

# A Hybrid Discrete Artificial Bee Colony for the Green Pickup and Delivery Problem with Time Windows

Djebbar Amel Mounia

University of science and technology of Oran Mohamed-Boudiaf

El Mnaouar, BP 1505, Bir El Djir 31000, Oran, Algeria

E-mail: mounia.djebbar@univ-usto.dz

Bachir Djebbar

University of science and technology of Oran Mohamed-Boudiaf

El Mnaouar, BP 1505, Bir El Djir 31000, Oran, Algeria

E-mail: bachir.djebbar@univ-usto.dz

**Keywords:** CO<sub>2</sub> emissions, combinatorial optimization, fuel consumption

**Received:** November 1, 2019

*This paper formulates a new optimization pickup and delivery problem with time windows which take into account CO<sub>2</sub> emissions. This new NP-hard combinatorial optimization problem is called green pickup and delivery problem with time windows (GPDPTW), the recent development in the vehicle routing problem and its variants, which extends PDP and PDPTW with respect to several constraints. The objective is to find a set of routes for a fleet of vehicles in order to serve given transportation requests with a minimization of fuel consumption and CO<sub>2</sub> emission to ensure the preservation of a clean and green environment. This paper presents a mathematical formulation and proposes a hybrid discrete artificial bee colony algorithm (HDABC) as a meta-heuristic algorithm which combines a discrete artificial bee colony with neighborhood operators to solve the GPDPTW model. To the best of our knowledge, this is the first time that an emission of CO<sub>2</sub> for the PDPTW is proposed. We performed computational experiments to evaluate the effectiveness of the proposed method, which provides the best result and can effectively find an optimal tour. Our results show that, (1) the shortest route is not necessarily the route that consumes the least fuel; (2) the fuel consumption is affected by the load and the number of vehicles.*

*Povzetek: Članek predstavi novo metodo za optimizacijo prevzema in dostave s časovnimi okni z minimalizacijo porabe goriva in emisij CO<sub>2</sub>.*

## 1 Introduction

Nowadays, the amount of CO<sub>2</sub> emission caused by transportation is significant for wider environmental and social impacts rather than just economic costs. It has direct effects on human health, e.g., pollution, and indirect ones, e.g., climate change. The objective of harmonizing the environmental and economic costs is to implement an effective strategy to meet the environmental concerns and financial indices. One of the most important decisions concerns the routing of vehicles with the minimum amount of CO<sub>2</sub> emissions, since it offers great potential to reduce the fuel consumption and to ensure the preservation of a clean and green environment. Thus, reducing fuel consumption can directly reduce carbon emissions. In addition, fuel consumption accounts for as much as 60% of the operating cost of a vehicle, according to [1]. Therefore, reducing fuel consumption can also reduce operating costs.

The vehicle routing problem (VRP) is a generic name given to a class of problems to determine a set of vehicle

routes, in which each vehicle departs from a given depot, serves a given set of clients, and returns back to the same destination. The basic VRP involves a single depot, a fleet of identical vehicles stationed at the depot, and a set of clients who require delivery of goods from the depot. The objective of basic VRP is to minimize the total routing cost, subject to the capacity constraints on the vehicles [2] [3].

One variation of the classical vehicle routing problem considers clients that require pickup and delivery service [4]. This problem is called the Pickup and Delivery Problem with Time Windows (PDPTW). In this variant, PDPTW deals with a number of client requests that are to be served by a fleet of vehicles, while a number of constraints must be observed. Each vehicle has a limited capacity (the capacity constraint). A vehicle route usually starts and ends at a central depot. A request must be picked up from a pickup location to be delivered to a corresponding delivery location. The pickup and delivery pair must be served by the same vehicle (the coupling

constraint) and the pickup must precede the delivery (the precedence constraint). In addition, every request must be served within a predetermined time window interval (the time window constraint). If the vehicle arrives earlier than the allowed service time, it should wait until the beginning of the specified period. A vehicle may never arrive to a location after the end of the time window of the location. The PDPTW mainly involves transportation activities to complete/serve a set of requests, but the transportation has an impact on the environment due to pollution and CO<sub>2</sub> emissions.

In this paper, we consider the Green Pickups and Deliveries problem with Time Windows (GPDPTW) which is an extension of the PDPTW. The objective is to construct valid routes for the vehicles without violating vehicle capacity, time window, precedence and coupling constraints with minimal CO<sub>2</sub> emissions.

The contributions of this paper are twofold. First, the paper addresses the GPDPTW problem with precedence, coupling, capacity, time windows constraints and we introduce the factor of CO<sub>2</sub> emission, which is applied to this variant of VRP for the first time. Second, we develop and implement the Hybrid discrete artificial Bee Colony (HDABC) algorithm to solve it. To the best of our knowledge, this study is the first attempt at applying the discrete artificial bee colony meta-heuristic. Our aim is to minimize the amount of CO<sub>2</sub> emissions.

The remainder of this paper is structured as follows. Section 2 discusses some related works on Pickup and Delivery Problem with Time Windows and the Green vehicle routing problem. Section 3 focuses on the formulation of the GPDPTW. Section 4 presents a brief definition of artificial bee colony. Section 5 describes a proposed hybrid discrete artificial bee colony. Then, a case study is presented in Section 6. The last section is devoted to conclusions and the research perspectives related to the current work.

## 2 Related work

Dumas et al [5], were the first to use column generation for solving PDPTW. They proposed a branch and bound method that is able to handle problems with up to 55 requests. Sol et Savelsbergh [6] proposed a branch and price algorithm to solve the PDPTW with the objective to minimize the number of vehicles and the total travel distance. Nanry et Barnes [7] are among the first researchers to present a meta-heuristic for PDPTW. The meta-heuristic is based on a reactive tabu search. First, a feasible solution is constructed using greedy insertion method. Next, tabu search is used to improve the initial solution. Three neighborhood moves are proposed in this paper. They are: single pair insertion, swapping pairs between route and within route insertion. In order to evaluate their work, the authors created PDPTW test instances from standard vehicle routing problems with time windows proposed by Solomon [8]. Li et Lim [9] developed a hybrid meta-heuristic based on tabu search and simulated annealing to solve the PDPTW, and they also produced several test instances for the PDPTW which are generated from

