Logistics Research (2019) 12:4
DOI_10.23773/2019_4

Optimization of the Heterogeneous Vehicle Routing Problem with Cross Docking Logistic System

N. Al Theeb, O. Al-Araidah, M. H. Aljarrah

Received: 20 May 2018 / Accepted: 20 March 2019 / Published online: 08 April 2019

ABSTRACT

This paper introduces a new mixed integer model to provide efficient distribution plans to route a set of inbound/outbound heterogeneous vehicles in the cross-docking systems that are used to transfer different types of commodities from a manufacturing plant to retail warehouses with allowed split deliveries. The objectives are to minimize the total commodity deviations and the overall distribution time or cost of vehicles. Different data sets from different scales are randomly generated and solved by CPLEX-Concert Technology. As the proposed model is NP-hard; a new heuristic is constructed to solve the problem in reasonable computational efforts. Results show that CPLEX can obtain optimal solutions for small-scale sets and sub-optimal solutions for medium scale sets with a six-hour time limit, while it fails to provide any feasible solution for 35 out of 50 large scale sets. The proposed heuristic provides very competitive results compared to CPLEX in all scales in term of gaps and computational times.

KEYWORDS: Logistics · Cross dock · Vehicle routing · Integer programming · Heuristic

Nader Al Theeb (corresponding author),
naaltheeb@just.edu.jo

Omar Al-Araidah,
alarao@just.edu.jo@just.edu.jo

Manar H. Aljarrah,
ammanar14@eng.just.edu.jo

Industrial Engineering Department
Jordan University of Science & Technology
P.O. Box (3030) Irbid 22110 Jordan
Phone: +962 2 720 1000 Ext. 22538

1 INTRODUCTION

Recently, many industries have focused on improving the efficiency of logistics and distribution systems of material flow to satisfy more complicated customer demands in terms of timing, cost and quality (Lee et al. (2006) and Serrano et al. (2016)). For example, more than 30% of sales is incurred in the distribution process as stated by Apte and Viswanathan (2000). Therefore, one of the most important things in controlling logistics and distribution costs and reducing waste produced by high inventories compared to traditional warehousing systems while simultaneously maintaining the level of customer satisfaction is CD system (Sung and Song (2003)). Cross docking (CD) can be defined as moving commodities from a manufacturing plant and delivering them directly to a retail chain with little or no material handling or storage time in between. CD can decrease the warehousing costs up to 70 percent due to the reduction in storage space and order picking costs (Vahdani and Zandieh (2010)). Moreover, it can decrease the transportation costs due to the use of full trucks by consolidating different shipments (Apte and Viswanathan (2000)). In CD, commodities transported to the CD by inbound vehicles, are de-batched and re-batched in different quantities and combinations and then transported to their destinations by outbound vehicles in a short time, usually less than 24 h (Apte and Viswanathan (2000) and Kreng and Chen (2008)).

Two critical requirements of CD are simultaneous arrival and consolidation. Simultaneous arrival of the pickup vehicles fleet, consolidation can be easily achieved in a supply chain physical flow. Vehicles can leave the CD to distribute products to their destinations without any interruptions, and thereby the lead-time for delivery and the inventory level will be reduced. On the contrary, if pickup vehicles fleet cannot arrive at the CD simultaneously, the consolidation process is delayed until all products are collected and classified and then loaded to each vehicle in CD. Therefore, the waiting time of some vehicles and the inventory level is increased (Lee et al. (2006) and Liao et al. (2010)).

Although many studies on CD and vehicle routing problem (VRP) have considered them separately, dealing with both simultaneously is very important and common in practice, as proposed by Lee et al. (2006). The integration of VRP with CD strategy (VRPCD) has been increasingly appreciated and investigated in recent studies as an effective strategy for distribution management and logistics. Furthermore, most studies on the VRPCD problem have assumed that pickup and delivery tasks are sequentially carried out by a homogeneous vehicle fleet. However, few of them have assumed that the pickup and delivery tasks are sequentially carried out by a heterogeneous vehicle fleet. Since all the vehicles are non-identical, the assignment of vehicles to routes is required (Dondo and Cerdá (2015)). In a distribution system with CD and a heterogeneous fleet, with vehicles of different sizes and capacities, large quantities of inbound commodities are transported from plants using large vehicles to CDs where small vehicles are awaiting to be replenished, and then small vehicles transport outbound commodities to customers. This enhances the speed of delivery by avoiding large vehicles congestions inside cities and hence improves the overall efficiency of the distribution system.

This research effort aims to provide the following contributions. First, this research effort provides a realistic model that can be used efficiently in VRPCD problems. A new optimization model is proposed to provide detailed and efficient distribution plans for vehicles that are used to transfer commodities from manufacturing plants to retail warehouses. From literature, it can be clearly noticed that research that combines vehicle routing with different logistics decisions in cross docking systems is not fully investigated. The model in this research addresses some of shortcomings appear in literature by providing detailed optimal route and schedule for each vehicle that minimizes the total commodities deviations and the overall distribution cost alongside with quantities. Deviations represent the difference between the requested and delivered demand. Additionally, the proposed model is developed with many extensions from other models in the literature such as inbound/outbound heterogeneous fleet of vehicles with different capacities and speeds, multiple commodity types with different masses and quantities, multiple inbound/outbound (strip/stack) doors for CD with limited capacity, multiple replenishment, split deliveries, and deviation variable are added to represent the difference between desired demand and actual delivered quantities.

The second contribution is to develop an efficient solution approach that can provide high quality solutions for the proposed model in acceptable computational efforts. The developed solution approach will pass through two phases. In the first phase, a route will be constructed for each vehicle in a greedy manner. Then, in the second phase, the commercial software, i.e., CPLEX, will be used to solve other variables based on the constructed routes (fixed binary variables).

The remainder of this paper is organized in the following manner. In the following section, a review of relevant literature is presented. The problem statement and the mathematical formulation for parameters, decision variables and constraints used in developing the mixed integer linear programming formulation for the model are clarified in Section 3. In Section 4, a heuristic approach for solving the resulting model in reasonable computational effort is explained. Section 5 includes a numerical analysis of the mathematical model for different data sets. Finally, summarizing our work with suggestions for future research directions are given in Section 6.

2 RELATED LITERATURE

CD is an important strategy in logistics that benefits the supply chain in managing the flow of materials from suppliers to customers efficiently to improve customer satisfaction. CD reduces warehouse space requirements, reduces inventory handling risks, and hence reduces associated inventory and transportation costs (Apte and Viswanathan (2000)). Several CD problems were studied in literature including operational, tactical and strategic problems. Major decisions at the operational level are related to truck scheduling at the CD, dock door assignment and pickup/delivery vehicle routing and scheduling; tactical decisions are concerned with network flows, vehicle routing and distribution planning; and strategic decisions are focused on geographical location and shop-floor layout. The scheduling problems at cross docks with multiple doors were studied by (Song and Chen (2007)). The problem was limited to one CD with multiple inbound doors with a single outbound door. Larbi et al. (2009) considered multiple inbound and outbound dock doors as well as proposed heuristic methods to find the best solution for transshipment operations scheduling, hence, minimize the sum of inventory holding and truck replacement costs. Serrano et al. (2016) focused on operational and tactical issues in their research. The authors proposed a mixed integer linear programming model to treat the operation scheduling problem at the CD platform. This involved scheduling truck arrivals, shop-floor operation and truck departures to minimize the internal operation and outbound transportation costs. More review details of the latest research on CD problems can be found in (Boysen and Fliedner (2010), Agustina et al. (2010) and Van Belle et al. (2012)) articles. These articles reviewed what have been done in improving the CD operation from operational to strategic aspects in the previous researches.

There are many other problems in logistics, that utilize the cross docking operations, were considered in the past research. For example, the pickup and delivery with cross docking problem, which was formulated by Cortés et al. (2010) and Santos et al. (2013). The model of Santos et al. (2013) utilize a pre-defined routes that should be selected by the set of vehicles the are used to fulfill set of demand requests, i.e., single type of demand is considered. In their proposed model, split delivery is not allowed, homogeneous vehicles are used, no capacity limitations for the cross dock are considered, and branch and price is developed to solve the problem. Similarly, Cortés et al. (2010) developed pickup and delivery model for passengers transfer, but not for goods. Grangier et al. (2017) provide solution approach to the problem of pickup and delivery with cross docking systems by large neighborhood search and set partitioning based algorithm. The same problem was solved by Bodnar et al. (2015) with the same approach, but with addition of periodic calls. Other problems in logistics are also considered, as in Maknoon et al. (2017) who have provided a formulation for an integer model to minimize the cost of cross docking operations by making a proper schedule for the trucks at the terminal doors, where each truck is assigned for only one order (request).

Most studies investigated CD and vehicle routing problem VRP separately. However, the integration of VRP with CD strategy (VRPCD) has been increasingly appreciated and investigated in recent studies as an effective strategy for distribution management and logistics. The first study which took into consideration the VRPCD integration model was proposed by Lee et al. (2006). The authors investigated a variant of the VRP with synchronous arrival times of products and stable demand for consolidation. A tabu search (TS) algorithm was proposed to find the optimal number of vehicles and routing schedule with minimum transportation cost. Results from the proposed algorithm were compared to those obtained by enumeration methods; attained results were near optimal with an average percentage error of less than 4% of total cost within a reasonable amount of time. A new TS algorithm was proposed by Liao et al. (2010) for the set of VRPCD problems that were introduced by Lee et al. (2006) to minimize the sum of transportation and operational costs. The results showed improvements as high as 10–36% for various sizes of problems compared to the results obtained by Lee et al. (2006). Agustina et al. (2014) integrated routing, CD and scheduling to the distribution of food products to ensure that food products which are perishable with a short shelf life can be delivered to customers just in time to preserve their quality. A mixed integer linear programming (MILP) model was formulated in CPLEX software with the aim of minimizing inventory holding and transportation costs, as well as penalty costs of early and delayed deliveries. Dondo and Cerdá (2014) introduced a monolithic MILP formulation for VRPCD problems that integrated pickup/delivery points for a homogeneous fleet of vehicle routing and scheduling with both the assignment of inbound/outbound vehicles and the management of vehicle queues at strip/stack dock-doors. Additionally, Wen et al. (2009) considered the same VRPCD problem in (Lee et al. (2006)) with asynchronous arrival times for all homogeneous vehicles and an objective of minimizing the total distance traveled. Also, Tarantilis (2013) addressed the same problem presented by (Wen et al. (2009)) while considering the use of different inbound and outbound vehicles for pickup and delivery processes. An adaptive multi-restart procedure associated with a TS algorithm was applied, where the results showed better solution compared to the results obtained by Wen et al. (2009). Moreover, Morais et al. (2014) proposed three iterated local search solutions for (ILS) heuristics to solve VRPCD. The ILS heuristics provided the best solution and outperformed the TS heuristic and the adaptive multi-restart TS heuristic with better solutions than those of Wen et al. (2009) and Tarantilis (2013).

