International Journal of Supply and Operations Management

IJSOM

November 2015, Volume 2, Issue 3 pp. 833-855
ISSN-Print: 2383-1359
ISSN-Online: 2383-2525
www.ijsom.com

A memetic algorithm forthe vehicle routing problema with cross docking

Sanae Larioui [a], Mohamed Reghioui [a], Abdellah Elfallahi [a] and Kamal Eddine Elkadiri [a]

[a] Ensate, University of Abdelmalek Essaadi, Mhannech II, Tetouan, Morocco

Abstract

In this paper we address the VRPCD, in which a set of homogeneous vehicles are used to transport products from the suppliers to customers via a cross-dock. The products can be consolidated at the cross-dock but cannot be stored for very long as the cross-dock does not have long-term inventory-holding capabilities. The objective of the VRPCD is to minimize the total traveled distance while respecting time window constraints of suppliers and customers and a time horizon for the whole transportation operation. Following the literature on vehicle routing problems with cross-docking, it seems that few studies consider that customer will receive its requests from several suppliers; therefore, the present study is an attempt to investigate this case. A heuristic and a memetic algorithm are used to solve the problem. The proposed algorithms are implemented and tested on data sets involving up to 200 nodes (customers and suppliers). The first results show that the memetic algorithm can produce high quality solutions. It is able to find the optimal solution for small instances, for the large ones, it is very powerful comparing with the best insertion heuristic, the gap achieved 30%.

Keywords: Cross-docking; Vehicle Routing Problem; Pickup and Delivery; Memetic algorithm.

1. Introduction

The problem considered in this paper involves a set of known customer orders or requests, each one characterized by the cargo size, the pickup point and the place where it has to be delivered. These requests are picked up by a fleet of homogeneous vehicles, consolidated at the cross-dock, and immediately delivered to customers by the same set of vehicles, without intermediate storage. During the consolidation, goods are unloaded from the inbound vehicles and reloaded on outbound vehicles.

* Corresponding email address: sanae_lar@hotmail.com

In other words, a vehicle starting from the cross-dock first collects several requests at their pickup points, drives back to the cross dock, and unloads some but not necessarily all orders. Some loads may remain in the truck if the same vehicle will transport them to their destinations.

Then, the truck moves to the assigned shipping door, loads some additional requests and goes out again to serve the delivery locations. After completing their tours, delivery vehicles return to the cross dock. The objective is to determine the best pickup and delivery routes as well as the arrival times of pickup/delivery vehicles at the cross-dock so that all nodes are visited within their time windows at minimum total transportation cost, including variable and fixed costs.

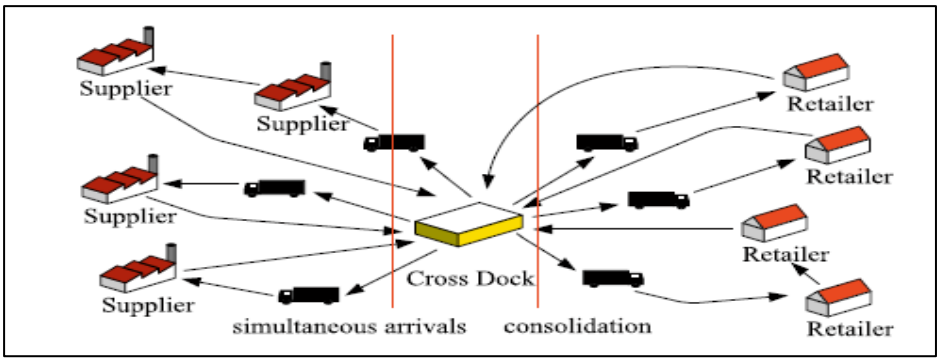

Figure1.The concept of cross docking in vehicle routing.

Note that in the case without consolidation, the solution of this problem can be found by solving two vehicle routing problems with time windows (one for pickup and one for delivery). But taking into account the consolidation, the pickup and delivery routes are not independent. Trying to minimize the distance of the pickup and delivery routes separately does not guarantee the global optimization; the exchanges of orders at the cross-dock also have to be taken into account (Wen et al.2008), in fact, the problem involves not only vehicle route design, but also a consolidation decision at the cross-dock.

We note that our problem has a difference with that of Wen et al. (2008), we have added an additional complexity which consists of allowing a customer to order from more than one supplier because we believe that a customer who requests one product from one supplier is a case a little far from what happens in reality. In figure 2 an example of the previously studied demand of customer is given.

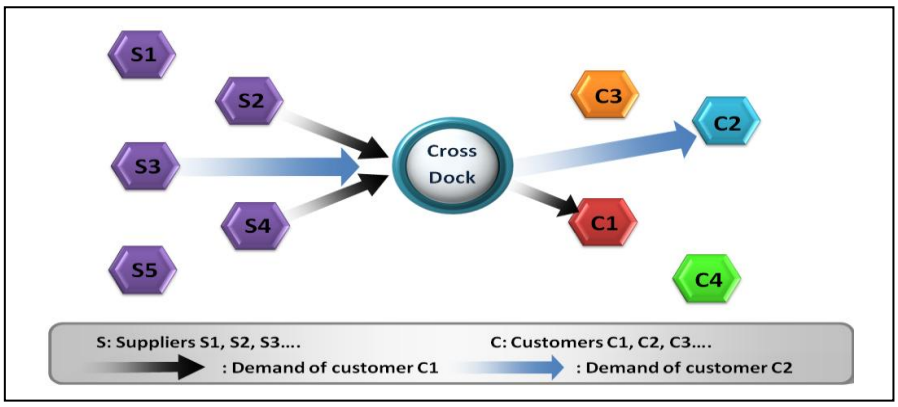

Figure2. Example of subdivision of client requests

In Figure 2 customer C1 have two types of products ordered from two suppliers S2 and S4.

Figure 3 illustrates the pickup and delivery routes for three vehicles; each vehicle starts and ends their routes at the cross-dock.

For example, the first vehicle makes pickups at nodes 1 and 2 and delivers to nodes 1', 2' and 5'. A delivery vehicle cannot leave the cross dock without loading all requests for all pickup vehicles. For instance, if one client (1') has two suppliers 1 and 3, then, the vehicle that will deliver 1' must await the pickup vehicles carrying his requests.

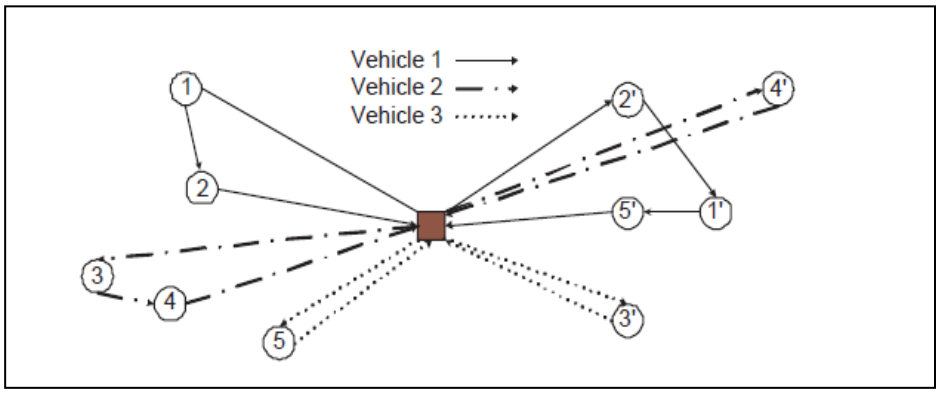

Figure 3. Example of pickup and delivery routes.

