

RESEARCH ARTICLE

Operations Research

Meta-heuristic method to schedule vehicle routing with moving shipments at the cross-docking facility

SR Gnanapragasam^{1,2*} and WB Daundasekera³

¹ Department of Mathematics, Faculty of Natural Sciences, The Open University of Sri Lanka, Nawala, Nugegoda, Sri Lanka.

² Postgraduate Institute of Science, University of Peradeniya, Peradeniya, Sri Lanka.

³ Department of Mathematics, Faculty of Science, University of Peradeniya, Peradeniya, Sri Lanka.

Submitted: 26 July 2022; Revised: 11 July 2023; Accepted: 28 July 2023

Abstract: Cross-Docking (CD) is a modern distribution strategy in a supply chain. The optimal scheduling of vehicle routing, known as the Vehicle Routing Problem (VRP), is one of the influential factors of the efficiency of a supply chain. In recent years, researchers and business consultants in different organizations have been interested in integrating the VRP with CD (VRPCD). Since VRPCD is a NP-hard problem, heuristic or meta-heuristic methods are always recommended to solve large-scale VRPCD. The Genetic Algorithm (GA) is a population based meta-heuristic algorithm and also, it is based on the principles of genetic and natural selections. The GA is capable of finding near optimal solutions to large-scale optimization problems which are extremely difficult to solve using traditional optimization algorithms. Therefore, in this study, a meta-heuristic approach based on the GA is proposed to solve the vehicle routing problem with moving shipments at the cross-docking facility (VRPCD&MS). The data are extracted from benchmark instances in the literature. The optimum solutions obtained to small-scale instances by the GA are compared with the exact solutions obtained by the Branch and Bound (BB) algorithm, which is a traditional algorithm to solve problems of this nature. The GA and BB algorithms are respectively coded in *MATLAB* and *LINGO*. The results reveal that the relative difference between the exact solution and the near-optimal solution is below 5%. Therefore, it can be concluded that the proposed GA is a better alternative method, considering its overall performance, to solve the VRPCD&MS models. Moreover, since the computational time is low, the proposed GA can be used to schedule the vehicles to the routes of VRPCD&MS at the last moment prior to the start of the time horizon.

Keywords: Cross-dock, genetic algorithm, meta-heuristic, moving shipments, vehicle routing.

INTRODUCTION

The efficiency of supply chains is crucial to maintain competitiveness in the globalised market around the world. The optimal scheduling of vehicle routing, known as the Vehicle Routing Problem (VRP), is one of the influential factors for a fast supply chain. Also, the VRP is one of the well-studied problems in the area of operations research and since the study of Dantzig and Ramser (1959), VRP has been subject to very intensive research by considering the different variants of the problem. In order to satisfy the demand of the customers in terms of time, quality, and cost, organizations always attempt new logistics strategies. Cross-docking is one of the key strategies in which business consultants in different organizations are interested. A Cross-Docking Facility (CDF) is an intermediate component of a supply chain. Generally, at a CDF, incoming products, after the consolidation process, are immediately dispatched to outgoing vehicles with minimum time delay. By doing so, cross docking reduces the cost incurred by traditional warehousing up to 70%, by eliminating some costly operations, such as storing and order picking, at those respective warehousing centres (Vahdani & Zandieh, 2010). In 1980, after a successful implementation of Cross

* Corresponding author (srgna@ou.ac.lk;  <https://orcid.org/0000-0003-1411-4853>)



This article is published under the Creative Commons CC-BY-ND License (<http://creativecommons.org/licenses/by-nd/4.0/>). This license permits use, distribution and reproduction, commercial and non-commercial, provided that the original work is properly cited and is not changed in anyway.

Docking at *Walmart*, which is a major multinational retail corporation (Apte & Viswanathan, 2000), other giant companies such as *Goodyear GB Ltd*, *Toyota*, *Eastman Kodak Co*, *Dots*, and *LLC* also successfully implemented Cross Docking in their supply chains (Van Belle *et al.*, 2012).

Initially ‘Vehicle Routing Problem’ and ‘Cross Docking’ have been treated separately in most studies, but after the study on the Vehicle Routing Problem with Cross Docking (VRPCD) by Lee *et al.* (2006), studies on the integration of the VRP with Cross Docking have attracted more attention among researchers and practitioners. For an effective VRPCD, there should be synchronization among the incoming vehicles which collect products from the suppliers and outgoing vehicles which distribute products to the customers. According to the literature, several characteristics of the VRP with Cross Docking technique with different methods of solution have been experimented with. However, the internal operations were not taken into consideration in almost all the past studies on VRPCD. In the literature review survey by Buakum and Wisittipanich (2019), it was recommended to consider the internal operations at the CDF to develop the models for VRPCD. Therefore, Gnanapragasam and Daundasekera (2022) integrated the component of ‘Moving Shipments inside a CDF’ with VRPCD and referred it as ‘VRPCD&MS’.

In the study (Gnanapragasam & Daundasekera, 2022), not only the moving shipments from indoors to outdoors of the CDF, but also loading and unloading shipments at all the nodes (suppliers or customers), including the CDF, were taken into consideration in VRPCD&MS. However, the solution method applied in this study was only capable of obtaining optimal solutions to small-scale instances, in which the total number of suppliers and customers was only up to 20. In general, reaching an exact optimal solution when the number of suppliers and customers exceeding 20, which is considered as a large-scale instance, is time consuming and hence, impractical in the business world. Therefore, Gnanapragasam & Daundasekera (2022) recommended applying meta-heuristic methods to reach near-optimal solutions to large-scale instances of VRPCD&MS. In this current study, a modified version of the Mixed Integer Non-Linear Programming model formulated by Gnanapragasam & Daundasekera is developed without considering time related constraints. Therefore, in this study, a meta-heuristic method based on the Genetic Algorithm is developed to reach near-optimal solutions to the large-scale instances of VRPCD&MS. It should be

emphasized that, in the development of the Mixed Integer Linear Programming model to VRPCD&MS, this study focuses only on the operation cost and does not include any time related constraints which were considered in the previous study.