Other studies assumed that the pickup and delivery tasks are sequentially carried out by a heterogeneous vehicle fleet. As vehicles are not identical, the assignment of vehicles to routes is required. Hasani-Goodarzi and Tavakkoli-Moghaddam (2012) considered the vehicle fleet while splitting orders for deliveries and pickups to different nodes in the network. Also, these vehicles did not necessarily have the same capacity, so capacity constraints were applied to multi-product cross-docks. Dondo and Cerdá (2015) presented new solution approaches for VRPCD to determine truck scheduling, vehicle routing dock assignment all at once as well as the routing and scheduling of a heterogeneous fleet. Ahmadizar et al. (2015) considered two-level routing in a network, the first-level involved the routing of heterogeneous inbound vehicles between CDs and suppliers in the pickup process, and the second level involved the routing of heterogeneous outbound vehicles between CDs and retailers in the delivery process. The problem was modeled considering multiple product types while taking into consideration the possibility of the total volume assigned to a supplier (or demanded by a retailer) being greater than the capacity of inbound (or outbound) vehicle, so each supplier (or retailer) might be visited by several inbound (or outbound) vehicles. Birim (2016) developed a VRPCD model in which a heterogeneous fleet of vehicles without considering splitting orders for pickup and delivery processes is considered to find the routes that minimize the total distribution costs including pickup and delivery.

Determining the optimal solution from VRP is an NP-hard problem because of the combinatorial nature and complexity of this problem and is usually solved by heuristic approaches (Vincent et al. (2016)). Different heuristics were proposed to solve such models, including sweep-heuristic algorithm (Dondo and Cerda´ (2013)), iterated local search (ILS) (Morais et al. (2014)), tabu search (TS) (Lee et al. (2006)), particle swarm optimization (PSO) (Kachitvichyanukul et al. (2015)), simulated annealing (SA) (Wang et al. (2015)), etc. Recently, some studies employed hybrid heuristic algorithms which provided optimal or near-optimal solutions for large-scale NP-hard routing problems within a competitive computational time as in (Al Theeb and Murray (2017) and Küçükoğlu and Öztürk (2015)).

Our proposed model is a mixed integer linear programming model that builds on the classical NP-hard vehicle routing problem with the incorporation of other distribution decisions. Additionally, a new heuristic, different than the approaches that were used in literature, will be constructed to solve the resulting model in a reasonable computational effort based on solving the model iteratively and optimally in the last phase according to some specific variable values which are determined greedily a previous phase.

A summary of the previous studies which focused on the integration of VRP with CD strategy appears in Table 1, which also contrasts several key features of these studied with the model proposed in this paper. For more details about the warehouse research, including CD, Davarzani and Norrman (2015) provided a review article which includes the most recent research in this field.

3 PROBLEM DESCRIPTION AND MATHEMATICAL FORMULATION

The problem under consideration involves a distribution system consisting of a single manufacturing plant, one CD with multiple inbound and outbound doors, a heterogeneous inbound fleet of vehicles (trucks), a heterogeneous outbound fleet of vehicles, multiple commodity types with different masses and quantities and multiple customers. This distribution system can be described as follows. Each customer orders a specific quantity from each type of commodity at the beginning of a time horizon, then the plant makes decisions of how to fill out the available large vehicles and decides on quantities. After this, filled large vehicles travel to the CD where small vehicles wait to be replenished by commodities. Filled out small vehicles deliver their loads to customers warehouses and come back to CD to get more loads, if needed.

Several realistic assumptions are made in the proposed model. Large vehicles are used to transfer commodities from plant store to CD which keeps these large vehicles away from the congestions inside cities where such vehicles have restricted access. Because organizations always update and extend their distribution fleets, heterogeneous vehicles are expected, rather than homogeneous ones. The heterogeneity of vehicles is represented by different speeds and capacities. Each large vehicle is assumed to visit CD once during the time horizon, to transfer commodities picked up from the manufacturing plant. However, each small vehicle can revisit CD to replenish commodities, and thus deliver as many quantities as possible to customer nodes, and it can visit the same customer node more than once as well. This could reduce the vehicle's idle time and improve the overall efficiency of a distribution system.

Split deliveries of commodities through multiple vehicles at separate times are allowed in this system. If a customer demand exceeds the capacity of a vehicle, it is necessary to serve that customer more than once by a fleet of outbound vehicles. In contrast, in most VRP problems, it is assumed that the demand of each customer is less than or equal to the vehicle capacity, and thus each customer is served by exactly one vehicle.

Multiple types of commodities are available, such that each type may have a unique mass. Additionally, quantities of demanded commodity types that should be picked up and delivered are determined. It is important to note that not all demand can be satisfied due to some factors in the logistic system that may prevent the delivery of some units such as, vehicle capacity limitations, lack of functional manufacturing plants, CD availability issues, time window conflicts, or time horizon limitations. Therefore, the quantity of unsatisfied demand requested by a particular customer is taken into consideration to be determined in this system.

One of the crucial assumptions is that the commodities must pass directly from inbound to outbound dock doors. The reasons behind this is that the CD can not hold any inventory because of limited storage area, so both the inbound and outbound trucks must be located at CD simultaneously, which ensures that the appropriate quantities of commodities are transferred at the proper time. Thus, commodities are picked up by a fleet of heterogeneous inbound vehicles from the plant, consolidated at CD and immediately delivered to customers by a fleet of heterogeneous outbound vehicles, without intermediate storage. Therefore, the problem involves not only vehicle route design, but also a consolidation decision at the CD.

The objective of the proposed model is to determine the optimal routing and scheduling of inbound and outbound vehicle fleets by minimizing the total commodities deviations and the overall distribution time while managing the number of movements for each vehicle. In addition, it provides each outbound vehicle a preference to serve a prior commodity type customer combination to decrease the quantity of unmet demand for such case as much as possible. Finally, as the cost is considered as an important issue in such a problem, the proposed objective can indirectly help to cover this issue, as minimizing the unmet demands requires the vehicles to visit more customers with minimum distance to deliver more commodities, and thus minimizing the traveled distance (cost).

Table 1: A comparison between the proposed VRPCD model and existing literature

Author	Multiple Commodities	Multiple Replenishments from CD	Split Delivery	Multiple Time Periods	Different Vehicle Capacities	Different Vehicle Speeds	Different Fleets of Inbound/ Outbound vehicles	Multiple Inbound/Outbound Dock-doors	Limited Capacity of CD	Time Windows	Obser-vations
Lee et al. (2006)											Minimize transportation cost plus the fixed cost of vehicles
Wen et al. (2009)										✓	Minimize total travel time
Liao et al. (2010)											Minimize the sum of the operational cost of vehicles and the transportation cost.
Tarantilis (2013)							✓			✓	Minimize traveling distance
Dondo and Cerdá (2013)								✓		✓	Minimize total transportation cost
Mousavi and Tavakkoli-Moghaddam (2013)							✓		✓		Minimize the fixed and total transportation costs
Santos et al. (2013)										✓	Minimize the cost of routes and load changing costs
Dondo and Cerdá (2014)								✓		✓	Minimize the cumulative vehicle routing cost, cumulative distribution time, and the total makespan
Agustina et al. (2014)							✓	✓		✓	Minimize the earliness penalty cost of the orders, the tardiness penalty cost of the orders, the inventory holding cost of the products, and the transportation cost of delivery
Morais et al. (2014)										✓	Minimize the travel cost
Dondo and Cerdá (2015)					✓			✓		✓	Minimize the vehicle routing cost, the distribution time, and the total makespan
Ahmadizar et al. (2015)	✓		✓		✓		✓		✓		Minimize transportation costs, the purchasing, and holding costs
Küçükoğlu and Öztürk (2015)	✓									✓	Minimize the total distance or cost
Birim (2016)					✓						Minimize the fixed and total transportation costs
Maknoon et al. (2017)					✓						Minimize the total processing time
This paper	✓	✓	✓	✓	✓	✓	✓	✓	✓		Minimize the total prioritized commodities deviations (unsatisfied demand)

3.1 Notation – Parameters

In this subsection, the sets and parameter notations are presented as follows. The time horizon T is partitioned into discrete integer periods with equal length, such that $T = \{1, 2, \ldots, |T|\}$. Denote the set of commodity types by C and the amount of commodity type $c \in C$ demanded at node $i \in N$ by parameter d_{ic}. Each commodity type $c \in C$ has a mass of m_c and a priority value denoted as p_{ic}, such that p_{ic} is 2-dimensional which represents the priority of both the customer $i \in N$ and the commodity type $c \in C$. Parameter p_{ic} will have a higher value if a customer $i \in N$ has a high priority for the organization and a specific commodity type $c \in C$ is of great importance for that customer.

Further, the set N represents all nodes in the network including CD and the imaginary node, where $N = \{0, 1, 2, \ldots, N\}$. The CD is the first node in the N set and denoted by 0 which has multiple inbound and outbound doors given by D and D', respectively. Also, it has a limited storage capacity which is determined by the number of dock doors and the maximum loading capacity at each time $t \in T$, which is denoted by l and depends on the handling system. The imaginary node i' is the last node in the N set which is considered to facilitate the problem-solving process. It is used to force outbound vehicles to return to CD after delivery at the end of a planning period instead of performing unnecessary movements, which makes them available for reassignment in the next tour. The remaining nodes in the N set are customer nodes $i \in N$, where $i \neq 0 \& i'$. Each customer is able to receive the maximum unloading capacity at each time $t \in T$.

A fleet of heterogeneous inbound vehicles, defined by the set F, of different capacities w_f, is available to transfer commodities from manufacturing plant to CD. In addition, the set of outbound vehicle fleet V composed by heterogeneous trucks $v \in V$, of different capacities w'_v, is available to distribute commodities from CD to destinations. Note that inbound and outbound vehicles have different speeds, where the parameters τ_f and τ'_{vij} represent the integer numbers of time periods required by inbound vehicle $f \in F$ to travel from manufacturing plant to CD or outbound vehicle $v \in V$ to travel from node $i \in N$ to node $j \in N$. Table 2 summarizes the aforementioned sets and parameters.

3.2 Notation – Decision Variables

Numerous decision variables are required to provide the more detailed solutions provided by the proposed VR-PCD. These variables may be categorized into four main types: pickup, delivery, deviation, and binary routing variables.

The pickup variable, Q^P_{cvft}, determines the quantities of loaded commodity type $c \in C$ onto outbound vehicle $v \in V$ from inbound vehicle $f \in F$ at dock doors for each time $t \in T$. Similarly, the delivery variable, Q^D_{civt}, determines the delivered quantity from outbound vehicle $v \in V$ to customer node $i \in N$ at time $t \in T$.

The deviation variable, v^*_{ic}, depends on the values of the pickup and delivery variables which captures the quantity of unsatisfied demand requested by customer $i \in N$ from commodity type $c \in C$. Finally, the binary variables assign outbound vehicle routes and relate them to other variables using big-M constraints, such that the pickup and delivery variables for each vehicle can have non-zero values only if a particular binary routing variable equals one. This will be described more clearly after the model is formulated. These binary variables are x_{ft}, y_{vt}, and x_{ijvt}. The definitions of all decision variables are presented in Table 3.

