Accepted Manuscript

Asia-Pacific Journal of Operational Research

Article Title:	A Decision Support System for Data-driven Driver-Experience Augmented Vehicle Routing Problem
Author(s):	Qitong Zhao, Chenhao Zhou, Giulia Pedrielli
DOI:	10.1142/S0217595920500189
Accepted:	19 February 2020
To be cited as:	Qitong Zhao, Chenhao Zhou, Giulia Pedrielli, A Decision Support System for Data-driven Driver-Experience Augmented Vehicle Routing Problem, *Asia-Pacific Journal of Operational Research*, doi: 10.1142/S0217595920500189
Link to final version:	https://doi.org/10.1142/S0217595920500189

A Decision Support System for Data-driven Driver-Experience Augmented Vehicle Routing Problem

Qitong Zhao*
School of Business, Singapore University of Social Sciences
qzhao@suss.edu.sg
Chenhao Zhou
Department of Industrial Systems Engineering and Management, National University of Singapore
zhou_chenhao@u.nus.edu
Giulia Pedrielli
School of Computing Informatics and Decision Systems Engineering, Arizona State University
Giulia.Pedrielli@asu.edu

Abstract

Logistics delivery companies typically deal with delivery problems that are strictly constrained by time while ensuring optimality of the solution to remain competitive. Often, the companies depend on intuition and experience of the planners and couriers in their daily operations. Therefore, despite the variability-characterising daily deliveries, the number of vehicles used every day are relatively constant. This motivates us towards reducing the operational variable costs by proposing an efficient heuristic that improves on the clustering and routing phases. In this paper, a decision support system (DSS) and the corresponding clustering and routing methodology are presented, incorporating the driver's experience, the company's historical data and Google map's data. The proposed heuristic performs as well as k-means algorithm while having other notable advantages. The superiority of the proposed approach has been illustrated through numerical examples.

Keywords: Decision Support System, Vehicle Routing Problem, Heuristics, Clustering, Routing

1. Introduction

With the tremendous growth of e-commerce around the world, it has become a great challenge for the logistics companies to improve operational efficiency and customer satisfaction. As the logistics and economic hub in the south-east Asia, Singapore deals with a very high volume of deliveries and pick-ups every day. One of the world's leading logistics company in Singapore (henceforth referred to as 'the company') can receive an average of 10,000 delivery parcels from the inbound flight and 3,000 pick-up parcels daily. Most of the times, in the real practice, parcels are manually assigned to each vehicle and the vehicle routing is determined by the driver. The inefficiency derived from such an approach can lead to, inter alia, decreasing service level due to the delivery delays, workload imbalances between vehicles and avoidable unnecessary use of resources. Therefore, it is important to develop an efficient and effective method to handle the pick-up and delivery processes while optimizing the resources at hand.

For the company, the couriers are classified into two categories to handle commercial and residential customers separately. Couriers handling commercial customers, due to contractual deadlines, generally perform all deliveries prior to noon, whilst reserve the remainder of the day for pick-ups. On the other hand, residential customers generally do not raise any pick-up jobs, and may be unable to receive packages in the day owing to them being away from their respective residences for work. Hence, residential couriers perform deliveries throughout the day.

Currently, the company relies on intuition and experience-based clustering followed by a routing methodology. The results are often not optimal, and lead to wastages that can be avoided. With the development of digitalization, it is much easier for the logistics companies to gather delivery and pick-up data. The company should use this data to make informed decisions. The switch from the current intuition-based scheduling to data-driven scheduling means that results will be consistent, supported by hard facts, and hence, reliable. The objective of this paper is to formulate a heuristic that incorporates and takes advantage of the large amount of data the company collected to solve their daily VRP with considerably better results within acceptable computational times.

The remainder of the paper is organized as such: Section 2 provides a literature review on exact methods and heuristics which help in solving the VRP. Section 3 contains the problem description. Section 4 describes the methodology used and outlines the proposed Decision Support System (DSS). Section 5 explains the dynamic clustering and routing algorithms. Section 6 deals with the experiments and results. Finally, Section 7 contains the conclusion and possible future work.

2. Literature Review

For the vehicle routing problem, methodologies range from exact approaches, which guarantee optimality in small scale problems, to heuristics that find acceptable solutions in large scaled problems (Talbi, 2009). In this literature review, we will focus on some heuristics, and determine how effective they are in finding the optimal solution to large-scale vehicle routing problems.

When the routing problem size is very large (e.g. number of deliveries is in the thousands), it is best dealt with by clustering to have multiple moderate-sized problems. A cluster-first-route-second heuristic by Özdamar and Demir (2012) divided demand clusters using the k-means partitioning heuristic. K demand nodes were randomly selected as cluster centroids, and the remaining nodes were assigned to the nearest centroid, recalculating cluster centroids iteratively. Another cluster-first approach proposed by Mitra (2008), was different from Özdamar and Demir (2012), where the cluster centroids were not randomly selected. The cluster centroids were identified based on distance. The customer located furthest from the depot would be in the first cluster. Subsequent clusters were assigned based on the furthest sum of distances from already allocated clusters. The k-means clustering algorithm generally uses Euclidean distance as a measure of 'closeness'. The Euclidean distance functions well only if the covariances of the clusters are generally homogeneous (Nelson, 2012). Replacing the Euclidean distance metric with a Mahalanobis distance metric can possibly improve the analysis of clusters, as the Mahalanobis distance takes into account the covariances between variables.

The most commonly known route construction heuristics was the Savings Heuristic by Clarke and Wright (1964), which many other heuristics were based upon. Heuristic of Lu and Dessouky (2006) was one of them. They first identified impossible links between customers to find the minimal number of vehicles. Subsequently, customers were allocated to vehicles based on a saving function which taken reduction in maximal postponed time into consideration. Visual attractiveness (no overlapping of routes) was also proposed as it possibly created trust amongst the operators. Pillac et al. (2012) proposed an optimization framework for dynamic and stochastic vehicle routing which was event- driven, parallelized and flexible. Dynamic and stochastic problems included partial stochastic knowledge on the dynamically revealed information. The methodology was parallelized to take advantage of modern multi-core and multi-threaded computing architectures. The paper presented the design and implementation of an object-oriented event-driven framework for the Multiple Scenario Approach (MSA). The optimization of scenarios was performed by an Adaptive Variable Neighborhood Search (AVNS) which improved an initial solution generated with a randomized Clarke and Wright heuristic. Santos et al. (2011) presented a problem in which demand occurs along the arcs, some arcs in the network might not require service (i.e., have no demand along them) and the vehicles had a capacity on the total demand that they could serve. The authors integrated optimization methodologies (e.g., heuristics and ant-colony meta-heuristics) for this multiple vehicle routing problem. These methodologies were adapted to satisfy several additional constraints of actual problems before their integration into the system. The developed solution was tested on a real-world multiple vehicle routing problem: trash collection in the City of Coimbra, Portugal. A two-stage heuristic was proposed by Qu and Bard (2012), where a greedy randomized adaptive search procedure (GRASP) constructor was followed by an adaptive large neighborhood search. This heuristic used multiple insertion/removal heuristics to shuffle customers between iterations. Furthermore, instead of a GRASP which was simply random, the proposed heuristic was adaptive where it changed the number of iterations allocated for diversification and intensification, based on past performance. Hu et al. (2013) developed a methodology to counter the challenges of unexpected events, such as customer's demand changes, delivery time window changes, disabled roads induced by traffic accidents or traffic jams, and vehicle breakdowns, where schedulers were needed to readjust vehicle routes in real time to improve vehicles' efficiency and enhance service quality. The research proposed a modeling approach named PAM (disruption-handling Policies, local search Algorithms and object-oriented Modeling) to handle disruptions in real-time vehicle routing problems. The modeling approach was combined with the scheduling knowledge (i.e. disruption-handling policies) of experienced schedulers with the optimization knowledge concerning algorithms and models, which could exert the advantages of them and eliminate their disadvantages. Mendoza, Medaglia, and Velasco (2009) proposed a decision support system (DSS) to help with integrating commercial systems with a custom-made distance-constrained routing module. This module included a modified Clarke and Wright savings heuristic and two memetic algorithms, along with two integer-programming clustering models whose function was to balance the workload. It was tested on ten realistic distance-constrained vehicle routing instances ranging around hundreds of nodes.