Since our VRPCD is based on the VRPTW, it is necessary to address this problem which is a generalization of the well-known capacitated vehicle routing problem in which the service of each customer must begin within a specified time window. It is defined on a complete undirected graph G = (V,E) with a node set V = {0, 1, 2, ..., n} and an edge set E. Node 0 represents a depot where a fleet of identical vehicles of capacity W is located. The n other nodes correspond to the customers. Each customer i has a demand q_i and a time window $[e_i, l_i]$, where e_i and l_i are respectively the earliest and the latest service time. Arriving at i earlier than e_i is allowed but induces a waiting time a_i, while arriving later than l_i leads o infeasibility. A traversal cost (distance) $d_{ij} = d_{ji}$ and a traversal time $t_{ij} = t_{ji}$ are associated with each edge [i, j]. The objective is to build a set of vehicle trips of minimum total cost, such that each trip starts and ends at the cross dock and services a subset of customers within their time windows. Each customer must be visited by a single trip, i.e., split deliveries are not allowed (Labadi et al.2008).

As in the Vehicle Routing Problem with Time Windows (VRPTW), each node must be served by exactly one vehicle within its time window, that is why we consider two VRPTWs (one for the pickup problem and one for delivery), the accumulated load of each route must not exceed the vehicle capacity, and the time horizon for the whole transportation operation must be respected. At the cross-dock, the unloading must be completed for each vehicle before reloading starts. Each vehicle can start unloading immediately after it arrives at the cross-dock from its pickup route. The duration of the unloading consists of a fixed time for preparation, and the time needed for unloading products, which is equal to the handling time for one pallet multiplied by the number of pallets.

The considered network consists of a set of collecting nodes (Suppliers) P = {1,..., n} and a set of delivery nodes (clients) D = {1,. ..., m}. Each demand D_{ij} is identified by the pair (i, j) where i is the pickup node and j is the delivery one. The total demand of a customer *i* is the sum of his partial demands: The cross dock is represented by node 0; which is the departure and the return point of each route of pick up or delivery. Also, the cross dock has a time window which indicates the hours of opening. In addition, the suppliers and the customers must be visited in a specific time window. The objective is to minimize the total cost of transportation respecting the constraints listed above. The complexity of the problem lies in the synchronization required at the cross dock between pickup and delivery routes.

In that follows, the literature review is given in Section 2. Section 3 treats the mathematical model. Section 4 presents the memetic algorithm proposed to solve the problem. The instances used and the obtained results are discussed in section 5. Finally, section 6 presents the conclusion of the work.

2. Review of literature

In previous studies, several studies were conducted in different areas of supply chain management (SCM) aiming to improve its efficiency and effectiveness. A novel distribution strategy was proposed for helping in the routing of goods without inventory: the Cross-Docking (CD) warehouse (Apte et Viswanthan.2000). In fact, in traditional warehouses, products are received and stored, generally for a long time. This storage needs various operations witch increase the cost of goods without added value. The reduction of this kind of movements ensures an efficient gain margin. On the other hand, in the cross-docking system, fewer stocks are handled in a temporary storage. In fact, contrary to the traditional warehouse, cross docking eliminates the inventory holding function, while still allowing consolidation (Wen et al.2008).

A number of studies have been conducted to investigate on cross docking. For example, Sung and Song (2003) have discussed the problem of deciding whether to open a cross dock or not, and the problem of assigning vehicles for transportation from a supplier to a single destination via one of the open cross-docks. They have proposed a tabu search algorithm for the transportation problem. Another work of the same authors performed on location of cross-docks. In their work, goods have to be transported from supply to customers essentially via a cross-dock (direct shipments are not allowed).
The cross-dock can be chosen from a set of possible cross-dock locations, each with an associated fixed cost. The objective is to find which cross-docks should be used and how many vehicles are needed on each link in order to minimize the total cost. This total cost consists of the fixed costs of the used cross-docks and the transportation costs. The authors present an integer programming model of the problem. Their model is similar to the model of Musa et al. (2010). Since the problem is NP-hard, a tabu search-based algorithm is proposed to solve the problem. The solutions determine how the goods flow through the network. Based on this flow, the number of vehicles can be obtained by solving a sub problem. Some computational experiments were

performed on generated test instances and indicate that the proposed algorithm finds good feasible solutions within a reasonable time.

Lee, Jung and Lee (2006) studied a problem which consists of a single cross-dock, multiple suppliers and multiple customers. The authors proposed a mixed integer programming formulation and a tabu search algorithm to assign tours to a set of vehicles at the cross dock so that suppliers and customers are visited within their time windows. They assume that all vehicles should arrive simultaneously at the cross-dock from their pickup routes.
Cross docking showed that it is one of the strategies that help to reduce the storage cost, delivery lead times, inventory holding and transportation costs. It is defined as the consolidation of products from incoming shipments so that they can be easily sorted at a distribution center (cross dock) for outgoing shipments. However, the cross-docking strategy cannot achieve its goal without a good organization of the distribution process. Its efficiency depends mainly on the quality of the vehicle routing solution. That is why, a variation of the well-known Vehicle Routing Problem (VRP) is present in the literature; this process is named Vehicle Routing Problem with Cross-Docking (VRPCD) and arises in many real cases.

The vehicle routing problem has been treated efficiently by several researchers during the last decade. The VRPTW (vehicle routing problem with time windows) can be especially helpful in treating the VRPCD, since the pickup and delivery routes must be synchronized at the cross-dock and time windows of suppliers and customers should be respected.

The VRPCD was studied first by Lee et al. (2006). The authors considered that split deliveries is not allowed and all pickup vehicles should arrive at the cross-dock simultaneously to prevent waiting times for the outbound trucks. This constraint is too restrictive and cannot be considered generally, except in some fewer cases. The authors present an integer programming model of the problem and a tabu search algorithm to solve this problem. They split the main problem into two sub-problems; the first one represents the pickup process while the second one corresponds to delivery. The second routing problem is considered only when the first one is finished and the complete process has to be finished within a certain planning horizon.

Wen et al. (2008) addressed the VRPCD similarly .They suppose that a time window is defined for all suppliers and customers and orders are not splittable. The authors present a mixed integer programming formulation of the problem in which the objective is to minimize the total travel time of all vehicles. They also propose a tabu search embedded within an adaptive memory procedure. This method was tested on data involving up to 200 supplier–customer pairs and obtained solutions less than 1% away from the optimum within short computing times (less than 5 s) for small problem instances. For larger instances, the gap with a lower bound was less than 5% while the computation time stays below 5 min.

Santos et al. (2011) presented in their study a novel column generation algorithm to solve the VRPCD. As wen et al.(2008), the authors considered couples supplier customer, the difference is that in their problem, costs to load/unload trucks at CD are introduced and time windows constraints at suppliers, customers and CD, on the other hand, have been neglected (see also the work of Sung and Song(2003)). The resulting column generation sub problem of their algorithm

implies a huge computational complexity and for this reason they implemented a branch-and-cut algorithm to obtain optimal solutions and a heuristic algorithm based on GRASP metaheuristic to approximate such solutions faster. The table below summarizes the characteristics of VRPCD of few authors who have dealt with this problem.