Table 2: Summary of parameter notation

Notation	Description
F	Set of inbound vehicles
V	Set of outbound vehicles
T	Set of time periods
C	Set of commodity types
N	Set of nodes $i \in \{0, 1, 2, \ldots, N\}$
0	CD
D	Inbound doors of CD
D'	Outbound doors of CD
l	Maximum loading capacity at each time $t \in T$
l'	Maximum unloading capacity at each time $t \in T$
w_f	Capacity of inbound vehicle $f \in F$
w'_v	Capacity of outbound vehicle $v \in V$
m_c	Mass of commodity type $c \in C$
d_{ic}	The amount of commodity type $c \in C$ demanded by customer $i \in N$
p_{ic}	Priority value of both customer $i \in N$ and commodity type $c \in C$
τ_f	Time periods required by inbound vehicle $f \in F$ to travel to CD from manufacturing plant
τ'_{vij}	Time periods required by outbound vehicle $v \in V$ to travel from node $i \in N$ to node $j \in N$

Table 3: Summary of decision variables, categorized by type

Notation	Description
Pickup variable	
Q^P_{cvft}	The quantity of commodity type $c \in C$ picked up by outbound vehicle $v \in V$ from inbound vehicle $f \in F$ at time $t \in T$
Delivery variable	
Q^D_{civt}	The quantity of commodity type $c \in C$ delivered to node $i \in N$ by outbound vehicle $v \in V$ at time $t \in T$
Deviation variable	
v^*_{ic}	The amount of unsatisfied demand of commodity type $c \in C$ at node $i \in N$
Binary routing variables	
x_{ft}	Binary variable, such that $x_{ft} = 1$ if inbound vehicle $f \in F$ parks at CD at time $t \in T$; $x_{ft} = 0$ otherwise
y_{vt}	Binary variable, such that $y_{vt} = 1$ if outbound vehicle $v \in V$ parks at CD at time $t \in T$; $y_{vt} = 0$ otherwise
x_{ijvt}	Binary variable, such that $x_{ijvt} = 1$ if outbound vehicle $v \in V$ reaches at node $j \in N$, coming from node $i \in N, i\&j \neq i'$, at time $t \in T$; $x_{ijvt} = 0$ otherwise

3.3 Mathematical Model

The MILP model of this problem is divided into three categories of constraints. These include routing constraints providing a detailed optimal route for each inbound and outbound vehicle, pickup and delivery constraints describing the appropriate pickup and delivery of commodities from CD to customers, and CD constraints addressing the coordination of inbound and outbound vehicles at dock doors. The objective function of this problem will be described after the clarification of these constraints.

3.3.1 Routing Constraints

The following set of constraints ensure that each outbound vehicle v is assigned to valid routes. However, the inbound vehicles are assigned to one defined route, beginning from manufacturing plant and ending with CD and vice versa, which will be described later.

The first constraint prohibits multitasking by ensuring that each outbound vehicle v cannot exist in more than one place at each time t, and thus it may serve only one customer at any given time t, as described in constraint (1) below:

$$\sum_{i \in N} \sum_{\substack{j \in N \\ i\&j \neq i'}} x_{ijvt} \leq 1 \quad \forall\, v \in V,\ t \in T \tag{1}$$

The constraints for scheduling and routing of outbound vehicles between CD and customers over time while maintaining the feasibility and continuity of routes in the delivery process are as follows:

$$\sum_{\substack{i \in N \\ i \neq i'}} \sum_{t \in T} x_{i'ivt} = 1 \quad \forall\, v \in V \tag{2}$$

$$\sum_{\substack{i \in N \\ i \neq i'}} \sum_{t \in T} x_{ii'vt} = 1 \quad \forall\, v \in V \tag{3}$$

$$\sum_{t \in T} x_{i'0vt} = 1 \quad \forall\, v \in V \tag{4}$$

$$\sum_{t \in T} x_{0i'vt} = 1 \quad \forall\, v \in V \tag{5}$$

$$\sum_{\substack{i \in N \\ i\&j \neq i' \\ i \neq k}} \sum_{\substack{r \in T \\ r+\tau'_{vjk} \leq t}} x_{ijvr} \geq x_{jkvt} \quad \forall\, v \in V,\ j \in N,\ k \in N,\ j \neq i' or\ j\&k \neq i',\ t \in T \tag{6}$$

$$\sum_{\substack{s \in T \\ s \leq t}} \sum_{\substack{i \in N \\ i\&j \neq i'}} x_{ijvs} \geq \sum_{\substack{k \in N \\ k\&j \neq i'}} \sum_{\substack{r \in T \\ r \leq t+\tau'_{vjk}}} x_{jkvr} \quad \forall\, v \in V,\ j \in N,\ j \neq i',\ t \in T \tag{7}$$

Constraints (2) – (5) address the case that each outbound vehicle v must leave and return to the imaginary node i' just once. As such, these leaving and returning movements must be assigned to CD node only, not the customer nodes. Specifically, constraint (2) implies that each vehicle v should start from i' and leave it just once in all time periods as well as it should return to i' only once in all time periods, as represented in constraint (3). Similarly, constraints (4) and (5) are used for the same purpose, which is to force outbound vehicle v to leave i' to CD and return from it to i' also once during the time horizon. In other words, each vehicle v must be assigned to routes, starting (from i' to CD) as an initial route for vehicle v at the beginning of its time horizon and ending with a route (from CD to i') at the end of its time horizon.

Constraints (6) is required to keep the continuity of vehicle routes. It states that each vehicle v visits node k coming from node j only if it comes from another node i, given that it reaches j in a time less than or equal (current time (t)) – (travel time between j and k, (τ'_{jk})). Similarly, constraint (7) informs that if vehicle v visits customer j, it should leave it for another customer. This constraint is used to maintain the feasibility of routes and balance the model. Note that each vehicle v is permitted to replenish commodities from CD and deliver those commodities to customers more than once within time horizon T.

The last constraint, (8) below, enforces vehicle v to return to CD before the ending of time horizon T and this constraint allows vehicles to work within part of T, with no need to consume all T.

$$t \sum_{k \in N} x_{k0vt} \leq | T | \quad \forall\, v \in V,\ t \in T \tag{8}$$

3.3.2 Pickup and Delivery Constraints

The following pickup and delivery constraints ensure that the appropriate quantity of commodities is transferred from CD to customers at a proper time. The first four constraints are concerned with demand and flow of both picked and delivered quantities of commodities.

$$d_{0c} \geq \sum_{v \in V} \sum_{t \in T} \sum_{\substack{j \in N \\ j \neq 0 \& i'}} Q^D_{cjvt} \quad \forall\, c \in C \tag{9}$$

$$v^*_{jc} = d_{jc} - \sum_{v \in V} \sum_{t \in T} Q^D_{cjvt} \quad \forall\, c \in C,\ j \in N,\ j \neq 0 \& i' \tag{10}$$

$$\sum_{\substack{s \in T \\ s \leq t-1}} \sum_{f \in F} Q^P_{cvfs} \geq \sum_{\substack{s \in T \\ s \leq t}} \sum_{\substack{j \in N \\ j \neq 0 \& i'}} Q^D_{cjvs} \quad \forall\, v \in V,\ c \in C,\ t \in T \tag{11}$$

$$\sum_{t \in T} \sum_{f \in F} Q^P_{cvft} = \sum_{t \in T} \sum_{\substack{j \in N \\ j \neq 0 \& i'}} Q^D_{cjvt} \quad \forall\, v \in V,\ c \in C \tag{12}$$

Constraint (9) states that the quantity of a commodity delivered to all customers by all vehicles at all times should not exceed the available supply of such a commodity at CD. Constraint (10) captures the fact that each customer expects to receive the required quantity of demand. However, in some cases it is infeasible to satisfy the desired quantity of customers due to time window conflicts, vehicle capacity limitations, lack of functional manufacturing plants, or service of more equitable numbers of customers such that an outbound vehicle v may distribute its load to meet multiple customer needs in the same trip. Thus, this constraint addresses the deviation in quantity ordered by a particular customer which is defined as the difference between desired demand level and the total actual delivered quantities. Constraints (11) and (12) balance the flow of both picked and delivered quantities of commodities between CD and customers, such that the quantity of commodities delivered to customers is restricted by the quantity that is picked up at CD. In other words, constraint (11) states that the quantity of commodity type c picked up from inbound vehicles at the inbound dock to be loaded into outbound vehicle v at the outbound dock should be greater than the delivered quantity to all customers at all pickup or delivery times s, where $s \leq t$. Furthermore, Constraint (12) ensuresthat total picked up quantities are equal to the delivered ones at the end of time horizon T for that vehicle.

Additional constraints are required to check the capacity feasibility of inbound and outbound vehicles as well as the unloading rate of commodities at each customer node, as described in constraints (13), (14) and (15) below:

$$\sum_{t \in T} \sum_{c \in C} \sum_{v \in V} m_c\, Q^P_{cvft} \leq w_f \quad \forall\, f \in F \tag{13}$$

$$\sum_{c \in C} \sum_{\substack{s \in t \\ s \leq t}} \sum_{f \in F} m_c\, Q^P_{cvfs} - \sum_{c \in C} \sum_{\substack{s \in t \\ s \leq t}} \sum_{\substack{j \in N \\ j \neq 0 \& i'}} m_c\, Q^D_{cjvs} \leq w'_v \quad \forall\, v \in V,\ t \in T \tag{14}$$

$$\sum_{v \in V} \sum_{c \in C} m_c\, Q^D_{civt} \leq l' \quad \forall\, i \in N,\ i \neq 0 \& i',\ t \in T \tag{15}$$

Constraint (13) implies that the total mass of commodities transferred to CD by inbound vehicle f does not exceed its capacity. Similarly, constraint (14) ensures that the difference between masses of picked up quantities and the delivered ones at any time t to all customers by outbound vehicle v does not exceed its capacity at each pickup or delivery time s, where $s \leq t$. Also, the maximum unloading rate of delivered commodities at customers warehouses is taken into consideration by constraint (15).

Finally, the following three constraints, which are defined by big-M constraints complete the requirements for valid pickup and delivery processes:

$$\sum_{\substack{c\in C\\ k\neq 0}} Q^{D}_{ckvt} \leq M_{16} \sum_{\substack{i\in N\\ i\&k\neq i'}} x_{ikvt} \quad \forall\, v\in V,\ k\in N,\ k\neq i',\ t\in T \tag{16}$$

$$\sum_{v\in V}\sum_{c\in C} Q^{P}_{cvft} \leq M_{17}\, x_{ft} \quad \forall\, f\in F,\ t\in T \tag{17}$$

$$\sum_{f\in F}\sum_{c\in C} Q^{P}_{cvft} \leq M_{18}\, y_{vt} \quad \forall\, v\in V,\ t\in T \tag{18}$$

These big-M constraints address the case that all pickup and delivery processes give a value to a node only if a vehicle visits that node. Specifically, constraint (16) ensures that vehicle v can deliver a quantity of commodity c to customer k only if it reaches customer k at time t from any node i. In this constraint, M_{16} is a sufficiently large number, and it may be calculated as $M_{16} = \sum_{c\in C} d_{Kc},\ \forall\, k\in N$. Note that the maximum quantity of commodity c that may be delivered to any customer k should only be his demand, and thus this is the best value for M_{16}. Constraint (17) states that the quantity of commodity c picked up from inbound vehicle f at the inbound dock by outbound vehicles at the outbound dock can be greater than zero only if inbound vehicle f parks at inbound dock at time t. Additionally, constraint (18) allows vehicle v to pick up a quantity of commodity c from CD only if it parks at outbound dock at time t. The big-M values in constraints (17) and (18) may be chosen such that $M_{17} = w_f$ and $M_{18} = w'_v$, where w_f and w'_v are the capacities of inbound vehicle f and outbound vehicle v, respectively.

