Available online at www.sciencedirect.com

ScienceDirect

Transportation Research Procedia 00 (2018) 000–000

Transportation Research Procedia

www.elsevier.com/locate/procedia

World Conference on Transport Research - WCTR 2019 Mumbai 26-31 May 2019

Order First Split Second Algorithm for One to One Pickup and Delivery Problems with Multiple Full Truckload Demands

Tejas Ghorpade, Narayan Rangaraj

Industrial Engineering and Operations Research, Indian Institute of Technology, Bombay

Abstract

A variation of the Pickup and Delivery Problem with customer specified origin and destination locations and demands for multiple Full Truckload demands by each customer is described in this paper. Such requests arise during rail freight or inland container movement. We present an Exact Formulation for this problem that allows multiple vehicle trips over arcs and gives individual cycles for each vehicle. A modified Order First Split Second Algorithm is applied to solve the deterministic version of this problem. We formulate a constructive heuristic to generate a giant tour by forming a greedy path at each feasible location. Next, a splitting algorithm is given that forms multiple cycles based on maximum cycle time and depot return constraints. We also obtain an exact solution for a single vehicle case as the giant tour and split it using the same algorithm for comparison with the heuristic. Considering the possibility of deviation of demand from average, an adaptive algorithm that modifies the cycles to accommodate small changes in demand is given.

Computations show that the heuristic gives a feasible and near-optimal solution to the problem in terms of net revenue and number of vehicles required. The idea of having planned cycles with an adaptive plan is tested on networks with dynamic demands. Possible practical application to the rail freight industry and inland container movement with large data sets are also discussed in this paper.

Keywords: Order First Split Second; Full Truckload, Pick up and Delivery; Exact solution; Giant Tour; Splitting Algorithm; Updating Algorithm

1. Introduction

A major variant of the Vehicle Routing Problem (VRP) is the Pick-up and Delivery Problem (PDP) where goods or people are transported from one location to other over the network. The classification of PDPs is given in the survey by Parragh et al. [2008a,b]. The first part includes the variations of VRP with backhaul which has initial loading at the depot and unloading at the customer location, from where the return load is delivered back to the depot. The second part of the survey describes variations of the general PDP where loading and unloading of single or multiple commodities can occur at any of the locations.

PDPs are also classified as Full Truckload (FTL) or Less than Truckload (LTL) problems. In the FTL case, a vehicle can satisfy only one request at a given time. However, with the LTL demands, each customer requires only part capacity of the vehicle and thus each vehicle can carry multiple orders during a single trip.

Our framework is motivated by two problems, one being the transportation of coal by rail in India. Coal is an important commodity required for electricity generation. Powerhouses are located throughout the country and are located at a significant distance from the mines. Coal is required regularly in large quantities and has to be transported repeatedly from the mines to the powerhouses. Also, the imported or exported coal has to be moved to and from the ports. To facilitate such repeated movements, there is a set of dedicated rakes moving over the rail network. These rakes undergo timely maintenance at certain depots. Customers place requests in the online system and state the pickup and drop location as well as the number of rakes required. The demand is reasonably predictable throughout the year.

Another application is the intermodal transportation of goods in standardized containers which can be loaded into trains, trucks, barges, ships, etc. In the hinterland operations, the export containers are loaded at the customer locations or the Inland Container Depots and sent to the ports, whereas the imported loaded containers are delivered to the customers where unloading occurs. With relaxation in the government policies, these containers can also be used for movement of domestic loads but have to be returned to the shipping companies within a specified period. There are a few one time customers but a majority of regular customers of the container shipping companies who export or import goods on a periodic basis. In both these cases, though the demands are not completely constant over time, there is a cyclic pattern of deployment of rolling stock helpful in deciding the number of vehicles and their potential base.

Currently, for the rail or container assignment in India, decisions are taken by individuals based on their experience, by allocating available resources to immediate requests without a long term route plan. As we assume that the demands are predictable, tentative cycles can be planned for existing resources that satisfy the upcoming requests along with certain recourse strategy to accommodate variations in the demand pattern. This will also reduce the task of making allocation decisions repeatedly.

If the vehicles are based at a particular depot and need to return periodically for maintenance or other purposes then cycles can be planned around this depot. The significance of the depot node could be very specific as in rail freight where the rakes have to be maintained at the depot within a given time period. In the container movement, the depot node is the point of entry or exit for international ocean movements. This could be a port or a container freight station where import or export containers are received, destuffed and customs-examined. If the vehicles are not associated with any depot, we can have a reference location to plan the cycle. Such routing decisions can be applied to freight transportation or even passenger transport like taxi services where one customer has to be served at a given time and demands are predictable over the given time period.

Problem Description and Contribution:

In this paper, we consider the specific case of multiple Full Truckload demands for One to One Pick-up and Delivery Problems. In this problem, the depot corresponds to the location where the vehicles are based. The customer requires a commodity to be picked from a location and delivered to some specific location. The customer requests also specify the number of vehicles they require. This demand can be served by a single vehicle performing repeated trips over a given arc or by multiple vehicles satisfying one of the demands in a given trip. Such, multiple requests are modeled as separate customers in the literature, increasing the size of the problem to be solved. In this paper, we provide a model to solve the problem without altering its physical characteristics and interpretation.

First, we present an exact mathematical formulation of the problem described above that allows multiple vehicle trips over arcs and gives individual cycles for each vehicle. The insertion heuristic methods used to solve such multiple vehicle routing problems are classified as Cluster First Route Second (CFRS) or Order First Split Second (OFSS). The OFSS method has been explored in the past few decades for VRP solutions. However, this method has not been frequently applied in a context of PDPs which are closer to Arc Routing Problem than to the Node Routing Problem under which most of the Vehicle Routing Problems (VRP) are classified. A modified Order First Split Second Algorithm is applied in this paper to solve the deterministic version of this problem. We formulate a constructive heuristic to generate a giant tour by forming a greedy path at each feasible location. Next, a splitting algorithm is given that forms multiple cycles based on maximum cycle time constraints. We also obtain the exact solution for the single vehicle case as the giant tour and determine cycles using the same splitting algorithm for comparison with the heuristic. Considering the uncertainty in demand, an adaptive algorithm is presented that modifies the cycles to accommodate small variation in the demand.

Organization of paper:

This paper is organized as follows. Section 2 gives a brief literature review. A mathematical formulation of the problem resulting in an exact solution is given in section 3. The Order First Split Second Heuristic approach used to solve the problem is described in section 4. The exact solution to giant tour formation is given in section 4.1 and a heuristic solution for the same is given in section 4.2. The splitting algorithm is discussed in section 4.3. This method is used to determine cycles over the network for deterministic demands. As a recourse in case of variation in demand, we suggest a Cycle Updating algorithm in section 5 that modifies cycles based on observed demands at a particular time period. The detailed outputs of these methods are explained with an example in section 3.1. The computations for testing the performance of the heuristic using performance parameters given in section 6 are described in section 7. Motivating examples from the rail freight industry in India and inland container movement planning as discussed above are also explored in this paper in section 8.

2. **Literature review**

The general PDP has been reviewed by Berbeglia et al. [2010] and classified into Many to Many, Many to One and One to One Problems. They mention a strategic approach to solve the Dynamic PDPs, provide performance measures for Dynamic problems and explore the effects of having future information and intermediate decisions like vehicle diversion. Parragh et al. [2008a,b] give an extensive survey of Less than Truckload (LTL) PDPs. In these papers, the exact, heuristic and metaheuristic methods used to solve the respective problems have been discussed. These papers state that limited work is done on the Dynamic Pick up and Delivery problems between two customer specified locations.

We classify the papers according to the problems they solve in Table 1. The first column identifies the paper and if it considers full truckload (FTL) or less than truckload (LTL) is given in the second column. The third column specifies mapping in each of the papers. The FTL demands require One to One movement from a pickup to a delivery location. However, the LTL problems could be One to One where the products from Pick up locations are distinguishable and have to be transported to a specified destination, Many to Many when the products are same and demand at a location can be fulfilled from any pick up location and lastly, One to Many where demands are picked from the depot and transported to the customer and vice versa. The Pickup and Delivery locations, (customer specified or depot) are given in the fifth column. The sixth column states if the pickup and drop nodes in the model are classified separately or the same node serves both the purposes. The last column lists the solution methodology used in the papers. In most of these works, the nodes are classified differently as pickup or delivery nodes or both nodes are combined into a single node for an order. Among the review papers, only Bianchessi and Righini [2007] consider pickup and delivery from the same location and Bruggen et al. [1993] does not specify any such conditions.

The exact solutions are obtained by Baldacci et al. [2011] modeling it as a set partitioning problem and Wang and Regan [2002] where the problem is modeled as an assignment problem. Baldacci et al. [2011] first generate sets of all feasible paths for every vehicle and then these routes are assigned to vehicles by set partitioning thus giving a static solution. Wang and Regan [2002], only assign one customer at a time and solves the dynamic problems myopically.