Table1. Summary of the characteristics of the problems studied

Author	Multiple cross dock	Split deliveries	Time Window	Heterogenous vehicles
Sung and song(2003	Yes	No	No	Yes
Chen et al.(2006)	Yes	No	Yes	No
Liao et al.(2010)	No	No	No	No
Musa et al.(2010)	Yes	No	No	No
Soltani and Sajdadi(2010)	No	No	No	No

In this paper, a model integrating cross-docking with vehicle routing is treated. The version considered here is different from those proposed in literature. We consider a new version where a customer may order different products from several suppliers which is more realistic. The objective is to determine the optimal vehicle routing schedule in order to minimize the total cost. Since this problem is known to be NP-hard, a memetic algorithm and a heuristic are developed to solve it.

3. Mathematical formulation

In this section, an integer linear programming formulation for the VRPCD is proposed; the objective is to minimize the total cost of transportation.

We denote the set of pickup nodes (Suppliers) with P = {1, ..., n) and the set of delivery nodes (customers) with D = {1, ... m}. Each request is identified by the pair (i, j) where i is the pickup node and j is the delivery one. The cross-dock is represented by four nodes and noted by the set O = (o_1, o_2, o_3, o_4), the first two nodes represent the beginning and end points of the pickup.

The last ones are for delivery.

N = is defined PUOUD, all nodes.

The set E indicates all the arcs of the network. They consist of arcs:

{(i,j): i,j ∈ P U { o_1; o_2},i≠ j}and the arcs {(i,j): i,j ∈ D U { o_3; o_4}, i≠ j}

K is the set of vehicles

The parameters are:

C_{ij} = the travel time between node i and node j ((i, j) ∈ E);

$[a_i, b_i]$ = the time window of node i (i ∈ N);

D_{ij} = the amount of demand requested by the customer j to the supplier i (i ∈ P) (j ∈ D);

Q = the vehicle capacity;

A = the fixed time for unloading and reloading at the cross-dock;

B = the handling time of one pallet.

The variables are:

$$x_{ij}^k = \begin{cases} 1 \;\; if\ vehicle\ k\ travels\ from\ node\ i\ to\ node\ j\ ((i;\ j)\ \in E;\ k\ \in\ K) \\ 0 \qquad otherwise \end{cases}$$

$$u_{ij}^k = \begin{cases} 1\ if\ vehicle\ k\ unloads\ request\ (i,j)\ at\ the\ crossdock\ (i\ \in P;\ j\ \in D, k\ \in\ K) \\ 0 \qquad otherwise \end{cases}$$

$$r_{ij}^k = \begin{cases} 1\ if\ vehicle\ k\ reloads\ request\ (i,j) at\ the\ crossdock\ (i\ \in P; j\ \in D, k\ \in\ K) \\ 0 \qquad otherwise \end{cases}$$

$$g_k = \begin{cases} 1\ if\ vehicle\ k\ has\ to\ unload\ at\ the\ crossdock, k\ \in\ K \\ 0 \qquad otherwise \end{cases}$$

$$h_k = \begin{cases} 1\ if\ vehicle\ k\ has\ to\ reload\ at\ the\ crossdock, k\ \in\ K \\ 0 \qquad otherwise \end{cases}$$

$s_i^k = the\ time\ at\ which\ the\ vehicle\ k\ leaves\ the\ node\ i\ (i\ \in N, k\ \in\ K)$

$t_k = the\ time\ at\ which\ vehicle\ k\ finishes\ unloading\ at\ the\ crossdock, (k\ \in\ K)$

$w_k = the\ time\ at\ which\ vehicle\ k\ starts\ reloading\ at\ the\ crossdock\ k \in\ K)$

v_{ij}=*the time at which request* D_{ij} *is unloaded by its pickup vehicle at the cross dock* $(i\ \in P; j\ \in D)$.

In addition, M is an arbitrarily large constant.

The VRPCD can be formulated as follows:

Minimise $\sum_{(i,j)\in E} \sum_{k\in K} C_{ij} x_{ij}^k$

Subject to

$$\sum_{j:(i,j)\in E}\sum_{k\in K} x_{ij}^{k} = 1 \qquad \forall i \in P\ U\ D \tag{1}$$

$$\sum_{i\in P}\sum_{j\in D}\sum_{l:(i,l)\in E} D_{ij}x_{il}^{k} \le Q \qquad \forall k \in K \tag{2}$$

$$\sum_{j:(h,j)\in E} x_{hj}^{k} = 1 \qquad \forall h \in \{o_1, o_3\}, k \in K \tag{3}$$

$$\sum_{j:(j,h)\in E} x_{jh}^{k} = 1 \qquad \forall h \in \{o_2, o_4\}, k \in K \tag{4}$$

$$\sum_{i:(i,h)\in E} x_{ih}^{k} - \sum_{j:(h,j)\in E} x_{hj}^{k} = 0 \qquad \forall h \in P\ U\ D, k \in K \tag{5}$$

$$s_j^k \ge s_i^k + c_{ij} - M\left(1 - x_{ij}^{k}\right) \qquad \forall (i,j) \in E, k \in K \tag{6}$$

$$a_i \le s_i^k \le b_i \qquad \forall i \in N, k \in K \tag{7}$$

$$u_{ij}^{k} - r_{ij}^{k} = \sum_{o\in P\cup\{o_2\}} x_{io}^{k} - \sum_{o\in D\cup\{o_4\}} x_{oj}^{k} \qquad \forall i \in P, j \in D, k \in K \tag{8}$$

$$u_{ij}^{k} + r_{ij}^{k} \le 1 \qquad \forall i \in P, \mathrm{j} \in D, k \in K \tag{9}$$

$$\frac{1}{M}\sum_{i\in P}\sum_{j\in D} u_{ij}^{k} \le g_k \le \sum_{i\in P} u_{ij}^{k} \qquad \forall k \in K \tag{10}$$

$$t_k = s_{o_2}^{k} + Ag_k + B\sum_{i\in P}\sum_{j\in D} D_{ij}u_{ij}^{k} \qquad \forall k \in K \tag{11}$$

$$w_k \ge t_k \qquad \forall k \in K \tag{12}$$

$$w_k \ge v_{ij} - M\left(1 - r_{ij}^{k}\right) \qquad \forall i \in P, \forall j \in D, k \in K \tag{13}$$

$$v_{ij} \ge t_k - M\left(1 - u_{ij}^{k}\right) \qquad \forall i \in P, \forall j \in D, k \in K \tag{14}$$

$$\frac{1}{M}\sum_{i\in P}\sum_{i\in D} r_{ij}^{k} \le h_k \le \sum_{i\in P}\sum_{i\in D} r_{ij}^{k} \qquad \forall k \in K \tag{15}$$

$$s_{o_3}^{k} = w_k + Ah_k + B\sum_{i\in P}\sum_{i\in D} d_{ij}r_{ij}^{k} \qquad \forall k \in K \tag{16}$$

$$\sum_{i,j\in S} x_{ij}^{k} \le |S| - 1 \ , S \in N \ \backslash\{\mathrm{o1}, \mathrm{o2}, \mathrm{o3}, \mathrm{o4}\}, |\mathrm{S}| \ge 2 \tag{17}$$

$$x_{ij}^{k}, u_{ij}^{k}, r_{ij}^{k}, g_k, h_k \in \{0,1\} \quad \forall i \in P, \forall j \in D, (\mathrm{i},\mathrm{j}) \in E, k \in K \tag{18}$$

$$s_i^k, t_k, w_k \ge 0 \qquad \forall i \in N, k \in K \tag{19}$$

$$v_{ij} \geq 0 \qquad \forall i \in P, j \in D \qquad (20)$$

The objective is to minimize the total cost of transportation.