Solomon's 56 benchmark instances [8]. A two phase method proposed by Lau et Liang [10] was developed. In the first phase, they applied a novel construction heuristic to generate an initial solution. In the second phase, a tabu search method is proposed to improve the solution. Lim et al [11] applied "Squeaky wheel" optimization and local search to the PDPTW. Another approach to this problem was proposed by Pankratz [12], who used a grouping genetic algorithm, and this is extended to a multi-strategy grouping genetic algorithm by Ding, Li et Ju [13]. Lu et Dessouky [14] presented a new insertion-based construction heuristic to solve the multi-vehicle pickup and delivery problem with time windows. Their main contribution was to define new criteria to evaluate requests insertion based on reduction of time slack compared to the classical one based on the incremental distance measure. Bent and Van Hentenryck [15] proposed a two-stage hybrid algorithm where the first stage uses a simple simulated annealing algorithm to decrease the number of routes, while the second stage uses a large neighborhood search to decrease the total travel cost. The heuristic was tested on the problems proposed by Li et Lim [9]. In addition, Dergis et Dohmer [16] showed that the approach of indirect local search with greedy decoding gives results which are competitive with both Li et Lim [9] and Pankratz [12]. Ropke et Cordeau [17] presented a new branch and cut and price algorithm in which the lower bounds are computed by the column generation algorithm and improved by introducing different valid inequalities to the problem. More recently, ant colony System was applied by Carabetti, De Souza et Fraga [18]. Harbaoui et al [19] presented an approach based on genetic algorithms and Pareto dominance method to give a set of satisfying solutions to the PDPTW minimizing total travel cost, total tardiness time and the vehicles number.

After that, the green concept emerged as one of the latest extensions of the VRP literature in recent years. Researchers suggest that there are possibilities for reducing carbon dioxide (CO<sub>2</sub>) emissions by extending the traditional VRP objectives to account for wider environmental and social impacts rather than just economic costs [20] [21] [22]. Until now, little research for minimizing energy consumption in transportation planning has been carried out, such as the PhD dissertation of Palmer [23] that presented an integrated routing and emissions model for freight vehicles and investigates the role of speed in reducing CO<sub>2</sub> emissions under various congestion scenarios and time window settings. However, Palmer did not take into account the vehicle loads in his model, although this was offered as a future research topic. Kara et al [24] considered a more realistic cost of transportation that is affected by the load of the vehicle as well as the distance of the arc travelled. They defined energy minimizing vehicle routing problem as the capacitated vehicle routing problem with a new objective of cost, in which the cost function is a product of the total load (including the weight of the empty vehicle) and the length of the arc. However, they used their work to represent the energy so as to simplify the relationship between minimizing the consumed energy

and the variables of the vehicle conditions. Details of the formulation of fuel consumption are not provided. Maden et al [25] considered a vehicle routing and scheduling problem with time windows in which speed depends on the time of travel. Fagerholt et al [26] proposed an alternative solution methodology in which the arrival time was divided and the problem was solved as a shortest path problem on a directed acyclic graph. Xiao et al [27] proposed a Fuel Consumption Rate (FCR) considered Capacitated Vehicle Routing Problem (CVRP), which extends CVRP with the objective of minimizing fuel consumption. In their paper, both the distance traveled and the load are considered as the factors which determine the fuel costs. FCR is taken as a load dependent function, where FCR is linearly associated with the vehicle's load. Kuo et Wang [28] developed a tabu search heuristic in order to find feasible vehicle routes while minimizing the total fuel consumption. They incorporated the effect of vehicle speed into the fuel cost. They took the travel speed as a parameter and observed the effect of this by conducting experiments on four different data sets with differing travel speed patterns. Zhang et al [29] studied the capacitated vehicle routing problem from an environmental perspective and introduced a new model called environmental vehicle routing problem (EVRP) with the aim of reducing the adverse effect on the environment caused by the routing of vehicles. The environmental influence is measured through the amount of carbon dioxide emission. They designed the hybrid artificial bee colony algorithm to solve the EVRP model. Zhang et al [30] studied a vehicle routing problem (VRP) with the consideration of fuel consumption and carbon emission. They developed an improved tabu search algorithm named RS-TS for solving the model. In the RS-TS algorithm, they introduced a novel route encoding and decoding algorithm named WSS, in which three neighborhood search methods are applied. Poonthaler et Nadarajan [31] introduced a bi-objective Fuel efficient Green Vehicle Routing Problem (F-GVRP) with varying speed constraint. The problem is solved using Particle Swarm Optimization with Greedy Mutation Operator and Time varying acceleration coefficient. Liu et Jiang [32] introduced the load-dependent vehicle routing problem with time windows. They designed a new constraint relaxation-based algorithm and they presented an effective execution scheme of local search procedures. Other VRP-related studies that aim at minimizing total fuel consumption include Apaydin et Gonullu [33], Maraš [34], Nanthavanij et al [35] and Tavares et al [36].

Another problem that considers fuel consumption is the pollution routing problem (PRP). The PRP was proposed by Bektas et Laporte [22]. Its aims are to find a set of vehicle routes and vehicle speeds over the routes that minimize the operational and environmental costs, while respecting constraints on time and vehicle capacities. The PRP was addressed with a two-phase heuristic in Demir et al [37]. In the first phase, the vehicle routing problem with time windows is solved by means of an adaptive large neighborhood search, including five insertion operators and twelve removal operators. In a

second phase, vehicle speeds are optimized using a recursive algorithm. A bi-objective variant considering fuel and driving time minimization is presented in Demir et al [38] and Franceschetti et al [39] considered the time-dependent PRP. Kramer et al [40] proposed a method which combines a local search-based meta-heuristic with an integer programming approach to solve the Pollution Routing Problem. This approach was also used to solve two other environmental based VRPs, namely the fuel consumption vehicle routing problem and the energy minimizing vehicle routing problem, as well as the well-known Vehicle Routing Problem with Time Windows (VRPTW) with distance minimization. Xiao et Konak [41] presented a Green Vehicle Routing and Scheduling Problem (GVRSP) considering general time-dependent traffic conditions with the primary objective of minimizing CO<sub>2</sub> emissions and weighted tardiness. They proposed a new mathematical formulation to describe the GVRSP with hierarchical objectives and weighted tardiness.