The VRP was proven to be NP-Hard (Cordeau, Laporte, Savelsbergh, & Vigo, 2007). This meant that as the number of customer increases, the amount of time taken to reach an optimal solution to the problem increased rapidly. Exact algorithms were only able to optimally solve problems with around 100 customers (Baldacci, Mingozzi, & Roberti, 2012), albeit with a high computational time. The company delivers to thousands of customers daily, and hence, exact approaches were considered largely ineffective. Due to this, a heuristic approach has been selected over exact approaches.

3. Problem Description

With regards to the company, a fleet of up to 200 identical vans travel on Singapore's road network. A cost, in terms of time, is incurred whenever a van traverses a road segment. The vans must fulfil a set of pre-determined customers by delivering packages to their respective destinations prior to their cut-off time. Deliveries are loaded onto the vehicles from the depot, whilst pickups are ad-hoc from varying customer locations. From the company's experience, van capacity is a non-binding constraint, while the time-deadline constraint is binding. Further analysis on the time available shows that it is the time available for deliveries that are binding, not pick-ups.

Currently, the company uses a cluster-first, route-second method to approach their daily scheduling problem. Their method of clustering is manual and decided based on historical data and personal experience of the planning person. Furthermore, the updating of the clusters occurs only occasionally. This results in a system that is backward-looking and tedious. Once the clusters have been decided, it means that specific addresses are allocated to specific vehicles. However, this fixed setting brings about situations where some vehicles may have very small number of deliveries. Figure 1 shows the trend of jobs and vehicles over the weeks in a month, and Table 1 provides some analysis on the utilization of the vehicles to fulfill the job orders.

Referring to Figure 1, visually, the number of vehicles dispatched looks relatively constant when compared to the number of daily jobs. From Table 1, it is noticeable that when there was a 32% decrease in number of jobs performed, there was only a corresponding 9% reduction in the number of vehicles dispatched. This ineffective method of resource allocation is an opportunity for improvement.

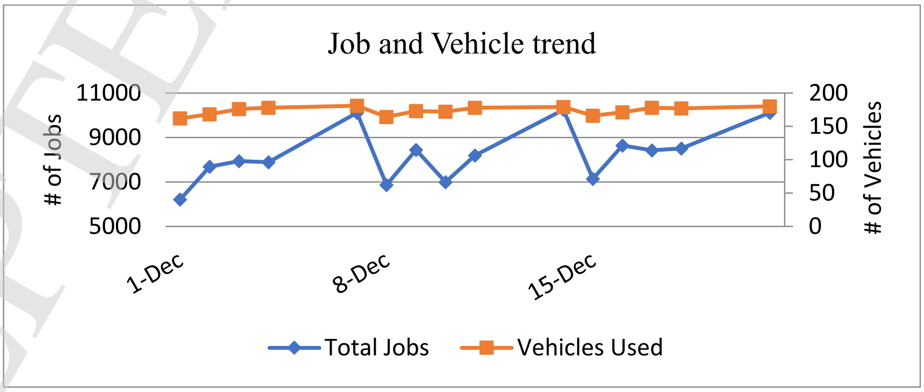

Fig. 1. Job/Vehicle Trend

Table 1. Unequal Variations in Jobs fulfilled & Vehicles Utilized

	Jobs	Vehicles

7th December	10113	181
8th December	6865	164
% Change from 7th to 8th Dec	↓ 32%	↓ 9%

4. Courier Experience based DSS

4.1 Driver Experience in DSS

Currently in the company, when the clustering decision is taken and is conveyed to the couriers, they negotiate amongst themselves prior to being dispatched. The outcome is either a balancing of workload or an increase in workload for some couriers which allows specific courier to remain at the depot to perform other tasks. When it comes to routing, couriers' experience could also influence the choice of the routes as the decision to take a specific route is left on the courier. The reason is that the experienced drivers know very well about the city and its traffic, and how to visit buildings by buildings faster, even better than the Google Maps. For example, courier can park at a central location and walk to the buildings in the vicinity. Couriers can also determine the routes based on some unpredictable information. For example, if there is construction work in progress on a certain route, they can choose the other one to reduce the travel time even though the new route is longer as per Google Maps. In short, the courier's experience is very important in the decision-making procedure as their decisions are more reflective of the real situations. The decision support system of the company should include the courier's experience to reduce their operating costs.

4.2 DSS for clustering and routing

Based on the arguments presented in the previous sections, a decision support system is presented in the subsequent text. Figure 2 is a diagrammatic representation of the proposed DSS.

The DSS consists of a user interface that helps the operator input the daily address postal codes for both the deliveries and pick-ups. This input contains the daily delivery information which includes: (1) the addresses of destinations, (2) their postal codes, (3) the quantity of parcels to be delivered, and (4) their respective deadlines. This information is used to identify the exact location of customers, compute their service times, and ensure customer satisfaction by meeting deadlines. The clusters and routes estimated by the DSS will also be communicated to the operator via this interface.

In addition to these inputs given dynamically by the operator, the DSS will also have access to historical data from the company and Google Maps traffic data at the beginning of the day.

Company historic data includes details about past deliveries, pick-ups and the timings. It refers to the sequences in which the company's couriers have travelled in the past and the accompanying time stamps at each destination. This data will also give a sense of the driver's choice with respect to a particular route which may not be in accordance to the route being suggested by Google. This data is utilized in two ways to make sure the data reflect the past situation and guide the heuristic. Firstly, the sequence is able to identify the road segments that are traversed frequently. A premium will be placed on these frequented paths whenever comparisons are made within the heuristic. This is because such data reflects the actual situation 'on the ground' such as traffic conditions or long-term construction activities. Further elaboration on how it affects the proposed heuristic have been made in Section 5.1 and 5.2. Secondly, the time stamps at each destination will be used in estimating the time between destinations. When the delivery origins and destinations are given every day, we can use linear functions derived from the historical data to estimate the delivery time for the daily routing implementation. This is further explained in Section 5.3. Currently the estimation is implemented on a static basis, in a long run, the rolling horizon estimation approach could be used.