3.3.3 *Cross Dock Constraints*

The final set of constraints shows the scheduling of inbound and outbound vehicles to dock-doors over time and the capacity feasibility of CD as follows:

$$t\, x_{ft} \leq |\, T\, | \quad \forall\, f\in F,\ t\in T \tag{19}$$

$$x_{ft} = 0 \quad \forall\, t < \tau_f,\ f\in F,\ t\in T \tag{20}$$

$$y_{vt} \leq \sum_{i\in N} x_{i0vt} \quad \forall\, v\in V,\ t\in T \tag{21}$$

$$\sum_{f\in F} x_{ft} \leq D \quad \forall\, t\in T \tag{22}$$

$$\sum_{v\in V} y_{vt} \leq D' \quad \forall\, t\in T \tag{23}$$

$$\sum_{f\in F}\sum_{v\in V}\sum_{c\in C} m_c\, Q^{P}_{cvft} \leq l \quad \forall\, t\in T \tag{24}$$

Constraint (19) states that each inbound vehicle f can park at CD during any time in T. However, constraint (20) ensures that each inbound vehicle f cannot park at CD unless it passes from manufacturing plant to CD. This process should be performed during time horizon T to prevent any delivery process by outbound vehicle v from being started unless the travel time τ_f for that inbound vehicle f from plant to CD is finished; otherwise, the value of $x_{ft}= 0$ at any time is less than that required for travelling by inbound vehicle f. Similarly, constraint (21) prohibits outbound vehicle v from being parked at CD unless it reaches CD at time t. Moreover, constraints (22) and (23) manage the inbound/outbound vehicle queues at inbound/outbound dock doors to ensure that the number of vehicles parked at CD at time t does not exceed the number of dock doors.

Once inbound vehicle f is coordinated at CD, constraint (24) ensures that the total mass of commodities transferred to CD at time t does not exceed the maximum loading rate of commodities, since its maximum storage area for temporary storage of commodities is limited, and the maximum capacity of the handling system is also limited.

3.3.4 *Objective Function*

The objective of this problem is to provide efficient distribution plans by better scheduling the routings of the heterogeneous trucks in the CD system which minimize the total commodities deviations and the overall distribution time and cost for vehicles, subject to the aforementioned constraints. The MILP formulation of the VRPCD is presented as follows.

$$\text{Min} \quad \sum_{\substack{i\in N\\ i\neq 0}}\sum_{c\in C} v^{*}_{ic}\, p_{ic} + \mu_1 \sum_{v\in V}\sum_{t\in T} t\, x_{oi'vt} + \mu_2 \sum_{i\in N}\sum_{\substack{j\in N\\ i\&j\neq i'}}\sum_{v\in V}\sum_{t\in T} x_{ijvt} \tag{25}$$

s.t. Constraints (1) – (24)

The objective function consists of three terms. The first term is the major part in the objective function which provides the outbound vehicle v a preference p_{ic} to serve a specific commodity type c for a specific customer i. This term is related to constraint (10), it decreases the deviation v^{*}_{ic} in quantity ordered by a particular customer as much as possible, hence constraint (10) addresses the unsatisfied demand of the commodities. The second term is to force vehicle v to return to its depot (the imaginary node) after delivery as early as possible, instead of performing unnecessary movements. This can minimize the total traveling time for a whole tour of vehicle v and can be a great help in reassigning vehicle v in the next tours. Similarly, the third term is to prohibit extra tours of v to any node such that v may visit customer i for delivering commodities, or visit a customer i if this offers a shorter travel time and allows more deliveries to the next customer in the tour. As such, v may return to CD only for replenishing commodities. The second and third terms are considered as auxiliary terms in the objective function which are added to manage the number of movements for vehicles, and ensures that they only perform the needed movements, and thus minimize excessive movements and reduce wasted time and cost. Additionally, if v returns to CD to replenish commodities but the outbound dock-doors are booked up by other vehicles or there is no inbound vehicle f parked at CD at the same time, it should wait in the queue at dock-door for a specific time instead of performing unnecessary visits to customers.

Values of μ_1 and μ_2 should be small portion from the maximum (worst) possible value of the first term, and thus do not effect the first term which is the main objective. Accordingly, if the size of a data set increases (number of customers increases and demand increases), the value of μ_1 and μ_2 increases. To make this clear, the maximum possible value of the first term is shown in Equation 26, where no demand is delivered.

$$\text{Maximum possible total deviations} = \sum_{\substack{i \in N \\ i \neq 0}} \sum_{c \in C} d_{ic} \; p_{ic} \tag{26}$$

The maximum possible value of the summation part in the second term is $|V| \times |T|$, if we assume that all vehicles return to the depot at the last time period. Similarly, the maximum possible value of summation in the third term is $|V| \times |T|$, if we assume that each vehicle make a move at each time slot. According to this, and based on some experimental work, the values of μ_1 and μ_2 are taken to make the maximum possible value of the second and third term equal at most 5% of the first term, as in the next equation:

$$\mu_1 \text{ and } \mu_2 = \frac{\sum_{\substack{i \in N \\ i \neq 0}} \sum_{c \in C} d_{ic} \; p_{ic}}{|V| \times |T|} \times 5\% \tag{27}$$

In literature, most of studies consider traditional objective functions such minimizing cost or distance. In the results section, short experimental analysis is performed to show some possible advantages to consider objective function of minimizing the total prioritized deviation variables, as in this research.

4 HEURISTIC SOLUTION APPROACH

Finding an optimal solution for any data set in the VRPCD model is extremely hard by using exact solution procedures, and sometime it becomes impossible to find any feasible solution for large scale sets. Thus, in this section, the greedy randomized adaptive search procedure (GRASP) is proposed to find feasible solutions for the problems with different scales. During the validation stage, it is found that small-scale problems might be solved optimally using CPLEX. However, larger-scale problems require the use of a customized heuristic to obtain solutions with high quality in reasonable computational time compared to the time needed by CPLEX. There are two phases in the proposed iterative heuristic for the VRPCD. In the first phase, vehicle routes are constructed in a greedy construction procedure. In the second phase, CPLEX–Concert Technology is used to determine the optimal values of the other variables used in the model based on the predefined routes which are found in Phase I. Additionally, an iterative procedure is applied to generate different routes at each iteration. At the end of Phase II, the best solution over all iterations is kept as the final result.

4.1 Phase I: Route Construction

The major purpose of the route construction is to set the values of the original x_{ijvt} binary decision variables to zeros or kept them without setting. The pseudo-code of the `Construction` function is provided in Algorithm 1. The route-construction procedure produces a proper route for each outbound vehicle, starting from and ending with its initial depot (imaginary node), such that each vehicle could decide whether to go through such route or not.

The heuristic approach begins by calculating the total priority value (TP_i) and the total mass of the requested commodity types (TM_i) for each customer node $i \in N$, as in equations 28 and 29. Then, these nodes are arranged in descending order based on their total priority value. Similarly, the importance value (Imp_v) of each outbound vehicle $v \in V$ is calculated based on their capacities, as in equation 30. Then, these vehicles are arranged in descending order based on their total priority value. Value of parameter R that appears in Equations 28 and 30 is a uniform random number generated within a range of [0.8,1.2]. The benefit of using R is to add randomness to the calculations and to get slightly different results in each iteration. The range of R is selected based on experiments to be around 1, it provides some randomness without violating the role of equations producing priority and importance values of nodes and vehicles.

$$TP_i = \frac{\sum_{c \in C} [\frac{p_{ic} \times d_{ic}}{m_c}]}{R \times r_{CD,i}} \qquad \forall\, i \in N,\; i \neq CD \tag{28}$$

$$TM_i = \sum_{c \in C} m_c \times d_{ic} \qquad \forall\, i \in N \tag{29}$$

$$Imp_v = R \times w'_v \tag{30}$$

After that, the movement of the selected outbound vehicles can be performed at a given time ($start_v$), by satisfying two conditions. The first condition is to check if the inbound vehicle $f \in F$ reaches CD coming from the manufacturing plant in a time that equals its travel time (τ_f). This means that such a vehicle can park at the inbound door of CD and deliver quantities to an outbound vehicle which is already parked at the outbound door at the same time, thus the starting time of the selected outbound vehicle $v \in arranged_v$ equals the travel time of that inbound vehicle. The second condition is to synchronize the number of parked inbound/outbound vehicles with the inbound/outbound doors, which also determines the earliest starting time of pick up from each inbound vehicle.

Next, in line 7, the outbound vehicle loop begins considering the maximum number of visits for the current selected vehicle before ending its tour and returning to its depot or returning to CD for resupply again. The number of visits is represented by factor (des), and is mainly determined by vehicle capacity w'_v and the loading capacity l' with some slight randomness, as in Equation 31. The maximum number of visits performed by such vehicle equals the number of nodes, while the minimum number of visits equals 1. The value of R is an integer and is randomly selected from a uniform range of [-1,1].

After that, the time loop begins for such vehicle considering the feasibility condition of visiting the selected customer node $i \in arranged_i$ by satisfying many sub-conditions, such as testing if this node is feasible to be visited before the end of time horizon, if it is reachable by the selected vehicle v at time t, and if v can return to CD before the end of time horizon time $|T|$. If these conditions are satisfied, the visit for the highest prioritized customer node is accomplished.

$$des = R + ceil(\frac{w'_v}{l'}) \tag{31}$$

When a customer node is visited, this implies that it received all (or some) of the requested demand. Hence, the mass value of the requested demand for the visited node should be reduced based on four cases. The first case, if the capacity of the selected vehicle (w'_v) is less than the total commodity masses requested by the visited customer node (TM_i), and less than or equal to the unloading rate of commodities at this node (l'). This implies that the visited node cannot receive the whole demand and can only receive the truckload of commodities at the visiting time. Thus, TM_i is decreased by subtracting w'_v, where w'_v is slightly randomized by multiplying it by a random number between [0.6, 1], as in line 22. The second case, if the capacity of the selected v is greater than or equal to the total commodity masses requested by this node, but less than or equal to the unloading rate of commodities at this node. This means that the visited node can receive its full demand by the visiting time, and TM_i is decreased by dividing by 3, as in line 25.

The third case, if the unloading rate of commodities at the visiting node is less than the total commodity masses requested by such node, and less than the capacity of the selected v as well. So that the visited node cannot receive the whole demand and can only receive the amount of unloaded commodities by the visiting time. Thus, TM_i is decreased by subtracting l', where l' is slightly randomized by multiplying it by a random number between [0.6, 1], as in line 28. Finally, if the unloading capacity of commodities at the visiting node is greater than or equal to the total commodity masses requested by such node, but less than the unloading capacity of commodities at this node. This implies that the visited node can receive its full demand by the visiting time, and TM_i is decreased by dividing by 3, as in line 31. Note that the visiting node in the second and fourth cases may be served later by one or more vehicles to deliver its remaining demand, since split deliveries are permitted.

After reducing the mass value of the requested demand for the visited customer node, the total priority value for that customer node should be decreased. This is because each time a customer node is visited, the importance of such node is decreased to avoid being revisited by other vehicles in the same period of time unless its priority justifies multiple visits. The new reduced value TP_i can be calculated as in Equation 32.

$$= \frac{TM_i \times R \times Avg_P}{Avg_M \times R' \times r_{current,i}} \quad \forall\, i \in N \tag{32}$$

In equation 32, the average (not the total) priority for both customer nodes and commodity types and the average commodity masses requested by the customer nodes are used. This is because the reduction was in the total mass of commodities, regardless the types of these commodities and their specific masses. The Avg_P and Avg_M values are calculated in Equations 33 and 34, respectively. The numerator and denominator are slightly randomized by multiplying it by a random number R between [0.8, 1.2]. Then, these nodes are re-arranged again in descending order based on their new total priority value (TP_i).