Other papers treated another variant of PDPTW that considered dynamic pickup and delivery problems with time window uncertainties [42], the same problem with time windows and electric vehicles was studied by [43] and [44] considered a setting in which a company not only has its own fleet of vehicles to service requests, but may also use the services of occasional drivers.

### 3 GPDPTW formulation

The GPDPTW can be formally defined as follows. Let  $G = (N, A)$  be a graph. The node set is  $N = \{i \in N / i = 0, 1, 2, \dots, m\}$ , such that  $m$  denotes a location. The node 0 denotes the depot. Since for each request we have a pair of pickup and delivery locations, the set  $N^+ = \{i \in N / i = 1, 2, \dots, m/2\}$  represents pickup locations, and the set  $N^- = \{i \in N / i = (m/2) + 1, \dots, m\}$  represents delivery locations.

Each location  $i$  is associated with:

- A demand  $q_i$ , such that  $q_i > 0$  for a pickup location,  $q_i < 0$  for a delivery location and  $q_i + q_j = 0$  for the same customer's pickup and delivery locations ( $q_0 = 0$ ).
- A service time  $s_i$  ( $s_0 = 0$ ), which is the time needed to load or unload a pickup or a delivery demand.
- A time window  $[e_i, l_i]$  during which the location must be served, and  $l_i \geq e_i$

For each pair of nodes  $(i, j)$ , travel time  $t_{ij}$  and a travel distance  $d_{ij}$  are specified.

The GPDPTW consists of designing a set of routes such that:

1. Each route starts and ends at the depot;
2. Each location is visited exactly once by exactly one vehicle;
3. The total vehicle load in any arc does not exceed the capacity of the vehicle assigned to it;
4. The total duration of each route (including travel and service times) does not exceed a duration limit.
5. If a vehicle arrives before the earliest pickup or delivery time of a location, it is allowed to wait until the start of the time window.

- 6. The precedence constraint requires that each pickup location must precede the corresponding delivery location.
- 7. The coupling constraint requires that the same pickup and delivery locations must be served by the same vehicle.

Savelsbergh et al [45] showed that the VRP is a NP-hard problem. Since the GPDPTW is a generalization of the VRP, it's a NP-hard combinatorial optimization problem, and the presence of many constraints makes the problem particularly complicated. The mathematical formulation of GPDPTW is a combination of Christofides et al [2], Savelsbergh et Sol [45], Xiao et al [27] and Zhang et al [29].

- $x_{ijk}$  a binary variable indicating whether arc (i, j) is traversed by vehicle k
- $x_{ijk} = 1$  if vehicle k traverses arc (i, j)
- $x_{ijk} = 0$  if vehicle k does not traverse arc (i, j)
- $y_{ik}$  load of vehicle k while visiting node i
- $Q_k$  capacity of vehicle k
- $D_i$  departure time from the node i /  $D_i \in [e_i, l_i]$ , where  $D_i = \max\{A_i, e_i\}$
- CE the CO<sub>2</sub> emission rate
- FCR the fuel consumption rate
- $\rho_0$  the empty load FCR
- $\rho^*$  the full load FCR
- $\rho$  the FCR provided that load is q
- $q_{ijk}$  the load of vehicle while k traverses arc (i, j)

In this paper, we consider that each vehicle emits a certain amount of CO<sub>2</sub> when traveling over an arc (i, j). This amount is dependent on a number of factors, such as load, number of vehicle and distance travelled, among others. Whereas the CO<sub>2</sub> emission (CE) is fixed, it is estimated at 2.61 kg of CO<sub>2</sub> for each liter of diesel consumed [46]. The formulation of fuel consumption is provided in [27]. It is determined by both the distance traveled and the load of vehicle. Our objective is to serve all client requests while minimizing the total cost of transport. This cost is related to the CO<sub>2</sub> emission rate, the number of vehicles used and the distance travelled.

$$\text{Minimiser } f_1 = \sum_{k \in K} \sum_{i \in N} \sum_{j \in N} x_{ijk} * d_{ijk} \tag{1}$$

$$\text{Minimiser } f_2 = \sum_{k \in K} \sum_{i \in N} \sum_{j \in N} CE * \left( \rho_0 + \frac{\rho^* - \rho_0}{Q_k} q_{ijk} \right) * x_{ijk} * d_{ijk} \tag{2}$$

Subject to

$$\sum_{i=1}^N \sum_{k=1}^K x_{ijk} = 1, \quad j = 2, \dots, N \tag{3}$$

$$\sum_{j=1}^N \sum_{k=1}^K x_{ijk} = 1, \quad i = 2, \dots, N \tag{4}$$

$$\sum_{i=1}^N x_{i0k} = 1, \quad \forall k \in K \tag{5}$$

$$\sum_{j=1}^N x_{0jk} = 1, \quad \forall k \in K \tag{6}$$

$$\sum_{i=1}^N x_{iuk} - \sum_{j=1}^N x_{ujk} = 0, \quad \forall k \in K, \forall u \in N \tag{7}$$

$$x_{ijk} = 1 \Rightarrow y_{jk} = y_{ik} + q_i, \quad \forall k \in K, \forall i, j \in N \tag{8}$$

$$y_{0k} = 0, \quad \forall k \in K \tag{9}$$

$$0 \leq y_{jk} \leq Q_k \quad \forall k \in K, \forall j \in N \tag{10}$$

$$D_p \leq D_d \quad \forall p \in N^+, \forall d \in N^- \tag{11}$$

$$D_0 = 0 \tag{12}$$

$$x_{ijk} = 1 \Rightarrow D_i + t_{ijk} \leq D_j, \quad \forall k \in K, \forall i, j \in N \tag{13}$$

$$t_{0i} + s_i + t_{ij} < l_j, \quad \forall i, j \in N, i \neq j \tag{14}$$

where  $K$  is the total number of vehicles,  $d_{ijk}$  is the travel distance from customer  $i$  to customer  $j$  by vehicle  $k$ .

Constraints (3) and (4) form the feasible routes of vehicles, so that every customer is visited by exactly one vehicle, and every vehicle that arrives to a location must leave that location. Constraints (5) and (6) ensure that each vehicle is used to serve at most one route. Constraint (7) ensures the route continuity. Constraints (8), (9) and (10) show that the total demands of any route must not exceed the capacity of the vehicle. Constraints (11), (12) and (13) ensure the precedence constraint. Constraint (14) ensures that only edges satisfying the time window constraint are allowed.