Constraints consist of two types:

The first one concerns vehicle routing while the second is related to the consolidation decisions at the cross-dock.

We noticed that both the pickup and the delivery parts can be formulated as VRPTWs, if the cross dock was absent. Constraint (1) requires each pickup or delivery node to be visited once by a single vehicle. Constraints (2) indicate that the vehicle capacity is never exceeded during both pickup and delivery processes. The constraints (3) ensure that each pickup route starts from O_1 and that of delivery from O_3. Constraints (4) show that each pickup route ends at O_2 and every delivery route ends at O_4. Constraints (5) are flow conservation constraints. Constraints (6) compute the traveling time between two nodes if they are visited consecutively by the same vehicle. Constraints (7) ensure that each node is visited within its time window. Constraints (8) and (9) are of consolidation type; they show the existing link between the pickup and delivery. Constraints (10) force g_k to be 1 if the vehicle needs to unload at the cross dock. Constraints(11) calculate the time required for unloading a vehicle which is equal to the fixed time of unloading preparation (A) added to the unloading time which is equal to the number of unloaded pallets to be multiplied by the time for discharging a unique pallet (B). Constraints (12) and (13) ensure that a vehicle cannot start reloading until it finishes unloading, and all the products to be reloaded on it are ready. The ready time of product *i* is represented by constraint (14), which depends on the time at which the last pickup vehicle of product *ij* finishes unloading. Constraints (15) and (16) for the reloading are similar to (11) and (12). Constraints (17) are subtours elimination constraints.

The model was tested on Cplex to solve exactly small instances, since the solver was not able to find optimal solutions for instances with more than 12 nodes.

4. Problem resolution

This section presents the methods used for the resolution of the problem: A constructive heuristic and a memetic algorithm. The obtained results are compared with those obtained by Cplex on small instances.

4.1.Constructive heuristic

A Best Insert Heuristic (BIH) has been developed for the VRPCD. This heuristic starts with an empty tour. At each iteration, it calculates the minimum insertion cost for each customer previously untreated and performs better integration. When the remaining capacity is insufficient to accept new customers, a new tour is created. In our case, at each iteration of the BIH, the pickup node and its delivery nodes are respectively inserted in a collection and distribution tour. The cost of the insertion is evaluated for all possible positions in all trips and the best location is selected for each node. The feasibility of insertion is evaluated respecting the vehicle capacity and time windows. The specificity of the best insertion heuristic in this case is that the vehicle which

serves the delivery node cannot start before all pickup vehicles returns to the cross dock. Otherwise, when inserting a pickup node, the feasibility of the insertion of the delivery node must be checked. In fact, for a given delivery node, if the last pickup vehicle returns too late to the cross dock, it may be impossible to serve it, even with a dedicated vehicle.

To check the time windows feasibility of the pickup part, we must go through all the nodes located after the location of the insertion and check if the offset of arrival dates of these nodes does not cause time windows violations. Solomon (1987) introduced a method based on the storage of a number of variables to achieve these feasibility tests in O (1) instead of O (n). These techniques remain valid for the pickup part. However, for delivery, the insertion of a node can delay the arrival even for the nodes located before the insertion position, when the availability date of this delivery at the cross dock is greater than all availability dates of the other nodes of this trip. We introduced a technique inspired from the one of Solomon that allows us to do feasibility tests in O (1) even for this special case.

Figure 4 shows an example of 3 suppliers and 3 customers; it is assumed that S1 is the first one to be visited in the pickup part. The vehicle starts its trip from the cross-dock at 6 am and arrives after an hour at S1 (7 am). It starts its service at 7 and loads 10 pallets, (its capacity is not yet reached). After 20 minutes the vehicle leaves S1 (Given that the fixed time for preparation is ten minutes and the time for loading each pallet is one minute, the total loading duration is 20 (10 + 10).
Before inserting a second node in the trip, we must check the following three conditions:

- To generate a minimal cost.
- The amount taken must respect the capacity of the vehicle.
- Neither the time window of supplier nor those of its customers are violated.

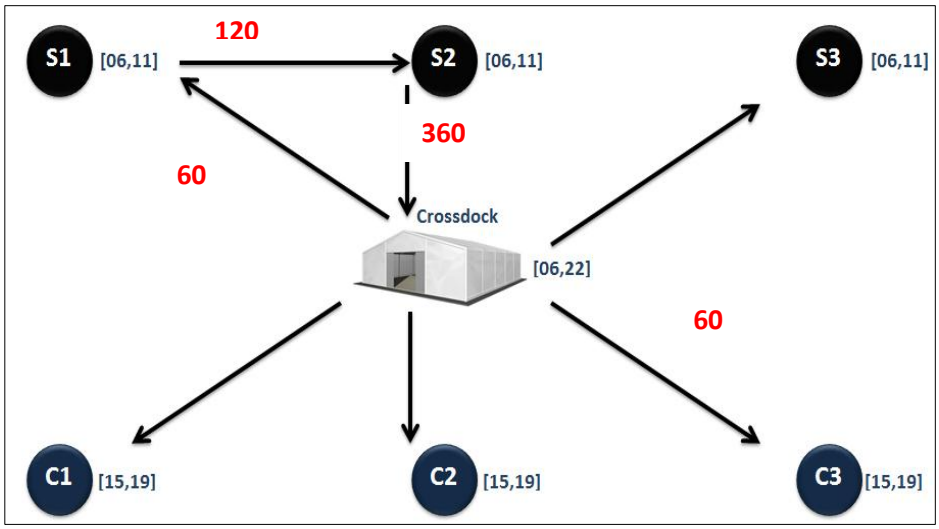

Figure 4. Example of insertion selection

To get to S_2, the vehicle needs two hours; it arrives at 9:20 am and leaves at 9:40 am.

It remains to check if delivery trips are affected. Assume that the S2 delivers C3, the vehicle will need 6 hours to return to the cross dock, it gets there at 3:40 pm (9:40+6). As the vehicle will leave the cross dock at 4:10 pm (since the discharge and the reload at the cross dock will take 30 minutes) if the travel time between the cross dock and C_3 is more than 3 hours and 50 minutes, the time window of C_3 will not be respected. To sum, before any insertion of a node, we must check the conditions mentioned before to ensure the feasibility of this insertion.

