IOE 591 Project: Vehicle Routing Problem applied to grocery delivery at Ann Arbor

Krishna Rao, Siddarth Maheshwari

Optimization in vehicle routing is a major focus to reduce variable costs for any industry. Now, even small businesses have realized that their future success may depend on the location-distribution success and making it as efficient as possible while making quick and reliable decisions. In this project we aim to understand how multiple alternatives vehicle routing problem (VRP) can be solved quickly and optimally, and help decision makers in choosing a good alternative of VRP for practical implementation.

For this purpose we use the city of Ann Arbor with Kroger stores as the depot for our vehicles, and based on the residential areas we distribute the customers throughout the region. The goal is to determine the optimal route from the depots to each customer location which generates the minimum travel cost. Two approaches have been discussed here, one with vehicles being allocated through bin-packing applied to different items, and the other by assuming all item demands into a single overall demand. The problem is then divided into three parts - Single-depot VRP, Multi-depot VRP and heuristic approach through k-Means clustering - to understand trade-offs between alternatives. Through our results we try to highlight what factors come into play in achieving a good objective value in each alternative and the choice of best performing model based on CPU runtime and achieved objective value.

Key words: Bin-packing, Vehicle Routing Problem (VRP), k-Means clustering, Heuristics

1. Introduction

Grocery e-commerce sales are forecast to account for 9.5% of total U.S. grocery sales of \$1.097 trillion in 2021, up from 8.1% of \$1.137 trillion in 2020. It is further forecasted that the sales will grow upto 20.5% of \$1.285 trillion in 2026 (Mercatus Centre 2021). Companies like Instacart, Walmart and Amazon Fresh command a major share in the market in the segment. However, in order to ensure an economically feasible model and in order to retain the market share by charging the customer a minimum amount for the delivery, it is essential to come up with a delivery model that provides the quickest and cheapest service.

Transportation is generally supplied by a fleet of vehicles based out of a single (or sometimes multiple) depot(s). The aim is to plan a set of minimum cost vehicle routes capable of accommodating as many requests as possible, under a set of possible constraints some of which can be identified as multi-depot vehicle routing (Renaud et al.1996), delivery with time windows (Braysy and Gendreau, 2005a) or inter-depot replenishment (Crevier et al., 2007) . However before proceeding with the vehicle routing, it is necessary to determine the optimum number of vehicles required to cater the demand. This can be determined by Bin Packing or the Cutting stock problem . Here, given the depots have multiple bin/vehicles available, the aim is to determine the minimum number of bins/vehicles needed given we have multiple units of different items, each item having some weight w_i, b_i units of item i, and bin capacity (Q) (Gimore and Gomory,1961; Barnhart, Cynthia, et al. 1998).

1.1. Focus on this project

In this project we use the location data of four Kroger stores in the Ann Arbor area and based on the residential areas the customer nodes have been distributed throughout the city. Random demand for each item has been assigned to each node. Here we assume that at the start of the day we have the demand for all customers. We first start by determining the number of bins/vehicles to be used by two approaches (i) by solving the Integer Programming formulation directly (ii) using column generation . Once the number of vehicles have been determined we use this data to find the optimum vehicle routes for catering to all the customers considering (i)single-depot (ii) multi-depot. Single depot capacitated vehicle routing problem and multi-depot vehicle routing problem is further solved using two approaches. The first approach being, solving the linear programming problem directly and in the second approach in order to speed up the computation we use a heuristic approach based on K-means clustering.Further we compile the model of bin packing included in the vehicle routing formulation itself and then the results for the respective approaches are then compared.

1.2. Organization of this project

The remainder of this project is organized as follows.Section 2 reviews the most relevant literature on bin-packing problem, vehicle routing problem and clustering. Section 3 covers the model formulation . Section 4 covers the data used. Section 5 covers the computational results of the models. Section 6 concludes the paper and states future research directions.