---

#### Initialization algorithm

---

- 1 Let  $k = 0$  {k is the number of vehicles used}
  - 2 Let list not empty {list contains all pickup node}
  - 3
  - 4 **repeat**
  - 5 Initialize an empty route  $r$
  - 6  $k = k + 1$
  - 7 **for** (All unassigned requests) **do**
  - 8 A request  $p_i$  is randomly selected from list
  - 9 Insert  $p_i$  at the end of the current route  $r$ ;
  - 10 Insert his corresponding  $d_i$  request into route  $r$ ;
  - 11 Call the IsFeasibleSolution algorithm to improve  $r$
  - 12
  - 13 **if** ( $r$  is a feasible route) **then**
  - Mark  $p_i$  as inserted
  - until** (All requests have been inserted)
- 

## 4 Artificial bee colony

The Artificial Bee Colony (ABC) algorithm is a swarm intelligence technique inspired by the intelligent foraging behavior of honey bees. This algorithm was proposed by Karaboga et al [47] [24] [48] [49] [50] based on the foraging behaviour of honey bees. The ABC algorithm classifies the foraging artificial bees into three groups; namely, employed bees, onlookers and scouts. A bee that is currently exploiting a food source is called an employed bee. A bee waiting in the hive to make a decision in choosing a food source is named as an onlooker. A bee carrying out a random search for a new food source is called a scout. In the ABC algorithm, each solution to the problem under consideration is called a food source and represented by an  $n$  dimensional integer valued vector, whereas the fitness of the solution corresponds to the nectar amount of the associated food resource. Similar to the other swarm intelligence based approaches, the ABC algorithm is an iterative process. It starts with a population of randomly generated solutions or food sources.

The ABC algorithm is usually used for continuous optimization problems. To apply it to discrete combinatorial problems, modifications and adjustments are needed. There are several strategies to implement in each part of the algorithm, and each combination can lead to a different Discrete Artificial Bee Colony (DABC) algorithm. The point is to know which strategy and which combination among them have to be used in order to enhance the performance of the algorithm for the problem at hand [51].

## 5 Proposed HDABC for GPDPTW

The description of the proposed Hybrid Discrete Artificial Bee Colony (HDABC) is given as follows:

### 5.1 Initialization

The algorithm starts with the generation of N initial solutions. These solutions characterize the initial food sources that will be explored by the employed bees. Each food source in the discrete artificial bee colony algorithm is a feasible solution of GPDPTW, which consists of a list of routes. One route is associated with one vehicle. Each route consists of a sequence of request points (pickup and delivery) which are visited by the given vehicle. Figure 1 represents the solutions under the form of food sources where 0 represents the depot and the integer numbers represents pickup or delivery location.

0	3 <sup>+</sup>	2 <sup>+</sup>	2 <sup>-</sup>	1 <sup>+</sup>	3 <sup>-</sup>	1 <sup>-</sup>	0	4 <sup>+</sup>	5 <sup>+</sup>	5 <sup>-</sup>	4 <sup>-</sup>	0
---	----------------	----------------	----------------	----------------	----------------	----------------	---	----------------	----------------	----------------	----------------	---

Figure1: Encoding solution.

According to our method, firstly, a distributed initial population is generated. The method used for achieving initial solutions was set up in such a way that it led to achieve better quality solutions than random selection. In this paper, the initial population is created as follows; a random pickup point is selected as initial of the route, then the next requests consecutively are added to the route to ensure the constraints are satisfied and to get a feasible solution. The algorithm of initialization is defined as follows:

### 5.2 Employed bee phase

In the basic artificial bee colony algorithm, every employed bee determines a food source in the neighborhood of its currently associated food source and evaluates its nectar fitness. We know that each employed bee  $x_{ij}$  generates a new food source  $\hat{x}_{ij}$  in the neighborhood of its present position as follows:  $\hat{x}_{ij} = x_{ij} + \varphi_{ij}(x_{ij} - x_{kj})$   $k = \text{int}(\text{rand FN}) + 1$  where  $\varphi_{ij} = (\text{rand} - 0.5) \times 2$ ,  $\phi_{ij}$  is a uniformly distributed real random number within the range  $[-1, 1]$ , FN is the number of food sources,  $i \in \{1, 2, \dots, FN\}$ ,  $k \in \{1, 2, \dots, FN\}$  and  $k \neq i$ ,  $j \in \{1, 2, \dots, FN\}$  are randomly chosen indexes. But this method cannot be applied to a hybrid discrete artificial bee colony. In this paper, we will propose hybrid neighborhood strategies for the HDABC algorithm to solve the GPDPTW problems. The details of

the hybrid neighborhood strategies are presented in Section 5.5.

Hybrid neighborhood strategies are used to obtain a new solution  $\hat{x}$  from the current solution  $x$  of the HDABC meta-heuristic. Each method for the generation of neighboring food sources may have different performance during the evolution process. The set of pre-selected operators is determined by experimental testing.

As for the selection, a new food source is always accepted if it is better than the current food source. The employed bee exploits the better solution.

### 5.3 Onlookers bee phase

After all employed bees complete their search, they come back to the hive and share their information about the nectar amount of their food sources with the onlookers waiting there; so the quality of the solutions are evaluated.

In this paper, a binary tournament is applied to choose some foods sources by onlookers. The term “binary tournament” refers to the size of two in a tournament, which is the simplest form of tournament selection [52]. Binary tournament starts by selecting two food sources at random. Then, fitness values of these food sources are evaluated. The one having more satisfactory fitness is then chosen. One advantage of the tournament selection is its ability to handle minimization problems without any structural changes. So, the onlookers play the role as an objective function to evaluate generated solutions. Obviously, when the fitness of the food source decreases, the probability with the preferred source by a looker bee decreases proportionally. The onlooker bee produces a new food source by the hybrid neighborhood strategies method presented in section 5.5, the same as the employed bee does. Then, the new source will be evaluated and compared to the primary food solution. If the new source has a better nectar amount than the primary food solution, the new source will be accepted.

