Multi-Period Vehicle Routing Problem with Cross-Docking using Split Pickups and Deliveries

Muhammad Mustafa Ismail Turner, Komarudin
Department of Industrial Engineering, Faculty of Engineering
Universitas Indonesia
Depok, Indonesia
muhammad.mustafa01@ui.ac.id, komarudin74@ui.ac.id

Abstract

Cross-docking provides benefits such as facilitating fast and direct unloading, consolidation, and reloading of goods from inbound to outbound trucks. The vehicle routing problem with cross-docking (VRPCD) is an extension of the vehicle routing problem with an additional constraint in which the depot has a role as cross-dock. The current paper provides a mixed-integer linear programming model for the VRPCD that considers multiple products, multiple vehicle types, and split pickups and delivery. Based on the literature, we generated two sets of instances, one with a homogenous vehicle and one with heterogeneous vehicles. The instances are then solved using the Gurobi Solver. Results show that instances with heterogeneous vehicles provide better results but require an increase in computation time (median of 37.3% for a 10-node instance with five types of vehicles). Split pickups and deliveries can produce lower objective values and generate solutions previously infeasible. The computational results also show that smaller instances of 10 nodes can be solved in a reasonable amount of time. However, larger instances of 30 nodes are not optimally solved given a limited computation time of 1 hour.

Keywords
Vehicle routing problem, Cross-docking, Delivery periods, Split pickup and delivery, Mixed integer.

1. Introduction

Cross-docking is a supply chain strategy in which products are consolidated in a distribution center without storing them before sending them to customers. The distribution center is referred to as a cross-dock, where products move through them with little to no dwelling period. Using the cross-docking strategy has the benefit of allowing consolidation while reducing holding times, which in turn reduces holding costs, order cycle time, and transportation costs (Apte & Viswanathan, 2000). The advantages of cross-docking have caused multiple companies to adopt and use the strategy, such as Walmart. In a commercial network such as that, multiple products by multiple suppliers need to be sent to different customers with varying demands (Su & Liao, 2021).

The Vehicle Routing Problem (VRP) is a problem that plays a prominent role in logistical and supply chain networks. The VRP is a problem where the objective is to find optimal delivery routes from at least one depot to several customers (Barbarosoglu & Ozgur 1999). Several additional constraints further categorize the VRP, such as Capacitated VRP, Heterogeneous VRP, Muti-Depot VRP, Split Delivery VRP, and VRP with Cross-Docking (Archetti & Speranza 2008; Han & Wang, 2018; Mavi et al. 2020). Previous studies on VRPCD linear models have provided focused on different factors such as split deliveries and multiple products (Gunawan et al. 2020; Hasani-Goodarzi & Tavakkoli-Moghaddam 2012). Other studies on this topic use non-linear models to represent the problem and heuristics to generate solutions (Birim 2016; Lee et al. 2006).

1.1 Objectives

This study attempts to create a linear model VRPCD with a combination of factors from previous studies, namely split pickups, split deliveries, available routes, multiple products, and heterogeneous vehicles. The proposed model also aims to add multiple period deliveries, which means that vehicles don't need start the delivery at the same time, provided that they have enough inventory. Modified parameters to generate data proposed by Lee et al. (2006) are used by multiple other studies as a benchmark dataset (Birim 2016; Gunawan et al. 2020; Liao et al. 2010; Vahdani et al. 2012). Thus, this study also uses generated data with parameters based on Lee et al. (2006).

2. Literature Review

Lee et al. (2006) were the first in researching the application of cross-docking to the pickup and delivery processes found in the Vehicle Routing and Scheduling Problem. They developed a mathematical model which minimizes transportation and renting vehicle costs used to pick up and deliver products through a cross-dock (Lee et al. 2006). Because of the NP-hard classification of the problem, they used a Tabu Search algorithm to obtain a solution in a reasonable amount of time (Lee et al. 2006). The solution is then compared with solutions solved using Enumeration. Results show that their average quality is only 3.77% lower than the Enumeration method.

Subsequently, Wen et al. (2009) were the first to define the vehicle routing problem with cross-docking (VRPCD) as a problem of transporting orders using a cross-docking strategy. Orders are sent using a homogenous fleet of vehicles that pick them up from a set of suppliers, consolidate them in a cross-dock, and deliver them to customers. The method used was a Tabu Search Algorithm embedded within an adaptive memory procedure and aggressively narrowing down the neighborhood sizes within short computation times. The algorithm was tested using real-world datasets provided by a Danish consultancy firm. The results show that the aggressive approach decreases computational times significantly while maintaining almost the same quality of solutions.