Paper	Demands	Goods movement mapping	Pick up and Delivery locations	Pick up or Delivery customers	Solution Methods
Mosheiov, 1997	LTL	one to many	depot to cust	different	Iterated Tour Partitioning
					Tour Partitioning
Bianchessi, 2007	LTL	one to many	depot to cust	same	Neighbourhood search
					Tabu search
Gronalt, 2003	FTL	one to one	cust to cust	combined (order)	Modified Savings Algo
Fabri, 2005	LTL	many to many	cust to cust	different	A* Algorithm
Bruggen, 1993	LTL	one to one	cust to cust	not specified	Arc exchange Algorithm
Quilliot, 2010	FTL	one to one	cust to cust	different	Greedy insertion heuristic
Yu, 2007	LTL	many to many	cust to cust	different	Evolutionary Algorithm
Baldacci, 2011	FTL	one to one	depot to cust	different	Set partitioning (ILP)
Currie, 2004	FTL	one to one	depot to cust	different	Tabu search
Wang, 2000	FTL	one to one	cust to cust	combined(order)	Assignment Problem (ILP)
Caris, 2009	FTL	one to one	depot to cust	different	Insertion Heuristic
This paper	FTL	one to one	cust to cust	same	Order First Split Second

FTL: Full Truckload ; LTL: Less than Truckload

Table 1: Pick up and Delivery Variants

Bianchessi and Righini [2007] mention various alternate heuristic approaches to the VRP with Backhauls. They specify a tour partitioning method, a neighborhood search method with complex and variable neighborhoods and a tabu search method with adaptive lengths of tabu list. The savings algorithm originally developed for VRP has been extended by Gronalt et al. [2003] and applied to FTLPDP. Fabri and Recht [2006] use the A* algorithm to find the shortest path for a single Pickup and delivery vehicle and extended it further to multiple vehicles. Bruggen et al. [1993] have used an arc exchange algorithm to solve the single-vehicle PDP between customer locations. A tour partitioning heuristic for solving VRP is extended to solve the VRP with backhauls by Mosheiov [1998]. Two variants of Iterated Tour Partitioning are developed, as Exhaustive Iterated Tour Partitioning (EITP) and Full Capacity Iterated Tour Partitioning (FC-ITP) whose performances depend on the spread of the customers around the depot.

Customer to customer PDP is defined as preemptive stacker crane problem and solved by tree based greedy insertion and local transformation by Quilliot et al. [2010]. Yu and Yu [2007] give Evolutionary approach to solve the PDP in a case where the identical goods can be picked from and delivered to any location. The solution approach is a combination of an evolutionary algorithm based on special genetic operators and Pareto dominance method and uses local tabu search for improving the solutions. Currie and Salhi [2004] solve the full truckload problem for movement between depot and customer by using tabu search heuristic whereas Caris and Janssens [2009] use route construction by insertion heuristic followed by a local search to solve the similar problem. This paper defines the problem as hinterland operations of intermodal transportation using containers.

Most of the PDP literature focuses on VRP with backhauls, where the goods are transported from the depot to the customer and back to the depot. Li and Lu [2014] state that full truckload problem is less studied as compared to the less than truckload problem. We also observe that this problem is not thoroughly explored in the context of One to One PDP.

The One to One PDP is closer to the Arc Routing Problem (ARP) than the Node Routing Problem (NRP) including most variants of VRP as demand can be given over arcs rather than at nodes. To see the difference in the way these problems are approached, we studied certain papers where the methods for NRP were extended to accommodate ARP or the ARP problem was redefined as NRP. The CARP has been explored by Lacomme et al. [2004] where they have extended three heuristics path scanning, augment merge and Ulusoy's heuristic which uses tour splitting method to solve ARP. They also use a memetic algorithm with a split. Ulusoy [1985] initially form a giant tour by using the Chinese Postman Problem and breaks it down to multiple single vehicle feasible subtours. Over these subtours new networks are generated and shortest path problems are solved over them.

Lastly, we refer to the survey of Order first split second methods given by Prins et al. [2014]. They explain the basic split procedure for CVRP and its application to CARP by converting it to CVRP. Also, various extensions of the basic problem are illustrated. Besides this, the application of split delivery to PDP is given by Bianchessi and Righini [2007] and Mosheiov [1998].

3. Mathematical Formulation

The Full Truckload problem can be modeled on a directed graph G = (N; E), where the possible loading\ unloading locations are nodes and the complete path joining two locations are edges between the corresponding nodes. We define our Pick up and Delivery requests as demands over the arc joining the two locations. We consider Full Truckload problem as the vehicle cannot pick other any load on its way before serving the current customer. Hence, we do not need to consider the actual path between demand points and this problem can be defined as an Arc Routing Problem. We consider the problem with multiple FTL demands over arcs which is generally observed in freight transportation. The demand k over an arc can be served by k different vehicles or by a single vehicle making k trips over the given arc or any combination between these extremes.

In their mathematical formulations, Parragh et al. [2008a,b] and Yu and Yu [2007] classifies each pickup and delivery nodes separately with single demand at each location. Multiple demands are considered by Mosheiov [1997], but they are represented in the model as multiple requests with unit demands. The pickup and delivery request is modeled as a single node by Wang and Regan [2000] and Gronalt [2003] and these nodes are assigned to vehicle schedule. Considering these modifications of the real physical structure of the problem, the models restricts a single visit to each node. However, such modeling leads to an increase in the size of the problem. Thus, the existing formulations consider the problem as node routing problem and restrict repeated visits to the same node and hence multiple trips of a vehicle over any given arc. The model represented in this paper differs from existing models as we consider demands over arcs and allow multiple trips over each arc. Also, each node can serve as a pickup or delivery node for different requests.

The requests are specified in terms of the number of vehicles required to satisfy the request of transporting goods between two given locations. The loading and unloading locations are connected via requests and thus paired. This can be referred to as One to One Pick-up and Delivery Problem with Multiple Full Truckload demands. Given a network with known demands over each time period, we aim to obtain cycles that maximize the net revenue and minimize the number of vehicles required to meet the demand. The net revenue is the result of revenue earned from loaded movements and costs incurred during empty movement. As we assume that all the requests are served, the revenue earned from loaded movements is constant. In order to maximize net revenue, the empty running costs have to be minimized. Thus we consider that the objective function that minimizes the total empty running and the number of vehicles required.

As there could be multiple visits to each location we need to track order of movements, to obtain separate vehicle cycles. For this, we include a time index T as a set of discrete time periods with maximum element equal to the allowed cycle time ($|T|$). Thus $T = [0,1,...|T|]$. This limit on T, avoids separate constraints of total time period on each cycle. Also note that the actual time period can be specified in hours, days or weeks based on the frequency of demands and desired time to serve them. Travel time is discretized in order to determine balance equations at every node at every time period. Since the depot node can also be a pickup or delivery location, we add dummy nodes *s* and *t* to represent the source and sink at the depot. On the resulting graph, we determine the direct path from source to sink. By using the time index, we also eliminate the need for separate subtour elimination constraints.

General Parameters:

- N: Set of nodes denoting the locations on the network
- a_{ij}: Directed arc connecting two locations i and j, $A = \{a_{ij} \mid i \in N, j \in N\}$
- d_{ij}: Distance between nodes i and j, $D = \{ d_{ij} \mid ij \in A \}$
- t_{ij}: Travel time between nodes i and j, $T = \{ t_{ij} \mid ij \in A \}$
- c_{ij}: Revenue earned by loaded movement from location i to location j, $C = \{ c_{ij} \mid ij \in A \}$
- c'_{ij}: Cost of empty travel from location i to location j, $C' = \{ c'_{ij} \mid ij \in A \}$
- r_{ij}: Average number of requests over arc ij per time period, $R = \{r_{ij} \mid ij \in A \}$
- s: Dummy source node
- q: Dummy sink node
- d: Depot

Variables:
$x_{ij}{}^{t}$ - 1 if arc $ij \in A$ is traversed at time $t \in T$ and 0 otherwise
u_{ij} - Number of empty movements from node i to j.
m- Number of cycles starting from the depot