The study on the integration of the VRP with Cross Docking was initiated in 2006 by Lee *et al.* (2006). In this initial study on VRPCD, two different sets of homogeneous vehicles, one for the pickup process and one for the delivery process, were used. Also, the simultaneous arrival of inbound vehicles to the CDF was assumed. Wen *et al.* (2009) extended the initial study by assigning the Time Windows to all the suppliers and customers. Moreover, the assumptions regarding the simultaneous arrival of inbound vehicles and two different sets of vehicles, made by Lee *et al.* (2006), were removed. On the other hand, the consolidation decisions with dependency rules and the same set of vehicles for pickup and delivery processes were introduced by Wen *et al.* (2009). Hence in the literature, this model has become a more generalized version of VRPCD. Different methods of solution have been tried out in the past to solve the variants of VRPCD, which are mentioned in the following paragraphs.

Since the VRPCD is a NP-hard problem, obtaining the exact solution is possible only for small-scale instances. The exact solution to the small-scale instances of the variants of VRPCD were obtained in the following studies which are stated in this paragraph. Santos *et al.* (2011a) used the Branch and Price algorithm to the model developed by Wen *et al.* (2009). This Branch and Price method outperformed the Branch and Bound approach for the same instances. In the same year, a Column Generation method was introduced by the same authors (Santos *et al.*, 2011b) and the overall results outperformed the results of Santos *et al.* (2011a). Multi commodity, splitting and heterogeneous characteristics were incorporated by Hasani-Goodarzi & Tavakkoli-Moghaddam (2012). Dondo (2013) applied a Branch and Price algorithm to solve the model of Wen *et al.* (2009). In the study by Gnanapragasam & Daundasekera (2022), a Branch and Bound algorithm is employed to reach the optimal solution to the VRPCD&MS.

Since the VRPCD is an optimization problem with high complexity, reaching an exact solution to a large-scale problem is extremely challenging. Meta-heuristic approaches to reach near optimal solutions to large-scale problems were also examined in the integrated research on the VRPCD, and the following studies in this paragraph

highlights them. Initially in the study of Lee *et al.* (2006), a Tabu Search approach was used. Subsequently the model of Wen *et al.* (2009) was also tested using a Tabu Search algorithm with Adaptive Memory Procedure. Later Liao *et al.* (2010) made use of another Tabu Search algorithm to obtain a better optimal solution than the one obtained by Lee *et al.* (2006). The open versus closed configurations to the VRPCD, described by Wen *et al.* (2009), were compared by Tarantilis (2013), by using a Tabu Search algorithm. Morais *et al.* (2014) attempted six different Iterative Local Search methods to solve the model of Wen *et al.* (2009) and obtained more promising solutions than Tarantilis (2013). The soft Time Windows characteristic was adapted with the model of Wen *et al.* (2009) by Fakhrzad & Sadri Esfahani (2014), using the Tabu Search and Variable Neighbourhood Search methods, and the Tabu Search outperformed the Variable Neighbourhood Search method in this attempt.

Furthermore, hybrid methods built by combining two or more meta-heuristics methods were also applied to solve the variants of VRPCD. A method by hybridizing Ant Colony System and Simulated Annealing was employed by Moghadam *et al.* (2014), with the added characteristic: splitting the order in the delivery process, to the model of Wen *et al.* (2009). It was revealed that the hybridized approach outperformed the method of using Simulated Annealing alone. Another hybridized approach combining the methods Variable Neighbourhood Search, Simulated Annealing, and Particle Swarm Optimization to the model of Lee *et al.* (2006) was utilized by Vahdani *et al.* (2012), who obtained more encouraging solutions than the previous. It was observed that a remarkable improvement to the solutions was obtained using the Simulated Annealing algorithm by Yu *et al.* (2014), compared to the solutions obtained by Liao *et al.* (2010). A mix of open and closed network configurations with splitting characteristics were tested on the VRPCD system by Alinaghian *et al.* (2016). Yin & Chuang (2016) adapted the environmental factor with the heterogeneous property to the model developed by Lee *et al.* (2006). The Adaptive Memory Artificial Bee Colony (AMABC) technique was applied to solve the Green VRPCD model presented by Yin & Chuang (2016), and AMABC has produced a more promising solution than was obtained by Liao *et al.* (2010). The model formulated by Wen *et al.* (2009) was enhanced by Nikolopoulou *et al.* (2016) by incorporating some characteristics such as many-to-many correspondence between suppliers and customers, different sets of heterogeneous vehicles, and splitting. The results obtained by the Adaptive Memory Procedure with Tabu Search by Nikolopoulou *et al.* (2016) were

parallel to the results obtained by Wen *et al.* (2009), and better than the results obtained by Morais *et al.* (2014). In the study of Larioui & Reghioui (2017), the solutions were compared by a Tabu Search and Memetic Algorithm to the model of Wen *et al.* (2009) and it was observed that the Memetic Algorithm outperformed the Tabu Search.

Moreover, the solutions to the additional characteristics with VRPCD have been attempted in the past and are highlighted as follows. The Open VRPCD developed by Yu *et al.* (2016) and the heterogeneous property added by Birim (2016) were modified versions of the method of Lee *et al.* (2006). The Simulated Annealing strategies were examined to solve both models by Yu *et al.* (2016) and Birim (2016). Early arrivals were allowed in the model of Wen *et al.* (2009) by Grangier *et al.* (2017) and a Large Neighbourhood Search with Set Partitioning and Matching approach generated some better solutions than the results from the studies of Wen *et al.* (2009), Tarantilis (2013), and Morais *et al.* (2014). A Two-phase Genetic Algorithm was employed in the study by Baniamerian *et al.* (2018b) by adding customer satisfaction to the model of Wen *et al.* (2009). The solutions obtained by a method developed by Gunawan *et al.* (2020a) with the combination of Adaptive Large Neighbourhood Search (ALNS) with Set Partitioning Problem and by Gunawan *et al.* (2020b) using ALNS were better than the solutions obtained from the studies conducted by Lee *et al.* (2006), Liao *et al.* (2010), and Yu *et al.* (2016).

It can be observed from the literature on VRPCD, that several solution methods were applied not only to the small-scale instances but also to the large-scale instances. Many models were developed to different variants of VRPCD. However, the internal operations of CDF were not taken into consideration. Moreover, as per the recommendation made by the literature survey of Buakum and Wisittipanich (2019), the component of moving shipments from indoors to outdoors of a CDF was incorporated with VRPCD by Gnanapragasam & Daundasekera (2022). However, in the previous study, exact solutions were obtained only for small-scale (total number of suppliers and customers not exceeding 20) instances of VRPCD&MS using the Branch and Bound algorithm and it was recommended to apply meta-heuristic methods to solve large-scale instances of VRPCD&MS.