To obtain the solutions for vehicle routing problems with cross-docking, Liao et al. (2010) use a Tabu Search Algorithm, the same approach as Lee et al. (2006), with two significant differences. The first is arranging nodes to vehicles one at a time, and the second is to allow the removal of an empty vehicle. The proposed Tabu Search Algorithm provided solutions 10-36% higher quality than Lee et al. (2006), while also using much less computational time.

Santos et al. (2011) proposed a novel column generation algorithm to solve the VRPCD and compared a branch-and-cut, GRASP, and branch-and-price algorithm. The dataset used is a real-world dataset based on Wen et al. (2009). The computational results show that the previous branch-and-price had deep symmetry problems. The novel branch-and-price provides solutions closer to the optimal values for all instances.

Hasani-Goodarzi and Tavakkoli-Moghaddam (2012) introduced split deliveries and pickups to the VRPCD, allowing supply and demand nodes to be visited by different vehicles. A mixed-integer model is formulated and solved using the GAMS software. Ten test problems were generated with 3-6 supply nodes and 4-6 delivery nodes. The experiment shows that the proposed model is applicable in practice.

Vahdani et al. (2012)proposed a hybrid metaheuristic algorithm that combines parts of simulated annealing, particle swarm optimization, and variable neighborhood search algorithms. The Taguchi experimental design is also used to tune the remaining parameters of the algorithm. The problem set is generated based on Lee et al. (2006), and the result is compared with Enumeration and a Tabu Search Algorithm. The result illustrates that the proposed algorithm provides a better solution than Tabu Search in larger problems, while also being faster than using Enumeration.

Morais et al. (2014) proposed three heuristics for VRPCD based on the Iterated Local Search metaheuristic (ILS), which only explores the space of feasible solutions. The heuristics proposed are S-ILS, X-ILS, and SP-ILS. A new set of instances were also used alongside the instances proposed by Wen et al. (2009). Computational results show that the X-ILS outperforms S-ILS, and the intensification procedure of SP-ILS is more efficient than the intensification procedure of X-ILS.

Mokhtarinejad et al. (2015) includes the possibility of direct shipment from the supplier to the customer without going through a cross-dock. A machine-learning-based heuristic method based on the filter learning method is used to group together clusters of customer, supplier, and cross-dock nodes. To solve the problem, an exact method and a genetic algorithm were used. The genetic algorithm is used to solve larger instances. Statistical analysis shows promising results in both the performance and efficiency of the proposed model.

Yu et al. (2016) uses a mixed integer linear program to model the open VRP with cross-docking (OVRPCD). The OVRPCD is different from VRPCD in that it creates an open network where the flow starts from the supply nodes and ends in the customer node through a cross-dock without forming a loop. Like VRPCD, the objective function of the OVRPCD is to minimize transport and hiring cost of vehicles. A simulated annealing algorithm is proposed to solve the problem, and its solution is compared to the solution obtained from a CPLEX solver. Results show the gap of solutions between the two methods being less than 1% and the computational time of the SA algorithm being considerably faster.

Birim (2016) addressed a VRPCD with heterogenous capacity vehicles. All nodes can only be visited by one vehicle, and all routes starts and terminates at the cross-dock. The algorithm used to solve this problem is a simulated annealing algorithm, with 30 generated instances that were based on the smallest instance provided in Lee et al. (2006). Like the model in Lee et al. (2006), the objective is to minimize the transportation and operational cost of vehicles. However, time-based constraints weren't considered in this research.

Kaboudani et al. (2020) was the first to investigate the VRPCD regarding forward and reverse logistics by using two strategies. In the proposed mathematical model, products must go through the cross-dock before going to the customers and the process is divided into pickup and delivery processes. The fist strategy consists of three stages, a pickup stage, a delivery stage, and a stage using VRP to consider both forward and reverse logistics. The second strategy eliminates the third stage and merges into the first stage, allowing only two stages. The model is solve using a simulated annealing based heuristic algorithm and were compared with the solution from a solver from GAMS software. Results show that the second strategy produces better solutions, especially in smaller instances. The proposed algorithm gives competitive results while being significantly faster than the GAMS solver.

Gunawan et al. (2020) extends the work of Lee et al. (2006) by modeling a VRPCD with multiple products and a set of homogenous vehicles, being called a VRP-MPCD. A mixed integer linear programming model was proposed and solved using the CPLEX solver. The generated instances were derived from Lee et al. (2006), where the supply and demand parameter were changed to accommodate the proposed model. 2 sets of 10 and 30 nodes were generated, each with 30 instances. Results show that optimal solutions in the 10-node set can all be found using the solver in less than 2 seconds. However, only 2 instances could be solved optimally in the 30-node set.