A mathematical formulation for solving the OFTPDP problem is given below. We denote ILP giving the complete solution as I_c

$$Minimize \sum_{ij \in A} c'_{ij} \times u_{ij} + M \times m \qquad (1)$$

subject to

$$u_{ij} \geq \sum_{t \in T} x_{ij}{}^{t} - r_{ij} \qquad \forall\, ij \in A \qquad (2)$$

$$\sum_{t \in T} x_{sd}{}^{t} = m \qquad (3)$$

$$\sum_{ij \in A,\, i \neq s} x_{ij}{}^{0} = 0 \qquad (4)$$

$$\sum_{j:(ij \in A), j \neq d} \sum_{t \in T} x_{ij}{}^{t} = 0 \qquad for\ i = s \qquad (5)$$

$$\sum_{j:(ij \in A),\, i:(t^{*}+t'_{ij}=t)} x_{ij}{}^{t^{*}} = \sum_{j:(ij \in A)} x_{ij}{}^{t} \qquad \forall\, j \in N,\ t \in T \qquad (6)$$

$$\sum_{j:(ij \in A), j \neq d} \sum_{t \in T} x_{ij}{}^{t} = m \qquad for\ j = q \qquad (7)$$

$$\sum_{t \in T} x_{iq}{}^{t} = \sum_{t \in T} x_{si}{}^{t} \qquad \forall\, i \in N \qquad (8)$$

$$\sum_{t \in T} x_{ij}{}^{t} \geq r_{ij} \qquad \forall\, ij \in A \qquad (9)$$

$$x_{ij}^{\ t} \quad \in \quad \{0,1\} \qquad \forall ij \ \in A \qquad (10)$$
$$u_{ij} \quad \in \quad N \qquad \forall ij \ \in A \qquad (11)$$
$$m \quad \in \qquad (12)$$

The objective function 1 minimizes the empty distance traversed by the vehicle and the number of vehicles used. The variable u_{ij} is defined by the number of excess movements on given arc *ij* beyond existing demand in constraint 2. The arcs originating from *s* equals to the number of cycles is given by 3. The condition that no other arcs start at time *t*=0 except *s* is specified by constraints 4 and 5. The flow balance equation is given by 6. Similarly, the constraints on the arc from the depot to sink is given by constraint 7. The number of paths starting from source *s* are equal to paths terminated in sink *q* is guaranteed by constraint 8. Constraint 9 ensures that the entire demand is satisfied over the given time period.

At every node, we apply balance equation 6 such that the number of arcs leaving any node at any given time is equal to the number of arcs that enter into it at that particular time and having left previous nodes at the time given by (t_0- corresponding traveling time). Thereby, we track the individual cycles based on entering and leaving time of arcs at every node. From all arcs starting from *s* we form multiple cycles ending in *q*, each completed within time $|T|$. The exact solution obtained from this model is used as a reference to study the performance of the heuristic methods in section 7.

3.1. Example

We illustrate the solution techniques proposed in this paper with the help of an example specified in this section. Consider a network with 5 locations as nodes and paths joining them as edges. From this, we form a directed graph with nodes = {1, 2, 3, 4, 5}, the direct path distances between nodes are given in table 2 in kilometers. The demands per day over this network are given in Table 3 as the number of vehicles requested. The traveling time between any two given nodes is taken to be directly proportional to the distance between them such that $t_{ij} = ceil(0.05\ d_{ij})$.

n	1	2	3	4	5
1	0	55	23	27	27
2	55	0	58	40	79
3	23	58	0	46	25
4	27	40	46	0	54
5	27	79	25	54	0

Table 2: Distances in km between nodes

n	1	2	3	4	5
1	0	5	0	0	3
2	1	0	3	4	3
3	5	3	0	4	1
4	3	3	0	0	5
5	3	5	1	3	0

Table 3: Average Demands per day

Here we now have total of 47 demands and need to be served within 24 hour period.

3.2. Solution to the Example in 3.1 by I_c

The exact solution to the above example obtained by solving Ic is given in table 4. The second column determines cycles that satisfy the average demand. The third column denotes the type of movements i.e. empty (e) or loaded (l). The fourth column gives the time required to complete each cycle in hours where maximum available time is 24 hours and the last columns give net revenue earned in each cycle where revenue earned for each loaded movement is 0.5 times distance covered and the cost for empty travel is 0.25 times the empty distance traveled.

No.	Cycles	Movement	Time	Revenue
1	1, 3, 2, 3, 4, 2, 5, 4, 1	e, l, l, l, l, l, l, l	23	175.25

2	1, 3, 2, 3, 4, 2, 3, 1, 3, 1	e, l, l, l, l, l, l, e, l	23	141.5
3	1, 3, 1, 5, 2, 3, 2, 1	e, l, l, l, e, l, l	20	100.75
4	1, 2, 5, 4, 5, 2, 3, 1	l, l, l, l, l, e, l	23	157.5
5	1, 2, 5, 2, 3, 4, 5, 1,	l, l, l, e, l, l, l	23	155.5
6	1, 2, 4, 1, 5, 1, 5, 2, 3, 1	l, l, l, l, l, l, l, e, l	23	138
7	1, 2, 4, 5, 3, 4, 2, 3, 5, 1	l, l, l, l, l, l, e, l, l	23	141.5
8	1, 2, 4, 5, 2, 4, 5, 4, 1	l, l, l, l, l, l, l, l	23	201.5
		Total	181	1211.5

Table 4: Solution to example by I_c

We have considered node 1 to be the depot. However, at every time period, the vehicle can start from any node from this cycle if there is no depot or base node. Also if the start and end movements of cycle comprise of empty runs, the cycle can be considered starting from the first node from where loaded movement starts and the node before last empty movement can be directly connected to this node.

4. Order First Split Second Approach

An Order First Cluster Second method called Iterated Tour Partitioning (ITP) is used by Mosheiov [1997] to solve the less-than-truckload PDP. They obtain the giant tour by solving the Travelling Salesman Problem (TSP). Two splitting algorithms are described in their paper. The giant tour is broken at points where the capacity constraint is violated in one algorithm and the second algorithm splits tours by extracting the largest segments in the tour which satisfies the capacity constraints, repeatedly, till all the demands are covered. In their constructive heuristic to solve LTLPDP, Bianchessi and Righini [2007] also use ITP and split the Hamiltonian tour based on allowing strongly or weakly feasible tours based on capacity constraints being satisfied directly or after modifying original sequence.

The full truckload PDP, on the other hand, is solved by exact methods by Baldacci et al. [2011] and Wang and Regan [2002]. The problem is solved heuristically using modified savings method by Gronalt et al. [2003] Quilliot et al. [2010] use Greedy insertion with preemptive loads and Insertion Heuristic combining pick up and delivery nodes are tested by Caris and Janssens [2009]. Tabu search procedure is used by Currie and Salhi [2004] to solve a similar problem. Thus, inspite of being a commonly used approach for LTL cases none of the papers have used the Order First Split Second Approach for FTL problems.

We modify the Order First Split Second (OFSS) Approach to solve One to One the FTLPDP. This problem is constrained by the maximum cycle time and the capacity constraints are not applicable. The OFSS method consists of two steps. In the first step, we relax the cycle time constraint and form a giant tour assuming that a single vehicle has to satisfy the entire demand. In the second step, we split this giant tour to form multiple cycles which satisfy the time constraints. We use two alternate methods to generate the giant tour in the first step. In one method, we form a giant tour by using an ILP formulation giving an optimal tour. We denote this tour formation ILP as I_t. In the second method, we use a constructive heuristic to find a feasible solution to the problem by generating demand based approximate values for each node. This tour forming heuristic is denoted by H_t. In the second step, we formulate a splitting algorithm, H_s where this single path is broken to find cycles around a predefined location, preferably a depot. We denote the first method with optimal find giant tour and splitting algorithm as I_tH_s method and the second method using tour formation heuristic and splitting algorithm as H_tH_s method.

4.1. *Exact formulation for Giant Tour (It)*

In this section, we give the exact mathematical formulation to obtain a giant tour. This model differs from I_c given in section 3 as we ignore the time constraints and determine only the stagewise movements without considering the actual time when the vehicle reaches a location. We define a set of discrete indices K=1,2,3,... indicating the decision making stages. Along with the variables x^k_{ij} and u_{ij} we consider variable y^k to determine the number of stages required which is 1 if the vehicle moves in step k and 0 otherwise.

$$Minimize \sum_{ij \in A} c_{ij} \times u_{ij} + \sum_{k \in K} k \times y^k \qquad (1)$$

subject to

$$u_{ij} \geq \sum_{t \in T} x_{ij}{}^k - r_{ij} \qquad \forall\, ij \in A \qquad (2)$$

$$\sum_{ij \in A,} x_{ij}{}^k \leq y^k \qquad \forall\, k \in K \qquad (3)$$

$$y^{k+1} \leq y^k \qquad \forall\, k \in K \qquad (4)$$

$$x_{sd}{}^0 = 0 \qquad (5)$$

$$\sum_{j:(ij \in A)} x_{ij}{}^k = \sum_{j\,:\,(ij \in A)} x_{ij}{}^t \qquad \forall\, j \in N,\ k \in K \qquad (6)$$

$$\sum_{k \in K} x_{ij}{}^k \geq r_{ij} \qquad \forall\, ij \in A \qquad (7)$$

$$x_{ij}{}^k \in \{0,1\} \qquad \forall\, ij \in A,\ \forall k \in K \qquad (8)$$

$$y^k \in \{0,1\} \qquad \forall\, k \in K \qquad (9)$$

$$u_{ij} \in N \qquad \forall\, ij \in A \qquad (10)$$

In the above model Objective 1 aims to minimize the empty distances and the number of stages within which the movement occurs so as to choose consecutive stages for assigning movements. The condition that u_{ij} is the number of empty movements over arc *ij* is determined by Constraint 2. Constraints 3 and 4 ensure that a given stage is selected only if movement occurs and all movement occurs in all consecutive stages. The initial movement from a dummy source to the depot is ensured by 5. Constraint 6 is the flow balance constraint for two consecutive stages. Lastly, the constraint that the entire demand needs to be satisfied is given by 7.