It can be observed that the Genetic Algorithm was applied in some of the studies in the literature to solve VRPCD. For instance, VRPCD with multiple objectives

was considered in several studies (Hasani Goodarzi *et al.*, 2018; Kargari Esfand Abad *et al.*, 2018; Ieva *et al.*, 2022). Moreover, the study by Baniamerian *et al.* (2018a) followed an approach hybridized with the Genetic Algorithm to maximize the total profit of the system. A two-phase Genetic Algorithm was developed by Baniamerian *et al.* (2018b) by considering customer satisfaction and time windows with VRPCD. For a VRPCD with a queuing approach, the Genetic Algorithm was proposed by Hasani Goodarzi *et al.* (2022). Besides, there is no study found in the literature that uses the Genetic Algorithm to minimize the total cost with a single objective. Therefore, in this current study, the Genetic Algorithm is employed to solve the VRPCD&MS constructed by Gnanapragasam & Daundasekera (2022).

METHODS AND MATERIALS

The integrated problem of Vehicle Routing with Cross Docking and Moving Shipments is described in the next section. The ‘moving shipments’ is a component of the internal operations at the cross-docking facility. Also, the developed mathematical model for the integrated problem has been presented.

Vehicle routing problem with cross-docking and moving shipments

The Vehicle Routing Problem with Cross-docking and Moving Shipment (VRPCD&MS) is a problem already studied by Gnanapragasam & Daundasekera (2022). Also, it is an extension of the problem defined by Lee *et al.* (2006) by incorporating internal operations at the CDF. This study focuses on solutions to the large-scale instances of VRPCD&MS without considering the time related constraints considered by Gnanapragasam & Daundasekera (2022).

The phases of VRPCD&MS

The VRPCD&MS model is partitioned into three main phases as follows:

Phase 1: The process of collecting products from all the suppliers.

The first vehicle routing problem is that of picking up the products from the suppliers. Initially all the inbound vehicles should start their trips from the CDF simultaneously.

Phase 2: The process of internal operations at the CDF.

This is a consolidation process occurring inside the CDF. First, the products accumulated by every inbound vehicle should be unloaded indoors at the CDF. Then the unloaded products should be moved to the outdoors of the CDF. Ultimately, the moved products should be loaded to the outbound vehicles.

Phase 3: The process of distributing products to all the customers.

The second vehicle routing problem is that of delivering the products to the customers. Eventually, all the outbound vehicles should terminate their trips at the CDF.

The assumptions

The following specific assumptions are taken into account when formulating the model for the problem VRPCD&MS:

- A single product is cross docked at a single facility and all the inbound vehicles should unload their accumulated products at the CDF.
- All the vehicles in both Phase 1 and Phase 3 should start their tours from the CDF and at the end they should return to the CDF.
- Two different homogeneous fleets of vehicles are used so that the fleet with larger capacity belongs to Phase 1 and the fleet with smaller capacity belongs to Phase 3. The capacities of the vehicles are always higher than the supply/demand by any supplier/customer so that the splitting the orders is not necessary.

The elements of the total cost due to the VRPCD&MS model are described with their respective formulas.

Mathematical formulation

The VRPCD&MS is formulated as a Mixed Integer Linear Programming (MILP) model as follows, which is based on the model proposed in the previous study (Gnanapragasam & Daundasekera, 2022). The indices, sets, parameters and variables used to develop MILP for the VRPCD&MS are described in the following Table 1:

Routing constraints

$$\sum_{j \in N} x_{hj}^k \leq 1 \quad \forall k \in V, \forall h \in O \quad \dots(1)$$

Table 1: Notations and their descriptions

Notation	Description
i, j	Indices for suppliers in Phase 1 or customers in Phase 3
h	Index for receiving or shipping doors of CDF in Phase 2
k	Index for inbound vehicles in Phase 1 or outbound vehicles in Phase 3
$S = \{S_1, S_2, \dots, S_n\}$	Set of n suppliers in Phase 1
$C = \{C_1, C_2, \dots, C_{n'}\}$	Set of n' customers in Phase 3
$N = S \cup C$	Set of $(n + n')$ suppliers and customers
$V_S = \{v_1^S, v_2^S, \dots, v_m^S\}$	Set of m inbound vehicles used in Phase 1
$V_C = \{v_1^C, v_2^C, \dots, v_{m'}^C\}$	Set of m' outbound vehicles used in Phase 3
$V = V_S \cup V_C$	Set of $(m + m')$ inbound and outbound vehicles
$O = \{o, o'\}$	Set of receiving (o) and shipping (o') doors of CDF in Phase 2
tc_{ij}	Transportation cost between destinations i and j in Phase 1 and Phase 3
q_i	Quantity at supplier i in Phase 1 or customer i in Phase 3
Q_S	Maximum capacity of inbound vehicles in Phase 1
Q_C	Maximum capacity of outbound vehicles in Phase 3
OC_S^k	Operational cost of the inbound vehicle k in Phase 1
OC_C^k	Operational cost of the outbound vehicle k in Phase 3
SC_i^k	Service cost at supplier i in Phase 1 or customer i in Phase 3 by vehicle k
SC_h^k	Service cost at receiving or shipping door h by vehicle k in Phase 2
A_c	Fixed preparation cost for loading or unloading products in all 3 phases
B_c	Variable shipping cost for loading or unloading a unit of product in all 3 phases
x_{ij}^k	1, if vehicle k travels from supplier (customer) i to supplier (customer) j 0, otherwise

$$\sum_{i \in N} x_{ih}^k \leq 1 \quad \forall k \in V, \forall h \in O \quad \dots(2) \quad x_{ii}^k = 0 \quad \forall i \in N \cup O, \forall k \in V \quad \dots(5)$$

$$\sum_{i \in N \cup O} \sum_{k \in V} x_{ij}^k = 1 \quad \forall j \in N \quad \dots(3) \quad x_{ij}^k + x_{ji}^k \leq 1 \quad \forall i, j \in N \cup O, \forall k \in V \quad \dots(6)$$

$$\sum_{j \in N \cup O} \sum_{k \in V} x_{ij}^k = 1 \quad \forall i \in N \quad \dots(4) \quad \sum_{i \in S} q_i = \sum_{i \in C} q_i \quad \dots(7)$$

$$\sum_{\substack{i \in S \\ j \in S \cup \{o\}}} q_i x_{ij}^k \leq Q_S \quad \forall k \in V_S \quad \dots(8)$$

$$\sum_{\substack{i \in C \\ j \in C \cup \{o'\}}} q_i x_{ij}^k \leq Q_C \quad \forall k \in V_C \quad \dots(9)$$