### 5.4 Scout bee phase

In the standard ABC algorithm, if a solution does not improve for a predetermined number of trails “limit”, then this food source is abandoned by its employed bee and then the employed bee becomes a scout. The scout produces a food source randomly in the search scope. But this new solution cannot carry better information for the population. In Pan et al [53] advise that the scout generates a food source by performing several insert operators to the best food source in the population. In this paper, we will use the Insertion Inter-route operator to generate a new solution.

### 5.5 Hybrid neighborhood strategies

In this paper, to enrich the neighborhood structure and diversify the population, four neighboring approaches based on the Insertion Inter-route, Swap Inert-route, Swap Intra-route and Move operator are separately

utilized to generate neighboring food sources for the employed bees and onlooker bees. On the whole, we expect that the chosen neighborhood strategies can perform distinct advantages; therefore, they can be effectively combined to solve different instances of green pickup and delivery problem with time windows. The applications of these neighborhoods are as follows:

- 1 : Apply Swap Intra-route to a food source.
- 2 : Apply Move to a food source.
- 3 : Apply Insertion Inter-route to a food source.
- 4 : Apply Swap Inter-route to a food source.

Each operator for the generation of neighboring food sources may have different performances during the evolution process. Therefore, we believe hybrid neighborhood strategies can perfectly be solvable to the green pickup and delivery problem with time windows. Based on the above considerations, we proposed a new method called Hybrid Discrete Artificial Bee Colony (HDABC), the primary idea, is that at each generation, a new bee colony is created. The detail of each operator is as follows:

#### Swap Intra-route

The role of operator swap intra-route is to improve the quality of a route by changing the order in which request points are visited. One route is selected at random. For each request from that route, we try to find a better location inside the same route. If there is such a place, we move a request to that place which satisfies all constraints for the problem. The swap intra-route operator is shown in Figure 2 (see Appendix).

#### Move

The role of move operator is to find the best position by changing the order in which request points (pickup-delivery) are visited. For each request from that route we move a location inside or beside that route. If there is such a place, we move a request to that place which satisfies all constraints for the problem. The move operator is shown in Figure 3 (see Appendix).

#### Insertion Inter-route

The insertion inter-route moves a pickup-delivery pair from its current route to another route in the solution. They perform the following process for all pickup-delivery pairs in the current solution. An admissible placement is one where both requests (pickup and delivery) satisfy all the constraints of the problem. To reduce the number of routes, the search process should be biased such that it tries to remove the request pairs from the shorter routes and insert them into longer routes which satisfy all constraints for the problem. The insertion inter-route operator is shown in Figure 4 (see Appendix).

#### Swap Inter-route

Swapping randomly requested pairs, i.e., a pickup followed by a delivery node between two different routes.

For each pair, we check whether it can be relocated by exchanging its pickup and delivery positions with the pickup and delivery positions of any other request pair in another route which satisfies all constraints for the problem. The swap inter-route operator is shown in figure 5 (see Appendix).

## 5.6 Proposed HDABC

In this paper, we propose a new method called Hybrid Discrete Artificial Bee Colony (HDABC), the primary idea is to hybridize neighborhood strategies at each generation to create a new bee colony. The above idea is illustrated in figure 6.

In the proposed HDABC, there are four strategies to update the food sources. We present a food source as a route and apply the discrete operations to generate new neighborhood food source for three different bees. The heuristic in section 5.1 was used to initialize the population with certain quality and diversity. Then, the new hybrid neighborhood strategies were used to solve the green pickup and delivery problem with time windows. The procedure of HDABC proposed, is given as follows:

## 6 Case study

### 6.1 Data and parameters setting

A numerical example is used to illustrate the applications of the proposed model and the solution algorithm. The data sets for the problem are derived from the instances created by Li et Lim [9] which are related to the well known Solomon instances. The datasets are available at the following link:

<http://www.sintef.no/Projectweb/TOP/PDPTW>.

The graph consists of one depot and 40 nodes. In each instance, nodes are located in geographical clusters and have a small vehicle capacity and narrow time windows. Instances were generated considering only the first 40 nodes which are represented by a complete graph, and all distances are Euclidean distances satisfying the triangle inequality. The pickup and delivery requests are paired and therefore, the number of nodes in the network, without the vehicle depots, is even. Each node has  $x$  and  $y$  coordinates, a demand  $q_i$ , a time window  $[e_i, l_i]$ , a service time  $s_i$  and the corresponding pickup (succ) or delivery (pred) node. The following table (Table 1) represents an example of instance used for the problem. The CO<sub>2</sub> emission rate (CE) per liter of fuel, the fuel consumption rate for both empty-load ( $\rho_0$ ) and full-load ( $\rho^*$ ) situations are set to 2.61, empty load fuel consumption rate = 0.296 and full load fuel consumption rate = 0.390, respectively referring to a previous case study by [46].

**HDABC Algorithm:**

```

1  Set Popsiz = FN //Colony size = 2*FN
2  Set Max.iter = Maximum number of iterations
3  Set Max.trial = Maximum number of
   improvement trials
4   $E_i = \emptyset$ ;  $i = \langle 1, \dots, n \rangle$ ,  $E_i$  is the set of neighbor
   solution of food source
5  Generate FN food sources using the method
   presented in section 5.1 for initial population
6  Evaluate initial population // Calculate  $f(x_i)$  for
   each food sources
7  Memorize best food source  $x_i$  ;
8  Set iteration = 1
9  For each food source  $i$  do, Set  $trial_i = 0$  end
for
10 do while iteration  $\leq$  Max.iter
11 //*****EMPLOYED BEE PHASE*****
12 For each food source  $x_i$  do
13 Apply hybrid neighborhood strategies
   //Produce a new neighbor solution  $\hat{x}_i$ 
14 Evaluate  $\hat{x}_i$  //Calculate  $f(\hat{x}_i)$ 
15 If (  $f(x_i) > f(\hat{x}_i)$  ) then
16 replace  $x_i$  with  $\hat{x}_i$ 
17 Set  $trial_i = 0$ 
18 Else
19 Set  $trial_i = trial_i + 1$ 
20 EndIf
21 End For
22 //*****ONLOOKER BEE PHASE*****
23 For each onlooker do
24 Select a food source using the binary tournament
   selection method
25 Apply hybrid neighborhood strategies
   //Produce a new neighbor solution  $\hat{x}_i$ 
26  $E_i = E_i \cup \hat{x}_i$ 
27 End For
28 For each food source  $x_i$  and  $E_i \neq \emptyset$  do
29 If (  $f(x_i) > f(\hat{x}_i)$  ) then
30 replace  $x_i$  with  $\hat{x}_i$ 
31 Set  $trial_i = 0$ 
32 Else
33 Set  $trial_i = trial_i + 1$ 
34 End If
35 End For
36 //*****SCOUT BEE PHASE*****
37 Set  $i = \text{index of max}(trial)$  //Find the index that
   has the maximum trial value
38 If (  $trial_i \geq \text{Max.trial}$  ) then
39 replace  $x_i$  with Insertion
   Inter-route operator
40 End If
41 Memorize global best solution
42 end while