3. Method

This study uses mixed-integer linear programming to model the proposed VRP with Cross-Docking. The problem in this study is based on the work of Lee et al. (2006) and provides an extended and alternative model of the work done by Gunawan et al. (2020), by allowing the use split pickups, split deliveries, available routes, and heterogeneous vehicles. Furthermore, our proposed model allows vehicles to start the delivery process without waiting for all vehicles to reach the cross-dock, provided that the inventory in the cross-dock can fulfill the demand in the vehicle's route. Figure 1 shows the model representation.

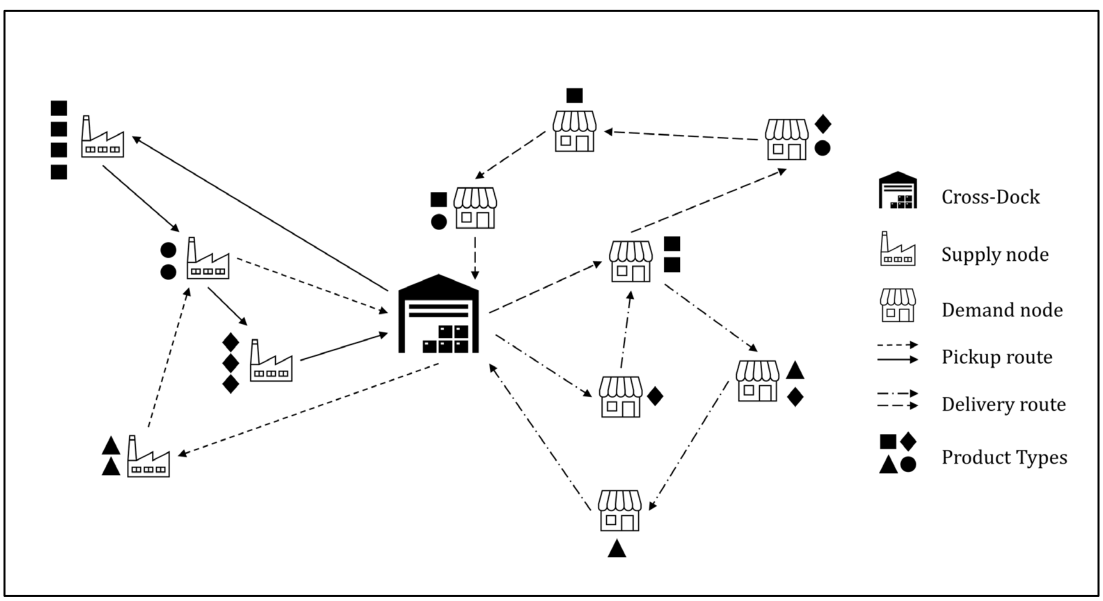

Figure 1. Model Representation

3.1 Parameters and Variables

Model parameters and variables are shown in Tables 1 and 2. The parameters and variables are based on the work of Lee et al. (2006), with additions that help the model come closer to real-world scenarios such as having different sets of products with varying volumes, different vehicle capacities and hiring costs, and available routes. Additions that enable the use of multiple delivery periods can be seen with the delivery period parameter and the variables which deals with the inventory.

Table 1. Parameter descriptions

Parameter	**Description**
S	Set of supply nodes
D	Set of delivery nodes
0	Cross-dock
N	All nodes ($N \in S \cup D \cup 0$)
V	Set of vehicles
P	Set of products
C_v	Capacity of vehicle v, ($v \in V$)
Q_p	Volume of product p, ($p \in P$)
s_{ip}	Supply quantity in supply node i for product p, ($i \in S, \quad p \in P$)
d_{ip}	Demand quantity in demand node i for product p, ($i \in D, \quad p \in P$)
tc_{ij}	Transportation cost from node i to node j, ($i, j \in N$)
et_{ij}	Transportation time from node i to node j, ($i, j \in N$)
c_v	Hiring cost of vehicle v, ($v \in V$)
R_{ij}	1: if route i to j is available; 0: otherwise ($i, j \in N$)
T	Time horizon
TP	Delivery periods
M	Large number

Table 2. Variable Descriptions

Variable	**Description**
y_{pvij}	Amount of product p in vehicle v at node i heading to node j, ($v \in V,\ i, j \in N,\ p \in P$)
z_v	1: if vehicle v is used; 0: otherwise ($v \in V$)
I_{pt}	Cross dock inventory for product p during period t, ($v \in V,\ p \in P,\ t \in TP$)
IA_{vpt}	Product p unloaded from vehicle v, arriving to cross dock at period t, ($v \in V,\ p \in P,\ t \in TP$)
ID_{vpt}	Product p loaded to vehicle v, $departing$ from cross dock at period t, ($v \in V\ p \in P\ t \in TP$)
DT_{vij}	Departure time of vehicle v at node i heading to node j, ($v \in V\ i, j \in N$)
CD_{vt}	1: if vehicle v arrives at cross dock during period t; 0: otherwise ($v \in V\ t \in TP$)
AT_t	Latest arrival time at cross-dock for period t, ($t \in TP$)
x_{vij}	1: if vehicle v moves from node i to node j; 0: otherwise ($v \in V\ i, j \in N$)