Inequalities (1) and (2), respectively, represent the closeness of the Vehicle Routing Problem; at the beginning of any route, vehicles (inbound or outbound) should depart from CDF to suppliers in Phase 1 or customers in Phase 3, and concurrently, all the vehicles (inbound and outbound) should arrive from suppliers in Phase 1 or customers in Phase 3 at the end of their routes. Furthermore, only one vehicle has to arrive at a supplier in Phase 1 or a customer in Phase 3; likewise only one vehicle has to depart from a supplier in Phase 1 or a customer in Phase 3. This is depicted in the equations (3) and (4), respectively. Moreover, equations (5) and (6), respectively, prevent the repeating routes and backward movements in routes. Also equation (7) controls the equilibrium condition of total supply in Phase 1 which must be equal to the total demand in Phase 3. In addition, the inequalities (8) and (9) depict the limitation of the capacities of inbound vehicles in Phase 1 and outbound vehicles in Phase 3, respectively.

The required output

After satisfying the aforementioned constraints, the following required outputs are obtained to calculate the total cost incurred by all three phases described in the previous section.

Required number of inbound vehicles to collect the total supply from all the suppliers:

$$m = \sum_{k \in V_p} \sum_{j \in P} x_{oj}^k$$

Required number of outbound vehicles to distribute the total demand by all the customers:

$$m' = \sum_{k \in V_d} \sum_{j \in D} x_{o'j}^k$$

Cost of transportation at the collecting and distributing processes:

$$TC_p = \sum_{k \in V} \sum_{i, j \in N \cup O} tc_{ij} x_{ij}^k$$

Service cost at the suppliers or customers:

$$SC_p = \sum_{k \in V} \sum_{\substack{i \in N \cup O \\ j \in N}} SC_j^k x_{ij}^k,$$

where

$$SC_j^k = A_c + B_c q_j x_{ij}^k$$

$$\forall i \in N \cup O, \forall j \in N, \forall k \in V$$

Service cost (cost of unloading) at the receiving doors of CDF:

$$SC_s = \sum_{k \in V_s} \sum_{i \in N} SC_o^k x_{io}^k,$$

where $SC_o^k = A_c + B_c \sum_{\substack{i \in N \\ j \in N \cup \{o\}}} q_i x_{ij}^k \quad \forall k \in V_s$

Service cost (cost of loading) at the shipping doors of CDF:

$$SC_c = \sum_{k \in V_c} \sum_{i \in N} SC_{o'}^k x_{o'i}^k, \text{ where}$$

$$SC_{o'}^k = A_c + B_c \sum_{\substack{i \in N \\ j \in N \cup \{o'\}}} q_i x_{ij}^k \quad \forall k \in V_c$$

Cost of moving shipments from the receiving doors to the shipping doors of CDF:

$$MC_T = \sum_{k \in V_s} \sum_{\substack{i \in S \\ j \in S \cup \{o\}}} q_i x_{ij}^k$$

Operational cost of required number of inbound and outbound vehicles:

$$OC_p = \sum_{k \in V_s} \sum_{i \in S} OC_S^k x_{io}^k + \sum_{k \in V_c} \sum_{j \in C} OC_C^k x_{o'j}^k$$

The single objective Mixed Integer Linear Programming (MILP) model is developed to minimize the total cost incurred through all three phases from Phase 1 to Phase 3. Therefore, the objective function of the developed MILP model can be articulated as follows:

$$\text{Minimizing Total Cost} = TC_p + SC_p + SC_s + SC_c + MC_T + OC_p$$

In the following sub-sections, the algorithms to solve the problem VRPCD&MS with the related software are described.

Branch and Bound (BB) Algorithm

The exact solutions to the fifteen small-scale instances (up to 14 suppliers and customers) are obtained using the traditional algorithm known as the Branch and Bound algorithm. Version 18 of *LINGO* optimization software is used in this regard to code the developed Mixed Integer Linear Programming model to obtain the optimal solution. Since the Branch and Bound algorithm always seeks to reach the exact optimal solution, the algorithm is more appropriate to solve small-scale instances and it performs poorly for large-scale instances. Therefore, to obtain near optimal solutions for the large-scale instances, the following meta-heuristic method based on Genetic Algorithm is employed.

Genetic Algorithm (GA)

In 1975, John Holland proposed the Genetic Algorithm (GA) which is based on the Darwinian revolution of 'survival of the fittest' (Holland, 1975), and it was popularized by Goldberg in 1989 (Goldberg, 1989). The Genetic Algorithm is a probabilistic search method which contains genetic operators such as reproduction, crossover, and mutation. In this study, to optimize the total cost, the proposed Genetic Algorithm is divided into three parts, one for each phase defined earlier. It must be emphasized that all these three parts are combined into a single programme. In detail, Part 1 is for the process of pick up the products from suppliers. The internal operations at the cross-dock are considered in Part 2. Part 3 is for the process of delivering the collected products to the customers. There are many combinations of genetic operators used to design a Genetic Algorithm. However, in this study, the Taguchi parameter design method is used to select the best combination of the genetic operators. Accordingly, in this study, the following selection procedures are adapted to design the Genetic Algorithm to solve the VRPCD&MS:

Solution representation

Each solution consists of a set of numbers representing permutation of suppliers and customers in Part 1 and part 3, respectively.

Generation of initial solution

The Genetic Algorithm starts with the initial population

based on the random generation method. Since this is one of the best methods to prevent the diversity of the population, the initial population is generated randomly in this proposed Genetic Algorithm. In this case, random permutations are generated based on the number of suppliers or customers as a giant chromosome in the initial population.

Selection procedure

The parent chromosomes are randomly chosen from the initial population based on the tournament selection method to create offspring to the next generation of the population. The minimum value of the chromosome is selected among four randomly chosen chromosomes from the previous generation of the population. To avoid accounting for bytes and without loss of generality, the second power of two is selected as the size of the tournament.