```

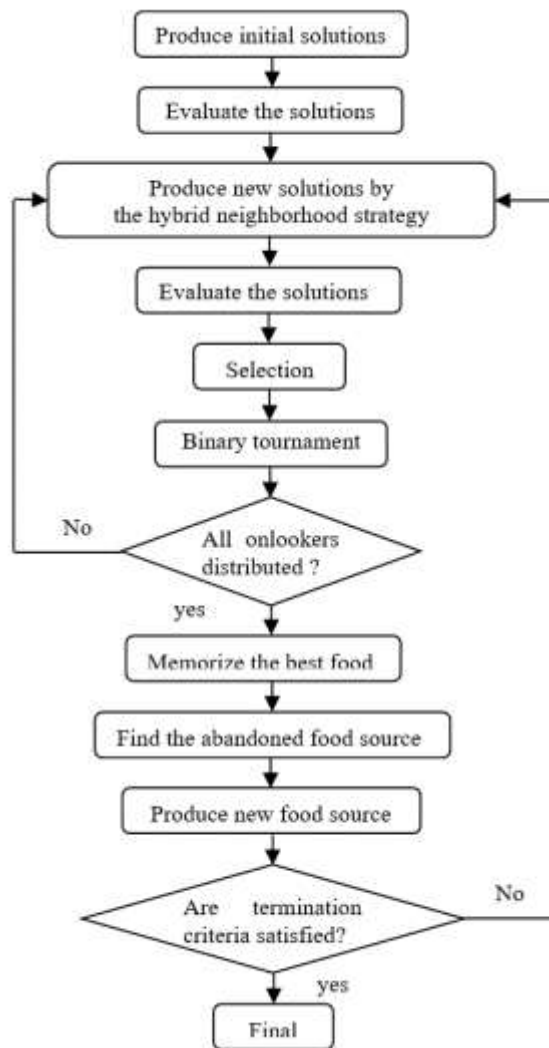


Figure 6: The flow chart of HDABC algorithm.

(Population of food sources), number of iterations, and limit (number of trails). Since an obvious correlation between the optimal settings of these parameters is observed, each one of them is tested individually for deciding on the standard parameter setting. After several preliminary experiments, the size of population is fixed to 100, the number of iterations is fixed as 200 and the limit is 20.

**6.2 Analysis and discussion of results**

In this section, we present a numerical example for the GPDPTW problem and the corresponding optimum solution that is obtained from the proposed HDABC algorithm described in Section 5 which is coded in C++ Builder 2010 software running on a personal computer using Intel Core i5, 2.60 gigahertz, 64-bits processor with 4 gigabyte RAM and Windows 8 OS. In order to analyze the performance of HDABC which is applying from the first time to the green pickup and delivery problem with time windows, we use problem instances with 40 nodes.

A robust parameter setting is required for the proposed HDABC algorithm to efficiently perform on different data sets. In order to select best parameter setting, tests are performed on three parameters: FN



Request	x	y	q	e	l	S	succ	Pred
0	35	35	0	0	110	0	0	0
1	41	49	10	55	88	10	0	5
2	35	17	7	59	89	10	0	20
3	55	45	-23	30	87	10	18	0
....								

Table 1: An example instance.

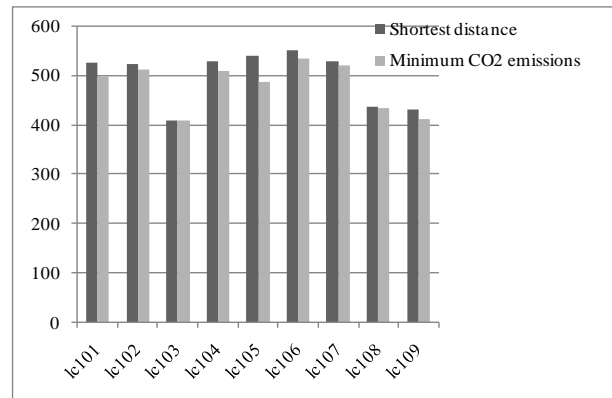
For proper comparison of the computational results, we conduct our experiments upon two cases; case one with the objective of the minimum amount of CO<sub>2</sub> emission and case two with the objective of shortest distance. We run the GPDPTW 10 times, and we compare the average minimum emission of CO<sub>2</sub> and the average distance obtained for the two cases during the ten runs. The results are tabulated in tables 2–4 (see Appendix).

All 40 nodes problem instances are solved to optimality. For our approach, we report a column CO<sub>2</sub> gap (%), which is the relative gap between the amount of CO<sub>2</sub> found in the case of minimum CO<sub>2</sub> emission and the amount of CO<sub>2</sub> found in the case of shortest distance. For example, assuming that the amount of emissions of CO<sub>2</sub> for an instance in case 1 is CO<sub>2min</sub> and the amount of emissions of CO<sub>2</sub> for the same instance in case 2 is CO<sub>2max</sub>, then gap (%) is calculated as  $100 \times (\text{CO}_{2\text{min}} / \text{CO}_{2\text{max}} - 1)$  and a column distance gap(%), which is the relative gap between the total travelled distance found in the case of minimum CO<sub>2</sub> emission and the total travelled distance found in the case of shortest distance. For example, assuming that the total travelled distance for an instance in case 1 is dist<sub>max</sub> and the total travelled distance for the same instance in case 2 is dist<sub>min</sub>, then gap(%) is calculated as  $100 \times (\text{dist}_{\text{max}} / \text{dist}_{\text{min}} - 1)$ . We can notice that comparing the case of minimum CO<sub>2</sub> emission and the case of the shortest distance, the amount of CO<sub>2</sub> could be reduced by 3,43% on average and the average on total distance traveling is increased by 4,60% , we can deduce that the amount of CO<sub>2</sub> emission is not guaranteed for the vehicles along the shortest path. Thus, the distances between customers are shorter and the loading rate and the number of vehicles used in the routes are higher.