The whole process is repeated for the same vehicle until the vehicle performs the predetermined number of visits. Then, it should go back to its depot (i'), or return to CD to resupply commodities again in case there is still available time slots in horizon to perform new set of visits. If the vehicle visits the CD, the total priority value for each customer node TP_i should be recalculated, nodes are arranged again in descending order based on their new total priority value, the number of time periods that the vehicle should park to replenish commodities from it should be determined based on free doors availability and based on loading capability, and the number of customer nodes might be visited, as discussed before. This procedure continues until the time horizon ends.

$$Avg_P = \frac{\sum_{i \in N} \sum_{c \in C} p_{ic}}{(N-1) \times C} \tag{33}$$

$$Avg_M = \frac{\sum_{c \in C} m_c}{C} \tag{34}$$

After that, a route for the selected vehicle $v \in arranged_v$ is constructed from the variables x^{new}_{ijvt}, such that the value of x^{new}_{ijvt} is updated to 1 after each visit, even the visit is to customer nodes or to CD.

The whole procedure is repeated until all vehicles have been selected. After the end of the time horizon for each vehicle, binary variables are known and set to zeros or kept as defined. If the value of the binary variable x^{new}_{ijvt} equals zero, then the value of the original binary variable x_{ijvt} which is defined in the existing VRPCD model is added as a constraint to such a model, such that $(x_{ijvt} = 0 \quad \forall\, i\&j \in N,\ i\&j \neq i',\ v \in V,\ t \in T)$. This facilitates coding, because x_{ijvt} is excluded from the solution process. However, if the value of x^{new}_{ijvt} equals one, then the value of x_{ijvt} is kept either zero or one. This approach yields highly competitive solutions in a short computation time, as demonstrated in the next section.

Algorithm 1: Construction Function

Calculate TP_i per Equation (28) // Total priority for all customer nodes.
Calculate Imp_v per Equation (30) // Importance for each outbound vehicle.
Calculate TM_i per Equation (29) // Total masses of requested commodities by customer nodes.
Arrange customer nodes in descending order based on their TP_i to get $arranged_i$.
Arrange outbound vehicles in descending order based on their Imp_v to get $arranged_v$.
Set starting time for each outbound vehicle $start_v$ based on the travel time of inbound vehicles, the number of inbound doors of CD, and the number of inbound vehicles.
for all $v \in arranged_v$ **do** // Start v of $arranged_v$ loop
  Calculate the maximum number of v visits to customer nodes (des) before returning to its depot or CD, as in Equation (31).
  if $des > N - 1$ **then**
    $des = N$
  end if
  if $des \leq 0$ **then**
    $des = 1$
  end if
  $t^{now} = start_v$ // Set the current time of the v to zero
  for all $t \in T$ **do** // Start t loop for each V of $arranged_v$, such that t starts at $start_v$
    if it is feasible to visit customer node i of $arranged_i$ **then**
      $x^{new}_{CD\rightarrow i,v,t} = 1$
      Set $t^{now} = t$
      Update $current = i$ // $i \in arranged_i$
      Reduce TM_i value for the visited node i based on vehicle capacity and unloading rate of commodities at i.
      if $(l' \geq w'_v$ And $w'_v < TM_i)$ **then**
        $TM_i = TM_i - R * w'_v$ // Random number [0.6, 1]
      end if
      if $(l' \geq w'_v$ And $w'_v \geq TM_i)$ **then**
        $TM_i = TM_i/3$
      end if
      if $(l' < w'_v$ And $l' < TM_i)$ **then**
        $TM_i = TM_i - R * l'$ // Random number [0.6, 1]
      end if
      if $(l' < w'_v$ And $l' \geq TM_i)$ **then**
        $TM_i = TM_i/3$
      end if
      Recalculate TP_i for all customer nodes per Equation (32) to avoid revisiting by other vehicles in the same period of time.
      Rearrange customer nodes in descending order based on their new TP_i to get new $arranged_i$.
    end if // End if feasible to visit i
    if Customer node visits are finished **then** // to decide the next visit
      if it is feasible to visit CD **then**
        $x^{new}_{i\rightarrow CD,v,t} = 1$
        Set $t^{now} = t$
        Update $current = CD$
        Recalculate TP_i for all customer nodes per Equation (32).

Rearrange customer nodes in descending order based on their new TP_i to get new $arranged_i$.
Decide if v should park at CD for other time periods to replenish more commodities or not.
end if // End if park time
end if // End if feasible to visit CD
end for // End for t loop
end for // End for v loop
Set the binary variable values to zeros and ones
if $x^{new}_{i,j,v,t} == 0$ **then**
Add $x_{i,j,v,t} = 0$ as a constraint to the existing model, $\forall\ i\&j \in N,\ i\&j \neq i',\ v \in V,\ t \in T$.
end if
if $x^{new}_{i,j,v,t} == 1$ **then**
Return x_{ijvt} $\forall\ i\&j \in N,\ i\&j \neq i',\ v \in V,\ t \in T$
end if

4.2 Phase II: Model Solution Based on Predefined Routes

When all binary variables are set to zeros or kept as defined using the construction function, the model is solved optimally using CPLEX optimizer to determine the other variables values such as pickup quantities, delivery quantities, unsatisfied quantities, and other binary variables which control the parking time of each inbound-outbound vehicle at CD. Then, the solution process is repeated fifty times to generate different vehicle routes at each run by changing specific parameters, and finally, the best solution found over all runs is selected and saved as the final result.

5 NUMERICAL ANALYSIS

In this section, numerical experiments on various data sets are performed using CPLEX within a six-hour time limit to evaluate the effectiveness of the suggested mathematical model. Then, an analysis is conducted on these data sets with the incorporation of the proposed heuristic. This can help compare the solutions generated using CPLEX against the proposed heuristic in terms of solution quality and the required computation time.

All experiments are implemented on an HP-Tower workstation Z240 with a Xeon-Intel processor running Ubuntu Linux 16.04.1 LTS in 64-bit mode. As the proposed model is new and no similar models or benchmarks are available in literature, the input parameters are programmed to be randomly generated using C++ programming language and follow a uniform distribution. Data sets are classified as small, medium, and large scale. Each scale consists of 50 generated data sets to test the proposed model and heuristic under different scenarios. Specifically, small-scale sets are randomly generated to compare the optimal results produced by CPLEX with the proposed heuristic results, such that the proposed approach will be fairly evaluated. Whereas, medium-scale sets are randomly generated to compare them with sub-optimal results. For large-scale sets, CPLEX produces sub-optimal results for 15 out of 50 sets, however, it fails to provide feasible solutions for the remaining ones within a 6-hour time limit. Therefore, large-scale sets are randomly generated to evaluate the performance and scalability of the proposed heuristic.

The parameter values for the number of inbound/outbound vehicles, time periods, commodity types, nodes, and inbound/outbound dock doors have a significant effect on the computation time, and thus are used to classify the data sets into different scales. Table 4 shows these input parameters along with their suggested values.

Table 4: Design of experiments

Parameter	Small-scale Range	Medium-scale Range	Large-scale Range
No. of inbound vehicles, $\|F\|$	~Unif(1,2)	~Unif(3,5)	~Unif(6,12)
No. of outbound vehicles, $\|V\|$	~Unif(3,5)	~Unif(6,9)	~Unif(10,18)
No. of time periods, $\|T\|$	~Unif(7,13)	~Unif(14,20)	~Unif(21,36)
No. of commodity types, $\|C\|$	~Unif(1,2)	~Unif(3,5)	~Unif(6,10)
No. of customer nodes, $\|N\|$	~Unif(3,5)	~Unif(6,12)	~Unif(13,25)
No. of inbound doors of CD, $\|D\|$	~Unif(1, 2)	~Unif(3,4)	~Unif(5,10)
No. of outbound doors of CD, $\|D'\|$	~Unif(2,4)	~Unif(5,7)	~Unif(8,16)

Other parameters are considered as fixed parameters. These parameters have less effect on computation time and are independent of the VRPCD scale, such as the capacity of each inbound/outbound vehicle which is generated in the range of 1200 to 3400 kg and 500 to 1500 kg, respectively, to cover most available vehicles suitable for CD system. The mass of commodities is selected to be between 1 and 6 kg, which covers most food and drink commodities. The maximum loading rate at CD and unloading rate at each customer node are generated in the range of 800 to 1800 kg and 300 to 1000 kg, respectively, in each time period based on some measurements done on handling systems in some locations, while considering the number of operators, equipment for loading, unloading and movement in the warehouse such as forklifts and hand trucks, etc., that are available at CD or customer nodes at each time period. Also, the demand of commodities at each customer node is selected in the range between 100 and 400 units for each commodity type. Moreover, it is assumed that the range of priority value for both customer and commodity type is between 1 and 4.

In the next section, analysis is conducted on the different sizes of data sets to compare the solutions generated using CPLEX against the proposed heuristic approach in terms of solution quality and the required computational effort and thus show the effectiveness of the proposed approach.

5.1 Small-scale Problems

In this subsection, CPLEX is able to produce optimal solutions for all small sets in a mean computation time of 602 seconds (about 10 minutes). These sets are used to test the optimal results produced by CPLEX versus heuristic. Therefore, a gap between the objective function value (OFV) obtained from the CPLEX solution and the OFV obtained from the heuristic solution is used for the sake of comparison. Mathematically, it can be calculated as follows.

$$gap = \frac{\texttt{Heuristic OFV - CPLEX OFV}}{\texttt{CPLEX OFV}} \times 100\% \tag{35}$$

Figure 1 compares the performance of CPLEX against the proposed heuristic for small sets. It shows the bar-plot of OFV produced by CPLEX solution and OFV produced by heuristic solution with their gaps and the bar-plot of the computation time. From figure, it can be inferred that the proposed heuristic provides very competitive results because in a time less than 2 seconds, it can give solutions for all sets with a mean gap of 0.84%. As such, it can give the same results compared to CPLEX for 42 out of 50 sets. Furthermore, for the gaps, there is a small spread of all data points which ranges from 0% to 7.83%.

In some instances, gaps are considered to be high. As only 3 instances out of 50 instances have gaps greater than 7% and the proposed heuristic provides solution with 0% gap for 42 instances, it can not be concluded that the trapping in local optima is a common for the proposed heuristic. Making more investigation on these instances, it is found that there is a possible reason for the high gaps for some instances which is that they need more solution time (higher number of iterations). For example, set number 29 can be solved by the proposed solution approach to get objective function value of 2491 with gap of 2.38%, but in 4.21 seconds instead of 1.3 seconds. Similarly, in set number 40, we can get objective function value of 3962 with gap of 4.07% in 6.72 sec instead of 4.08 sec. For consistency, results are not changed for these sets, as all the results were taken at the same number of iterations.

As small scale sets are solved optimally is short time, some of them are randomly selected to investigate the effect of using different objective functions, as in section 5.1.1.