4.2. *Giant Tour formation Heuristic (Ht)*

With the increasing size of the problems, the time to solve *It* increases even for a single vehicle case. Hence, we propose a heuristic that can give a feasible giant tour in lesser time. While making the assignments of a request to a constructed route we associate a value with every location available for vehicle movement. We determine the value of a location by constructing greedy paths from that location and calculating its cost.

4.2.1. *Value Determination*

We calculate the value at each node in the path by forming a greedy path of maximum length *lv* from the corresponding node satisfying the unassigned demands. This path is based on the prioritizing the location from where there is maximum to and fro demand. In case there is no return demand to a particular node, we next prefer a location to which there is an onward demand. If this is not true for any of the locations, we consider the node from which there is the maximum outward demand. These three cases are represented in figure 1. In case 1, we are at node 1 and the maximum outward and return demand is for node 4, thus we move to node 4. In case 2, there are no bidirectional demands from any of the nodes. Considering only the outward demands from 4, we can move to node 1 or 3, but the maximum revenue is earned when we go to 1. In case 3, there is no outward demand to any node from current node 3, thus we see that the maximum outward demand among nearby nodes is from 2 and thus we move to this location in the next step. We follow these steps until a path of maximum length l_v is obtained. The net revenue earned by traversing this path is assigned as the value of the starting node of the path.

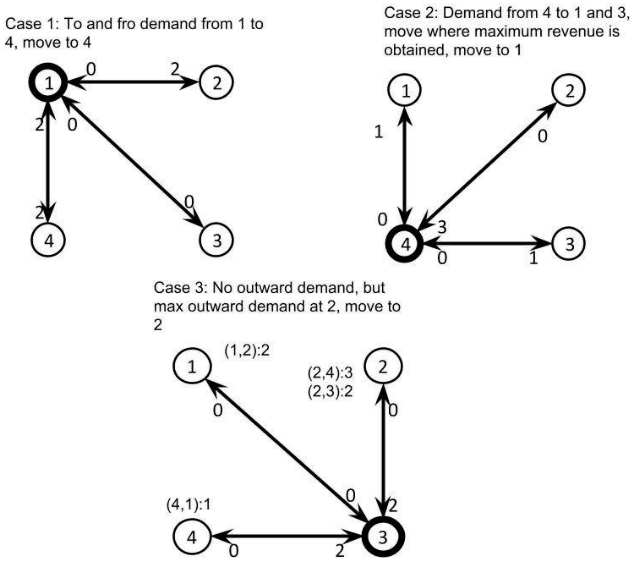

Figure 1: Greedy path formation for Value Function

4.2.2. *Giant Tour formation (Ht)*

We formulate a giant tour by a constructive heuristic method using the values obtained by the method described in section 4.2.1. The average demands at each node that have to be met within the given time period are known. First, we calculate the values for each starting node and select the one with the highest value. Further at every step, we calculate the benefit of moving to other available nodes say '*i*''s from the current location by calculating the revenue earned by following a greedy path from *i*. The node with the maximum value is then selected as the next location for the vehicle to visit. Also based on the existing demand on the arc traveled, we note if the vehicle traveled in empty or a loaded condition.

When we move from the current node to a new node, the demand over the corresponding arc is then updated and this pending demand value is used for any further calculations of greedy paths. We also consider a tabu list of a certain number of previous empty trips and avoid selecting them as the current trip to reduce the number of empty movements. Thus, a tabu search with changing list length similar to Bianchessi and Righini [2007] is used here.

4.3. *Tour Splitting (Hs)*

In the second phase of the OFCS method, we aim to form cycles by splitting the Giant Tour, such that each cycle starts and returns to a particular depot. In the splitting algorithms used in literature, the splitting is done based on vehicle capacity. This algorithm differs from other capacity based splitting algorithms as for Full Truckload PDP the capacity constraints are not relevant. The cycles are such that their travel time does not exceed Cycle Time $|T|$. We consider the Giant tour '*P*' obtained by *Ht* or *It*. In the first step, we split tour *P* every time it visits the depot to form a set of cycles starting and ending at depot C={ c_i | c_i ε C} . Further, we observe all the c_i's with traversal time larger than cycle time *Tc* are split by inserting an empty return to the depot. Finally, if possible the smaller cycles are combined to form a single cycle around the depot.

4.3.1. *Solution to Example in 3.1 by* I_tH_s *and* H_tH_s

We illustrate these methods using the example given in 3.1. It was solved by *Ic*. In this section we solve the example by forming a giant tour by *It* and then splitting this tour by *Hs* we obtain results as given in table 5. The solution obtained by forming the giant tour by *Ht* is given in table 6. We tried to test the parameter *lv* for multiple demand instances, but it showed no visible trend. So we decide to use 10 different values of *lv* and consider the best solution among them. This is a feasible option since the heuristic takes a few seconds to solve even large problems. We test the performance of these solutions using the optimal solution obtained in table 4 as standard. This is done in section 6.

No.	cycle	Movement	Time	Revenue
1	1, 5, 4, 5, 1, 2, 4, 2, 4, 5, 1	l, l, l, l, l, l, l, l, l, l	24	209.0
2	1, 5, 3, 4, 2, 4, 5, 1, 2, 3, 1	l, l, l, l, l, l, l, l, l, l	24	197.5
3	1, 2, 5, 2, 5, 2, 1	l, l, l, l, l, e	22	171.5
4	1, 2, 5, 4, 1, 3, 4, 1, 3, 1	e, l, l, l, e, l, l, e, l	23	102.75
5	1, 2, 4, 1, 3, 1, 3, 1	l, l, l, e, l, e, l	15	72.5
6	1, 3, 4, 5, 2, 3, 2, 3, 1	e, l, l, l, l, l, l, e	23	165.0
7	1, 3, 5, 4, 5, 2, 3, 2, 1	e, l, l, l, l, e, l, e	23	101.0
8	1, 3, 2, 3, 4, 2, 1, 2, 3, 1	e, l, e, l, l, l, l, e, l	24	103.75
9	1, 5, 2, 1	l, l, e	9	39.25
		Total	187	1162.5

Table 5: Solution by I_tH_s

No.	cycle	Movement	Time	Revenue
1	1, 5, 2, 5, 2, 5, 3, 1	e, l, l, l, l, e, l	22	156.5
2	1, 5, 2, 3, 4, 5, 2, 3,1	e, l, l, l, l, l, l, e	24	174.5
3	1, 4, 5, 4, 5, 2, 5, 1	e, l, l, l, l, l, e	21	146.5
4	1, 2, 4, 5, 1, 2, 3, 2,1	l, l, l, l, l, e, l, l	22	157.5
5	1, 5, 4, 1, 5, 4, 1, 5, 3,4, 1	l, l, l, l, l, l, l, l, l, l	23	170.5
6	1, 2, 3, 2, 3, 2, 4, 5,1	l, e, l, l, l, l, l, l	22	160.5
7	1, 2, 4, 3, 4, 3, 5, 1, 3, 1	l, l, e, l, e, l, l, e, l	22	79.25
8	1, 2, 4, 2, 4, 2, 4, 2, 3, 1, 3, 1	l, l, l, e, l, e, l, e, l, e, l	24	90.25
9	1, 3, 4, 3, 1	e, l, e, l	10	17.25
		Total	190	1152.75

Table 6: Solution by H_tH_s

5. Cycle Updation (Hu)

The cycles in the previous steps are designed to satisfy average demand. The observed demand in a given time period could differ from this average value. To cater to these demands, we propose a method, based on the difference between observed and average values, that updates the cycles obtained by *Hs* to satisfy the current demand.

For updating the cycles, we first calculate the differences in actual and average demands and note the arcs over which demands have increased (or decreased). In the first step, we remove the nodes that have reduced demands in two consecutive arcs from the cycle having the highest number of reduced demand arcs. We repeat this on all cycles. Next, the loaded trips are converted to empty trips for the remaining arcs. Once all the unrealized trips are removed from the cycles we start adding the new excess demands.

The steps used in this Algorithm are explained here:

1. Add the requests which can be included by replacing empty trip by loaded trip.
2. Replace the existing empty trips by a loaded trip followed by an empty trip from the given location.
3. Unserved requests can be added at the end of smaller cycles by adding an empty trip to the origin of the request, a loaded trip to the destination and then an empty trip back to the depot.
4. Demands that are not accommodated in existing cycles are combined to form separate cycles.
5. These extra cycles can be served depending on the availability of vehicles.

By having a plan to modify the cycles, we work with the initial route and adjust them to actual demands without affecting initial customers by rerouting in every time period. The changes in 3.1 with demand is shown with the example below.