3.2 Objective Function and Constraints

$$min. \quad \sum_{v \in V} \sum_{i \in N} \sum_{j \in N} tc_{ij} x_{vij} + \sum_{v \in V} c_v z_v \tag{1}$$

The objective function (1) calculates the minimum transportation costs and vehicle hiring costs.

$$\sum_{k \in N} x_{vjk} = \sum_{i \in N} x_{vij} \quad v \in V, \quad j \in N \tag{2}$$

$$x_{vii} = 0 \quad v \in V, \quad i \in N \tag{3}$$

$$x_{vij} \leq R_{ij} \quad v \in V, \quad i,j \in N \tag{4}$$

$$\sum_{i \in 0 \cup S} x_{vij} \leq 1 \quad j \in 0 \cup S \tag{5}$$

$$\sum_{i \in 0 \cup D} x_{vij} \leq 1 \quad j \in 0 \cup D \tag{6}$$

$$x_{vij} = 0 \quad v \in V, \quad (i,j) \in (S,D) \cup (D,S) \tag{7}$$

Constraints (2)-(7) are routing constraints. Constraint (2) ensures the consecutive movement of vehicles. Constraint (3) eliminates routes that only stay in one node. Constraint (4) ensures that vehicles can only use available routes. Constraints (5) and (6) ensures that each vehicle can only go through each pickup and delivery node once. Constraint (7) differentiates between pickup and delivery routes.

$$\sum_{i \in N} \sum_{j \in N} x_{vij} \leq z_v M \quad v \in V \tag{8}$$

$$\sum_{p \in P} y_{pvij} Q_p \leq C_v \quad v \in V, \quad i,j \in N \tag{9}$$

Constraints (8)-(9) are vehicle constraints. Constraint (8) counts the number of vehicles used. Constraint (9) ensures the amount of product picked up does not exceed vehicle capacity.

$$\sum_{v \in V} \sum_{i \in S} y_{pvio} \geq \sum_{i \in S} s_{ip} \quad o = 0, \quad p \in P \tag{10}$$

$$\sum_{v \in V} \sum_{i \in s} y_{pvio} \geq \sum_{v \in V} \sum_{j \in D} y_{pvoj} \quad o = 0, \quad p \in P \tag{11}$$

Constraints (10)-(11) are flow constraints. Constraint (10) ensures that all product in supplier nodes gets picked up. Constraint (11) ensures that the total supply is at least the same as the total delivered goods.

$$\sum_{v \in V} y_{pvoj} \quad o = 0, \quad p \in P, \quad j \in 0 \cup S \tag{12}$$

$$\sum_{v\in V}\sum_{k\in S\cup 0} y_{pvjk} - \sum_{v\in V}\sum_{i\in S\cup 0} y_{pvij} \le s_{jp} \quad p\in P, \quad j\in S \tag{13}$$

$$\sum_{v\in V}\sum_{i\in D\cup 0} y_{pvij} - \sum_{v\in V}\sum_{k\in D\cup 0} y_{pvjk} \ge d_{jp} \quad p\in P, \quad j\in D \tag{14}$$

$$\sum_{j\in S\cup 0} y_{pvij} \ge \sum_{j\in S\cup 0} y_{pvji} \quad v\in V, \quad p\in P, \quad i\in S \tag{15}$$

$$\sum_{j\in D\cup 0} y_{pvij} \le \sum_{j\in D\cup 0} y_{pvji} \quad v\in V, \quad p\in P, \quad i\in D \tag{16}$$

$$Mx_{vij} \ge \sum_{p\in P} y_{pvij} \quad v\in V, \quad i,j\in N \tag{17}$$

Constraints (12)-(17) deal with products in vehicles. Constraint (12) initializes the vehicle inventory when a vehicle goes from a cross-dock to a supply node. Constraint (13) ensures the number of products picked up in supply nodes does not exceed the amount of inventory in each supply node. Constraint (14) ensures the number of products delivered in demand nodes meets the demand in each demand node. Constraint (15) prohibits unloading products in pickup nodes while constraint (16) prohibits loading products in demand nodes. Constraint (17) ensures that loading and unloading can only be done in used routes.