2. Literature review

We focus on the literature available for solving the Bin Packing and the vehicle routing problem. We have reviewed the papers most relevant to our models. The bin-packing problem belongs to the set of NP hard problems and it is unlikely to solve the problem optimally in polynomial time. In order to solve the problem optimally (Gimore and Gomory,1961) first introduced the column generation approach for solving bin-packing where instead of looking at a large set of columns that will improve the solution, it is optimal to solve the problem by finding a column and solving it using auxiliary knapsack problem whereas the model proposed by (Valerio ´ de Carvalho, 2003) uses a branch-and-price procedure that combines deferred variable generation and branch-and bound. We solve the bin-packing problem through normal IP formulation and the other approach we use is through a combination of column generation and solving the auxiliary knapsack problem, which has been taught as well in class for course Advanced optimization methods. In the VRP problem, a set of identical or heterogeneous vehicles, based at a central depot, is to be optimally routed to supply customers with known demands subject to vehicle capacity constraints. We refer to (Laporte 1992) and (Baldacci, Battara and Vigo,2007) for the modeling of the VRP problem through multiple approaches that are through normal IP formulation, through two-commodity network flow approach. In order to solve the model in a much shorter time we use a heuristic approach of K-means clustering (AK Jain 2009) through clustering we find a partition such that the squared error between the empirical mean of a cluster and the points in the cluster is minimized

3. Problem Description and Modeling

In the absence of real time data, we have mapped the locations of the four Kroger stores in Ann Arbor area and considered them as the depot denoted by D. Based on the population density then the customers, denoted by C have been distributed throughout the region and have been plotted as nodes. (Refer Fig.1).

We then consider an online based ordering model where each customer places an order for a certain number of items denoted by i in multiple quantities, denoted by b_i. All the orders are received centrally and a cutoff time for receiving the orders is taken to be the prior day before the distribution. For simplifying the bin-packing model, we consider the dimension of the product to be only in 1-D, for our project we have taken the weight of the product and denoted it by q_i.

In the first case we consider only one depot and all the orders being centrally received at the depot. We then use bin-packing by (i) Direct IP formulation (ii) Column Generation to determine the number of vehicles to be used for the distribution where each vehicle has the same capacity and is denoted by Q. Once the number of bins has been determined we use this input for the Capacitated vehicle routing problem as the number of vehicles, denoted by (K).

In order to determine the route that minimizes cost we start with the single-depot capacitated vehicle routing problem. Here we assume the node $n+1$ corresponds to one of the four Kroger locations as the depot(D) and customers(C) are represented from node 1 to node n. Whereas all the nodes that are depot and customers can then be listed as a set of nodes denoted by (V). We consider the cost for unit distance travelled as \$1. Therefore the cost element cst_{ij} for arc (i, j) can then be considered as the distance between any nodes i and j. We make the assumption that each node can be visited by only one vehicle and every vehicle returns to the depot after completing the deliveries. The solver will then give an output which determines the most optimum route. However due to the high run time of the model we end the kernel after 3600s and plot the optimal route obtained.

We then complicate the problem by moving from single-depot vehicle routing problem to multi-depot vehicle routing problem. Through bin-packing we are able to determine the number of vehicles to be used and a set of vehicles(K) are allocated to each depot. Here the set of customers are denoted by C and depots by set D, whereas the set of all nodes is denoted by V. Here we assume that demand at each depot is zero, each customer will be catered by only one vehicle, each vehicle will return to the same depot it started from, the vehicle cannot travel from one depot to another, and all the depots are having sufficient stock of inventory of each item to be catered to all the customers. The solver will then give an output which determines the most optimum route. Due to the high run time of the model we end the kernel after 3600s and 7200 s and plot the optimal route obtained.

In order to obtain an optimal solution using a smaller run time and a scalable method we use a heuristic approach based on K-means clustering. To overcome the limitation of long solving times, a pre-processing stage is performed to cluster nodes together which gives a more compact cluster-based Mixed Integer Linear Programming (MILP) problem formulation, since all the customer nodes are now allocated to respective depots. Once the customers have been allocated to respective customers the MILP formulation only needs to determine the most optimum route for delivery to customers. We then proceed onto combine VRP with bin packing in a single formulation to compare the results obtained from the approaches.

3.1. Direct IP Formulation

We formulate a model to determine the number of bins needed where y_j a binary variable determines whether a bin j is used or not. The variable z_{ij} determines the allocation of item i to bin j. An upper limit is set on the maximum number of bins that can be used as T, which is determined by considering that a unit item is allocated to each bin.