5.1. Cycles for observed demands using Hu for the Example discussed in 3.1

We consider that the cycles are planned as per *HtHs* heuristic on average data given in Table 6. Now we observe actual demands prior to the beginning of operational time period, which is as shown in table 7. We obtain a new set of cycles shown in table 8 by using Cycle Updation. In these cycles, we see that four of the nine cycles (1,7,8,9) are exactly similar to the previous solution. The remaining cycles are modified to accommodate the changes in demand. Thus, based on the previous solution, we obtain the modified cycles that followed a similar structure and does not affect the existing route plan.

n	1	2	3	4	5
1	0	5	0	2	3
2	2	0	1	4	3
3	5	2	0	2	1
4	3	3	0	0	5
5	3	5	1	3	0

Table 7: Observed demands at some time period

No.	Cycle	Movement	Time	Revenue
1	1, 5, 2, 5, 2, 5, 3, 1	e, l, l, l, l, e, l	22	156.5
2	1, 5, 2, 1, 4, 5, 2, 1	e, l, l, e, l, l, e	21	147.5
3	1, 4, 5, 4, 5, 2, 5, 1	l, l, l, l, l, l, e	21	166.75
4	1, 2, 4, 5, 1, 2, 3, 1	l, l, l, l, l, e, l	18	112.5
5	1, 5, 4, 1, 5, 4, 1, 5, 3, 1, 4, 1	l, l, l, l, l, l, l, l, e, l, l	24	155.25
6	1, 2, 3, 2, 3, 2, 4, 5, 1	l, l, e, l, e, l, l, e, l	16	146.5
7	1, 2, 4, 3, 4, 3, 5, 1, 3, 1	l, l, e, l, e, l, l, e, l	22	79.25
8	1, 2, 4, 2, 4, 2, 4, 2, 3, 1, 3, 1	l, l, l, e, l, e, l, e, l, e, l	24	90.25
9	1, 3, 4, 3, 1	e, l, e, l	10	17.25
		Total	178	1071.25

Table 8: Solution by H_tH_s

6. Performance parameters

The performance of heuristics described in this paper are judged by comparing the Net Revenue and Number of Vehicles required to satisfy the demands. To study the effectiveness of the heuristics, we compare them with the optimal solution obtained from *Ic*. For this, we define two terms given below:

- Revenue Ratio (RR) is the total revenue earned from the solutions given by Heuristic as the fraction of maximum revenue that is indicated by the Exact solution.

- Number of Vehicle Ratio (NVR) is the Number of Vehicles determined by the heuristic as a multiple of the minimum number of vehicles specified by the Exact solution.

The cycle time constraint and the Pickup and Delivery demands from different nodes necessitate some minimum empty travel for vehicles to reach all loading locations or back to the depot. Since we assume that all the demands need to be served, the loaded running time is fixed and the variation in cycle times is only due to the empty runs. Thus, a higher value of total time implies longer empty running time.

- We measure Utilization (U) of the vehicle in terms of loaded running time as a fraction of the total running time of the vehicles.

To study the performance of the updating algorithm with demand diverging from average we use the concept of Degree of Dynamism. Berbeglia et al. [2010] mention the degree of dynamism (DoD) as the ratio of dynamic requests to the total number of requests. We assume that the average demand is realized with a certain probability, say, p and varies from expected demands symmetrically with probability $1-p$. Due to symmetric variation, the average number of dynamic demands over all instances remains the same and we have the degree of dynamism as $1-p$.

6.1. Comparison of Algorithms using Example 3.1

For the example studied above, we compare the solutions of different heuristics with respect to these performance parameters. In table 9 we can see results of the exact and two heuristic methods and also the solution obtained from *Hu*. However, as the demand satisfied by *Hu* is different from the one used in *Ic* solution, we do not compare the two solutions.

	Ic	***ItHs***	***HtHs***	***Hu***
Revenue	1211	1162.5	1152.75	1071.25
Revenue Ratio	1	0.96	0.95	NA
Number of Vehicles	8	9	9	9
Number of Vehicles Ratio	1	1.25	1.25	NA
(Total running time, Loaded run time)	(181,150)	(187,150)	(190150)	(178,147)
Utilization (U)	0.83	0.8	0.79	0.83

Table 9: Comparison of Methods for Example 3.1

As the *Ic* solution is taken as a standard for comparison, its *RR* and *NVR* equals 1. The *RR* of *HtHs* is less than that of *ItH*. The *NVR* of both solutions is the same indicating that *HtHs* does not lead to an increase in the number of vehicles in this case.

From the values of *U* we see that out of total empty movement minimum 17 percent empty running is inevitable as per the optimal solution. The *ItHs* solution has 20 percent and *HtHs* solution has 24 percent empty running in the overall cycle time. Due to variation in demand *Hu* has different total loaded running but it gives better utilization by eliminating unnecessary empty movements.

7. Computations

In order to test the performance of the heuristic methods, we compare them with the exact solutions obtained by Ic using parameters given in section 6. Since ILP takes more time to solve with the increasing number of demands, we make a comparison on instances with small demand size. Later we apply the heuristic to large sized problems to ensure that the heuristic gives feasible solutions in a reasonable time. The ILPs are modeled in PuLP in Python 2.7.13 and solved by COIN-OR solver. The other algorithms are also modeled and simulated in Python. All the computations are performed on a machine equipped with 4 Intel Xeon 2.13 GHz cores and 64 GB RAM.

All the reviewed papers concerning FTLPDP perform calculations over randomly generated data sets where the network is extracted either from some TSP instances or generated randomly or from similar to real test region, over which the pickup and delivery customers are generated randomly. Thus, no standard PDP demand data instance is available for comparison of our heuristic with the existing results. We generate multiple instances, by constructing several graphs with the number of nodes varying from 5 to 8 over a 100 X 100 grid. For each graph, we generate 10 demand instances by increasing the maximum number of demands over each arc from 1 to 4. Thus the demands vary from 10 to 100 overall 160 (i.e. 4X4X10) generated instances. We compare the heuristic solutions given by *ItHs* and *HtHs* with the exact solution obtained by *Ic* for every instance. However, as the number of demands increases, the time to solve the exact solution increases both in *Ic* (Multiple vehicle optimal solution) and *ItHs* (Single vehicle optimal solution). With demands more than 80 the ILPs take more than 30 mins to find the optimal solution. The *HtHs* heuristic gives a feasible solution within a minute even for a large number of demands.

7.1. Results

The *Revenue Ratio* of the two heuristic solutions with varying demand is shown in Figure 2. We can see that at smaller demand ranges, the *ItHs* solution is quite close to *Ic* solution. However, it decreases with increasing demand. The *HtHs* solution is further away from *Ic* by less than ten percent. Also as the demand increases, the difference between *ItHs* and *HtHs* solution also increases. Further, in Figure 3, we can see the variation of *NVR* with demands i.e. the total number of requests in the system. The number of vehicles is the same for all three methods at lower demand instances. However, with increasing demands it moves further away from the optimal values. The solution of *ItHs* is closer to *Ic* as compared to *HtHs*. As the demand increases the gap between the two solutions increases. However, for the given range, it is less than 1.25 times the optimal vehicle number. Thus if 10 vehicles are required in the optimal solution, the heuristic solution may require 11-13 vehicles.

The utilization parameter is shown in figure 4 as a ratio for the comparison of different methods. As *Ic* gives the optimal average utilization value at every instance, we report the ratio of utilization factor in heuristic to that with an optimal solution, as Utilization Ratio (*UR*). At lower demand values, the utilization of *ItHs* heuristic is closer to that of *Ic* which implies that it allows the only minimum required empty movements. The *HtHs* heuristic demonstrates a reduction in performance with increasing distance of empty movement, however, the performance is close to *Ic* for all given demands. Also, we see that with increasing problem size the empty movements increase in both *ItHs* and *HtHs* solutions.

7.2 Performance with Variations in Observed Demand

The initial cycles are planned based on average demands which may be realized with a certain probability. We then devise an algorithm H_u for updating the predetermined cycles. To study the effect of dynamism, we consider a network with 5 nodes and approximately 60 demands and form cycles over this network using H_tH_s heuristic given above. We generate demands using a triangular probability distribution. We assume that the average demand is realized with a certain probability, say, p that varies from expected demands by +1 or -1 with $1-p$ = 2/3 probability and by +2 or -2 with $1-p$ =1/3. The degree of dynamism here is $1-p$. We generate 100 different demand instances for different values of p. We determine the average revenue over these instances and compute its standard deviation. The variation of net revenue with $(1-p)$ is shown in figure 5. The average revenue remains constant as we have the same average demands but the deviation shows the variation over multiple instances. For DoD = 0, the solution is our heuristic solution for all cases as there is no change in demand. The deviation increases with increasing DoD. The fluctuation in the number of vehicles due to the variation of demand is reflected in figure 6. The increased demands can be managed within the existing number of vehicles as planned if DoD is less than 0.3.