Figure 7 shows that lc101, lc104, lc105, lc106 and lc109 have significant effects on the amount of CO<sub>2</sub> emissions. The savings percentages are respectively - 5,52%, 3,67%, 9,81%, 3,36% and 4,68%. In these instances, the second case, in addition to the total load of the vehicle, the number of vehicles used is reduced compared to the first case. Thus, the fuel consumption is reduced.

## 7 Conclusion

This paper proposes and develops a hybrid discrete artificial bee colony approach to solve and discuss the green pickup and delivery problem with time windows (GPDPTW), the recent development in the vehicle routing problem and its variants, which extends the classical vehicle routing problem by considering the coupling, the precedence, the time windows constraints

Figure 7: The difference between the amounts of CO<sub>2</sub> emissions in the two cases.

and the CO<sub>2</sub> emission by vehicles which make the NP-hard combinatorial and optimization problem. The major contribution of this paper is two-fold. First, the GPDPTW model is presented and formulated. Second, a hybrid discrete artificial bee colony for solving the HDABC model is developed which combines a discrete artificial bee colony meta-heuristic with neighborhood operators. The objective is to minimize the amount of fuel consumption to minimize the CO<sub>2</sub> emissions while respecting all constraints. Costs are based on fuel consumption which depends on many factors, such as travel distance, vehicle load and the number of vehicles. The solution approach is evaluated in terms of optimality to reach the best solution on the various test instances. Computational experiments show that the proposed method is effective and efficient and can solve the problem optimally.

Research perspectives in the field highlight the application of the proposed method to accommodate multiple depot and heterogeneous vehicles. The GPDPTW is formulated by assuming only one depot and homogeneous vehicles. However, in many cases, companies may have multiple depots and different kinds of vehicle for pickup and delivery operations.

## 8 References

- [1] B. Sahin, H. Yilmaz, Y. Ust, A. F. Guneri, et B. Gulsun, « An approach for analysing transportation costs and a case study », *European Journal of Operational Research*, vol. 193, n° 1, p. 1-11, févr. 2009. <https://doi.org/10.1016/j.ejor.2007.10.030>.
- [2] N. Christofides, A. Mingozzi, et P. Toth, « *The vehicle routing problem* ». Chichester; New York: Wiley, 1979.
- [3] P. Toth et D. Vigo, « *The Vehicle Routing Problem* », Édition Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2002, p. 1–26. <https://doi.org/10.1137/1.9780898718515>.
- [4] H. Xu, Z.-L. Chen, S. Rajagopal, et S. Arunapuram, « Solving a Practical Pickup and Delivery Problem », *Transportation Science*, vol. 37, n° 3, p. 347-364, août 2003. <https://doi.org/10.1287/trsc.37.3.347.16044>.