Objective Function

$$Z = Min \sum_{j=1} y_j$$

subject to

$$\sum_{j=1}^{T} z_{ij} = b_i \quad \forall i \in 1, 2, 3...m \tag{1}$$

$$\sum_{i=1}^{m} q_i z_{ij} \leq Q.y_j \quad \forall i \in 1, 2, 3...m \tag{2}$$

$$y_j \in \{0, 1\} \quad \forall j \in 1, 2, 3...T \tag{3}$$

$$z_{ij} : Integer \quad \forall j \in 1, 2, 3...T \quad \forall i \in 1, 2, 3...m \tag{4}$$

$$T = \sum_{i=1}^{n} b_i \quad \forall i = 1, 2, 3....n \tag{5}$$

In the direct IP formulation our objective is to minimize the number of vehicles/bins needed to transport all items. Constraint (1) ensures that all the items have been included in the bin. The total weight of all items in the bin cannot exceed the capacity Q of the bin which is ensured by constraint (2). The solution obtained will give an optimal solution on the number of bins needed to cater to all the customer demand , however the solution obtained results in large running time even for small instances. In order to obtain an optimal solution using a smaller running time we use a more scalable approach of solving the cutting stock problem using column generation.

3.2. Cutting Stock problem via Column Generation

Before defining the objective function for the problem we define **Pattern** here the j^{th} pattern is given by a vector $a_{1j}, a_{2j}, ..., a_{mj}$ which means that the bin contains a_{ij} units of item i (for items ranging from i = 1,..., m). In order for the pattern to be feasible, we require

$$\sum_{i=1}^{m} q_i a_{ij} \leq Q$$

In our model, d_j can be defined as the number of bins filled using pattern j.

Objective Function

$$Z = Min \sum_{j=1}^{n} d_j$$

subject to

$$\sum_{j=1}^{n} a_{ij}.d_j = b_i \qquad \forall i \in 1,2,3...m \tag{6}$$

$$x_j \geq 0 \;\; \forall \;\; j = 1,2,3....n \tag{7}$$

The objective of the above problem is to minimise the patterns. Constraint (6) ensures that the solution obtained has covered all the quantities needed to satisfy the customer demand.

We now focus on solving the above integer program with LP relaxation. In order to solve the LP we use the reduced cost to solve the matrix. The reduced cost of variable j can be defined as, $\overline{c}_j = c_j - c_B A_B^{-1} A_j$. We then can define, $p = c_B A_B^{-1}$ which is a m dimensional row vector. $\overline{c}_j$ can be rewritten as $\overline{c}_j = c_j - pA_j = c_j - \sum_{i=1}^{m} p_i.a_{ij}$.

Given profit p_i the reduced cost oracle can help us determine either that the there is no pattern which gives a negative reduced cost or it give the output of a column with negative reduced cost.

We can now formulate the oracle as an instance of knapsack problem with the following parameters.

1. Weight = q_i /unit
2. Profit =p_i /unit
3. No of items = m

4. Capacity = Q

Knapsack formulation can be explained as follows:

1. Stages = 1,2,3,4......m
2. State(t) = Total weight of first i items $t = 0, 1, 2, 3....Q$
3. Value Function($V_i t$)=Maximum total profit from i items, subjected to weight limit t
4. Optimal_knapsack =$V_m Q$
5. Initialization : $(V_1 t)$ = round down($\lfloor t/q_1 \rfloor$)P_1
6. Bellman Equation: In order to compute $(V_i t)$ from $(V_{i-1} t)$ we consider choosing e units of item i and the weight of the remaining items is $t - q_i e$ and the maximum profit from these items is $(V_i t)(t - q_i e)$.
7. Therefore :

$$(V_i(t)) = max[ep_i + v_{i-1}(t - e.q_i)]$$

$$0 \leq e \leq [t/q_i]$$

In order to find the optimal solution (and not just its value), we use backtracking. This uses selection of items and is done as follows. We just check whether $Vn(Q) > 1$. If yes, then the optimal solution to this knapsack instance is the pattern with negative reduced cost. If not, then there is no pattern with negative reduced cost. The algorithm can be summarized as below:

Algorithm for Column Generation for Bin Packing Problem

- We obtain a basic feasible solution of m patterns where each pattern contains $\lfloor L/q_i \rfloor$ units of item i and zero units of all other items.
- Initialize patterns $J \leftarrow${1, 2, ···m}.
- for $k = m+1, m+2,$ do
-     Solve LP to the patterns in J, i.e.
-     Let B denote the basis corresponding to the optimal LP solution.
-     Set profits p = (1, ···1) $A_B^{-}1$
-     Solve the knapsack instance with each item i having weight w_i and profit p_i.
-     let $(y_1^*, ...y_m^*)$ be the optimal solution (pattern) to the knapsack instance.
-     If $\sum_{i=1}^{m} p_i * y_i \leq 1$ then
-         Exit for-loop.