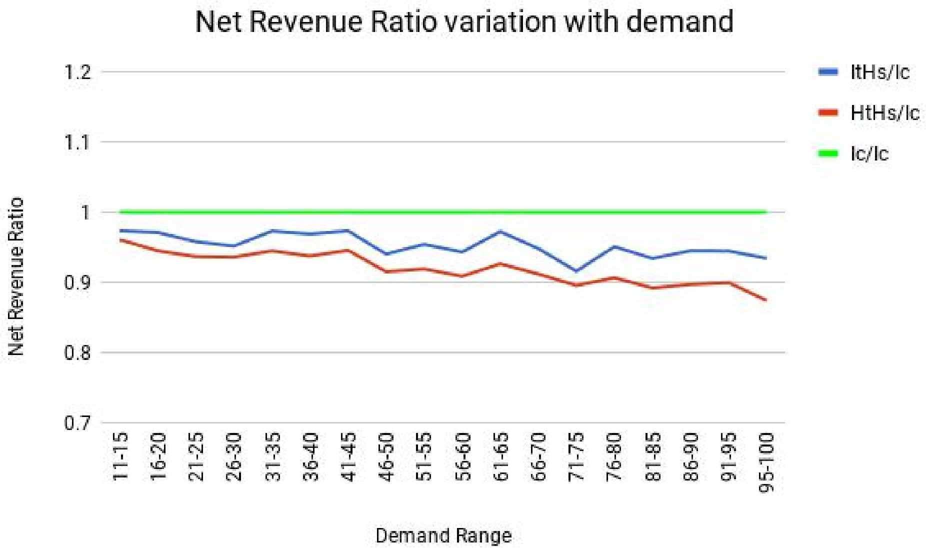

Figure 2: Net Revenue Ratio variation with demand

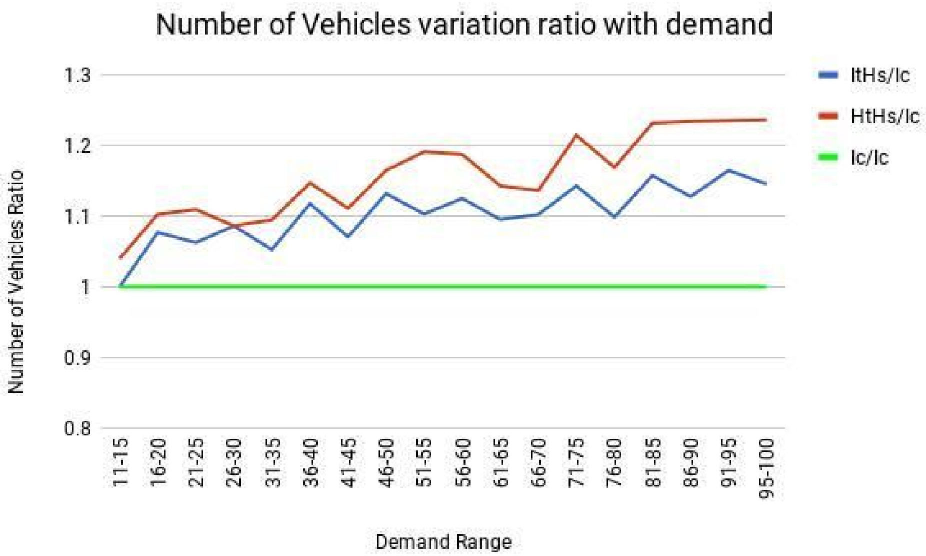

Figure 3: Ratio of Number of vehicles variation with demand

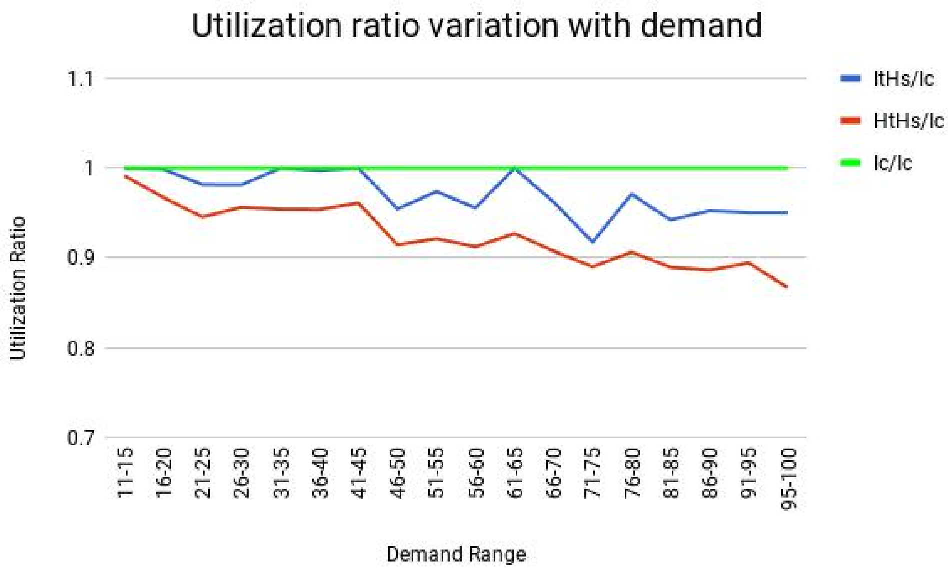

Figure 4: Average Utilization time variation with demand

We solve the HtHs heuristic multiple times with different *lv* values and determine the best solution. If we try to apply it to the new demands every time, we need to solve it multiple times to obtain a solution comparable to the existing one. However in the current case, we only modify the solution obtained previously.

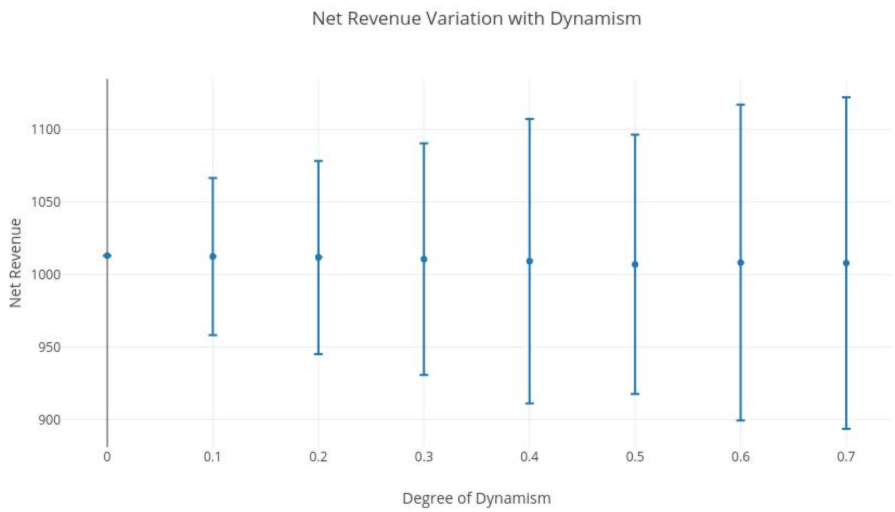

Figure 5: Revenue variation for varying demand data

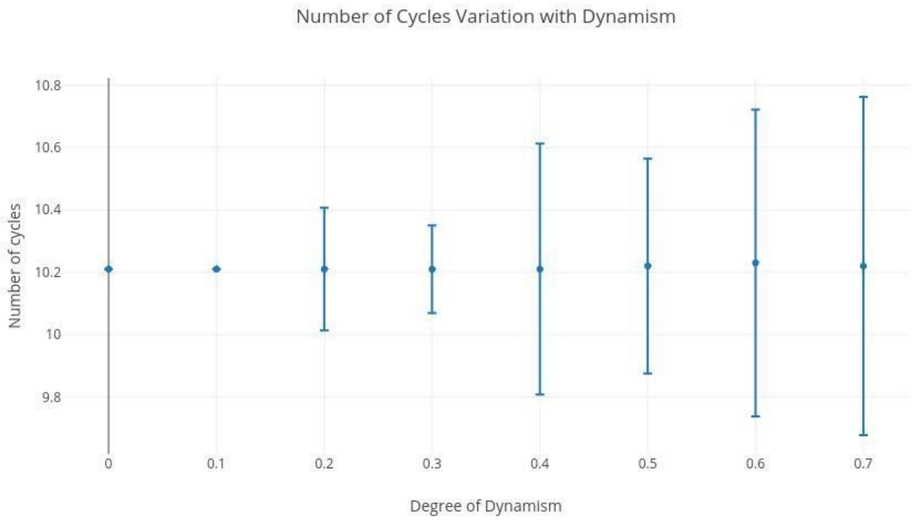

Figure 6: Number of Cycle variation for varying demand data

8. Application of the model to real-life examples

We studied the performance of the heuristic methods against the optimal solution for small sized data sets. Our aim is to implement this heuristic as a decision support system for some real-life applications. One to One Full Truck load Pick up and Delivery Demand structure is mainly seen in two major freight movement applications.