Crossover operation

To produce the offspring for the next generation, the order crossover method is applied by randomly selecting two parent chromosomes from the current generation of the population. An offspring is produced by copying the sub-route from the first chromosome into the corresponding positions. Then unfixed positions of the offspring are filled from the second chromosome (the chromosome modified by deleting the genes which are already in the first chromosome) from left to right according to the order of the sequence.

Mutation operation

To find a nearest solution in the search space, the swap mutation is used by selecting a random chromosome from the current generation of the population. The genes of two randomly chosen positions in the chromosome exchange their positions in the chromosome.

Elitism

A set of better solutions in the current generation of the population is transformed to the next generation as it is, so that the better solutions are always preserved.

Operations at the Part 1 and Part 3

Figure 1 given below explains Part 1 and Part 2 in the proposed Genetic Algorithm for the pickup and delivery processes of the VRPCD&MS model:

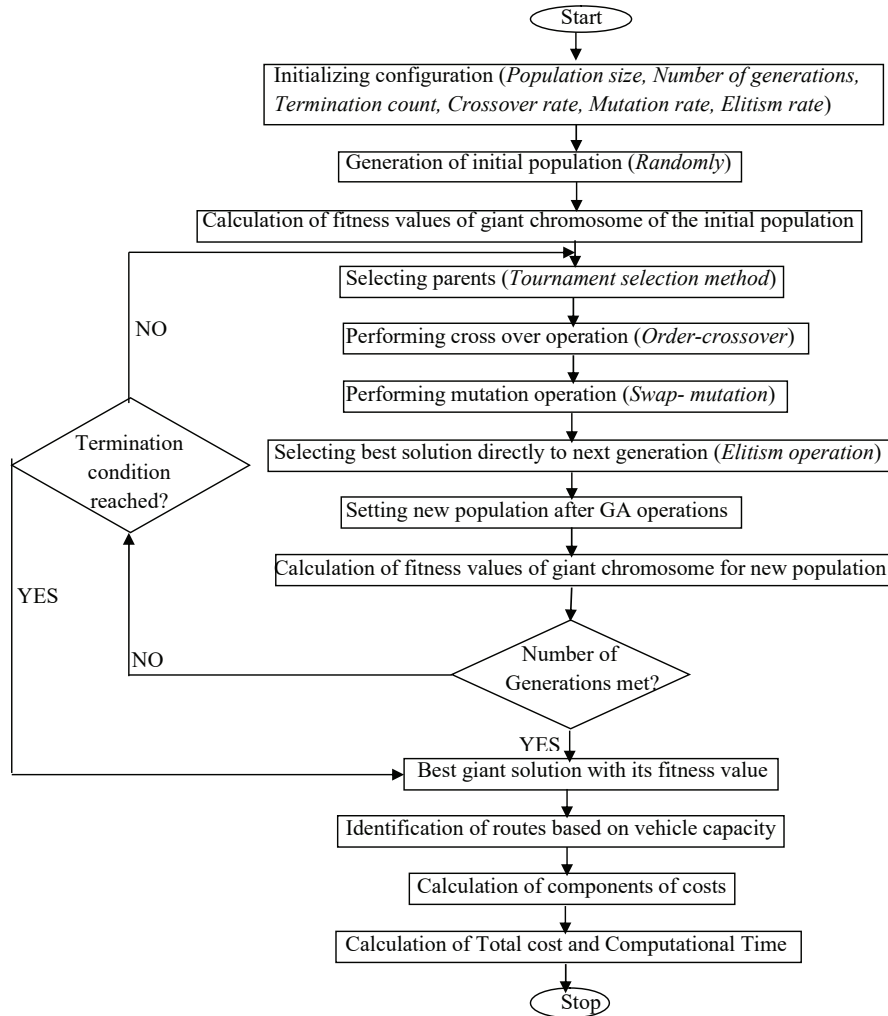


Figure 1: Flow chart for the part 1 and part 3 of the proposed Genetic Algorithm

It should be emphasized that the very first instruction is applicable only for Part 1. In this, the estimated values of the parameters of the Genetic Algorithm are to be included. Consequently, those parameters are tuned by the Taguchi parameter design method and are reported in Table 3.

Moreover, the very last instruction is applicable only for Part 3 and those calculations are performed based on the formulas described earlier. Furthermore, all the operations of the middle portion (except the first and the last instructions) given in the Flow chart shown in Figure 1 are common to both Part 1 and Part 3 of the proposed Genetic Algorithm. Besides, it can be observed from

Figure 1 that there are two termination criteria to reach the best solution. One criterion is after the execution of ‘total number of generations’ of new populations. The second criterion is meeting the ‘termination count’ defined as no improvement in the current solution after a predetermined number of iterations, and the current solution would then be considered as the best near optimal solution to the problem.

Internal operations at the cross-docking facility

The operations inside the CDF of VRPCD&MS are depicted in the Figure 2 below:

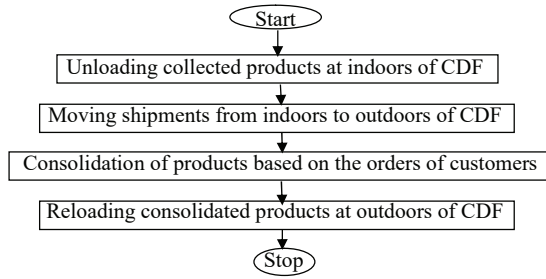


Figure 2: Flow chart for the part 2 of the proposed Genetic Algorithm

The aforementioned Genetic Algorithm is coded in *MATLAB* environment on an Intel Core i5 personal computer with 2.30 GHz CPU and 4 GB RAM. The data are extracted from the benchmark instances (Wen *et al.*, 2009) in the literature of VRPCD, in which the small-scale instances are used to test the compatibility of the developed VRPCD&MS model and to compare

the optimal solutions obtained by the Branch and Bound Algorithm and Genetic Algorithm.

RESULTS AND DISCUSSION

Benchmark instances given by Wen *et al.* (2009) are used to compare the solutions to small-scale instances as well as to large-scale instances of VRPCD&MS.

Parameter values

Table 2 below provides the values, which are chosen arbitrarily, for the problem parameters of the VRPCD&MS: Table 3 below provides the values of the parameters of the Genetic Algorithm for the meta-heuristic method, which are further tuned by the Taguchi method: The same estimated values are fixed in all the cases in the small-scale and large-scale instances considered in this study.