Google maps is used to access data related to routes and traffic. This data is dynamic as the traffic routes change with the hour of the day. Map coordinates are a necessity in order to transform the information from postal codes into pinpoint locations, and subsequently visual representations on a map. These coordinates are then used to estimate the distances between destinations, and ultimately, the travel times. Geographic longitude and latitude coordinates are retrieved from Google Maps and are used as substitutes for the X and Y coordinates on a two-dimensional map. These coordinates are stored offline to reduce the computation time required by removing internet dependency.

These multiple inputs are used by the processor for clustering the jobs based on the delivery addresses and estimating the travel times for each of the delivery. These inputs are used by the processor to calculate the clusters and the delivery routes which are the output of the whole exercise. These decisions regarding the routes and clusters are distributed to the drivers to commence their daily routine.

The methodology proposed in this paper consists of two distinct stages. In stage one, clustering of the destination nodes is carried out while stage 2 concerns with routing of the vehicles to reach the assigned clusters. A flowchart of the heuristic is represented in Figure 3. In the clustering stage, both merging of the smaller clusters and splitting of the larger clusters is carried out iteratively in order to reach optimal sized clusters. In the routing stage, similar iterations are performed on the routes to arrive at the routes that will result in least travelling and hence cost savings.

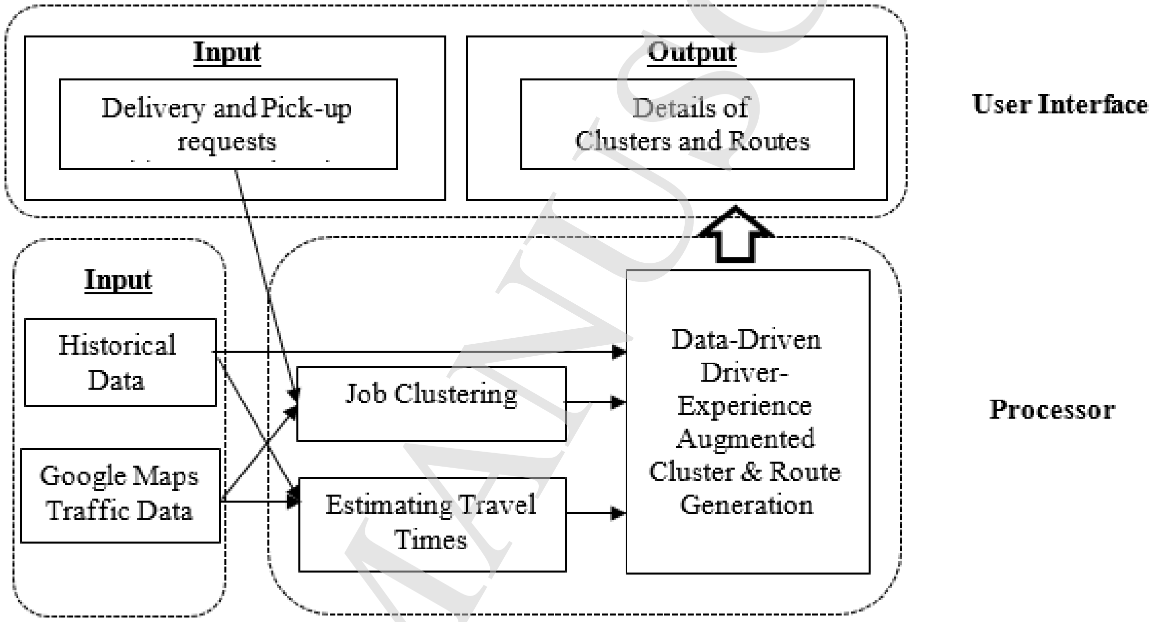

Fig. 2. Diagram of the Decision Support System (DSS)

To solve the company's problem, the proposed DSS will dynamically cluster and route path sequences for each vehicle to fulfil all customer demands. This will be done within the given constraints, whilst keeping costs minimal. The cost incurred from the number of vehicles dispatched (include vehicle equipment costs, vehicle maintenance costs, and manpower costs), possibly account for the largest share of operational costs. Hence, cost can be minimized by minimizing the number of vehicles. An example solution is shown in Figure 4, where couriers leave the company's depot for their clusters, perform their tasks, and return to the depot at the end of the day.

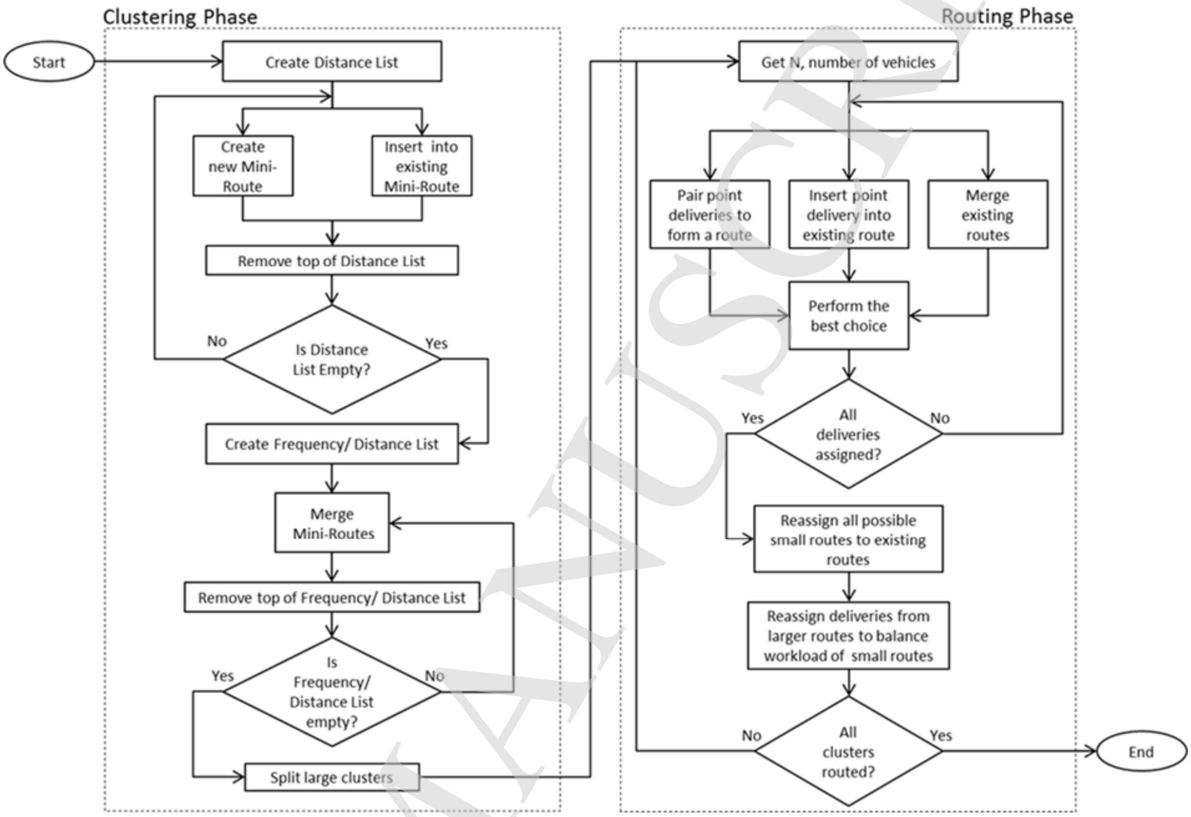

Fig.3. Overall Heuristic Flowchart

Fig. 4. Example Solution

5. Dynamic clustering and routing

5.1 Clustering

Currently postal codes are allocated to specific vans, meaning the number of vans used remains generally constant over the week as vans with fewer deliveries are also dispatched. This problem can be solved by clustering dynamically at the start of each day.