The first example is the transportation of commodities like coal via freight trains, where the rakes, associated with certain depots, are used to pick loads from mines and transport them to powerhouses and have to return to the depot for maintenance within a given time frame. Here each location is classified as either a loading or an unloading location and the vehicle travels empty from the unloading to the next loading point. The second example is that of movements of containers by trucks, where international containers have to be moved from port or depots to customers and vice versa and domestic containers have to be moved between two specified locations in the network. The nodes are not distinguished as loading or unloading locations in this application.

In both these cases, currently, the decisions are taken based on the availability of the resources. Despite there being predictable demands, there is no advance planning and the available resources at a given time are assigned to demands arising in the system. Proper route planning for the long term would lead to better utilization of resources as well as improved service levels. We generate test cases replicating the network and demands in these applications and apply our algorithm to these two cases with different demand patterns to demonstrate the feasibility of the algorithm.

8.1 Application to Freight Railways

A typical example of this type where the demands are at large predictable and constant is coal movement on Indian Railways. The example below shows average demands from pick up locations *L* to delivery locations *U*. The depot node 1 is the maintenance node and there are no demands to and from this node.

The actual railway network consists of more than 50 locations. However, it would be difficult to present the results of such large scale model here. So we consider a similar, but a smaller sized problem to define the problem. For this, we first generate a network with 15 nodes by randomly generating points on a plane. Next we determine set of locations like coal mines where loading occurs (L = [2; 4; 6; 8; 9; 12; 13; 14]) and the power plants where unloading takes place (U = [3; 5; 7; 10; 11; 15]). We generate random demands from L to U. We have a total of 67 demands over the network. As the rakes need to be maintained every 30 days we consider maximum cycle time to be 30 days. The travel time is also measured in days and is proportional to the distance between two locations. We use the H_tH_s heuristic to find the optimal routes for the given data. The results are as shown in table 10. As per the solution, we need 23 vehicles to satisfy 67 demands within 30 days. We see that every loaded trip is followed by an empty run. Also, as there are no demands from maintenance depot which leads to empty trips to and from the depot in every cycle. In some cycles, the net revenue is negative due to more empty travel of rakes than loaded, but the net revenue from all cycles is positive. If the restriction of serving all demands is removed we can skip the low revenue cycles which will reduce the number of vehicles required and also avoid unprofitable cycles.

No.	Cycle	Movement	Time	Revenue
1	1, 8, 15, 2, 3, 12, 10, 2, 3, 1	e, l, e, l, e, l, e, l, e	29	303
2	1, 13, 11, 13, 11, 1	e, l, e, l, e	20	127
3	1, 6, 15, 12, 15, 9, 5, 9, 5, 1	e, l, e, l, e, l, e, l, e	25	359.75
4	1, 14, 5, 2, 10, 6, 11, 1	e, l, e, l’, e, l, e	26	140
5	1, 6, 10, 14, 11, 1	e, l, e, l, e	19	228.75
6	1, 6, 5, 4, 3, 2, 5, 1	e, l, e, l, e, l, e	27	428.75
7	1, 9, 10, 12, 10, 8, 7, 1	e, l, e, l, e, l, e	23	301.25
8	1, 4, 5, 4, 7, 1	e, l, e, l, e	27	332.5
9	1, 12, 3, 14, 10, 9, 3, 13, 15, 1	e, l, e, l, e, l, e, l, e	30	244.5
10	1, 12, 3, 9, 11, 8, 3, 6, 11, 1	e, l, e, l, e, l, e, l, e	29	346.5
11	1, 4, 7, 6, 7, 1	e, l, e, l, e	25	310.5
12	1, 13, 7, 13, 10, 2, 11, 1	e, l, e, l, e, l, e	30	375.25
13	1, 4, 11, 4, 5, 1	e, l, e, l, e	24	359.75
14	1, 9, 15, 2, 15, 12, 11, 8, 5, 1	e, l, e, l, e, l, e, l, e	20	42
15	1, 13, 3, 13, 3, 1	e, l, e, l, e	21	86
16	1, 6, 3, 2, 11, 6, 7, 1	e, l, e, l, e, l, e	25	614.25
17	1, 8, 3, 14, 7, 14, 7, 1	e, l, e, l, e, l, e	25	454.25
18	1, 13, 5, 6, 10, 1	e, l, e, l, e	20	278.75
19	1, 8, 10, 14, 15, 14, 15, 9, 15, 1	e, l, e, l, e, l, e, l, e	25	-26.5
20	1, 14, 5, 13, 10, 6, 5, 1	e, l, e, l, e, l, e	29	421.5
21	1, 12, 7, 14, 3, 2, 15, 1	e, l, e, l, e, l, e	24	55.75
22	1, 9, 7, 4, 3, 1	e, l, e, l, e	22	112.75
23	1, 4, 10, 14, 11, 12, 15, 1	e, l, e, l, e, l, e	28	301
		Total	573	6197.25

Table 10: Cycles for freight railways example

The above cycle plan ensures that the rakes return to the depot on time for maintenance and satisfy the average demand per month.

8.2 Application to Container Movements

The inland container movement is carried out by trucks, which could be based at some terminal. The terminal can be located near a port or container depot to reduce excessive movements of trucks from its base to the depot. This node, together with a number of nodes in the vicinity, represent various Container Freight Stations (CFS) operated by the different container handling agencies generally located near the port. There is also is a set of hinterland nodes, representing manufacturing units or Inland Container Depots. The truck delivering international containers can also be used to transport domestic loads from nearby locations. Thus, from an unloading location where the container is dropped off, there could be pick up of another empty container or domestic loaded container. Both the loaded and the empty containers contribute to the load of the truck and an empty movement would mean relocating the truck without any container. Thus, the consecutive loaded running of trucks can be achieved without the need to travel empty if the routes are planned in advance. Here, the trucks start from the depot and move around the network to satisfy internal demands and to the depot after a certain time period based on the crew and maintenance requirements. Certain demands are repetitive over a given time period and the customers are classified and priced differently by container carrying companies owing to the long term contracts.

We construct a network by generating 15 nodes and considering the distances between them. We generate 8 nodes within a small area and the remaining nodes scattered away from it to replicate immediately near the port and some at inland locations. There could be demands from depots to customers, for empty repositioning of containers between depots and domestic container movements between customers. Also, each location can be a pickup or a delivery location, unlike the previous example. We have a total of 89 demands. The cycle time, in this case, is 15 days within which the truck has to return to the depot. Using *HtHs* heuristic we obtain the optimal routes for this case are shown in table 11. To satisfy the entire demand over 15 days 25 vehicles are required.

No.	Cycle	Movement	Time	Revenue
1	1, 6, 3, 14, 5, 4, 1	l, l, l, e, l, l	13	894.5
2	1, 2, 11, 3, 15, 9, 11, 6, 1	e, l, l, l, l, l, l, e	14	716.25
3	1, 7, 14, 3, 13, 6, 1	e, l, e, l, l, e	15	590
4	1, 7, 9, 2, 15, 1	e, l, l, l, e	13	480
5	1, 10, 4, 10, 7, 1	l, l, l, l, e	13	1040.5
6	1, 5, 13, 7, 2, 12, 7, 1	e, l, l, l, l, l, e	13	948.75
7	1, 5, 14, 1, 8, 5, 3, 1	e, l, l, e, l, l, l	12	806.25
8	1, 10, 6, 15, 5, 1	e, l, l, l, e	14	717.75
9	1, 6, 12, 15, 7, 10, 1	e, l, l, l, l, e	14	583.5
10	1, 14, 10, 5, 10, 1	l, l, l, l, l	15	1278.5
11	1, 11, 13, 5, 11, 4, 5, 1	e, l, l, l, l, l, e	15	782.75
12	1, 14, 11, 10, 11, 1	e, l, e, l, e	13	-98
13	1, 9, 7, 8, 14, 1	l, l, l, l, e	15	725.75
14	1, 4, 3, 4, 8, 15, 1	l, e, l, l, l, e	14	431.25
15	1, 10, 13, 2, 1	e, l, l, e	10	188.25
16	1, 2, 13, 11, 12, 14, 12, 1	e, l, l, e, l, l, l	13	720.25
17	1, 2, 14, 4, 12, 8, 1	l, l, l, l, l, e	14	986.5
18	1, 13, 1, 5, 2, 7, 1	e, l, e, l, l, l	12	406
19	1, 2, 10, 14, 8, 1	e, l, l, l, e	12	615.25
20	1, 3, 12, 5, 12, 9, 13, 12, 1	e, l, l, l, l, e, l, e	15	503
21	1, 13, 15, 11, 1, 2, 8, 1	e, l, e, l, e, l, l	13	86.75
22	1, 6, 9, 3, 10, 2, 1	l, l, l, l, l, e	14	995.25
23	1, 3, 6, 7, 15, 12, 11, 2, 1	e, l, l, l, e, l, l, l	14	692.25
24	1, 15, 13, 9, 3, 2, 8, 6, 2, 4, 6, 1	e, l, l, e, l, e, l, l, l, l, e	14	65.25
25	1, 7, 5, 7, 3, 11, 9, 5, 6, 1	e, l, e, l, l, l, e, l, e	12	146.25
		Total	336.0,	15302.75

Table 11: Cycles for container example

8.3 Observation

By the above two examples, we demonstrate the applicability and performance of the solution technique to real life problems. Since the data is randomly generated and not from any specific case study, we do not make any implications regarding the results here. We observe that the model gives expected results under different demand patterns. However, the time taken to solve the problem depends on the demands in the network. The solution of rake movement example is obtained in 20 seconds, whereas the container movement problem is solved in a one to two seconds. This time difference is due to the fact that in the absence of subsequent loads the algorithm has to decide the location where the empty vehicle has to be moved.