$$DT_{vij} + et_{ij} \le T \quad v\in V, \quad i,j\in N \tag{18}$$

$$DT_{vij} \le Mx_{vij} \quad v\in V, \quad i,j\in N \tag{19}$$

$$DT_{voj} = 0 \quad v\in V, \quad o=0, \quad j\in S\cup 0 \tag{20}$$

$$DT_{vjk} \le \sum_{i\in N}\left(et_{ij}x_{vij} + DT_{vij}\right) \quad v\in V, \quad j\in S\cup D, \quad k\in N \tag{21}$$

$$\sum_{k\in N} DT_{vjk} \ge \sum_{i\in N}\left(et_{ij}x_{vij} + DT_{vij}\right) \quad v\in V, \quad j\in S\cup D \tag{22}$$

Constraints (18)-(22) are time constraints. Constraint (18) ensures that all deliveries fall under the time horizon. Constraint (19) ensures that only used routes have their departure time calculated. Constraint (20) initializes the departure time in the starting node, which is the cross-dock. Constraints (21) and (22) calculates the departure time of vehicles in used routes.

$$\sum_{t\in TP} IA_{vpt} \le \sum_{i\in S} y_{viop} \quad v\in V, \quad p\in P, \quad o=0 \tag{23}$$

$$\sum_{t\in TP} ID_{vpt} \ge \sum_{j\in D} y_{vojp} \quad v\in V, \quad p\in P, \quad o=0 \tag{24}$$

$$\sum_{p\in P} IA_{vpt} \le CD_{vt}M \quad v\in V, \quad t\in TP \tag{25}$$

$$\sum_{p\in P} ID_{vpt} \leq CD_{vt}M \quad v \in V, \quad t \in TP \tag{26}$$

$$I_{pt} = \sum_{k\in K} IA_{kpt} - \sum_{k\in K} ID_{kpt} \quad p \in P, \quad t = 0 \tag{27}$$

$$I_{pt} = I_{pt-1} + \sum_{k\in K} IA_{kpt} - \sum_{k\in K} ID_{kpt} \quad p \in P, \quad t \in TP, \quad t \neq 0 \tag{28}$$

$$\sum_{t\in TP} CD_{vt} \geq 1 - M(1 - z_v) \quad v \in V \tag{29}$$

$$\sum_{t\in TP} CD_{vt} \leq 1 \quad v \in V \tag{30}$$

Constraints (23)-(30) deals with calculating inventory in the cross-dock. Constraints (23) and (24) calculates the arriving and departing inventory at the cross-dock that are loaded and unloaded from vehicles for each delivery period. Constraints (25) and (26) ensures that cross-dock inventory is only updated at the vehicles chosen delivery periods. Constraints (27) and (28) calculates the cross-dock inventory for each delivery period. Constraints (29) and (30) assigns each vehicle to a delivery period.

$$AT_t \geq AT_{t-1} \quad t \in TP, \quad t \neq 0 \tag{31}$$

$$AT_t \geq \sum_{i\in S} (et_{io}x_{vio} + DT_{vio}) - M(1 - CD_{vt}) \quad v \in V, \quad t \in TP \tag{32}$$

$$\sum_{j\in D} DT_{voj} \leq AT_t - M(1 - CD_{vt}) \quad v \in V, \quad o = 0, \quad t \in TP \tag{33}$$

$$\sum_{j\in D} DT_{voj} \geq AT_t + M(1 - CD_{vt}) \quad v \in V, \quad o = 0, \quad t \in TP \tag{34}$$

Constraints (31)-(34) deal with arrivals at the cross dock and starting delivery times. Constraint (31) ensures that the delivery period times are ordered chronologically. Constraint (32) finds the latest arrival time at the cross-dock for each delivery period. Constraints (33) and (34) ensures that the departure time of delivery routes are at least the time of the latest arrival at the cross-dock.

$$z_v, x_{vij}, CD_{vt} \in \{0,1\} \tag{35}$$

Constraint (35) shows the binary variables.

4. Results and Discussion

Computational results of the proposed model are run in a generated dataset are run using the Gurobi solver using the Python programming language. The computer used to run the model uses an AMD Ryzen 7 3700x CPU with a clock speed of 4.1 GHz and 32 GB RAM.

4.1 Generated Instances

Two sets of 10 instances are generated. The first uses the same method of instance generation as the first set by Gunawan et al. (2020). The second set of instances is a copy of the first set, except the vehicles are changed to be heterogeneous. Vehicles in the first set have a hiring cost of 1000 and a capacity of 70. In the second set, there are vehicles with hiring costs of 600, 800, 1000, 1200, and 1400, each having capacities of 50, 60, 70, 80, and 90, respectively. Each instance in the second set has access to two of each type of vehicle. All other parameters are generated in the same way, which can be seen in Table 3. It is also assumed that all routes are available to be used.