A novel dynamic clustering approach, Mini-Route Clustering, is proposed to minimize distances between destinations. The proposed heuristic also incorporates historical data from the company's couriers, and balances cluster workloads. The approach is named as 'Mini-Route Clustering' as shown in Figure 5 because it basically takes advantage of the closeness of the destinations, forming small-sized initial routes (mini-routes), and subsequently merges these mini-routes into clusters based on couriers' experiences, and closeness of mini-routes.

```
Algorithm 1: Mini-Route Clustering
  Input : Delivery requests, Google maps based traffic data (distance)
  Result: Job Clustering
  begin
    Form Descending Distance List between destinations;
    while Length of Distance List > 0 do
      if Existing mini route < 5 then
        Add a destination to an existing mini-route
      else
        Create a mini-route
      end
    end
    Create mini-routes pairs based on the frequency and distance;
    while Length of mini-routes List > 0 do
      if Cluster size < 100 then
        merges mini-routes based on the frequency and distance list
      else
        Create a new cluster
      end
    end
    if cluster size < 80 then
      Insert into another cluster
    end
    if cluster size >= 400 then
      Split into two using K-means algorithm, where K is 2
    end
  end
```

Fig. 5. Mini-route Clustering Algorithm

The detailed explanation of the Mini-route Clustering Algorithm is as follows. Firstly, generating a list of destination-destination pairs, and sorting them based on distances in descending order. Secondly, creating a mini-route from two individual destinations, or adding a destination to an existing mini-route with up to a maximum of size 5. A maximum size is necessary; else the mini-route could possibly stretch from one end of Singapore to the other. The terminating criterion is the exhaustion of the distance list. Thirdly, creating a list of mini-route pairs, including the frequency of travel between them, and the distance. If there is a frequented path between mini-routes, the distance computed is the distance of the frequented path; else, the distance is the Euclidean distance between mini-route centroids. Fourthly, merging mini-routes based on the frequency/distance list, which is sorted against frequency, and subsequently, distance. For example, mini-route 'A' has a distance of 1km to mini-route 'B' and a frequency of 5, while 'A' to 'C' is 0.7km with 0 frequencies. The selection will be such that a higher frequency is chosen first, because it indicates that the courier has encountered this situation before, and has chosen to take the longer path, in possibly a shorter time. A maximum initial size of 100 is imposed to ensure that no cluster is 'too large'. A 'too large' cluster size will result in a large computation time required for the routing heuristic. Then, inserting smaller clusters, sizes of less than 80, into another cluster to take advantage of the flexibility of routing. Larger cluster sizes may result in better routes, albeit at larger computation times. Lastly, all clusters are less than size of 400 by splitting large clusters larger or equal to 400 based on a K-means algorithm, where K is 2.

5.2 Routing

A greedy approach is taken during the construction of routes, and the algorithm is presented as Figure 6 and Figure 7.

Algorithm 2: Routing

Input : Historic Data, Google maps based traffic data (distance), Job Clustering
Result: Routing
begin
  while *Number of destination doesn't belong to any route* > 0 **do**
    Choose the shortest vehicle travelling time routes using <u>*Courier Experience based Selection*</u>
  end
  if *Small routes(size* < 10*) can be redistributed* **then**
    Redistribute small routes into other routes
  else
    Reallocates work from other routes to balance the workload
  end
end

Fig. 6. Routing Algorithm

Algorithm 3: Courier Experience based Selection

begin
  Join the cluster as a new route travelling time T_1;
  Insert individual destination to an exiting route travelling time T_2;
  Merge with existing routes travelling time T_3;
  Update the travel time using augmenting equation $T_i^{new} = (1 - \frac{frequency}{occurrence}) * T_i$;
  Choose the best route with $T = \min_{x \in i} T_x^{new}$
end

Fig. 7. Courier Experience based Selection Algorithm

To incorporate couriers' experience, this augmenting equation is proposed when comparing 'best' choices using historical data. The Courier Experience-based Selection selects the best choice based on shortest vehicle travelling time required, while adhering to delivery deadline constraints. The terminating criterion is that every destination is part of a route. The utilization of vehicles is increased by redistributing small routes into other routes and balance the workload between routes. In order to incorporate couriers' experience, an augmenting equation is proposed in Figure 7, where *Occurrence* is the number of times the destinations in comparison are present, and *frequency* is the number of times a path is chosen during the occurrences. For example, as illustrated in Figure 8, courier will choose to go from X to Y because the Google maps shows 5 minutes travel time as compared to 7 minutes travel to Z. However, due to long-term construction works, the historical data shows that the courier had chosen to go to Z from X 6 times out of 10. The final decision, based on the algorithm in Figure 7, will be to go from X to Z rather than going to Y. As seen from Figure 8, the original decision which was merely based on Google Maps has been changed with inputs from the driver's experience.

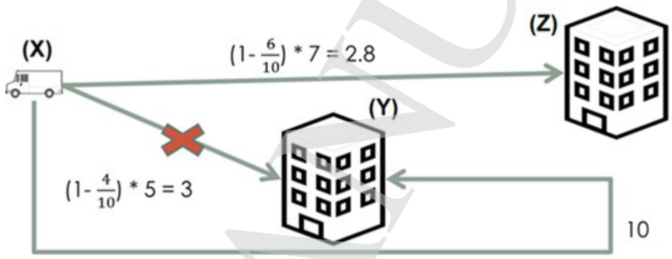

Fig. 8. Intelligent Routing Comparison Illustration

5.3 Processing (Estimate Travel Time)

An important component required in any VRP is the quantification of cost incurred when a route is taken. Commonly, the cost function is in terms of the distance between origins and destinations. However, since the company is more interested in customer satisfaction (which relates to deadlines - time) as opposed to distance travelled, the cost function will be in terms of time.

To set a baseline, it is assumed that the company's historical data of travelling duration is accurate to a certain extent. Due to the volume and tight deadlines of deliveries, mornings are the busiest periods of the day. This implies that the component of idle time is minimized, and hence, time stamps in the morning are most accurate. In view of this, the estimation of delivery times only utilized historical data collected in the morning.

The following equation is used to estimate time between deliveries in this heuristic:

$$Y(x) = V(x) + S(x) + \epsilon \quad (1)$$

where x represents delivery origin and destination pair. Y(x) is the time between delivery origin and destination which includes: V(x), the vehicle travelling time between delivery origin and destination, S(x), the total service time at destination building, and ϵ, the residual component. Y(x) is broken into these components because information regarding travel distance and the number of delivery parcels can be used to estimate V(x) and S(x) individually, providing more accurate results.