- else
- Define pattern k to be $(y_1^*...y_m^*)$.
- Update patterns $J \leftarrow J \cup k$.
- end if
- end for
- output the last LP solution found.

Algorithm for Dynamic program for knapsack with multiplicities

- set $V_1(t) = p_1.\lfloor t/q_1 \rfloor$ and $select_1(t) = \lfloor t/q_1 \rfloor$, for $0 \geq t \geq L$.
- for i = 2, ··· , m do
- for t = 0, 1, ··· , L do
- set Vi(t) $\leftarrow -\infty$.
- for e = 0, 1, ···$\lfloor t/q_1 \rfloor$ do
- set u to t - e $\cdot q_i$.
- if $V_i(t) < V_{i-1}(u) + ep_i$ then
- set $Vi(t) \leftarrow V_{i-1}(u)+$ ep_i
- $select_i(u) \leftarrow e$, Select e units of item i
- end if
- end for
- end for
- end for

Using backtracking we then obtain the most profitable pattern. Having now calculated the number of vehicles needed we move on to the next stage of the problem, the vehicle routing problem.

3.3. Single-Depot capacitated vehicle routing problem

We formulate a problem to determine the route of that each vehicle should take before returning to the same depot.In our formulation x_{ij} is a binary variable which determines whether the arc (i, j) is a part of the optimum route found.

Objective Function

$$Z = Min \sum_{ij \in A} cst_{ij} x_{ij}^k$$

Subject to

$$\sum_{i\in V}\sum_{k\in K} x_{ij}^k = 1 \quad \forall j \in C \tag{8}$$

$$\sum_{j\in V}\sum_{k\in K} x_{ij}^k = 1 \quad \forall i \in C \tag{9}$$

$$\sum_{(i)\in V} x_{ih}^k - \sum_{(j)\in V} x_{hj}^k = 0 \quad \forall h \in V \quad \forall k \in K \tag{10}$$

$$\sum_{i\in V} q_i \sum_{j\in V} x_{ij}^k \leq Q \quad \forall k \in K \tag{11}$$

$$\sum_{i\in D}\sum_{j\in C} x_{ij}^k \leq 1 \ \ \forall k \in K \tag{12}$$

$$\sum_{j\in D}\sum_{i\in C} x_{ij}^k \leq 1 \ \ \forall k \in K \tag{13}$$

$$\sum_{i\in s}\sum_{j\in s} x_{ij}^k \leq |s| - 1 \ \ [s \geq 2; s \in C] \tag{14}$$

In the above formulation constraint (8) and constraint (9) ensure that each customer is served by exactly one vehicle. Constraint (10) ensures the continuity of the vehicle. Constraint (11) ensures the capacity constraint of the vehicle are not exceeded. Constraints (12) and (13) ensure that the vehicle leaves and returns to the same depot. Constraint (14) eliminates sub-touring. Having now formulated the problem for one depot we then move to our actual problem which involves multi-depot (D), each having a set of K vehicles to cater to the demand of the set of customers (C).

3.4. Multi-depot capacitated vehicle routing problem We formulate a problem to determine the route of that each vehicle should take before returning to the same depot it started from .In our formulation x_{ij} is a binary variable which determines whether the arc (i, j) is a part of the optimum route found.

$$Z = Min \sum_{ij\in A} cst_{ij} x_{ij}^k$$

Subject to

$$\sum_{i\in V}\sum_{k\in K} x_{ij}^k = 1 \quad \forall j \in C \tag{15}$$

$$\sum_{j\in V}\sum_{k\in K} x_{ij}^k = 1 \quad \forall i \in C \tag{16}$$

$$\sum_{(i)\in V} x_{ih}^k - \sum_{(j)\in V} x_{hj}^k = 0 \quad \forall h \in V \quad \forall k \in K \tag{17}$$

$$\sum_{i\in V} q_i \sum_{j\in V} x_{ij}^k \leq Q \quad \forall k \in K \tag{18}$$