Table 3. Generated data parameter values

Parameters	Value
N	10
S	4
D	6
K	10
T	960
Q	70
c	1000
et_{ij}	Uniform (20,200)
tc_{ij}	Uniform (48,560)
s_{ip}, d_{ip}	Uniform (5,50)

4.2 Computational Results

Table 4 shows the computational results of the generated data solved by the Gurobi solver. All the generated instances could be solved optimally by the Gurobi Solver. The results also show that the computational time varies between instances, with solve times as high as 228 seconds in instance 8. The solving times tend to get longer when considering heterogeneous vehicles, with an average increase of 79.65% and a median increase of 37.3%. We also tested instances with 30 nodes, however, the Gurobi Solver failed to find optimal solutions within a time limit of 1 hour. This shows that computational times for harder problems cannot be solved within a reasonable amount of time using the Gurobi solver. For larger problems, an efficient heuristic method should be considered.

Table 4. Computational Results

Instance	Homogenous Vehicles			Heterogeneous Vehicles		
	Objective Value	Time (s)	Vehicle Utilization	Objective Value	Time (s)	Vehicle Utilization
1	5123	10.26	2/10	4723	15.79	2/10
2	4763	14.75	2/10	4375	10.54	2/10
3	6306	10.13	3/10	5178	28.49	2/10
4	7530	45.44	3/10	6302	19.89	2/10
5	4724	24.31	2/10	4268	16.83	2/10
6	8161	35.70	3/10	6780	21.35	2/10
7	5391	18.99	2/10	5176	40.42	2/10
8	5965	132.47	3/10	5204	228.67	3/10
9	5344	19.86	2/10	4856	121.27	2/10
10	5413	17.41	2/10	4972	21.01	2/10

4.3 Split Pickups and Deliveries

To show the effectiveness of split pickups and deliveries, we ran the model in all instances with an increase in supply and demand. The result of this can be seen in Figure 2. In some instances, during conditions with lower supply and demand, the solution given by the model using split pickups and deliveries is the same as a model without split pickups and deliveries. However, if we look at the average cost from all instances, having split pickups and deliveries helps lower the total cost. Split pickups also help with solution feasibility. In instances with a 150% increase in supply and demand, only 5/10 instances have feasible solutions without split pickups and delivery. Increasing the supply and demand even more removes solution feasibility entirely, as seen in the 170% and 190% increase. Using split pickups and deliveries, all tested instances have feasible solutions.

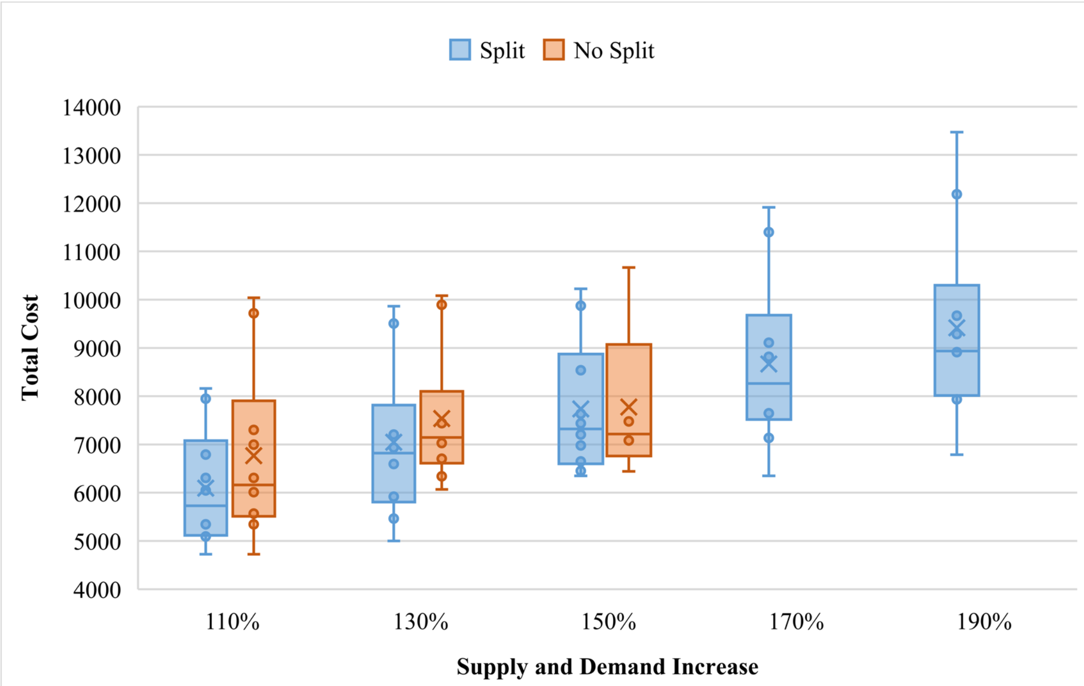

Figure 2. Effect of supply/demand increase towards total cost with and without split delivery

