivery, meantime, distance saving of 22% is gained, using only 3 vans within 3.5 hours. Correspondingly, shorter distances mean lower costs of fuel and wage.

In another respect, minimizing time used is not of concern for two reasons. Firstly, the data for total service time are not provided. Hence, comparisons with findings of Trapeze cannot be made. Nonetheless, the analysis does attempt to minimize the delivery time window. Secondly, computational runtime may be different depending on computing specification. Despite the above, depending on context, Trapeze can be used to optimize any dimension, including time minimization.

In a nutshell, the findings serve to enhance cost efficiency of a meal delivery operation. This work is another piece of evidence that demonstrates the link between scholarly literature and practical application.

References

[1] Applegate DL, Bixby RE, Chvatal V, Cook WJ (2006) The Traveling Salesman Problem: A Computational Study. Princeton University Press, Princeton

[2] Bektas T (2006) The multiple traveling salesman problem: an overview of formulations and solution procedures. Omega 34: 209-219

[3] Bräysy O, Dullaert W, Nakari P (2009a) The potential of optimization in communal routing problems: case studies from Finland. Journal of Transport Geography 17(6): 484-490

[4] Bräysy O, Dullaert W, Nakari P, Neittaanmäki P (2009b) An optimization approach for communal home meal delivery service: A case study. Journal of Computational and Applied Mathematics 232(1): 46-53

[5] Coventry1 http://www.coventry.gov.uk/ccm/cms-service/stream/asset/?asset_id=9625002

[6] Dantzig GB, Ramser JH (1959) The truck dispatching problem. Management Science 6: 80-91

[7] Gutin G, Punnen AP (2002) The Traveling Salesman Problem and Its Variations. Kluwer Academic Publishers, Netherlands

[8] Lawler EL, Lenstra JK, Rinnooy Kan AHG, Shmoys DB (1985) The Traveling Salesman Problem-A Guided Tour of Combinatorial Optimization. Wiley, Chichester

[9] Toth P, Vigo D (2002) The Vehicle Routing Problem. SIAM Monographs on Discrete Mathematics and Applications, Philadelphia

[10] www.trapezegroup.co.uk