5.1.1 *Effect of Objective Function*

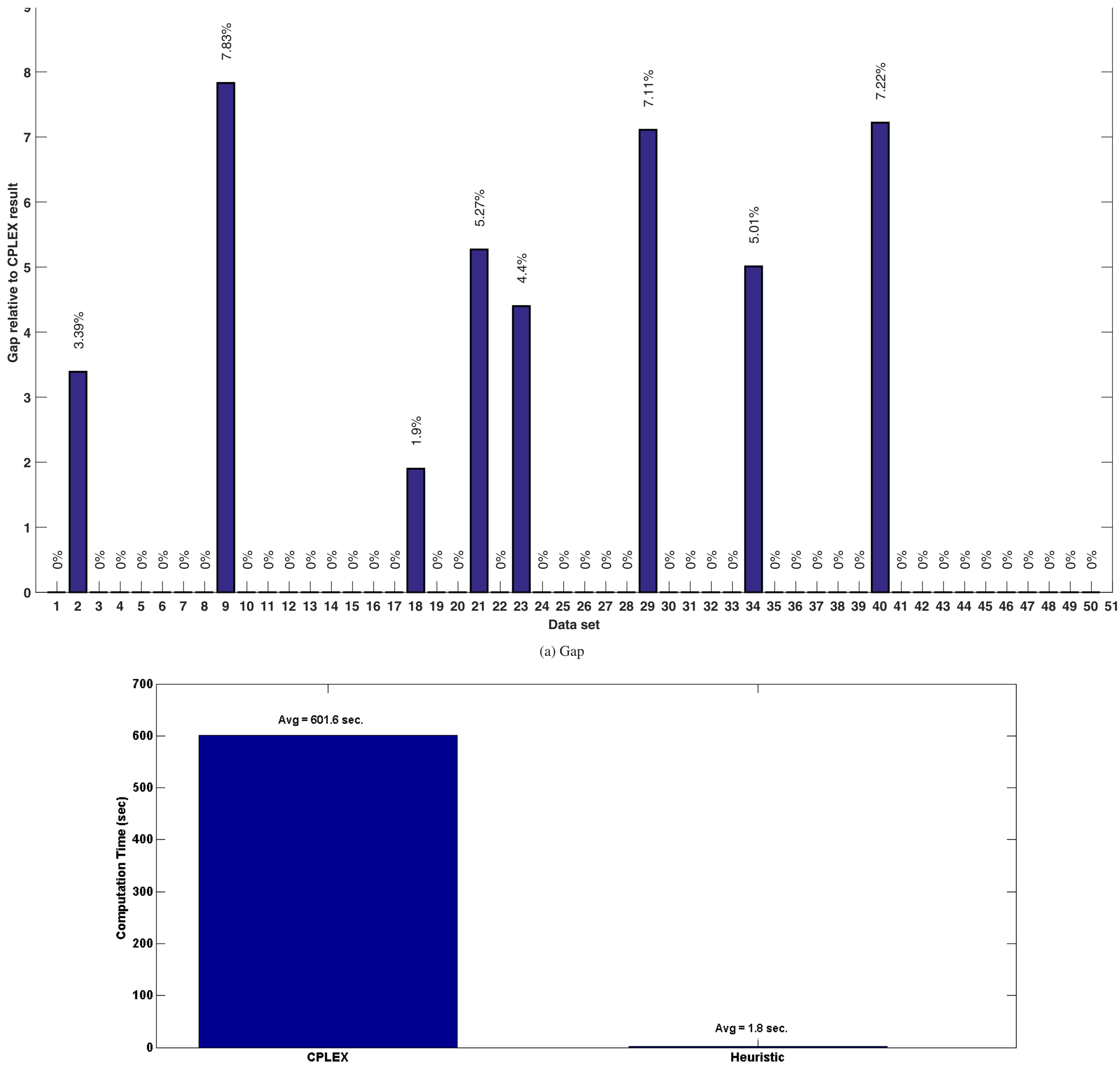
As briefly discussed in the formulation section, objective function plays a great role in the results of the model. In this section, some small sets are randomly selected to investigate the effect of the objective function. The first issue of the objective function is that most of research in the literature use traditional objective function such as minimizing the cost or minimizing the distance. In the model proposed in this research, minimizing the total cost (distance) will not work without extra constraints to enforce vehicles to supply the demand. In our problem, there are many limitations that prevent vehicles from supplying the whole requested demand. First limitation could be the short time horizon (allowable working hours) which limit the vehicles from performing more routes. The second limitation could be the available supply which is sometimes less than the requested demand. The third limitation could be the vehicle capacities or the cross dock capacity. In this case, time horizon is long enough and there is too much supplies but the vehicles are small to deliver all demand, or the handling system in the cross dock is not able to transfer high quantities from inbound to outbound vehicles.

Because all of the previous limitations, the objective function is used to be minimize the total prioritized deviation variables. In case of the minimizing the total cost is used as in 36, new constraints should be used, as the constraint set 37.

$$\text{Min} \quad \sum_{i \in N} \sum_{\substack{j \in N \\ i \& j \neq i'}} \sum_{v \in V} \sum_{t \in T} d_{ij}\, x_{ijvt} \tag{36}$$

$$\sum_{v \in V} \sum_{t \in T} Q^D_{cjvt} \geq \zeta_{jc} d_{jc} \qquad \forall\, c \in C,\; j \in N,\; j \neq 0 \& i' \tag{37}$$

(38)



(a) Gap

(b) Computation time

Figure 1: Gap and computation time produced by heuristic for small data sets

Constraints 37 enforce vehicle to supply portion of demand for each customer, where ζ_{jc} is a portion of demand type c that is expected to be supplied to customer j. The problem with such constraints is the value of ζ_{jc} which if it is selected to be high, the possibility of getting infeasible solution increases. If it is selected to be small, deviation variables will be large; giving that the objective function is to minimize the total cost. If the available supply is greater than the requested demand and the resources (time and capacities) are available, ζ_{jc} can be given a value of 1.

Other possible solution to make the objective function of minimizing the cost works well in the model is to penalize the deviation variables in the objective function without using constraints 37, as in 39. To investigate the effect of changing the objective function, sets solved with considering constraints 37 and ζ equal to 50%. Then, they are solved with penalizing the deviation variables in the objective function. The last issue to be investigated here is to solve the model with using the two auxiliary terms to show the effect of μ_1 and μ_2. Results are shown in table 5.

$$\text{Min} \quad \sum_{i\in N} \sum_{\substack{j\in N \\ i\&j\neq i'}} \sum_{v\in V} \sum_{t\in T} d_{ij}\, x_{ijvt} + \mu' \sum_{\substack{i\in N \\ i\neq 0}} \sum_{c\in C} v^{*}_{ic}\, p_{ic} \tag{39}$$

Table 5: Results with different objective functions

Set Number	Results with the original function: Total deviation variables	Total Traveled Distance	Results with the Min total distance and penalized deviations: Total deviation variables	Total Traveled Distance	Results with the Min total distance and constraints 37: Total deviation variables	Total Traveled Distance	Results with the original function without penalizing: Total deviation variables	Total Traveled Distance
1	1338	70	1338	70	Infeasible	Infeasible	110	1338
2	53	70	127	70	480	70	110	53
3	380	160	726	100	696	160	320	380
4	1484	90	1484	90	Infeasible	Infeasible	210	1484
5	253	160	713	110	1440	160	250	253
6	1556	50	1556	50	Infeasible	Infeasible	250	1556
7	2147	80	2147	80	Infeasible	Infeasible	270	2147
8	658	100	856	60	Infeasible	Infeasible	210	658

It can be observed that minimizing the total prioritized deviations with penalized distance (original objective function) will provide solution with the least deviation variables. However, using the objective function to minimize the total distance or cost can provide solution with less total traveled distance, but with greater undelivered demand, as in sets 3, 5, and 8. Some of sets have the same results regardless which function is used, as in 1, 4, and 6. Using the objective function of minimizing the total distance beside using the constraints 37 is not preferable because value of ζ should be accurately defined or many infeasible cases will be produced as shown in the table. This analysis leads to an excellent potential for future work which is considering multiple objective optimization.

Solving model without penalizing the traveled distance in the objective function provides the same deviation variables, but with noticeable greater distance, as there is no anything to enforce the vehicles to take the short distances and return back early, as sown in the last two columns in Table 5.

5.2 Medium-scale Problems

In this subsection, larger instances and the same definition of the gap as in Equation 35 are used. Figure 3 shows the performance of CPLEX against the proposed heuristic in terms of both resulting solutions and running time.

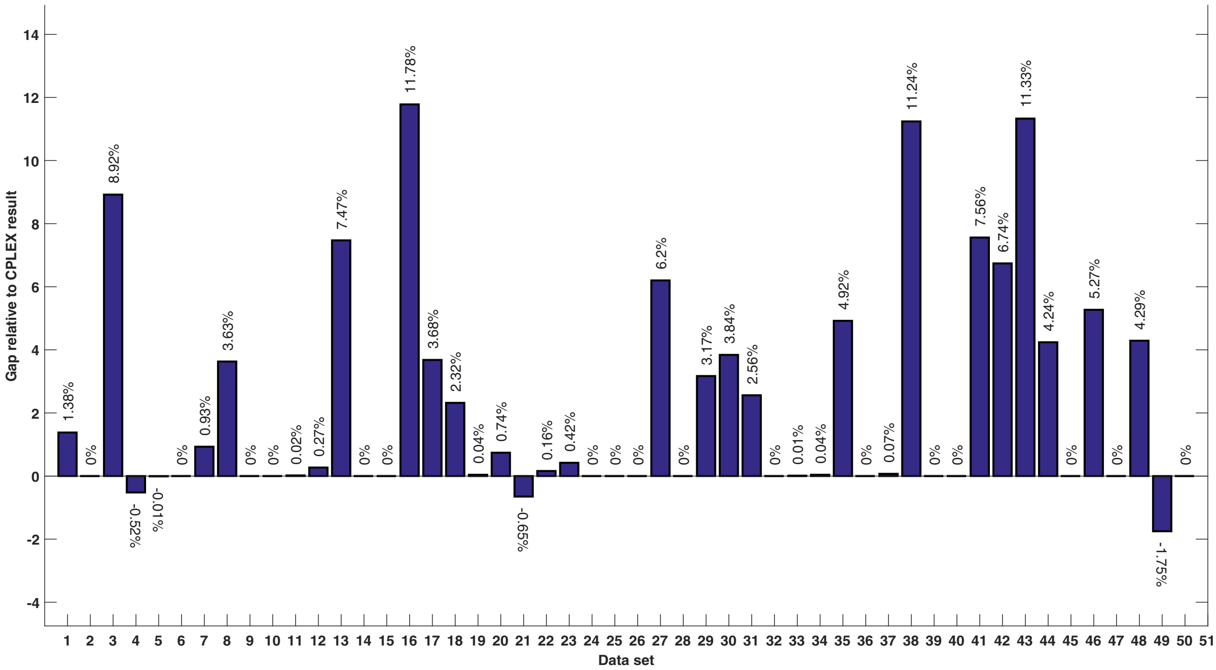
As shown, the proposed heuristic produces the same or better results compared to CPLEX for 21 sets out of 50, with a mean gap of 2.21% in a mean time less than 24 seconds for all medium sets. Conversely, CPLEX takes about 6 hours to determine sub-optimal solutions for these sets. Additionally, for the gaps, there is a small spread of all data points which ranges from -1.75% to 11.78%.

From this, it can be concluded that the proposed heuristic is highly recommended for medium-scale sets in both solution quality and computational effort. However, because the heuristic results are compared with sub-optimal solutions, further analysis is done to evaluate them more fairly, as in section 5.2.1.