$$\sum_{i\in D}\sum_{j\in C} x_{ij}^k \leq 1 \quad \forall k \in K \tag{19}$$

$$\sum_{j\in D}\sum_{i\in C} x_{ij}^k \leq 1 \quad \forall k \in K \tag{20}$$

$$x_{ij}^k = 0 \quad \forall k \notin D_k', \quad and\ i \in D_k', j \in C \tag{21}$$

$$x_{ij}^k = 0 \quad \forall k \notin D_k', \quad and\ i \in C, j \in D_k' \tag{22}$$

$$\sum_{i\in s}\sum_{j\in s} x_{ij}^k \leq |s| - 1 \quad [s \geq 2; s \in C] \tag{23}$$

In the above formulation constraint (15) and constraint (16) ensure that each customer is served by exactly one vehicle. Constraint (17) ensures the continuity of the vehicle. Constraint (18) ensures the capacity constraint of the vehicle are not exceeded. Constraints (19) and (20) ensure that the vehicle leaves and returns to the same depot. Constraint (21) prevents travel between nodes. Constraint (23) eliminates sub touring.

3.5. Heuristic Approach - k-Means Clustering For understanding how clustering could help in easing-off computational load of Multi-depot vehicle routing problem, we use k-Means Clustering to divide the data into smaller vehicle routing problems applied to individual depots. The k-Means algorithm clusters data by trying to separate samples in n groups of equal variance, minimizing a criterion known as the inertia or within-cluster sum-of-squares.

$$\sum_{i=0}^{n} min_{u_j\in C}(||xc_j - u_j||)^2 \tag{24}$$

After applying clustering to the nodes with depot points as our initial cluster centroids (using init parameter) we continue with the same formulation of Constrained VRP used previously to solve our routing problem.

4. Kroger Vehicle Routing Problem Design

4.1. Data Description

As introduced in section 3, we have modelled our problem based on Kroger depot in the city of Ann Arbor with customer demands distributed throughout the residential areas of the city. Considering varying population densities we have distributed the customer demands by following the below distributions for different neighbourhoods:

- Burns Park and nearby localities - 25%
- Eberwhite and nearby localities - 25%
- North Campus Housing and nearby localities - 25%
- Central Campus and nearby localities - 15%
- Random demands distributed over the grid - 10%

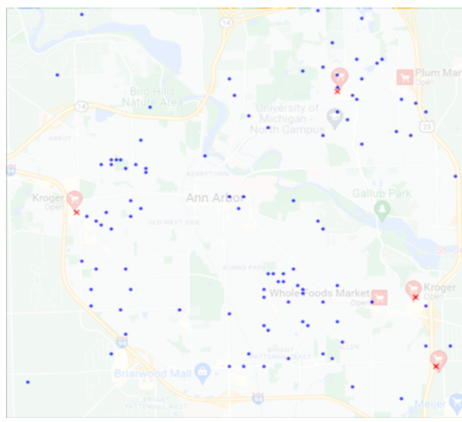

Figure 1: Depot and Customer Nodes in Ann Arbor

We have also assumed random demands, in the range of 1 to 5, for each item for each customer. For the purpose of amassed demands, we have assigned weights, in kg, to each item and have accumulated the whole demand on total weight by each customer. The vehicles are assumed to have a capacity of 400kg.

4.2. Modelling Design

As described in section 3, the project initializes 5 models. The first two models focus on single-depot VRP, one with bin-packing the other with amassed item demands into a single demand for each customer. The next two models are multi-depot VRPs with the similar setup as the previous two models, the difference being on the number of depots on in the model. The last model is a multi-depot VRP decomposed into multiple single-depot VRPs through k-Means clustering.

For the purpose of the project, two function have been scripted for multi-depot VRP

(MDVRP), and bin-packing supported multi-depot VRP (MDVRP-BP) has been scripted on Python with Gurobi. For the single-depot VRP only a single-depot has been passed in the function to get results of the model, while for the multi-depot case all the depots had been passed to the function along with the set of associated vehicles. The last model with k-Means algorithm divides the customer sets associated with individual depots, and these are passed to the MDVRP function for computation. The results of all five models are discussed in the following sections.