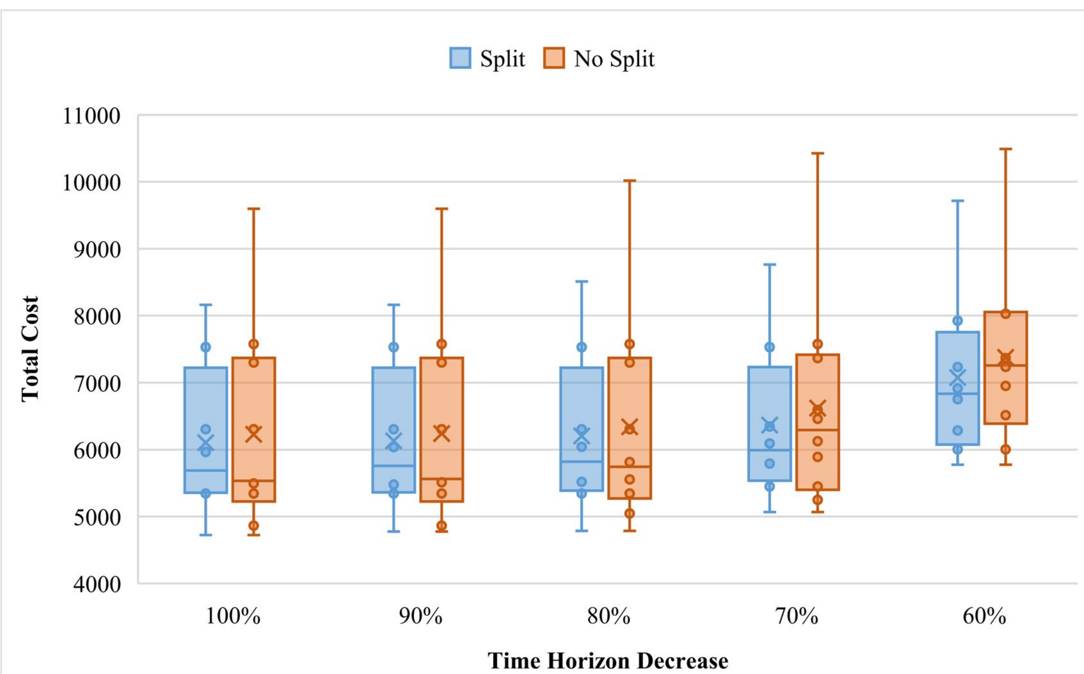

Figure 3 Effect of time horizon decrease towards total cost with and without split delivery

Other than increasing supply and demand, we also tested the effects of decreasing time horizon towards the total cost, with and without split pickups and deliveries, which can be seen in Figure 3. Like the previous case, the decrease in the time horizon causes an increase in the total cost. In the tested instances, split deliveries also enable previously infeasible cases. Also like the previous case, having split pickups and deliveries decreases the cost compared to not having them. Specifically, in the case of increasing supply and demand having split deliveries lowers the average cost by 4.45%, in the case of reducing time horizon, it reduces the average cost by 6.40% (infeasible cases are ignored in the calculation).

5. Conclusion

We proposed a mixed-integer linear model of a VRPCD problem with multiple products, heterogeneous vehicles, split pickup, split deliveries, and available routes. However, in this case, we assumed that all routes are available for use. We also modified the VRP-MPCD benchmark instances to accommodate heterogeneous vehicles. The instances are solved using our proposed mixed-integer linear model with the Gurobi solver. Results show that instances with heterogeneous vehicles provide better results with a median of 37.3% for a 10-node instance with five types of vehicles. However, having heterogeneous vehicles require a longer computational time. A more efficient heuristic method can be used to solve larger problems, which can be done as an opportunity for future work. We also tested the advantages of having split pickups and deliveries, which can help at higher demand numbers and even produce solutions that were infeasible for the model without split pickups and deliveries. Other avenues of future work may also focus on formulating a model which ties delivery speed to the objective function.