Though we attempt to use the current method only to plan movements and vehicle requirement in the current example, this can also be further extended to determine the best location of the depot by running the heuristics multiple times on different location of the depot and choosing the most profitable one. Such decisions cannot be made by using the exact methods as the time required to solve it might be quite large for this data size even for a single depot location.

9. Conclusion

In this paper, we consider a Full Truckload One to One Pickup and Delivery Problem with multiple demands. Each demand can be satisfied by multiple trips of a single vehicle or by multiple vehicles. The freight movements are generally not planned in advance due to uncertainties in the demand and travel time. In most cases, the demands are repetitive over a certain time period. It could be beneficial to plan movements to avoid the repeated decision-making process. We use the ILP to find an exact solution to the problem and develop two Order First Split Second Heuristics I_tH_s and H_tH_s. We first determine cycles based on average demand. Due to possible variations in actual demands, we have recourse function to update cycles according to the new demand.

With the increasing size of the problem, the time to obtain exact solution increases. The net revenue from heuristic is close to 0.9 times the exact solution and is calculated within much lesser time. Similarly, the number of required is 1.25 times the optimal value. However, the average cycle time increases with demands due to empty movements. The utilization ratio that quantifies the empty movements beyond the minimum in heuristic shows that the solution is also close to 0.9 times the optimal solution.

Also, for a smaller degree of dynamism in the system, the variation in demands can be managed with the existing number of vehicles by slight modifications in existing cycles. The plans made can be implemented with minor changes if the degree of dynamism of the system is less than 0.3.

Besides providing tactical-level planning of long term cyclic paths and resources required to real life large sized problems within a small time period, the H_tH_s heuristic also can be used for strategic planning determining the location of depots. The heuristic H_u that updates the cycles based on actual demand can be used to make the operational level decisions per time period without modifying the initial plan completely. Following planned routes facilitates managing maintenance cycles, crew movements and other activities.

As a part of the further analysis, we plan to apply this approach to the actual data from Indian Freight Railway movements and quantify the effect of long term cycle planning. Further, we aim to perform simulations with uncertain demand data to test the outcome of periodical changes in the path using the Updating Algorithm. The modified OFCS can be further extended to solve different variants of the PDP including time windows for pick up and deliveries or associating vehicles to multiple depots. This approach can be adapted in container movement and freight railways planning to give near-optimal solutions within a reasonable time. The planned routes will provide an added benefit over existing manual processes by eliminating the human factors in decision making leading to uniform decision patterns, less empty running and improved service levels for the final customers.

References

R. Baldacci, E. Bartolini, and A. Mingozzi. An exact algorithm for the pickup and delivery problem with time windows. Operations Research, 59(2):414–426, 2011. doi: 10.1287/opre.1100.0881. URL https://doi.org/10. 1287/opre.1100.0881.

G. Berbeglia, J.-F. Cordeau, and G. Laporte. Dynamic pickup and delivery problems. European Journal of Op-erational Research, 202(1):8 – 15, 2010. ISSN 0377-2217. doi: https://doi.org/10.1016/j.ejor.2009.04.024. URL http://www.sciencedirect.com/science/article/pii/S0377221709002999.

N. Bianchessi and G. Righini. Heuristic algorithms for the vehicle routing problem with simultaneous pick-up and delivery. Computers and Operations Research, 34(2):578 – 594, 2007. ISSN 0305-0548. doi: https://doi.org/10. 1016/j.cor.2005.03.014. URL http://www.sciencedirect.com/science/article/pii/S0305054805001097. Re-verse Logistics

L. J. J. Bruggen, J. K. Lenstra, and P. C. Schuur. Variable-depth search for the single-vehicle pickup and delivery problem with time windows. Transportation Science, 27(3):298–311, 1993. doi: 10.1287/trsc.27.3.298. URL https://doi.org/10.1287/trsc.27.3.298

A. Caris and G. Janssens. A local search heuristic for the pre- and end-haulage of intermodal container terminals. Computers and Operations Research, 36(10):2763 – 2772, 2009. ISSN 0305-0548. doi: https://doi.org/10.1016/j.cor.2008.12.007. URL http://www.sciencedirect.com/science/article/pii/S0305054808002633

R. H. Currie and S. Salhi. A tabu search heuristic for a full-load, multi-terminal, vehicle scheduling problem with backhauling and time windows. Journal of Mathematical Modelling and Algorithms, 3(3):225–243, Sep 2004. ISSN 1572-9214. doi: 10.1023/B:JMMA.0000038616.99798.f2. URL https://doi.org/10.1023/B:JMMA. 0000038616.99798.f2.

A. Fabri and P. Recht. On dynamic pickup and delivery vehicle routing with several time windows and waiting times. Transportation Research Part B: Methodological, 40(4):335 – 350, 2006. ISSN 0191-2615. doi: https://doi.org/ 10.1016/j.trb.2005.04.002. URL http://www.sciencedirect.com/science/article/pii/S0191261505000585

Gronalt, R. F. Hartl, and M. Reimann. New savings based algorithms for time constrained pickup and delivery of full truckloads. European Journal of Operational Research, 151(3):520 – 535, 2003. ISSN 0377-2217. doi: https://doi.org/10.1016/S0377-2217(02)00650-1. URL http://www.sciencedirect.com/science/article/pii/ S0377221702006501.

P. Lacomme, C. Prins, and W. Ramdane-Cherif. Competitive memetic algorithms for arc routing problems. Annals of Operations Research, 131(1):159–185, Oct 2004. ISSN 1572-9338. doi: 10.1023/B:ANOR.0000039517.35989.6d. URL https://doi.org/10.1023/B:ANOR.0000039517.35989.6d.

J. Li and W. Lu. Full truckload vehicle routing problem with profits. Journal of Traffic and Transportation Engineering (English Edition), 1(2):146 – 152, 2014. ISSN 2095-7564. doi: https://doi.org/10.1016/S2095-7564(15) 30099-4. URL http://www.sciencedirect.com/science/article/pii/S2095756415300994.

G. Mosheiov. Vehicle routing with pick-up and delivery: Tour-partitioning heuristics. 34:669–684, 07 1998.

S. N. Parragh, K. F. Doerner, and R. F. Hartl. A survey on pickup and delivery problems. Journal of Betriebswirtschaft, 58(1):21–51, Apr 2008a. ISSN 1614-631X. doi: 10.1007/s11301-008-0033-7. URL https: //doi.org/10.1007/s11301-008-0033-7.

S. N. Parragh, K. F. Doerner, and R. F. Hartl. A survey on pickup and delivery problems: Part ii: Transportation between pickup and delivery locations. Journal für Betriebswirtschaft, 58:81–117, Jun 2008b.

C. Prins, P. Lacomme, and C. Prodhon. Order-first split-second methods for vehicle routing problems: A review. Transportation Research Part C: Emerging Technologies, 40:179 – 200, 2014. ISSN 0968-090X. doi: https://doi. org/10.1016/j.trc.2014.01.011. URL http://www.sciencedirect.com/science/article/pii/S0968090X14000230.

A. Quilliot, M. Lacroix, H. Toussaint, and H. Kerivin. Tree based heuristics for the preemptive asymmetric stacker crane problem. Electronic Notes in Discrete Mathematics, 36:41 – 48, 2010. ISSN 1571-0653. doi: https://doi.org/ 10.1016/j.endm.2010.05.006. URL http://www.sciencedirect.com/science/article/pii/S1571065310000077. ISCO 2010 - International Symposium on Combinatorial Optimization.

G. Ulusoy. The fleet size and mix problem for capacitated arc routing. 22:329–337, 02 1985.

X. Wang and A. C. Regan. Local truckload pickup and delivery with hard time window constraints. Transportation Research Part B: Methodological, 36(2):97 – 112, 2002. ISSN 0191-2615. doi: https://doi.org/10.1016/ S0965-8564(00)00037-9. URL http://www.sciencedirect.com/science/article/pii/S0965856400000379.

X. G. Yu and X. P. Yu. An approach on solving the real time dynamic pickup and delivery problem. In 2007 IEEE International Conference on Control and Automation, pages 581–585, May 2007. doi: 10.1109/ICCA. 2007.4376422.