4.2. The memetic algorithm for the VRPCD

Genetic Algorithm (or GA) is a metaheuristic proposed by Holland (1975) and popularized by Goldberg (1989). It is known that the early version of the genetic algorithm is not aggressive enough on combinatorial optimization problems compared to other metaheuristics such as tabu search. This is why, Moscato (1999) proposed a more powerful version; hybridized with a local search algorithm called memetics (Memetic Algorithm, MA). Local search in the memetic algorithm brings intensification. Indeed, the resulting solutions from the crossover are improved by local search before undergoing mutation.

4.2.1. Population, chromosomes and evaluation

The proposed memetic algorithm starts with a population *Pop* containing ns chromosomes, sorted in increasing order of costs. The initial population includes random solutions and the BIH solution to introduce some intensification.

Each chromosome is coded as a list of suppliers followed by a list of customers. This list can be viewed as two giant tours which ignore vehicle capacity and time windows. The VRPCD solution is computed by an adaptation of a splitting procedure called split developed for the VRP by Prins(2004) and described in next section.

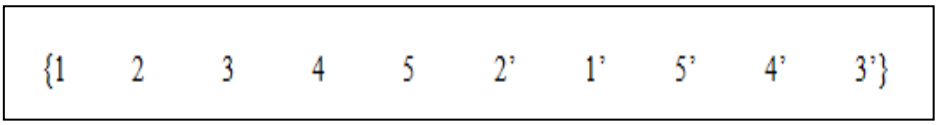

Figure 5. Example of coding the VRPCD

4.2.2. Chromosome evaluation: Splitting procedure

In general, the principal of split for the VRP is to build an auxiliary graph *H* containing one dummy node 0 and n other nodes corresponding to n customers. Each subsequence of customers (Si, Si+1, . . . , Sj) corresponding to a feasible trip is modeled by a weighted arc (i−1, j) in *H*. The best VRP solution subject to the order given by the chromosome is obtained by computing a shortest path between node 0 and node n.

For the VRPTW, a chromosome is encoded as a sequence S containing the n customers, but not including tour delimiters. It may be seen as a giant tour where the vehicle capacity and time windows are ignored. Lack tour delimiters allows to have chromosomes with the same length and

using simple crossovers, like those used in the GA for traveling salesman problem (Reghioui(2008)).

The best solution to the VRPTW, respecting the sequence , the capacity of vehicles and the time windows can be deduced by the splitting procedure: to calculate the shortest path in the auxiliary graph H containing a dummy node 0 and a node per customer. The shortest path from node 0 to any other node in the auxiliary graph H can be calculated using Bellman algorithm for graphs without circuits. This procedure was used to determine where the giant tour should be cut to obtain feasible tours.

For the VRPCD, this operation is realized in two steps. In the first step, the auxiliary graph is built for the pickup part. Each arc represents a trip which respects time windows and vehicle capacity, and the best pickup solution is deduced by computing the shortest path.

The principle of split in the pickup part is illustrated in Figure 5. The upper part of the figure shows an example of chromosome S = (S1, S2, S3, S4, S5). Numbers in parentheses indicate the amount to be collected and those in bracket represent time windows.

Table 2. Instance data used in Figure 6.

Node	a_i	b_i	r_i	Ridge	C_{ij}
CD	0	200	0	[CD,S1]	30
S1	0	50	15	[CD,S2]	25
S2	10	30	20	[CD,S3]	30
S3	20	60	20	[CD,S4]	40
S4	30	50	25	[CD,S5]	10
S5	80	120	20	[S1,S2]	20
				[S2,S3]	30
				[S3,S4]	20
				[S4,S5]	40

The auxiliary graph H is given in the middle of the figure, assuming vehicle capacity Q = 50. Each edge of the graph represents a feasible trip. Thus, arc S1 models a dedicated trip for supplier S1, the value 60 is a go-and-back from the cross dock.

We note that the arc S1S2 is missing since the trip visiting S1 and S2 violates the time window of S2. The trip S2S3c and S4S5 are also feasible because neither the capacity of the vehicle nor time windows are affected. All other trips that are not represented violate vehicle capacity or time windows of suppliers. The result of the procedure split is shown in the lower part in Figure 6.

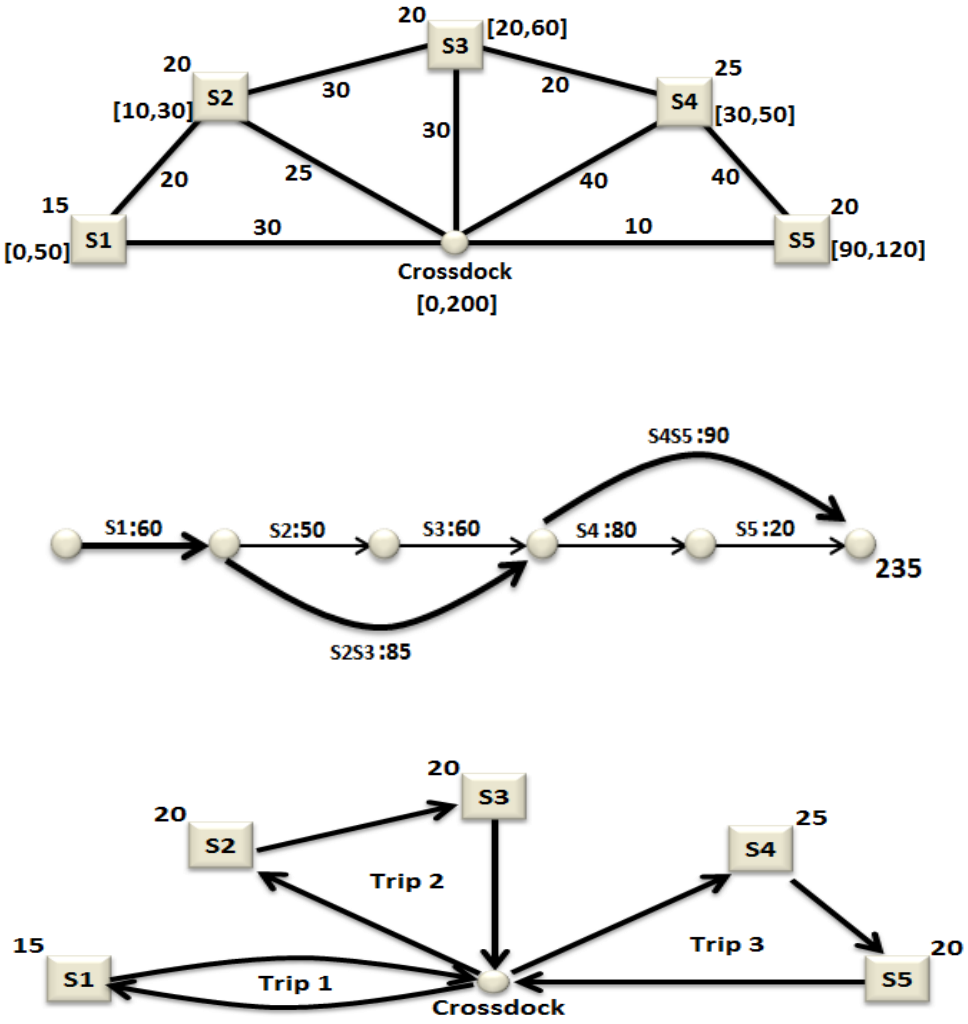

Figure 6. Example of coding the VRPCD

In the second step, the auxiliary graph for the delivery part is built, another constraint is added this time, which is the availability of demands, because delivery trips are constructed by taking into account availability dates of deliveries which depends on the arrival time to the cross dock of their pickup trips. Once the second auxiliary graph is built, the delivery solution is deduced in the same way as for the pickup one.