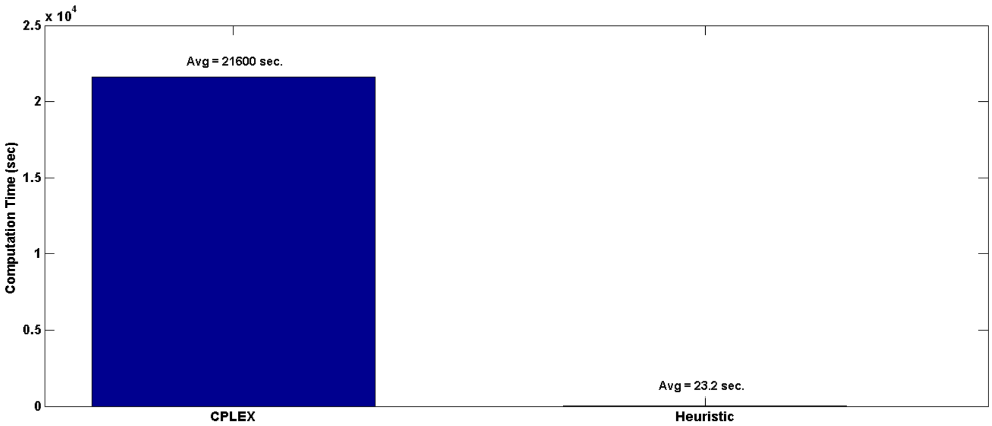
5.2.1 Relative LP gap analysis for medium scale sets

Figure 3 shows the gap and relative LP gap values. Some examples are taken here with different values of gaps and relative LP gaps to explain the issues, as in Figure 3. Note that the medium sets are independent of each other, but they are arranged in ascending order in Figure 3 based on the gap values to make the comparison easier to understand.

Some sets with high value of gaps, such as sets 47-50, have small relative gaps which means that if they are compared with optimal solutions, they will not potentially get worst, or will get worser by small percentage less than 1%. Some sets with gap of 0% have relative gap ranging from 0.03% to around 6%. Thus, we can claim that the heuristics provide solutions very close to the optimal solutions for sets 15-21. In the other hand, we can claim that heuristic will provide solutions that are not worser than 2-6% if they compared to the optimal solutions, as in sets 6, 9, and 33. All sets with negative gaps have small relative gaps, except set number 2. This means that heuristic provide excellent solutions for these sets, even if we get the optimal solutions of them.



(a) Gap



(b) Computation time

Figure 2: Gap and computation time produced by heuristic for medium data sets

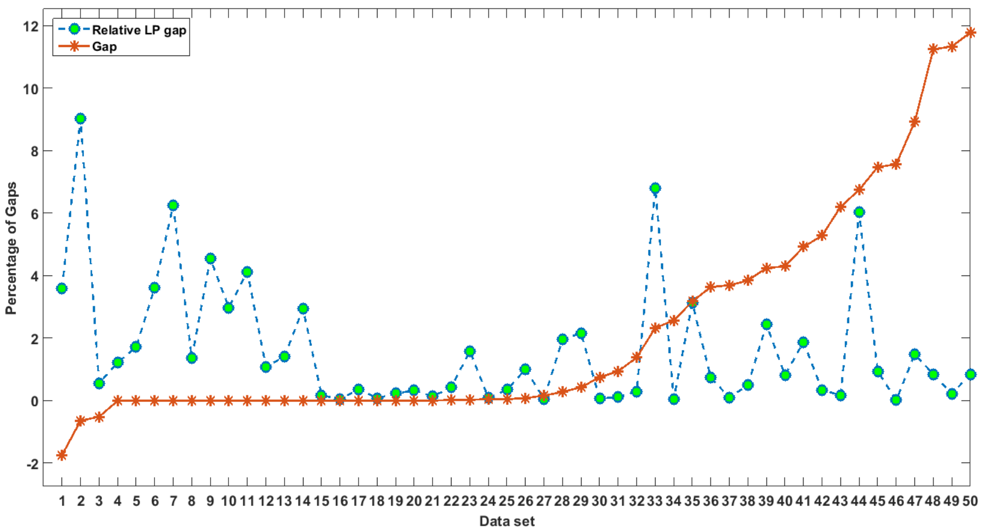
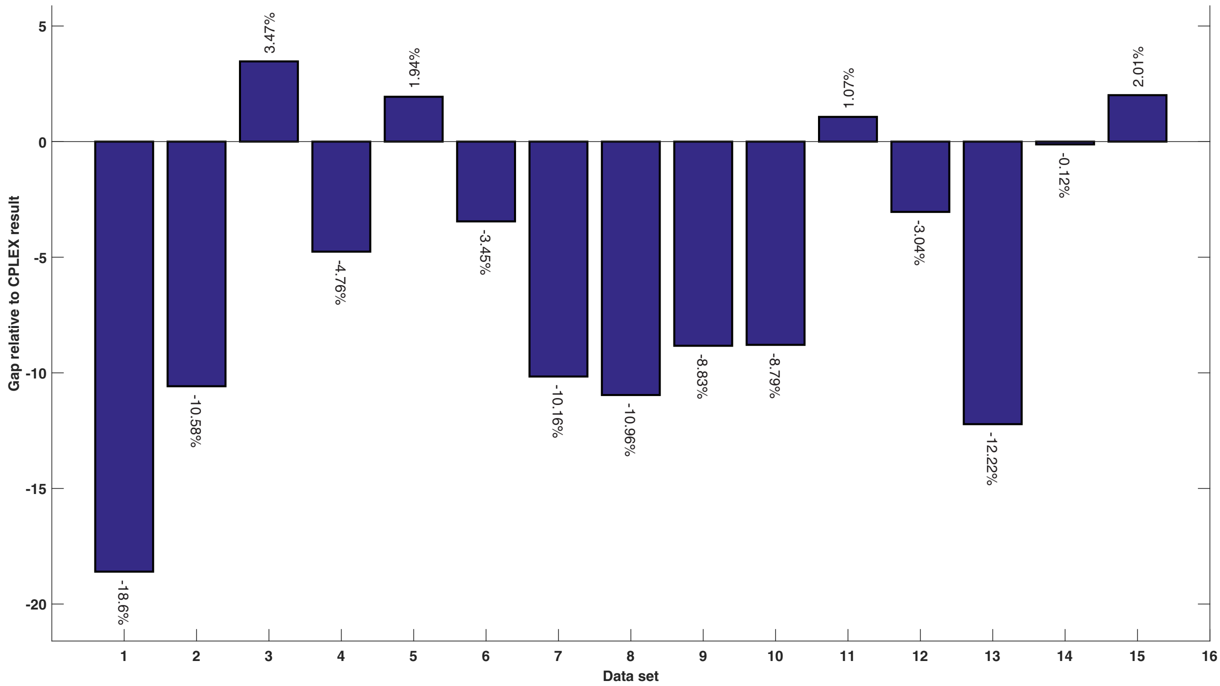


Figure 3: Gaps and Relative LP Gaps

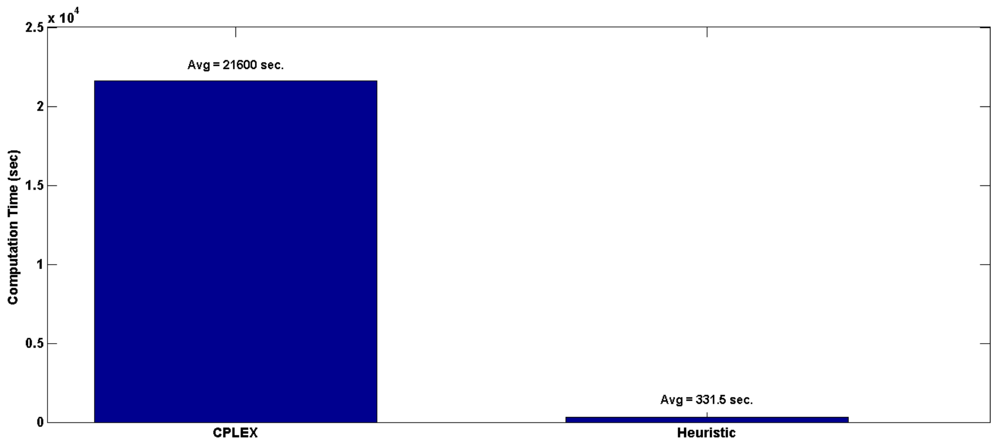
5.3 Large-scale Problems

For large scale problems, CPLEX is unable to find feasible solutions for many instances within 6 hours. Therefore, large-scale data sets are considered to evaluate the performance and scalability of the proposed heuristic in this subsection. The same definition of the gap as in Equation 35 is used. Figure 4 shows the performance of CPLEX against the proposed heuristic in terms of both resulting solutions and running time. Note that the gap is not applicable for many sets, since these sets have no feasible integer solutions obtained from CPLEX. Thus, these sets are excluded from the gap plot in Figure 4.

From results, it can be inferred that CPLEX takes the allowable 6 hours to produce sub-optimal solutions for only 15 out of 50 sets, and fails to find feasible solutions for the remaining ones. On the contrary, the proposed heuristic is able to produce solutions for all large-scale sets with a mean computation time of about 332 seconds (about 6 minutes). As such, it shows better results than that of CPLEX for 11 out of 15 sets, with a mean gap of -5.53% and a variability in data points, ranging from -18.60% to 3.47%. Therefore, the proposed heuristic results outperform CPLEX results in both solution quality and computation time. Similar to the further analysis done in medium scale section, more experimental analysis is performed to evaluate the results of large scale sets with considering the LP relative gaps, as in section 5.3.1.



(a) Gap



(b) Computation time

Figure 4: Gap and computation time produced by heuristic for large data sets

5.3.1 Relative LP gap analysis for large scale sets

For the large scale sets, more experimentally work is performed by repeating the run with time limit of 12 hours and get the results in the Table 6. With this time limit, CPLEX provides solutions for 22 sets. It can be can noticed that heuristic provides solution with gap range 3-6% away from the best possible solution that could be achieved by CPLEX. For example, in set number 1, the LP gap with 6 hour time limit is 23.32%, and with 12 hours time limit is 8.14%. Whereas the gap between the heuristic results and these results are -18.6%, -3.7%, respectively. Accordingly, if the optimal solution of this set is better than the CPLEX solution with 6 hours limit by 23.32%, which the maximum possible improvement, the heuristic solution will be 4.75% away from the optimal solutions. The same analysis can be done for many sets such as sets 2-4, 9, 11, 13-15, and 21-22.

Some other sets, heuristic finds almost the optimal solution, as in sets 16, 19, and 20. In these sets, the absolute value of the negative gaps almost equal to the relative LP gaps, which means that if the optimal solutions are found by giving the CPLEX very long time limit, they will be approximately equal to the heuristic solutions. The last type of sets is the sets with gaps greater than the LP relative gap by 7-15%, as in sets 6, 7, and 12. These sets are expected to have the worst gaps in case the optimal solutions can be found. Again, this analysis is just an expectation, as optimal solutions could equal the current incumbent solutions where the current gaps will be the best gaps can be found.

It is worth to mention that solution of the set number 6 provided by CPLEX in 6 hours is better than the solution provided in 12 hours, this could happen because of the branching procedure followed by CPLEX and it can not be justified. As conclusion, it can be claimed that the suggested solution approach is expected to produce optimal solutions, near optimal with low gaps, and near optimal with acceptable gaps to large scale problems.

At the end of the numerical analysis, it can be concluded that CPLEX and the proposed heuristic approach work well for small problem sizes. However, as the problem size increases, then the challenge in finding solutions with high quality in reasonable computational time also increases. Therefore, the proposed heuristic outperforms CPLEX for many instances.