5.3.1 Vehicle Travelling Time

Vehicle travelling time is categorized into travelling time between depot & cluster and travelling time between job destinations. This is done because of the usage of expressways for long distances. Hence, the time estimation models for long distances (depot to cluster) should be different from short distances (between job destinations).

Using Company's historical data of the time vehicles leave the depot, as well as the time they reach their first destination, a scatterplot of time taken against distance travelled is shown in Figure 9. For travelling time to clusters, the distance travelled is the Manhattan Distance between Company's depot and the cluster centroid. The estimation model derived is: $y = 1.2515x + 14.831$, $R^2 = 0.8764$.

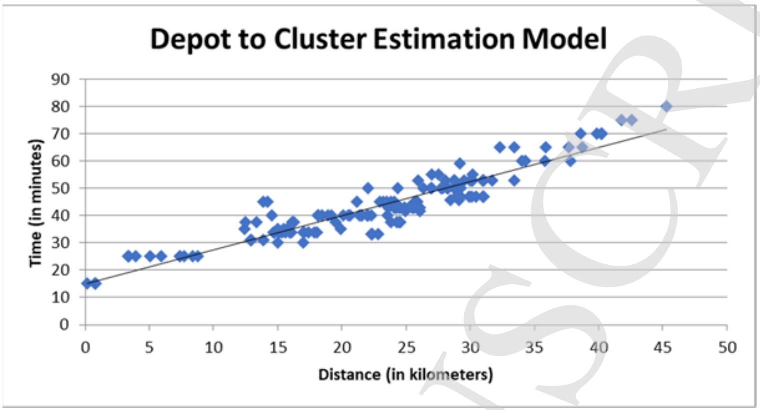

Fig. 9. Depot to Cluster Estimation Model

For traveling times between job destinations, no discernible source of data was available from the Company. This is because the time stamps of destinations in the historical data inherently include all components of equation (1). Hence, another approach was taken. A random sample of 300 origins and destinations was taken from the historical data, and their Euclidean distance was computed from Google coordinates. Subsequently, the travel time was retrieved from Google maps, and its scatter plot is shown in Figure 10. Based on the data, the estimation model is: $y = 0.0027x + 1.473$, $R^2 = 0.5648$.

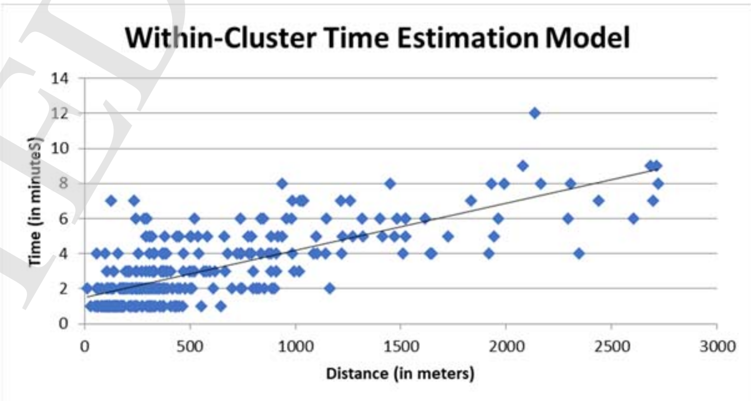

Fig. 10. Within-cluster Time Estimation Model

For consistency, traffic conditions were excluded when retrieving travel durations from Google Maps. This can be used as a baseline when creating the clusters and routes at headquarters. However, it would probably be better for the couriers to re-route their deliveries 'on-the-go', with real-time information from Google Maps, when they are negotiating various traffic conditions of the day.

5.3.2 Service Time per Building

Instead of the intuitive service time per customer, we will be using service time per building, which is represented by the sum of all individual service times in a postal code. This is because there is no way to identify the distance traversed by foot, and it is also difficult to identify the individual destinations within a single postal code.

By selecting historical data that had multiple deliveries within the same postal code, we are able to isolate S(x) from equation (1). With this data, the average time taken to deliver certain number of parcels was plotted against number of parcels in Figure 11. The estimation model derived is: $y = 1.0513x + 0.9891$, $R^2 = 0.8676$.

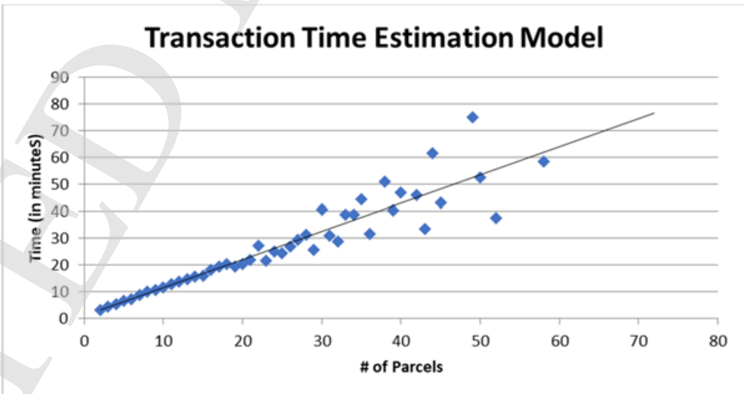

Fig.11. Transaction Time Estimation Model

This estimation makes sense because as the number of parcels increase, it is likely that the courier has to either make multiple stops within the building, and/or make multiple trips from the van to the customer.

5.3.3 Residual Component

The residual component is calculated by deducting the sample vehicle travelling time between delivery origin and destination $\hat{V}(x)$and the sample total service time at destination building $\hat{S}(x)$ from time between deliveries Y(x), and the histogram is shown in Figure 12.

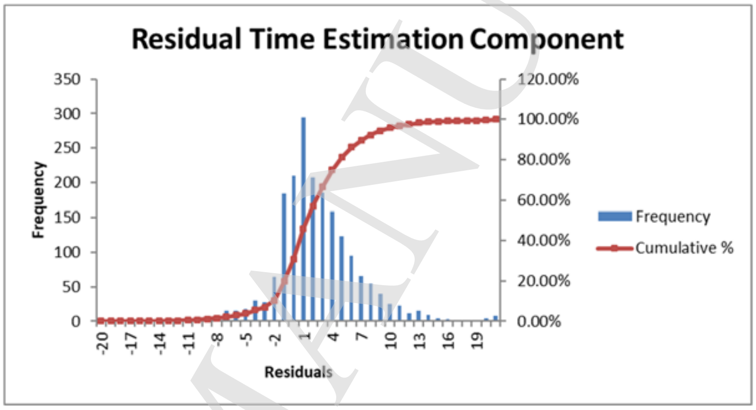

Fig.12. Residual Time Estimation Component

As expected of any residuals plot, the histogram looks somewhat normal. The normal probability plot and Chi-square goodness of fit test were conducted to ascertain the residual's normality.