4.2.3. Selection and crossover

Each iteration of the memetic algorithm selects two parents from the population with the binary tournament: two chromosomes are randomly selected from the population and the best one is kept as the first parent P1. This process is repeated to get the second parent P2. These parents are then combined using the order crossover or OX, a classical crossover for the travelling salesman problem which randomly selects two cutting points i and j in the first parent. Customers between these two cutting points are copied to the same positions in the child. Missing customers are then added by scanning circularly parent P2 from the second relative position j + 1 to the position j. The child is also filled circularly from j + 1. Crossover OX usually provides two children by

inverting parents order. In our version, we generate a single child by randomly choosing the order of parents.

For the VRPCD, OX is applied separately to the pickup or to the delivery parts with a probability of 0.5. Numerical experiments proved that if both pickup and delivery parts are considered by the crossover, the resulting child is too different from its parents. Figure 7 shows an example of crossover OX applied to the delivery part.

P1:	C1	C2	|C3	C4|	C5
P2:	C2	C5	|C1	C4|	C3
C:	C5	C1	|C3	C4|	C2

Figure7. Example of crossover OX for the delivery part

4.2.4. Local search

Local search for the VRPCD is based on an adaptation of some classical VRP moves (Or-opt, exchange,...). For the pickup solution, if we consider the move or-opt that remove one supplier from a trip T1 and reinsert it in another trip T2, the arrival time to the cross dock of the vehicle of trip T2 may be delayed, and then it will shift the availability date at the cross dock of all the deliveries of this trip. This may also make the delivery impossible because of time windows. In such case, for each node of trip T2, if its new availability date at the cross dock will shift the departure time of its delivery trip, the feasibility of time windows must be checked for all the deliveries of this trip.

The improved child integrates the population and replaces one of the worst solutions to ensure the conservation of the best solution.

5. Computational results

5.1. Implementation and instances

To illustrate the potential of the proposed memetic algorithm on providing high-quality solutions with a remarkable computational efficiency, an extensive number of examples have been studied. Our algorithms were implemented in Delphi and executed on a PC with an Intel core 2 dual processor, 2 GHZ, and 2 GB of RAM.

Our test set is built upon the instances (real-life instances) introduced in the work of wen et al(2008); size of these instances is between 30 and 200. Since instances of this size are still out of reach for our exact solution approach, we randomly extracted other instances with a smaller size 8, 12 and 20. Coordinates of suppliers and customers were kept as in Wen et al. (2008).Since our problem is different in terms of distribution of customer orders to suppliers, these instances have been adapted to our problem. These instances can be retrieved from http://www.dcc.ufmg.br/~fsantos/instances.

The first test data is divided in two subsets, the first subset contains 5 small instances used to compare the MA with the exact method. The second consist of three Euclidean sets, each one containing 10 instances denoted by *Ia*, *Ib*, *Ic*, *Id* , *Ie* , *If* , *Ig* , *Ih*, , *Ii* and *Ij*, respectively, where I stands for the number of suppliers. Each instance has the same number of suppliers and customers known by their pickup and delivery locations (x; y). The time window for each pickup node is limited between 6:00 and 11:00 am. For delivery it is between 3:00 and 7:00 pm. The choice of dividing the day into two parts was motivated by example systems of some of our partners. The time horizon for the whole transportation operation is from 6:00 am to 10:00 pm. The demand transported from each pickup location to a delivery location is given in number of pallets. Vehicles drive at a constant speed of 60 km/h and have a capacity of 33 pallets. It takes ten minutes to prepare a vehicle, plus an additional one minute for each pallet to be loaded or unloaded.

Taking into account the arrival time of the last pickup vehicle at the cross dock, the unload/load time and the travel time between the cross dock and the delivery node, the delivery vehicle must arrive before the end of the customer's time window.

5.2. Results analysis

5.2.1. Parameters settings

The objective of our memetic algorithm is to minimize the total distance. The parameters used in this study are: A population containing ns = 30 chromosomes, a local search rate $p_{ls} = 0.1$, a minimum cost spacing $\Delta = 0.2$, a maximum number of crossovers = 3000, a maximum number of crossovers per phase without improvement =3000.

5.2.2. Evaluation of the used memetic algorithm

Our MA was first evaluated by solving a series of small size problems and comparing the results obtained with those found using the exact method. Table 2 shows the results of the MA compared to the ones obtained by the solver Cplex. Cplex column provides the result of the branch and cut algorithm obtained by the simulation of the proposed mathematical formulation presented in section 3, the MA column provides the result of the memetic algorithm and the computational time.

Table 3. Comparison between memetic algorithm and branch and cut algorithm.

Data Set	Nodes	CPLEX		MA	
		Endcost	Time (s)	Endcost	Time (s)
DATA4_A	8	1069,802	0,05	1069,811	0,49
DATA5_A	10	1143,929	0,08	1143,929	0,35
DATA5_B	10	994,631	0,11	994,631	0,39
DATA6_A	12	1286,172	0,16	1286,179	0,33
DATA6_B	12	1403,315	0,48	1403,315	0,49
DATA10_A	20	*		1906,224	0,90
DATA10_B	20	*		2276,194	0,54
DATA10_C	20	*		1856,265	0,78
DATA10_D	20	*		2239,171	1,35
DATA10_E	20	*		2029,263	0,74
DATA10_F	20	*		1790,135	1,10
DATA10_G	20	*		2000,594	0,85
DATA10_H	20	*		1956,153	1,29
DATA10_I	20	*		2075,500	1,00
DATA10_J	20	*		1948,906	0,85
DATA30_A	60	*		7474,786	4,39
DATA30_B	60	*		7555,811	2,23
DATA30_C	60	*		6148,789	6,95
DATA30_D	60	*		7042,106	6,70
DATA30_E	60	*		5986,499	5,02
DATA30_F	60	*		7131,365	5,45
DATA30_G	60	*		6774,495	2,61
DATA30_H	60	*		7062,248	3,51
DATA30_I	60	*		6863,495	4,42
DATA30_J	60	*		6303,393	6,04
DATA50_A	100	*		11096,202	7,42
DATA50_B	100	*		10796,803	13,62
DATA50_C	100	*		11281,220	11,41
DATA50_D	100	*		10831,557	6,90
DATA50_E	100	*		10445,996	10,56
DATA50_F	100	*		10998,258	10,54
DATA50_G	100	*		10675,237	9,44
DATA50_H	100	*		10574,697	9,01
DATA50_I	100	*		10129,632	10,77
DATA50_J	100	*[1]		10008,955	10,78

1 Cplex goes out of memory for these instances

For the first subset of small instances, Cplex was able to find the optimal solution, and the memetic algorithm found exactly the same results. For the second subset, Cplex goes out of memory because of the large number of variables and constraints. The tests proved the efficiency of the MA compared to exact methods like the branch and cut algorithm.