Table 6: Large Scale sets results

	CPLEX Results 6 hours time limit		CPLEX Results 12 hours time limit		Heuristic Results		Gaps results	
Set Number	Objective Function value	Relative gap	Objective Function value	Relative Gap	Computational time	Objective Function value	Gap Compared to 6 hours time limit results	Gap Compared to 12 hours time limit results
1	29662	23.32	25072	8.14	241.34	15540	-18.6	-3.7
2	81768	14.25	71703	2.2	290.9	73121	-10.58	1.98
3	60912	10.21	58326	9.43	153.16	63028	3.47	8.06
4	26948	11.52	25738	7.36	44.24	25665	-4.76	-0.28
5	46414	10.81	44112	6.23	96.69	47313	1.94	7.26
6	82807	13.02	88588	18.82	218.71	79950	-3.45	-9.75
7	97592	26.22	91584	21.65	547.41	87677	-10.16	-4.27
8	36574	18.22	36193	17.78	1025.23	32564	-10.96	-10.03
9	100799	10.52	97380	7.44	192.14	91897	-8.83	-5.63
10	23042	21.43	22782	18.55	134.1	21017	-8.79	-7.75
11	102831	9.1	98539	4.72	142.76	103933	1.07	5.47
12	42306	16.32	39420	10.54	519.14	41022	-3.04	4.06
13	59988	14.56	53549	4.31	135.82	52660	-12.22	-1.66
14	62090	13.76	57031	6.1	99.89	62013	-0.12	8.74
15	12659	9.74	12242	6.6	73.21	12913	2.01	5.48
16	No Soln	—	34570	40.69	603.79	20547	NA	-40.56
17	No Soln	—	87091	23.49	52.77	70426	NA	-19.14
18	No Soln	—	57322	51.19	61.37	33516	NA	-41.53
19	No Soln	—	110117	19.12	72.85	89137	NA	-19.05
20	No Soln	—	76569	28.63	83.13	54755	NA	-28.49
21	No Soln	—	79094	22.15	35.52	62760	NA	-20.65
22	No Soln	—	68607	27.3	10.77	52022	NA	-24.17

6 CONCLUSION

In this research, the integration of VRP with CD strategy is presented. A new optimization model is introduced to provide detailed and efficient distribution plans for vehicles that are used to transfer commodities from a manufacturing plant to retail warehouses. The proposed MILP model provides a detailed optimal route for each outbound vehicle that minimize the amount of unsatisfied quantities requested by customers, and the overall distribution cost. The entire problem is based on a CD environment with allowable split deliveries and replenishment of multiple commodity types with different masses and quantities, using a heterogeneous fleet of inbound/outbound vehicles with different capacities and speeds. The research work carried out to date addresses the VRPCD with only a subset of the different logistics decisions mentioned above.

The proposed model is validated using CPLEX. During the validation stage, it is found that as the size of the problem increased, the complexity of the problem increased. Thus, CPLEX is not able to generate at least one feasible integer solution. Furthermore, CPLEX consumes longer run times to produce any feasible solution to the problem (although not optimal). Hence, CPLEX cannot produce solutions to realistically sized problems within a reasonable time. This motivated the need to develop a heuristic solution approach, which provides high quality solutions within a short computation time. The proposed heuristic consists of two phases. In the first phase, a greedy route construction procedure is used to determine the values of binary decision variable x_{ijvt} in the existing model, which are added as constraints to the existing constraints in the model or kept as defined (binary variables). In the second phase, CPLEX is used to determine the values of pickup quantities, delivery quantities, unsatisfied quantities, and other binary variables for a given collection of vehicle routes.

Numerical analysis is conducted on different scale data sets. Each scale consists of 50 generated data sets to test the performance of the proposed model and heuristic under different scenarios. All of test data sets are first solved by CPLEX within a maximum time limit of six hours. Then, all of these sets are solved using CPLEX with the incorporation of the proposed heuristic. For small scale sets, CPLEX produces optimal results in an average time of 602 seconds (about 10 minutes), while the proposed heuristic provides results with an average time of less than 2 seconds and a mean gap of 0.84%. For medium scale sets, CPLEX produces sub-optimal results in an average time of 6 hours, while the proposed heuristic is highly recommended because it produces the same or better results for many sets, with an average time of less than 24 seconds and a mean gap of 2.21% for all of these sets. For large scale sets, CPLEX fails to find feasible solutions for 35 out of 50 sets and it produces sub-optimal results for only 15 sets in an average time of 6 hours. However, the proposed heuristic provides results for all large sets with an average time of about 332 seconds (about 6 minutes) and a mean gap of -5.53% At the end of the numerical analysis, it is concluded that CPLEX and the proposed heuristic approach work well for small problem sizes. However, for larger sizes, the challenge in finding solutions with high quality in reasonable computational time is increased. Therefore, the proposed heuristic outperforms CPLEX for many instances.

Finally, the research work can be extended by developing a mathematical model which allows multiple manufacturing plants, multiple CD platforms, and a time window for each customer node. Additionally, allowing storage of inventory at the CD by addressing the constraints that limit the quantity of stored commodities and limit the time interval that the commodities remain in the CD. Furthermore, a transfer time of commodities from the inbound to the outbound doors of the CD can be introduced alongside with the limits introduced in this research. Different objective functions can be used besides minimizing the deviations, and multiple objective function models can also be used. Suggested objective functions are minimizing the total distribution time, the number of vehicles used, and the maximum tour duration. Additionally, although the presented heuristic approach is capable of providing quality solutions in a reasonable computation time for different problem sizes, local search can be incorporated as a third phase of such an approach because of its potential to enhance the performance of the solution approach to find higher quality solutions in a short computation time, and some of parameters that are used in this heuristics can be calibrated in statistical experimental design to provide better results.

REFERENCES

Agustina, D., Lee, C., Piplani, R., 2010. A review: Mathematical modles for cross docking planning. International Journal of Engineering Business Management 2, 13.

Agustina, D., Lee, C., Piplani, R., 2014. Vehicle scheduling and routing at a cross docking center for food supply chains. International Journal of production economics 152, 29–41.

Ahmadizar, F., Zeynivand, M., Arkat, J., 2015. Two-level vehicle routing with cross-docking in a three-echelon supply chain: A genetic algorithm approach. Applied Mathematical Modelling 39 (22), 7065–7081.

Al Theeb, N., Murray, C., 2017. Vehicle routing and resource distribution in postdisaster humanitarian relief operations. International Transactions in Operational Research 24 (6), 1253–1284.

Apte, U. M., Viswanathan, S., 2000. Effective cross docking for improving distribution efficiencies. International Journal of Logistics 3 (3), 291–302.

Birim, Ş., 2016. Vehicle routing problem with cross docking: A simulated annealing approach. Procedia-Social and Behavioral Sciences 235, 149–158.

Bodnar, P., de Koster, R., Azadeh, K., 2015. Scheduling trucks in a cross-dock with mixed service mode dock doors. Transportation Science 51 (1), 112–131.

Boysen, N., Fliedner, M., 2010. Cross dock scheduling: Classification, literature review and research agenda. Omega 38 (6), 413–422.

Cortés, C. E., Matamala, M., Contardo, C., 2010. The pickup and delivery problem with transfers: Formulation and a branch-and-cut solution method. European Journal of Operational Research 200 (3), 711–724.

Davarzani, H., Norrman, A., 2015. Toward a relevant agenda for warehousing research: literature review and practitioners input. Logistics Research 8 (1), 1.

Dondo, R., Cerdá, J., 2013. A sweep-heuristic based formulation for the vehicle routing problem with cross-docking. Computers & Chemical Engineering 48, 293–311.

Dondo, R., Cerdá, J., 2014. A monolithic approach to vehicle routing and operations scheduling of a cross-dock system with multiple dock doors. Computers & Chemical Engineering 63, 184–205.

Dondo, R., Cerdá, J., 2015. The heterogeneous vehicle routing and truck scheduling problem in a multi-door cross-dock system. Computers & Chemical Engineering 76, 42–62.

Grangier, P., Gendreau, M., Lehuédé, F., Rousseau, L.-M., 2017. A matheuristic based on large neighborhood search for the vehicle routing problem with cross-docking. Computers & Operations Research 84, 116–126.

Hasani-Goodarzi, A., Tavakkoli-Moghaddam, R., 2012. Capacitated vehicle routing problem for multi-product cross-docking with split deliveries and pickups. Procedia-Social and Behavioral Sciences 62, 1360–1365.

Kachitvichyanukul, V., Sombuntham, P., Kunnapapdeelert, S., 2015. Two solution representations for solving multi-depot vehicle routing problem with multiple pickup and delivery requests via pso. Computers & Industrial Engineering 89, 125–136.

Kreng, V. B., Chen, F.-T., 2008. The benefits of a cross-docking delivery strategy: a supply chain collaboration approach. Production Planning and Control 19 (3), 229–241.

Küçükoğlu, İ., Öztürk, N., 2015. A hybrid meta-heuristic algorithm for vehicle routing and packing problem with cross-docking. Journal of Intelligent Manufacturing, 1–17.

Larbi, R., Alpan, G., Penz, B., 2009. Scheduling transshipment operations in a multiple inbound and outbound door crossdock. In: Computers & Industrial Engineering, 2009. CIE 2009. International Conference on. IEEE, pp. 227–232.

Lee, Y. H., Jung, J. W., Lee, K. M., 2006. Vehicle routing scheduling for cross-docking in the supply chain. Computers & Industrial Engineering 51 (2), 247–256.

Liao, C.-J., Lin, Y., Shih, S. C., 2010. Vehicle routing with cross-docking in the supply chain. Expert Systems with Applications 37 (10), 6868–6873.

Maknoon, M., Soumis, F., Baptiste, P., 2017. An integer programming approach to scheduling the transshipment of products at cross-docks in less-than-truckload industries. Computers & Operations Research 82, 167–179.

Morais, V. W., Mateus, G. R., Noronha, T. F., 2014. Iterated local search heuristics for the vehicle routing problem with cross-docking. Expert Systems with Applications 41 (16), 7495–7506.

Mousavi, S. M., Tavakkoli-Moghaddam, R., 2013. A hybrid simulated annealing algorithm for location and routing scheduling problems with cross-docking in the supply chain. Journal of Manufacturing Systems 32 (2), 335–347.

Santos, F. A., Mateus, G. R., Da Cunha, A. S., 2013. The pickup and delivery problem with cross-docking. Computers & Operations Research 40 (4), 1085–1093.

Serrano, C., Moral, J., Delorme, X., Dolgui, A., 2016. Cross-docking operation scheduling: Truck arrivals, shop-floor activities and truck departures. IFAC-PapersOnLine 49 (12), 1353–1358.

Song, K., Chen, F., 2007. Scheduling cross docking logistics optimization problem with multiple inbound vehicles and one outbound vehicle. In: Automation and Logistics, 2007 IEEE International Conference on. IEEE, pp. 3089–3094.

Sung, C. S., Song, S. H., 2003. Integrated service network design for a cross-docking supply chain network. Journal of the Operational Research Society 54 (12), 1283–1295.

Tarantilis, C. D., 2013. Adaptive multi-restart tabu search algorithm for the vehicle routing problem with cross-docking. Optimization letters 7 (7), 1583–1596.

Vahdani, B., Zandieh, M., 2010. Scheduling trucks in cross-docking systems: Robust meta-heuristics. Computers & Industrial Engineering 58 (1), 12–24.

Van Belle, J., Valckenaers, P., Cattrysse, D., 2012. Cross-docking: State of the art. Omega 40 (6), 827–846.

Vincent, F. Y., Jewpanya, P., Redi, A. P., 2016. Open vehicle routing problem with cross-docking. Computers & Industrial Engineering 94, 6–17.

Wang, C., Mu, D., Zhao, F., Sutherland, J. W., 2015. A parallel simulated annealing method for the vehicle routing problem with simultaneous pickup–delivery and time windows. Computers & Industrial Engineering 83, 111–122.

Wen, M., Larsen, J., Clausen, J., Cordeau, J.-F., Laporte, G., 2009. Vehicle routing with cross-docking. Journal of the Operational Research Society 60 (12), 1708–1718.