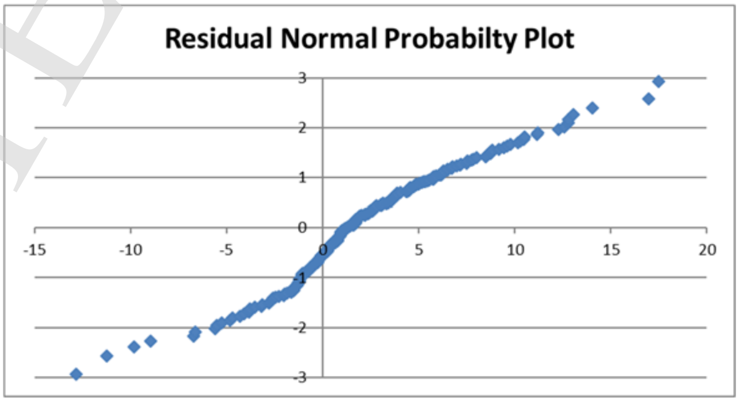

Fig. 13. Residual Normal Probability Plot

The normal probability plot in Figure 13 looks close to a straight line, indicating that indeed the residuals are normally distributed. This is further confirmed by the statistical Chi-square test with hypothesis, and the $X^2 = 64.4 < X^2_{100,0.05} = 124.3$ indicates that we do not reject the null hypothesis that residuals are normally distributed.

An example of positive residuals is a small number of parcels taking a long time to deliver (large $Y(x)$, small $\hat{S}(x)$). This could be due to complications 'on the ground', such as an unavailable receptionist. Negative residuals could be because of a large number of parcels delivered to a single customer (small $Y(x)$, large $\hat{S}(x)$).

To ensure repeatability of the heuristic, the first moment of the distribution is used to characterize the residuals and this average is 2 minutes. This positive representation of residuals can be conceptualized as the time taken by the courier to plan the route to his next destination, delays due to customer, or even some traffic delays not captured by vehicle travelling time.

6. Experiment and Results

This section has the following three parts: firstly, a comparison between the proposed clustering heuristic and the traditional K-means approach; secondly, a numerical study to ensure that the heuristic can be completed within the allocated amount of time; thirdly, a comparison to test the efficacy of the heuristic against the actual historical data of the company.

6.1 Comparison with Traditional K-Means Approach

By varying the clustering method (proposed heuristic and traditional K-means), and keeping the routing method constant, Figure 14 shows the number of vehicles required to serve the deliveries that the company had performed over the three weeks of study.

Visually, it seems indiscernible as to which clustering produces a better practical result. Hence, a paired t-test calculation was done:

$$H_0: \mu_{Heuristic} - \mu_{Kmeans} = 0$$

$$H_1: \mu_{Heuristic} - \mu_{Kmeans} \neq 0$$

$$t = 0.139, t_{14,0.05} = 2.145$$

$$Since\ -t_{crit} < t < t_{crit}\ , we\ do\ not\ reject\ H_0$$

Since the t-statistic is within the limits, we conclude at a 95% confidence level that there is no difference between the result of the heuristic and the traditional K-means.

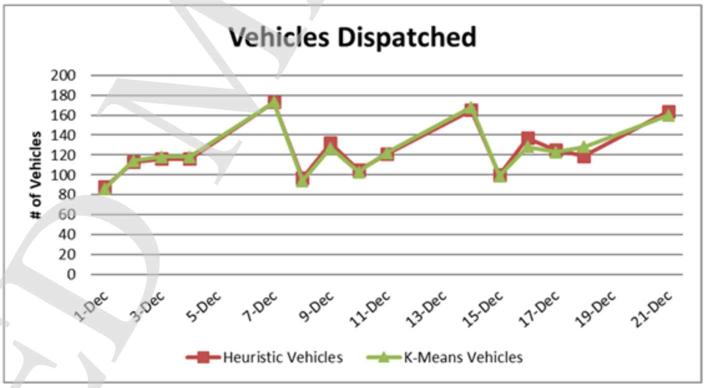

Fig. 14. Vehicles Dispatched (Heuristics vs K-Means)

However, there are some discernible advantages of the proposed heuristic over the traditional K-means clustering and these are enumerated in the following text.

1. The traditional K-means minimizes the distances between points in a cluster and their respective cluster centroids. This generally means that clusters will be circular in nature, ignoring covariances. The proposed clustering heuristic tries to minimize the distances between the points themselves. Intuitively, this has more meaning, as

a courier would rather have the distance he has to travel to the next point minimized, as opposed to the distance from the centroid which is pretty irrelevant. Distance from a point to the cluster centroid has no physical applicability.

2. How do we choose K (number of clusters) in the K-means approach? How do we then approach to initialize these K points? This in itself is a problem, which highlights the large computational time of the K-means approach. The K-means can be exactly solved in time $O\ (n^{(dk+1)} * log\ n)$, where n is the number of points, k the number of clusters, and d is the number of dimensions (Inaba, Katoh, & Imai, 1994). This does not include the time taken for selection of k (k could range from 1 to n). The proposed heuristic has a complexity of $O(n^2)$, which is always smaller than $O\ (n^{(dk+1)} * log\ n)$.
3. The proposed heuristic further takes into account historical data provided by the company. This historical data allows for the heuristic to capture information, such as traffic and road conditions, based on travel frequencies.

It seems that although the results are similar, the heuristic should be chosen over the K-means in this situation because of the advantages it has over the K-means approach: (1) better physical representations of distance, (2) faster computational times, and (3) incorporation of historical data.

6.2 Numerical Study on Computation Durations

The company's operations team would like to keep computational times to within an hour, so as to facilitate their dissemination of information to couriers and provide for some additional buffer time. This next section shows the two types of times required in the algorithm: (1) Time for clustering, and (2) time for routing in Figures 15 and 16 respectively.