A detailed description of the best sets of pickup and delivery routes for examples 10a and 30a found with the memetic algorithm are shown in Tables 3 and 4, respectively. Such tables include the pickup and delivery routes and the total load picked up or delivered by each vehicle. In addition, the pickup and delivery trip costs are reported. Graphical representations of the best solutions for examples 10a and 30a provided by the MA are displayed in Figures 8 and 9 respectively(Appendix). Where r_{ij} is the request of customer i to supplier j.

It is obvious in the representation of solution of instance 10a presented in the appendice that requests are not picked up and delivered by the same vehicles. furthermore, they must be unloaded at the cross-dock, justifying non-use of direct passages from pickup points to those of delivery without going to the cross dock.

5.2.3. Comparison between BIH and MA

Table 4 shows the results of comparing MA with BIH .The BIH column shows the result of the Best Insertion Heuristic for the set of instances, The MA column presents the result of the memetic algorithm. The saving column gives the percentage of improvement over the solution of BIH; the last one is for computational time.

Table4 . Comparison between memetic algorithm and BIH.

Data set	Nodes	BIH	Time(s)	MA	Time(s)	GAP (%)
DATA10_A	20	2394,807	0,17	1906,224	0,90	4,89
DATA10_B	20	2799,268	0,01	2276,194	0,54	5,23
DATA10_C	20	2612,165	0,17	1856,265	0,78	7,56
DATA10_D	20	2540,801	0,90	2239,171	1,35	3,02
DATA10_E	20	2613,486	0,26	2029,263	0,74	5,84
DATA30_A	60	9966,764	3,09	7474,786	4,39	24,92
DATA30_B	60	9468,676	1,29	7555,811	2,23	19,13
DATA30_C	60	8706,521	6,18	6148,789	6,95	25,58
DATA30_D	60	9072,882	6,39	7042,106	6,70	20,31
DATA30_E	60	8351,863	3,75	5986,499	5,02	23,65
DATA50_A	100	14691,297	5,67	11096,202	7,42	35,95
DATA50_B	100	14773,067	13,54	10796,803	13,62	39,76
DATA50_C	100	14782,854	11,04	11281,220	11,41	35,02
DATA50_D	100	14787,624	5,01	10831,557	6,90	39,56
DATA50_E	100	14261,985	10,14	10445,996	10,56	38,16

The results show that the MA performs better than BIH. It improves strongly the result of the BIH in all cases, especially when the size of problems increases (over 30 customers), it exceeds 30% with reasonable computational time not exceeding 15 seconds for all instances.

These tests confirm the efficiency of the memetic algorithm and the need of use of such metaheuristics to solve the problem instead of using simple rules or heuristics.

6. Conclusion

Although cross-docking has been widely practiced within both manufacturing and retailing companies and brings benefits to companies, there are very few studies on the integration of vehicle routing problems and cross-docking. In this study, two different solution approaches have been developed: A mathematical formulation and a memetic algorithm. The latter is based on the solution given by the best insertion heuristic.

Our MA was first validated by solving a series of small size problems and comparing the results obtained with the exact formulation's results (Cplex). For these examples, MA practically provides optimal solutions in a short CPU time. After validation; the MA was applied to larger problem instances involving up to 50 customer requests and the least transportation cost as the problem objective. In all cases, the MA improved strongly the solution of the BIH within a small CPU time especially for larger instances.

In future work, additional constraints will also be taken into consideration, such as direct connections (suppliers to clients without going through the cross dock). A lower bound is needed to be developed to effectively evaluate the performance of the proposed methods.

Rererences

Yang H.L., Sarker B. and Chang C.T. (2013). A two-warehouse partial backlogging inventory model for deteriorating items with permissible delay in payment under inflation. *Applied Mathematical Modelling*, Vol. 37, pp. 2717–2726.

M.Wen, J.Larsen, J.Clausen, J.F Cordeau, and G Laporte(2008), Vehicle routing with Cross-Docking, *Journal of Operational Research Society* ,vol.60, pp 1708–1718.

Y. H Lee, W. J Jung, and K.M Lee (2006), Vehicle routing scheduling for cross docking in the supply chain.*Computer and Industrial Engineering*, vol.51, pp.247–256.

F.A Santos,G.R Mateus, and A.S Da Cunha (2011).A novel column generation algorithm for the vehicle routing problem with cross-docking. International conference on Network optimization,5th. pp 412-425.

Solomon. M (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research*,vol.35,pp.254–265.

J.H. Holland (1975). *Adaptation in natural and artificial systems*. The University of Michigan Press, Ann Arbor, MI, USA.

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

P. Moscato(1999). *Memetic algorithms: a short introduction*. In D. Corne, M. Dorigo, and F. Glover, editors, New ideas in optimization, pp 219–234: McGraw-Hill.

C. Prins(2004), A simple and effective evolutionary algorithm for the vehicle routing problem, *Computer Operations Research*, vol.31pp.1985–2002.

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

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

Apte,M.U., Viswanathan,S(2000).Effective cross docking for improving distribution efficiencies.*International Journal of Logistics: Research and Applications*,vol.3, pp.291–302,

N.Labadi ,C.Prins,M.Reghioui(2008). A memetic algorithm for the vehicle routing problem with Time windows , *RAIRO Operations Research*,vol.42(3),pp.415-431.

Musa R, Arnaout J P, Jung H (2010). Ant colony optimization algorithm to solve for the transportation problem of cross-docking network. *Computers &Industrial Engineering*, vol.59 (1), pp.85–92.

M.Reghioui(2008). Problèmes de tournées de véhicules avec fenêtres horaires ou préemption des taches. *Thèse de doctorat de l'Université de Technologie de Troyes*.185 p.

Appendix

Table 5. The best cost solution for the instance 30 a using the MA algorithm.

Pickup			
Vehicle	**Trip**	**Qty Loaded**	**Cost**
V1	CD – S9 – S4 – CD	25	264,32
V2	CD – S2 – S8 – CD	24	361,83
V3	CD – S3 – S6 – S10 – S5 – CD	32	298,22
V4	CD – S7 – S1 – CD	17	449,05

Delivery			
Vehicle	**Trip**	**Qty Delivered**	**Cost**
V1	CD – C10 – C3 – C6 – C8 – C1– CD	31	532,80
V2	CD – C9 –C7 – CD	24	142,45
V3	CD – C2 – CD	14	104,75
V4	CD – C4 – C5 – CD	29	166,33

Table 6. The best cost solution for the instance 10 a using the MA algorithm.