References

Apte, U. M., & Viswanathan, S., Effective Cross Docking for Improving Distribution Efficiencies, *International Journal of Logistics Research and Applications*, vol. 3, no. 3, 2000

Archetti, C., & Speranza, M. G, The split delivery vehicle routing problem: A survey, *Operations Research/ Computer Science Interfaces Series*, vol. 43, 2008

Barbarosoglu, G., & Ozgur, D., A tabu search algorithm for the vehicle routing problem, *Computers & Operations Research*, vol. 26, no. 3, pp. 255–270, 1999

Birim, Ş., Vehicle Routing Problem with Cross Docking: A Simulated Annealing Approach, *Procedia - Social and Behavioral Sciences*, vol. 235, 2016

Gunawan, A., Widjaja, A. T., Siew, B. G. K., Yu, V. F., & Jodiawan, P., Vehicle routing problem for multi-product cross-docking, *Proceedings of the International Conference on Industrial Engineering and Operations Management*, pp. 66–77, Dubai, UAE, March 10-12, 2020

Han, M., & Wang, Y., A Survey for Vehicle Routing Problems and Its Derivatives, *IOP Conference Series: Materials Science and Engineering*, vol. 452, no. 4, 2018

Hasani-Goodarzi, A., & Tavakkoli-Moghaddam, R., Capacitated Vehicle Routing Problem for Multi-Product Cross-Docking with Split Deliveries and Pickups, *Procedia - Social and Behavioral Sciences*, vol. 62, 2012

Kaboudani, Y., Ghodsypour, S. H., Kia, H., & Shahmardan, A., Vehicle routing and scheduling in cross docks with forward and reverse logistics, *Operational Research*, vol. 20, no. 3, 2020

Lee, Y. H., Jung, J. W., & Lee, K. M., Vehicle routing scheduling for cross-docking in the supply chain, *Computers and Industrial Engineering*, vol. 51, no. 2, 2006

Liao, C. J., Lin, Y., & Shih, S. C., Vehicle routing with cross-docking in the supply chain, *Expert Systems with Applications*, vol. 37, no. 10, 2010

Mavi, R. K., Goh, M., Mavi, N. K., Jie, F., Brown, K., Biermann, S., & Khanfar, A. A., Cross-docking: A systematic literature review, *Sustainability (Switzerland),* vol. 12, 2020

Mokhtarinejad, M., Ahmadi, A., Karimi, B., & Rahmati, S. H. A., A novel learning based approach for a new integrated location-routing and scheduling problem within cross-docking considering direct shipment, *Applied Soft Computing Journal*, vol. 34, 2015

Morais, V. W. C., Mateus, G. R., & Noronha, T. F., Iterated local search heuristics for the Vehicle Routing Problem with Cross-Docking, *Expert Systems with Applications*, vol. 41, no. 16, 2014

Santos, F. A., Mateus, G. R., & Salles Da Cunha, A., A novel column generation algorithm for the vehicle routing problem with cross-docking, *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 6701 LNCS, 2011

Su, T. S., & Liao, T. W., The Delivery Strategy in Cross Docking Operations with Overlapped Temperature Region and Time Windows Constraints, *2021 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, pp. 32–36, Singapore, Singapore, December 13-16, 2021

Vahdani, B., Reza, T. M., Zandieh, M., & Razmi, J., Vehicle routing scheduling using an enhanced hybrid optimization approach, *Journal of Intelligent Manufacturing*, vol. 23, no. 3, 2012

Wen, M., Larsen, J., Clausen, J., Cordeau, J. F., & Laporte, G., Vehicle routing with cross-docking, *Journal of the Operational Research Society*, vol. 60, no. 12, 2009

Yu, V. F., Jewpanya, P., & Redi, A. A. N. P., Open vehicle routing problem with cross-docking, *Computers and Industrial Engineering*, vol. 94, 2016

Biographies

Muhammad Mustafa Ismail Turner, S.T., is a master's student studying systems design and management at the Department of Industrial Engineering, Faculty of Engineering, Universitas Indonesia. Mustafa graduated with a bachelor's degree in Industrial Engineering from Universitas Indonesia. His research interests are in operations research, with an undergraduate thesis about the Facility Location Problem for Humanitarian Logistics and is currently researching the Vehicle Routing Problem.

Dr. Komarudin, S.T., M.Eng, is a Lecturer and Head of Department Industrial Engineering, Faculty of Engineering, Universitas Indonesia. Komarudin graduated from Vrije Universiteit Brussel. His research interest are optimization, mathematical programming, facility layout, scheduling, game theory, and many others. He currently researches the applications and developments of optimization methods and algorithms for solving industrial problems. He has published his research papers in, among others, European Journal of Operations Research, Annals of Operations Research and Journals of Operational Research Society, and many International Conferences.