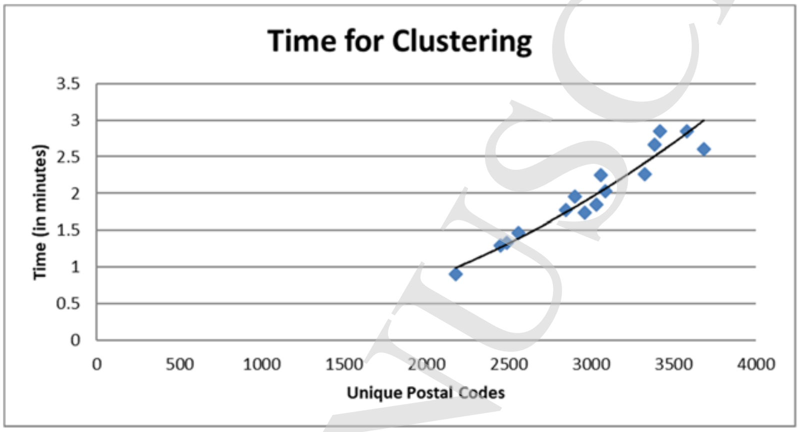

Fig.15. Time for Clustering

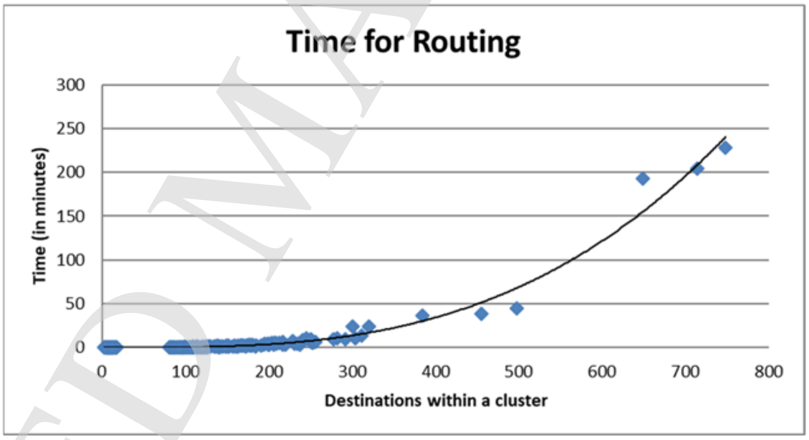

Fig.16. Time for Routing

From both figures it can be deduced that as the number of deliveries increase, the time required to cluster and route them increases rapidly. This is because of the increasing number of permutations available as the number of deliveries grows. For clustering, there is nothing much to be done, as the number of deliveries is an input based on customer demand. The number of deliveries in each cluster, however, is something that the algorithm is able to control. By restricting a cluster from growing beyond a certain size, we can ensure that the time taken to route them stays within a certain range. By maintaining a maximum cluster size of 400, all the experiments in Section 6.1, as well as in 6.2, managed to keep total computational durations below 1 hour, thus, adhering to the Company's requirements.

6.3 Comparison with the Company

For this section, two experiments were performed to test the efficacy of the proposed heuristics. The experiments are illustrated in Table 2. The first experiment keeps the company's clustering method constant, while varying the routing method, to compare routing techniques. Subsequently, the new routing method is kept constant, while changing the clustering method, to compare clustering techniques.

Table 2. Experiment Design

Experiment	Number	Clustering Strategy	Routing Strategy
Routing Method Comparison	1	The Company Clustering Heuristics	The Company Routing Heuristics
	2	The Company Clustering Heuristics	New Routing Heuristics
Clustering Method Comparison	1	The Company Clustering Heuristics	New Routing Heuristics
	2	New Clustering Heuristics	New Routing Heuristics

6.3.1 *Routing Methods Comparing*

From the company's historical data, we are able to tell the sequences which their couriers chose. This sequence includes pick-up stops, and in order to provide a fair comparison, these pick-up stops were removed. Subsequently, using the time estimation model from Section 5.3, we are able to identify the time required by the company's clustering and routing methods to complete all deliveries (denoted by 'Actual' in Figures 17 and 18).

Using the company's clusters, and the proposed routing heuristic, the time required to complete all deliveries are also shown in Figures 17 and 18, for commercial and residential deliveries respectively (denoted by 'Heuristic').

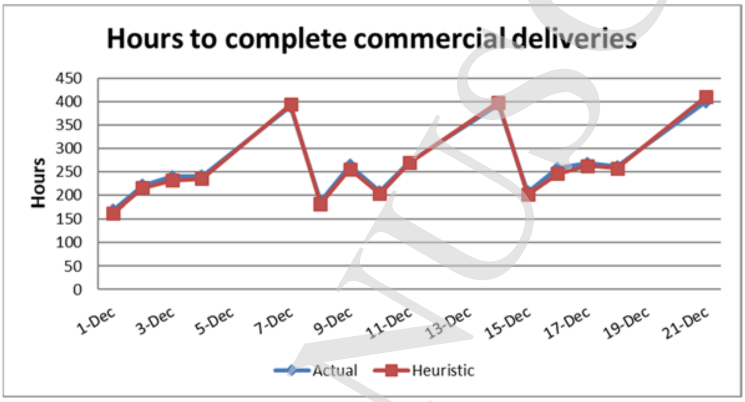

Fig. 17. Hours to Complete Commercial Deliveries

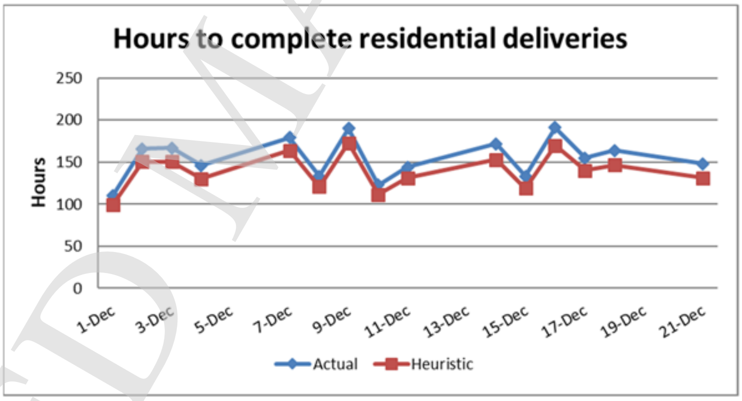

Fig. 18. Hours to Complete Residential Deliveries

The average reduction in time is 1 min per vehicle for commercial deliveries, and 28 min for residential deliveries. Commercial cluster sizes are relatively small (~12 deliveries), while residential clusters are much larger (~40 deliveries). This means that Company's method of routing, intuition and personal experience is able to handle small-sized routes well but fails to perform as route size increases. Therefore, the proposed routing heuristic is at par with Company's method at the minimum, and it improves as route size increases.

6.3.2 Clustering Methods Comparing

This next experiment compares the company's clustering method with the proposed clustering heuristic, while keeping the routing method constant. The performance measure used to compare the two methods is the number of vehicles dispatched to complete all deliveries, as opposed to time required to complete all deliveries. This is because as the number of clusters increase, the time required will decrease as vehicle travelling time per vehicle reduces.

These assumptions are also included in the experiment: commercial and residential deadlines are 12pm and 6pm respectively and there is a 1-hour lunch break in the afternoon. The results are shown in Figure 19.

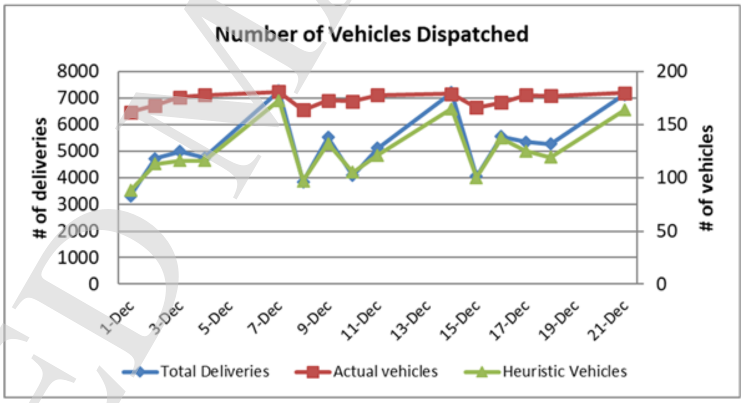

Fig. 19. Number of Vehicles Dispatched (Heuristics vs Actual)

The number of vehicles dispatched by the heuristic clearly follows the trend of total deliveries per day, and this reduces the number of vehicles on Tuesdays-Fridays significantly. This is because the clustering heuristic does not allocate specific address to vehicles, and the routing heuristic allows for tightly-packed schedules. However impressive the results, this experiment only accounts for deliveries, and not pick-ups, and hence, a conclusion cannot be reached yet.