Pickup			
Vehicle	**Trip**	**Qty Loaded**	**Cost**
V1	CD– S19 – CD	24	500,26
V2	CD – S24 – S13 – CD	31	294,16
V3	CD– S29 –S3- CD	31	292,52
V4	CD – S1– CD	27	178,54
V5	CD – S4-S6– CD	29	447,10
V6	CD – S16 –S8– CD	28	181,32
V7	CD – S27-S30-S11 – CD	32	417,11
V8	CD – S17 – S18 – CD	33	405,32
V9	CD – S12-S2 – CD	28	428,12
V10	CD – S15 – S28 – CD	31	310,18
V11	CD – S14 – CD	32	428,53
V12	CD – S9 – S20 – CD	21	494,53
V13	CD – S7 – S23 – CD	30	425,44
V14	CD – S22 –S25-S26– CD	32	293,27
V15	CD – S10-S21-S5 – CD	32	299,47

Delivery			
Vehicle	**Trip**	**Qty Delivered**	**Cost**
V1	CD – C5 – CD	25	56,95
V2	CD – C18-C11 – CD	32	106,58
V3	CD – C1-C2– CD	32	97,90
V4	CD – C30 – C6 – CD	33	163,07
V5	CD – C25 – CD	20	65,63
V6	CD – C16 – CD	16	56,93
V7	CD – C24 – C26-C27-C19 – CD	28	110,70
V8	CD – C23 – CD	30	108,98
V9	CD – C7 – C13-C20 – CD	29	107,63
V10	CD – C3 – CD	10	82,34
V11	CD – C14– CD	22	106,10
V12	CD – C29-C9 – CD	29	168,71
V13	CD – C17 – CD	21	171,59
V14	CD – C12– CD	27	110,42
V15	CD – C8-C15-C10-C4 – CD	31	198,62
V16	CD – C21 – CD	24	168,31
V17	CD – C28 – C22 – CD	32	178,44

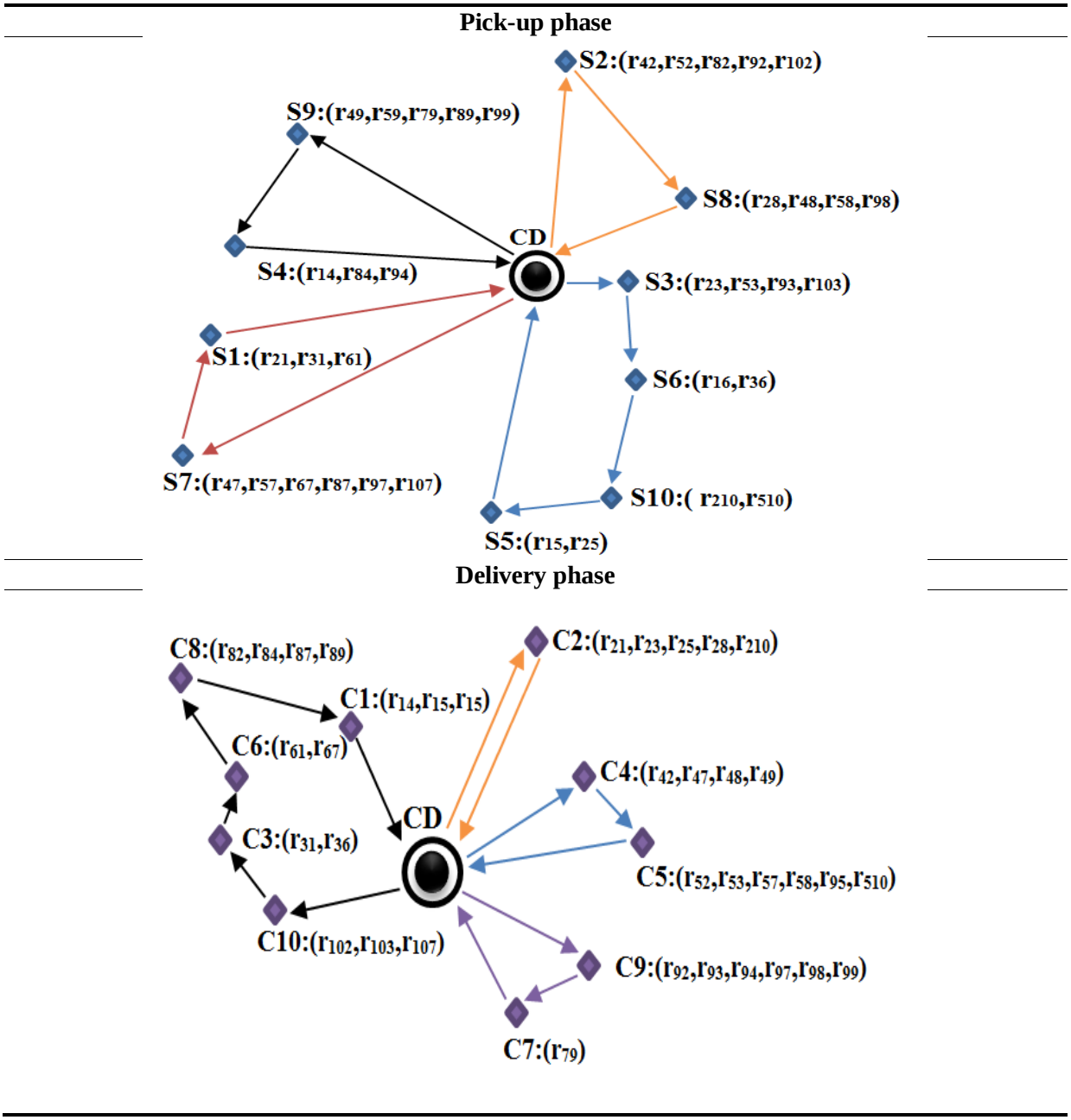

Figure 8. The best routing-cost solution to the instance 10a using the MA algorithm.

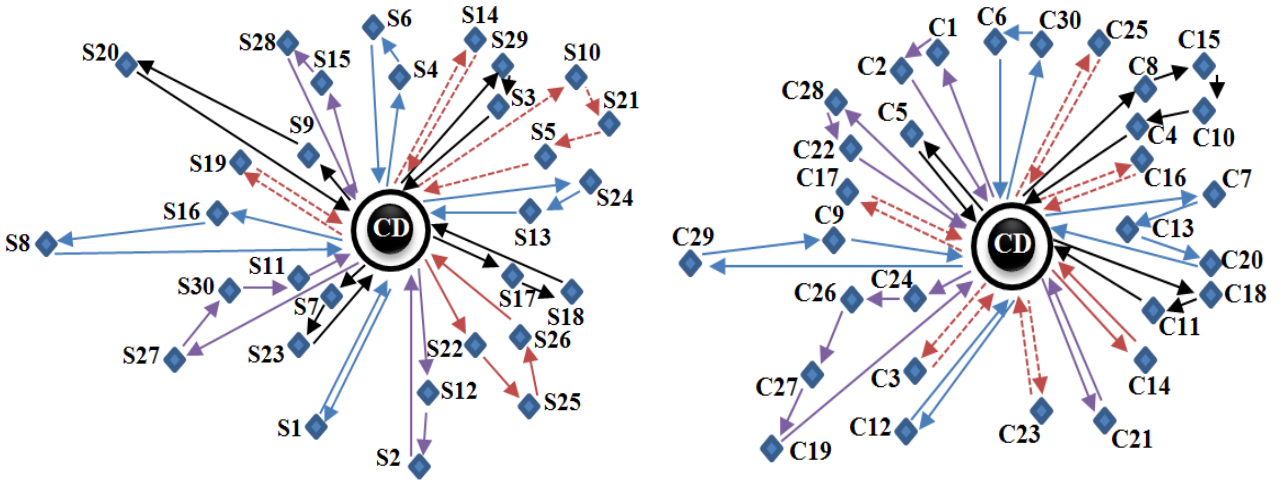

Figure 9. The best routing-cost solution to the instance 30 a using the MA algorithm.