5. Computational Study and Analysis

5.1. CPU Time and Objective Value

Table 1 presents the CPU Time, Objective values and % optimality gap achieved by the models. For all the instances the cost of travelling from one node to another is set to $1 per kilometer, equivalent to the distance travelled by the vehicle. We fix the CPU time to two levels, 1 hour and 2 hours, and analyse the results produced by each model. The computations are done on Python with Gurobi 9.1.2 on an Intel(R) Core(TM) i7-8665U CPU with 16GM RAM, the following results were obtained.

Models	CPU Time (hrs)	Objective Value ($)	% Gap
Single-Depot VRP	1	1349	62.1
	2	1220	58.1
Single-Depot VRP with Bin Packing	1	2230	76.1
	2	1372	61.6
Multi-Depot VRP	1	1042	51.9
	2	990	49.4
Multi-Depot VRP with Bin Packing	1	1265	54.9
	2	1009	42.9
VRP with k-Means Clustering	1	791	13.01
	2	790	13.01

Table 1: Results of different alternatives for VRP

5.2. Summary of Results

Although there is no way to directly comment on which model to choose based on just the run-time of the program, it is evident that clustering, or in general dividing the problem into subsets of the main problem, tends to produce better results within the same time frame. This is a direct outcome of the ‘curse of dimensionality’.