6.3.3 Overall Results

After accounting for pick-up jobs, the resulting number of vehicles is shown in Figure 20.

The tangible benefits for the company, if they use this proposed heuristic is that on a daily average, they have to dispatch 15 fewer vehicles. Assuming it costs $1,000 a month for vehicle operations and maintenance, and another $1,000 for manpower, the estimated annual savings from this could be $360,000.

The other non-tangible benefits include increased customer satisfaction and the removal of couriers' 'gut feel' in routing. Customer satisfaction increases due to the deadline-constraints imposed in the routing heuristic. This ensures that deadlines will always be met, provided no serious incidents occur, such as vehicle breakdown, unexpected traffic jams or a natural disaster.

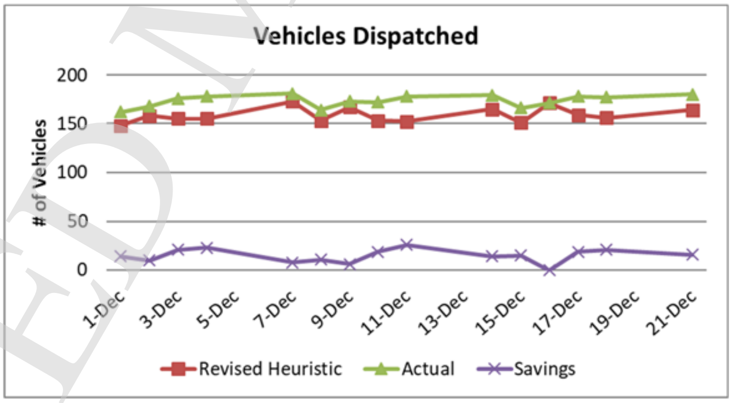

Fig. 20. Actual Vehicles vs Revised Heuristics Vehicles

7. Conclusion and Future Research

The current method of clustering and routing deliveries by the company is manual and caters mainly to peak workloads on Mondays. It does not change as workload varies over the week. As the workload will decrease from the peak load, the company will end up employing greater than optimal number of resources in the course of the week, leading to an avoidable loss of money. The DSS presented as a part of this research can be an important optimizing tool for the company. The proposed heuristic can help by dynamically clustering and routing, so as to ensure that the number of vehicles dispatched follows the workload trend. On top of this, the machine-learning capability of the heuristic is able to learn from couriers' experiences, in order to recommend more suitable paths. In effect, the company representative who is planning the morning dispatch of the vehicles for pick-ups and delivery has to simply load a file containing the scheduled target destinations known to him at that point of time. The DSS will make use of this data, historic company data and Google Maps data in the heuristic algorithm to arrive at different clusters. Each of the cluster is then serviced by one of the company vehicles. In addition to this, the DSS will also generate the routes for each of the vehicle. The DSS ensures that the planning of vehicle and route assignment for the day is done dynamically, based on the current data on each day. In doing so, an average daily reduction of 15 dispatched vehicles is possible (annual savings of $360,000), and the shorter delivery time will reduce the customers' waiting time and the deadline-constraints in the routing approach will ensure that every parcel is delivered to increase customer satisfaction. It is recommended that the company departs from its traditional intuitional and 'gut feel' ways of operations, and adopts the systematic approach proposed by our heuristic.

Despite the fact that the proposed methodology has been developed specifically for a package delivery company, the underlying heuristic can be used alternatively for other logistic delivery operations. Logistic delivery companies often deal with slightly different variant of the VRP in terms of time windows. The product variety may also differ in size, shape and shelf life. These deviations from the described problem can easily modelled in the proposed DSS, thereby demonstrating the versatility of the method. The developed DSS can be used by the other companies for optimizing their resources while ensuring timely delivery of services.

Although the benefits of this proposed heuristic are high, there are some limitations which can be addressed in further studies. First, the routing heuristic is greedy, with only a construction phase. As stated by Hosny & Mumford (2012), many good solutions could get neglected. Therefore, an improvement/feedback phase is necessary to provide better routing sequences. Secondly, from the couriers' perspective, they will need to be familiar with a larger work-area than previously, as they may be assigned to any route within the cluster. In future works, couriers' preferences on clusters and routes could be taken into consideration. Thirdly, we only use Google Maps for static information. It would be interesting to further investigate the implementation of real-time routing to include traffic conditions.

Acknowledgments

This study was partially funded by National University of Singapore, Center for Next Generation Logistics. The authors appreciate Mr. Samuel Lee and Dr. Pankaj Sharma for their assistances in this study.

Reference

Baldacci, R., Mingozzi, A., & Roberti, R. (2012). Recent exact algorithms for solving the vehicle routing problem under capacity and time window constraints. *European Journal of Operational Research,* 218(1), 1-6.

Clarke, G., & Wright, J. W. (1964). Scheduling of vehicles from a central depot to a number of delivery points. *Operations research,* 12(4), 568-581.

Cordeau, J.-F., Laporte, G., Savelsbergh, M. W., & Vigo, D. (2007). Vehicle routing. *Handbooks in operations research and management science,* 14, 367-428.

Hosny, M. I., & Mumford, C. L. (2012). Constructing initial solutions for the multiple vehicle pickup and delivery problem with time windows. *Journal of King Saud University-Computer and Information Sciences,* 24(1), 59-69.

Hu, X., Sun, L., & Liu, L. (2013). A PAM approach to handling disruptions in real-time vehicle routing problems. *Decision Support Systems,* 54(3), 1380-1393.

Inaba, M., Katoh, N., & Imai, H. (1994). *Applications of weighted Voronoi diagrams and randomization to variance-based k-clustering.* Paper presented at the Proceedings of the tenth annual symposium on Computational geometry.

Lu, Q., & Dessouky, M. M. (2006). A new insertion-based construction heuristic for solving the pickup and delivery problem with time windows. *European Journal of Operational Research,* 175(2), 672-687.

Mendoza, J. E., Medaglia, A. L., & Velasco, N. (2009). An evolutionary-based decision support system for vehicle routing: The case of a public utility. *Decision Support Systems,* 46(3), 730-742.

Mitra, S. (2008). A parallel clustering technique for the vehicle routing problem with split deliveries and pickups. *Journal of the operational Research Society*, 59(11), 1532-1546.

Nelson, J. D. (2012). *On K-Means clustering using Mahalanobis distance.* North Dakota State University.

Özdamar, L., & Demir, O. (2012). A hierarchical clustering and routing procedure for large scale disaster relief logistics planning. *Transportation Research Part E: Logistics and Transportation Review,* 48(3), 591-602.

Pillac, V., Guéret, C., & Medaglia, A. L. (2012). An event-driven optimization framework for dynamic vehicle routing. *Decision Support Systems*, 54(1), 414-423.

Qu, Y., & Bard, J. F. (2012). A GRASP with adaptive large neighborhood search for pickup and delivery problems with transshipment. *Computers & Operations Research,* 39(10), 2439-2456.

Santos, L., Coutinho-Rodrigues, J., & Antunes, C. H. (2011). A web spatial decision support system for vehicle routing using Google Maps. *Decision Support Systems,* 51(1), 1-9.

Talbi, E.-G. (2009). Metaheuristics: from design to implementation (Vol. 74): John Wiley & Sons.