Table 2: Values of the problem parameters of the VRPCD&MS model

Parameter	Value	Parameter	Value
Fixed preparation cost (A_c)	10 cost units	Variable unit shipping cost (B_c)	1 cost unit
Operational cost of inbound vehicle (OC_s^k)	150 cost units	Operational cost of outbound vehicle (OC_c^k)	100 cost units
Inbound vehicle capacity (Q_s)	80 units	Outbound vehicle capacity (Q_c)	50 units

Table 3: Values of the parameters of the Genetic Algorithm

Parameter	Value	Parameter	Value
Population size	50	Mutation rate	0.3
Termination count	50	Crossover rate	0.7
Number of generations	100	Elitism rate	0.1

Results for small-scale instances of VRPCD&MS

The results of the small-scale instances of the VRPCD&MS model obtained from both the Branch and Bound (BB) algorithm and the Genetic Algorithm (GA) are compared in Table 4 below. The instances are described here in terms of number of suppliers and

customers in Phase 1 and Phase 3 respectively. In Table 4, the optimal solutions found for fifteen instances by the method of the Branch and Bound algorithm are compared with the near optimal solutions found by the Genetic Algorithm. In the Genetic Algorithm, the best solution for 10 repetitions of the same instance and the average solution of those 10 solutions are included in Table 4. The average time is the average computational time calculated by executing repeatedly the same instance 10 times under the same conditions. Since the other output values such as the required number of vehicles in Phase 1 and Phase 3 are the same in both methods, it is mainly the solutions in terms of the total cost, and the time required to obtain those results, that are compared in Table 4 below. The GAP value in Table 4 is calculated as follows:

$$GAP = \frac{(Average\ near\ Optimal\ Solution\ by\ GA\ method - Optimal\ Solution\ by\ BB\ method)}{Optimal\ Solution\ by\ BB\ method} \times 100$$

Table 4: Comparison of results between BB algorithm and GA for small-scale instances

Problem no.	Instance		BB algorithm			GA		GAP in %
	No. of suppliers	No. of customers	Optimal solution	Average time (s)	Best solution	Average solution	Average time (s)	
01	02	02	786.76	0.11	786.76	786.76	3.50	0.00
02	02	03	831.76	0.12	831.76	831.76	3.47	0.00
03	02	04	788.28	0.14	788.28	788.28	3.34	0.00
04	03	03	922.38	0.13	922.38	922.38	3.35	0.00
05	03	04	853.90	0.15	853.90	853.90	3.27	0.00
06	03	05	926.02	0.25	926.02	926.02	3.51	0.00
07	04	04	817.88	0.23	817.88	817.87	3.50	0.00
08	04	05	840.00	0.36	840.00	839.99	3.55	0.00
09	04	06	1052.96	2.17	1104.70	1104.70	3.40	4.91
10	05	05	1116.74	1.69	1146.70	1150.18	4.51	2.99
11	05	06	1080.80	2.48	1105.50	1107.78	4.00	2.50
12	05	07	1179.18	4.24	1191.20	1198.94	4.90	1.68
13	06	06	1091.87	3.90	1106.90	1110.20	4.84	1.68
14	06	07	959.92	10.31	1005.06	1005.73	5.47	4.77
15	06	08	1173.37	93.38	1228.40	1228.82	6.40	4.73

When comparing the solutions in terms of total cost exhibited in Table 4, it can be seen that, if the size of the problem is less than 10 (the first 8 instances), then the Genetic Algorithm also gives the same exact optimal solution as is given by the Branch and Bound algorithm. But, it can be observed from problems 10 to 15, when the size of the problem increases, in each and every instance, only a near optimal solution is reached by the Genetic Algorithm with a GAP of nearly 5%.

In the Branch and Bound algorithm, in spite of the average computational time to reach the optimum solution for the first 13 instances being comparatively less, it gradually increases as the size of the problem increases. Also it can be observed that, for the first 8 instances, the averages of the computational time taken by the Genetic Algorithm method are close to the computational time taken by the Branch and Bound algorithm. This occurs mainly as the values of the two parameters (number of generations = 100 and termination count = 50) of the Genetic Algorithm (described in Table 3) are fixed in the *MATLAB* program. However, for the first 8 small-scale instances, if those two particular parameter values are reduced, the same solutions can be obtained with less computational time than that specified in Table 4. Furthermore, it can be seen from Table 4 that in the last two instances, a near optimal solution can be reached in

less computational time by the Genetic Algorithm, than by the Branch and Bound algorithm. Therefore, with respect to the computational time, it can be concluded that to solve the VRPCD&MS model, the proposed Genetic Algorithm is more appropriate.

It can be observed from Table 4 that the average computational time rapidly increases to reach the optimal solution with the increase of the scale of the problem. Also, the study by Gnanapragasam & Daundasekera (2022) showed that the convergence time increased exponentially to reach the optimal solutions for the moderately large-scale instances. It recommends that, for the large-scale instances, a meta-heuristic method is more appropriate. Therefore, in this study, the Genetic Algorithm is used for further analysis for the large-scale instances of VRPCD&MS model and the corresponding results have been reported.

Results obtained by the GA for the large-scale instances of VRPCD&MS

Table 5 shows the number of suppliers and customers in each instance, which is considered as the size of the problem. The required fleets of vehicles in Phase 1 and Phase 3 are indicated as inbound and outbound vehicles respectively. In each instance, the average of the best

optimal solutions obtained in 10 replicates and finally the average computational time of those 10 replicates are exhibited in Table 5. From Table 5, the required number of inbound vehicles in Phase 1 and outbound vehicles in Phase 3 can be seen in each of the fifteen problems. It can be concluded that from the average computational

time given in Table 5, a near optimal solution to each problem can be reached in less than 6 seconds even when the size of the problem is 200 in both phases. Since it is relatively a better in performance, this model can be used to schedule the routes of the vehicles at the last moment prior to the start of the time horizon.