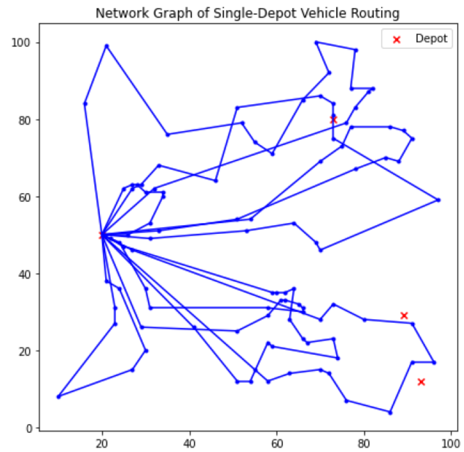

Figure 2: Vehicle Routes through Single-Depot VRP

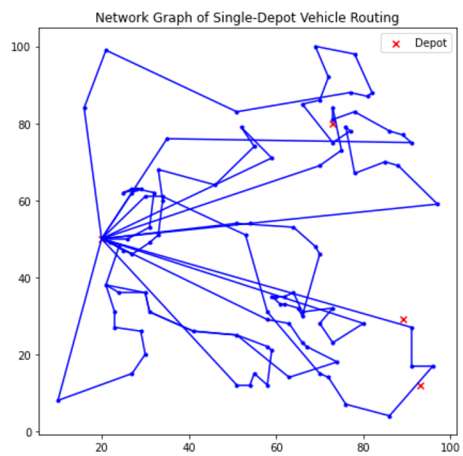

Figure 3: Vehicle Routes through Single-Depot VRP

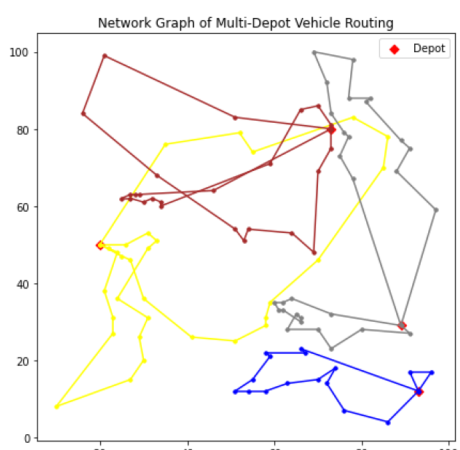

Figure 4: Vehicle Routes through Multi-Depot VRP

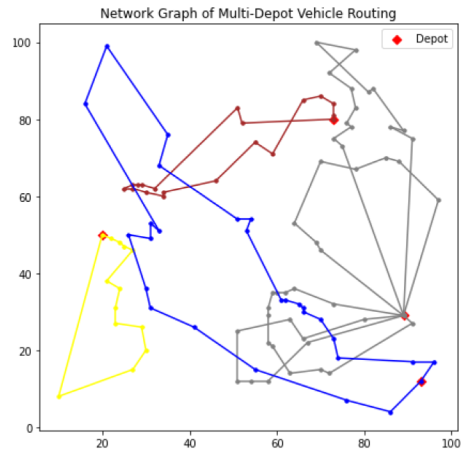

Figure 5: Vehicle Routes through Multi-Depot VRP with bin-packing

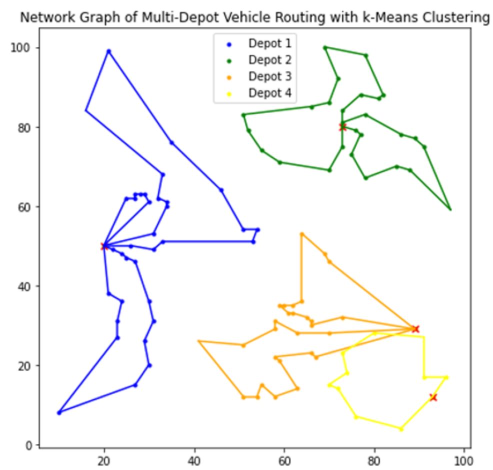

Figure 6: Vehicle Routes through k-Means clustering

The models give a fairly good idea about choosing between the five alternatives for an actual implementation. From Table 1 we can see that single-depot VRPs give a relatively higher cost for deliveries, which is fairly evident as further customers which could have been served by a closer depot are now being served by a farther depot. Naturally, the multi-depot VRPs gives better results compared to the single-depot VRP with most close-by customers being served together. And the best results are achieved by the k-Means clustering, heuristic approach, by segregating customers within the vicinity of a nearby depot.

An observation which can be made is the influence of bin-packing on the VRP. When comparing both the single-depot and multi-depot models, it is evident that for the same program runtime we achieve a higher objective value for the bin-packing alternative than the aggregated demand alternative. This comes from the fact that the vehicles may have to travel further in order to satisfy demands based on the individual items by each customers. Also, multiple vehicles have to travel to the same customer just to satisfy his demand for each item, hence the costs are higher. The graphs of bin-packing routes only represent vehicles deployed for one kind of item to each customer as all routes combined will create a big clutter.

6. Conclusion and Future Scope

The models show that the heuristic method gives better objective values compared to the single-depot VRP and multi-depot VRP models for the same amount of program runtime. It can be agreed that if there is a time limitation to solving a VRP, heuristic approach with k-Means clustering will provide a better result for the same dataset as compared to the traditional approach of vehicle routing with linear optimization.

Further developments which can be made on the model include, extending the model to account for time-window based constraints on delivery to customers, integrating better bin-packing approaches with Vehicle Routing Problem to account for optimal packing of items, and incorporating distances between nodes from Open Street Map or Google Maps to account for real/actual travel distance between points.

References

Crevier, B., Cordeau, J-F., Laporte, G., 2007. The multi-depot vehicle routing problem with inter-depot routes. European Journal of Operational Research 176, 756–773.

Renaud, Jacques, Gilbert Laporte, and Fayez F. Boctor. "A tabu search heuristic for the multi-depot vehicle routing problem." Computers Operations Research 23.3 (1996): 229-235.

Braysy, O. and Gendreau, M. (2005a). Vehicle routing problem with time windows, Part I: Route construction and local search algorithms. Transportation Science, 39(1):104–118.

Braysy, O. and Gendreau, M. (2005b). Vehicle routing problem with time windows, part ii: Metaheuristics.Transportation Science , 39(1):119–139.

Gilmore, Paul C., and Ralph E. Gomory. "A linear programming approach to the cutting-stock problem." Operations research 9.6 (1961): 849-859.

Barnhart, Cynthia, et al. "Branch-and-price: Column generation for solving huge integer programs." Operations research 46.3 (1998): 316-329.

Laporte, Gilbert. "The vehicle routing problem: An overview of exact and approximate algorithms." European journal of operational research 59.3 (1992): 345-358.

Baldacci, Roberto, Maria Battarra, and Daniele Vigo. "Routing a heterogeneous fleet of vehicles." The vehicle routing problem: latest advances and new challenges. Springer, Boston, MA, 2008. 3-27.

Jain, Anil K., M. Narasimha Murty, and Patrick J. Flynn. "Data clustering: a review." ACM computing surveys (CSUR) 31.3 (1999): 264-323.

Barreto, Sérgio, et al. "Using clustering analysis in a capacitated location-routing problem." European Journal of Operational Research 179.3 (2007): 968-977.

De Carvalho, JM Valério. "LP models for bin-packing and cutting stock problems." European Journal of Operational Research 141.2 (2002): 253-273.