Table 5: Results by the Genetic Algorithm for the large-scale instances of VRPCD&MS

Problem no.	No. of suppliers	No. of customers	Inbound vehicles	Outbound vehicles	Best solution	Average solution	Average time (s)
01	10	10	02	03	2741.10	2742.31	4.70
02	10	15	03	04	3778.40	3806.87	4.88
03	20	20	04	06	5569.80	5629.09	4.80
04	20	25	05	07	6700.30	6782.84	5.21
05	30	30	06	08	8278.30	8377.96	5.18
06	30	35	07	10	9383.10	9663.48	5.06
07	40	40	08	11	10987.00	11198.00	5.08
08	40	45	08	12	12319.00	12454.60	5.17
09	50	50	09	14	14146.00	14308.20	5.40
10	50	55	10	15	15100.00	15380.20	5.51
11	60	60	11	17	16392.00	16535.90	5.31
12	70	70	13	19	19285.00	19483.00	5.71
13	80	80	15	22	22362.00	22595.90	5.88
14	90	90	16	25	25605.00	25872.60	5.91
15	100	100	18	28	28443.00	28900.30	5.95

Table 6: Best solution to the problem with 10-suppliers and 10-customers

Phase	Routes	Elements of total cost					Route-wise cost	
		TC_p	SC_p	SC_s	MC_T	SC_c		OC_p
1	CDF - S ₂ - S ₄ - S ₃ - S ₁ - S ₉ - CDF	211.94	99	59	49	N/A	150	568.94
	CDF - S ₁₀ - S ₈ - S ₅ - S ₆ - S ₇ - CDF	123.64	101	61	51	N/A	150	486.64
3	CDF - C ₃ - C ₅ - C ₁ - C ₄ - CDF	337.12	79	N/A	N/A	49	100	565.12
	CDF - C ₇ - C ₂ - C ₆ - C ₁₀ - CDF	417.40	73	N/A	N/A	43	100	633.40
	CDF - C ₈ - C ₉ - CDF	301.02	48	N/A	N/A	38	100	487.02

Table 6 above elaborates the best solution to the first problem with 10 suppliers in Phase 1 and 10 customers in Phase 3. Here it should be noted that the calculation of the total cost is explained in a previous section.

As per the best solution reported in Table 6 to the problem with 10 suppliers and 10 customers (the first

problem in Table 5), it can be observed that the individual route-wise results in both phases can be obtained from the proposed Genetic Algorithm in an average of 4.7 seconds.

It can be noticed that the costs due to the highlighted elements (SC_s , MC_T and SC_c) in Table 6 are actually

incurred in Phase 2 as they are due to the three internal operations considered in this study. The total cost can be obtained by summing up all the elements of the total cost reported in Table 6 as 2741.12 units. Since there are two routes in Phase 1 and three routes in Phase 3, two inbound and three outbound vehicles are required to complete the task in the specified cost units which is same as the results obtained to the first problem given in Table 5.

CONCLUSIONS AND RECOMMENDATIONS

In this study, the Genetic Algorithm, which is known as a meta-heuristic method, is proposed to solve the VRPCD&MS model. In the developed Mixed Integer Linear Programming model, the cost of internal operations at the Cross Docking Facility (CDF) is added to the total cost. The unloading of the collected products from all the suppliers at the indoors of CDF, moving unloaded shipments from indoors to outdoors at the CDF, and loading the moved shipments onto the outbound vehicles to deliver the shipments to all the customers are the internal operations considered at the CDF. In the case of small-scale instances, the near optimal solutions obtained from the Genetic Algorithm are compared with the exact solutions obtained by the Branch and Bound algorithm. The results obtained for small-scale instances reveal that the GAP between exact solution and the near optimal solution is approximately 5%. Therefore, it can be concluded that, the proposed Genetic Algorithm is a better alternative to solve the VRPCD&MS model. It is observed that the computational time to reach a near optimal solution to a large-scale instance is under six seconds. Hence, it can be recommended that the proposed Genetic Algorithm can be used to schedule the routes of the vehicles at the last moment prior to the start of the time horizon. Since the Genetic Algorithm focuses only on the cost, it is recommended in future studies to take into account the time related constraints including arrival time of vehicles and time windows of each supplier/customer. This study further recommends that to incorporate additional constraints to the model, such as available vehicles to transport products and limitation of temporary storage capacity and other resources at the CDF, to generalize the study under investigation. Moreover, other existing heuristic or meta-heuristic methods can be exploited to investigate the adaptability in solving the VRPCD&MS model.

REFERENCES

Alinaghian M., Rezaei Kalantari M., Bozorgi-Amiri A. & Golghamat Raad N. (2016). A novel mathematical model

for cross dock open-close vehicle routing problem with splitting. *International Journal of Mathematical Sciences and Computing* **3**(2): 21–31.

DOI: <https://doi.org/10.5815/ijmsc.2016.03.02>

Apte U.M. & Viswanathan S. (2000). Effective cross docking for improving distribution efficiencies. *International Journal of Logistics Research and Applications: A Leading Journal of Supply Chain Management* **3**(3): 291–302.

DOI: <http://dx.doi.org/10.1080/713682769>

Baniamerian A., Bashiri M. & Zabihi F. (2018a). A modified variable neighbourhood search hybridized with genetic algorithm for vehicle routing problems with cross-docking. *Electronic Notes in Discrete Mathematics* **66**: 143–150.

DOI: <https://doi.org/10.1016/j.endm.2018.03.019>

Baniamerian A., Bashiri M. & Zabihi F. (2018b). Two phase genetic algorithm for vehicle routing and scheduling problem with cross-docking and time windows considering customer satisfaction. *Journal of Industrial Engineering International* **14**(1): 15–30.

DOI: <https://doi.org/10.1007/s40092-017-0203-0>

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

DOI: <https://doi.org/10.1016/j.sbspro.2016.11.010>

Buakum D. & Wisittipanich W. (2019). A literature review and further research direction in cross-docking. *Proceedings of the International Conference on Industrial Engineering and Operations Management MAR* 471–481.

Dantzig G. & Ramser J. (1959). The truck dispatching problem. *Management Science* **6**(1): 80–91.

DOI: <https://doi.org/10.1287/mnsc.6.1.80>

Dondo R. (2013). A branch-and-price method for the vehicle routing problem with cross-docking and time windows. *Iberoamerican Journal of Industrial Engineering* **5**(10): 16–25.

DOI: <https://doi.org/10.13084/2175-8018.v05n10a02>

Fakhrzad M.B. & Sadri Esfahani A. (2014). Modeling the time windows vehicle routing problem in cross-docking strategy using two meta-heuristic algorithms. *International Journal of Engineering, Transactions A: Basics* **27**(7): 1113–1126.

DOI: <https://doi.org/10.5829/idosi.ije.2014.27.07a.13>

Gnanapragasam S.R. & Daundasekera W.B. (2022). Optimal solution to the capacitated vehicle routing problem with moving shipment at the cross-docking terminal. *International Journal of Mathematical Sciences and Computing* **8**(4): 60–71.

DOI: <https://doi.org/10.5815/ijmsc.2022.04.06>

Goldberg D.E. (1989). *Genetic Algorithms in Search Optimization and Machine Learning*. Addison -Wesley, Reading, MA, USA.

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

DOI: <https://doi.org/10.1016/j.cor.2017.03.004>

Gunawan A., Widjaja A.T., Vansteenwegen P. & Yu V.F. (2020a). A matheuristic algorithm for solving the vehicle routing problem with cross-docking. *Proceedings of the*

- 14th Learning and Intelligent Optimization Conference, 12096 LNCS (Lion), pp. 9–15.
DOI: https://doi.org/10.1007/978-3-030-53552-0_2
- Gunawan A., Widjaja A.T., Vansteenwegen P. & Yu V.F. (2020b). Adaptive large Neighbourhood search for vehicle routing problem with cross-docking. *2020 IEEE Congress on Evolutionary Computation (CEC)*, 19–24 July.
DOI: <https://doi.org/10.1109/CEC48606.2020.9185514>
- 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**(2010): 1360–1365.
DOI: <https://doi.org/10.1016/j.sbspro.2012.09.232>
- Hasani Goodarzi A., Diabat E., Jabbarzadeh A. & Paquet M. (2022). An M/M/c queue model for vehicle routing problem in multi-door cross-docking environments. *Computers and Operations Research* **138**: 105513.
DOI: <https://doi.org/10.1016/j.cor.2021.105513>
- Hasani Goodarzi A., Nahavandi N. & Zegordi S.H. (2018). A multi-objective imperialist competitive algorithm for vehicle routing problem in cross-docking networks with time windows. *Journal of Industrial and Systems Engineering* **11**(1): 1–23.
- Holland J.H. (1975). *Adaptation in Natural and Artificial Systems*. The University of Michigan Press, Ann Arbor, MI, USA.
- Ieva M.-K., Nihal S., Figen Y., Shahryar G. & Renata C. (2022). Optimizing multi cross-docking systems with a multi-objective green location routing problem considering carbon emission and energy consumption. *Energies* **15**(1530).
DOI: <https://doi.org/10.3390/en15041530>
- Kargari Esfand Abad H., Vahdani B., Sharifi M. & Etebari F. (2018). A bi-objective model for pickup and delivery pollution-routing problem with integration and consolidation shipments in cross-docking system. *Journal of Cleaner Production* **193**: 784–801.
DOI: <https://doi.org/10.1016/j.jclepro.2018.05.046>
- Larioui S. & Reghioui M. (2017). Resolution of the vehicle routing problem with cross-docking. *Proceedings of the International Conference on Industrial Engineering and Operations Management*, April, pp. 1184–1194.
- Lee Y.H., Jung J.W. & Lee K.M. (2006). Vehicle routing scheduling for cross-docking in the supply chain. *Computers and Industrial Engineering* **51**(2): 247–256.
DOI: <https://doi.org/10.1016/j.cie.2006.02.006>
- 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.
DOI: <https://doi.org/10.1016/j.eswa.2010.03.035>
- Moghadam S.S., Ghomi S.M.T.F. & Karimi B. (2014). Vehicle routing scheduling problem with cross docking and split deliveries. *Computers and Chemical Engineering* **69**: 98–107.
DOI: <https://doi.org/10.1016/j.compchemeng.2014.06.015>
- Morais V.W.C., 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.
DOI: <https://doi.org/10.1016/j.eswa.2014.06.010>
- Nikolopoulou A.I., Repoussis P.P., Tarantilis C.D. & Zachariadis E.E. (2016). Adaptive memory programming for the many-to-many vehicle routing problem with cross-docking. *International Journal of Operational Research* **19**(1): 1–38.
DOI: <https://doi.org/10.1007/s12351-016-0278-1>
- Santos F.A., Mateus G.R. & Salles Da Cunha A. (2011a). A branch-and-price algorithm for a vehicle routing problem with cross-docking. *Electronic Notes in Discrete Mathematics* **37**(C): 249–254.
DOI: <https://doi.org/10.1016/j.endm.2011.05.043>
- Santos F.A., Mateus G.R. & Salles Da Cunha A. (2011b). 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)*, 6701 LNCS, pp. 412–425.
DOI: https://doi.org/10.1007/978-3-642-21527-8_47
- Tarantilis C.D. (2013). Adaptive multi-restart Tabu Search algorithm for the vehicle routing problem with cross-docking. *Optimization Letters* **7**(7): 1583–1596.
DOI: <https://doi.org/10.1007/s11590-012-0558-5>
- Vahdani B., Reza T.M., Zandieh M. & Razmi J. (2012). Vehicle routing scheduling using an enhanced hybrid optimization approach. *Journal of Intelligent Manufacturing* **23**(3): 759–774.
DOI: <https://doi.org/10.1007/s10845-010-0427-y>
- Vahdani B. & Zandieh M. (2010). Scheduling trucks in cross-docking systems: Robust meta-heuristics. *Computers and Industrial Engineering* **58**(1): 12–24.
DOI: <https://doi.org/10.1016/j.cie.2009.06.006>
- Van Belle J., Valckenaers P. & Cattrysse D. (2012). Cross-docking: State of the art. *Omega* **40**(6): 827–846.
DOI: <https://doi.org/10.1016/j.omega.2012.01.005>
- 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.
DOI: <https://doi.org/10.1057/jors.2008.108>
- Yin P.Y. & Chuang Y.L. (2016). Adaptive memory artificial bee colony algorithm for green vehicle routing with cross-docking. *Applied Mathematical Modelling* **40**(21–22): 9302–9315.
DOI: <https://doi.org/10.1016/j.apm.2016.06.013>
- Yu V.F., Jewpanya P. & Redi A.A.N.P. (2014). A simulated annealing heuristic for the vehicle routing problem with cross-docking. In: *Logistics Operations, Supply Chain Management and Sustainability*, pp. 15–30. Springer.
DOI: <https://doi.org/10.1007/978-3-319-07287-6>
- Yu V.F., Jewpanya P. & Redi A.A.N.P. (2016). Open vehicle routing problem with cross-docking. *Computers and Industrial Engineering* **94**: 6–17.
DOI: <https://doi.org/10.1016/j.cie.2016.01.018>