- [5] Y. Dumas, J. Desrosiers, et F. Soumis, « The pickup and delivery problem with time windows », *European Journal of Operational Research*, vol. 54, n°1, p. 7-22, sept. 1991. [https://doi.org/10.1016/0377-2217\(91\)90319-Q](https://doi.org/10.1016/0377-2217(91)90319-Q).
- [6] M. Sol et M. W. P. Savelsbergh, « A branch-and-price algorithm for the pickup and delivery problem with time windows », *Memorandum COSOR*, vol. 9422, 1994.
- [7] W. P. Nanry et J. Wesley Barnes, « Solving the pickup and delivery problem with time windows using reactive tabu search », *Transportation Research Part B: Methodological*, vol. 34, n° 2, p. 107-121, févr. 2000. [https://doi.org/10.1016/S0191-2615\(99\)00016-8](https://doi.org/10.1016/S0191-2615(99)00016-8).
- [8] M. M. Solomon, « Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints », *Operations Research*, vol. 35, n° 2, p. 254-265, avr. 1987. <https://doi.org/10.1287/opre.35.2.254>.
- [9] H. Li et A. Lim, « A metaheuristic for the pickup and delivery problem with time windows », in *Proceedings 13th IEEE International Conference on Tools with Artificial Intelligence. ICTAI 2001*, nov. 2001, p. 160-167. <https://doi.org/10.1109/ICTAI.2001.974461>.
- [10] H. C. Lau et Z. Liang, « Pickup and delivery with time windows: algorithms and test case generation », in *Proceedings 13th IEEE International Conference on Tools with Artificial Intelligence. ICTAI 2001*, nov. 2001, p. 333-340. <https://doi.org/10.1109/ICTAI.2001.974481>.
- [11] H. Lim, A. Lim, et B. Rodrigues, « Solving the pickup and delivery problem with time windows using "squeaky wheel" optimization with local search », 2002, p. 2335-2344.
- [12] G. Pankratz, « A Grouping Genetic Algorithm for the Pickup and Delivery Problem with Time Windows », *OR Spectrum*, vol. 27, n° 1, p. 21-41, janv. 2005. <https://doi.org/10.1007/s00291-004-0173-7>.
- [13] G. Ding, L. Li, et Y. Ju, « Multi-strategy grouping genetic algorithm for the pickup and delivery problem with time windows », in *Proceedings of the first ACM/SIGEVO Summit on Genetic and Evolutionary Computation - GEC '09*, Shanghai, China, 2009, p. 97. <https://doi.org/10.1145/1543834.1543849>.
- [14] Q. Lu et M. M. Dessouky, « A new insertion-based construction heuristic for solving the pickup and delivery problem with time windows », *European Journal of Operational Research*, vol. 175, n° 2, p. 672-687, déc. 2006. <https://doi.org/10.1016/j.ejor.2005.05.012>.
- [15] R. Bent et P. V. Hentenryck, « A two-stage hybrid algorithm for pickup and delivery vehicle routing problems with time windows », *Computers & Operations Research*, vol. 33, n° 4, p. 875-893, avr. 2006. <https://doi.org/10.1016/j.cor.2004.08.001>.
- [16] U. Derigs et T. Döhmer, « Indirect search for the vehicle routing problem with pickup and delivery and time windows », *OR Spectrum*, vol. 30, n° 1, p. 149-165, janv. 2008. <https://doi.org/10.1007/s00291-006-0072-1>.
- [17] S. Ropke et J.-F. Cordeau, « Branch and Cut and Price for the Pickup and Delivery Problem with Time Windows », *Transportation Science*, vol. 43, n° 3, p. 267-286, juin 2009. <https://doi.org/10.1287/trsc.1090.0272>.
- [18] E. G. Carabetti, S. R. d Souza, M. C. P. Fraga, et P. H. A. Gama, « An Application of the Ant Colony System Metaheuristic to the Vehicle Routing Problem with Pickup and Delivery and Time Windows », in *2010 Eleventh Brazilian Symposium on Neural Networks*, oct. 2010, p. 176-181. <https://doi.org/10.1109/SBRN.2010.38>.
- [19] I. Harbaoui Dridi, R. Kammarti, M. Ksouri, et P. Borne, « Genetic Algorithm for Mulicriteria Optimization of a Multi-Pickup and Delivery Problem with Time Windows », in *INCOM'09 IFAC*, Russia, juin 2009, p. 1521-1526. <https://doi.org/10.12700/APH.12.8.2015.8.9>.
- [20] A. C. McKinnon et M. I. Piecyk, « Measurement of CO2 emissions from road freight transport: A review of UK experience », *Energy Policy*, vol. 37, n° 10, p. 3733-3742, oct. 2009. <https://doi.org/10.1016/j.enpol.2009.07.007>.
- [21] A. Sbihi et R. W. Eglese, « Combinatorial optimization and Green Logistics », *4OR*, vol. 5, n° 2, p. 99-116, juill. 2007. <https://doi.org/10.1007/s10288-007-0047-3>.
- [22] T. Bektaş et G. Laporte, « The Pollution-Routing Problem », *Transportation Research Part B: Methodological*, vol. 45, n° 8, p. 1232-1250, sept. 2011. <https://doi.org/10.1016/j.trb.2011.02.004>.
- [23] A. Palmer, « The development of an integrated routing and carbon dioxide emissions model for goods vehicles », *Ph.D. Dissertation*, School of Management, Cranfield University, nov. 2007.
- [24] İ. Kara, B. Y. Kara, et M. K. Yetis, « Energy Minimizing Vehicle Routing Problem », in *Combinatorial Optimization and Applications*, 2007, p. 62-71. [https://doi.org/10.1007/978-3-540-73556-4\\_9](https://doi.org/10.1007/978-3-540-73556-4_9).
- [25] W. Maden, R. Eglese, et D. Black, « Vehicle routing and scheduling with time-varying data: A case study », *Journal of the Operational Research Society*, vol. 61, n° 3, p. 515-522, mars 2010. <https://doi.org/10.1057/jors.2009.116>.
- [26] K. Fagerholt, « Optimal fleet design in a ship routing problem », *International Transactions in Operational Research*, vol. 6, n° 5, p. 453-464, 1999. <https://doi.org/10.1111/j.1475-3995.1999.tb00167.x>.
- [27] Y. Xiao, Q. Zhao, I. Kaku, et Y. Xu, « Development of a fuel consumption optimization model for the capacitated vehicle routing problem », *Computers & Operations Research*, vol. 39, n° 7, p. 1419-1431, juill. 2012. <https://doi.org/10.1016/j.cor.2011.08.013>.
- [28] Y. Kuo et C.-C. Wang, « Optimizing the VRP by minimizing fuel consumption », *Management of Environmental Quality: An International Journal*, juin 2011. <https://doi.org/10.1108/14777831111136054>.

- [29] S. Zhang, C. K. M. Lee, K. L. Choy, W. Ho, et W. H. Ip, « Design and development of a hybrid artificial bee colony algorithm for the environmental vehicle routing problem », *Transportation Research Part D: Transport and Environment*, vol. 31, p. 85-99, août 2014. <https://doi.org/10.1016/j.trd.2014.05.015>.
- [30] J. Zhang, Y. Zhao, W. Xue, et J. Li, « Vehicle routing problem with fuel consumption and carbon emission », *International Journal of Production Economics*, vol. 170, p. 234-242, déc. 2015. <https://doi.org/10.1016/j.ijpe.2015.09.031>.
- [31] G. Poonthalir et R. Nadarajan, « A Fuel Efficient Green Vehicle Routing Problem with Varying Speed Constraint (F-GVRP) », *Expert Syst. Appl.*, vol. 100, n° C, p. 131–144, juin 2018. <https://doi.org/10.1016/j.eswa.2018.01.052>.
- [32] R. Liu et Z. Jiang, « A constraint relaxation-based algorithm for the load-dependent vehicle routing problem with time windows », *Flexible Services and Manufacturing Journal*, vol. 31, n° 2, p. 331-353, 2019. <https://doi.org/10.1007/s10696-018-9323-0>.
- [33] O. Apaydin et M. T. Gonullu, « Emission control with route optimization in solid waste collection process: A case study », *Sadhana*, vol. 33, n° 2, p. 71-82, avr. 2008. <https://doi.org/10.1007/s12046-008-0007-4>.
- [34] V. Maraš, « Determining Optimal Transport Routes of Inland Waterway Container Ships », *Transportation Research Record*, vol. 2062, n° 1, p. 50-58, janv. 2008. <https://doi.org/10.3141/2062-07>.
- [35] S. Nanthavanij, P. Boonprasurt, W. Jaruphonga, et V. Ammarapala, « Vehicle Routing Problem with Manual Materials Handling: flexible delivery crew-vehicle assignments », Bali, Indonesia, 2008.
- [36] G. Tavares, Z. Zsigraiiova, V. Semiao, et M. da G. Carvalho, « A case study of fuel savings through optimisation of MSW transportation routes », *Management of Environmental Quality: An International Journal*, juin 2008. <https://doi.org/10.1108/14777830810878632>.
- [37] E. Demir, T. Bektaş, et G. Laporte, « An adaptive large neighborhood search heuristic for the Pollution-Routing Problem », *European Journal of Operational Research*, vol. 223, n° 2, p. 346-359, déc. 2012. <https://doi.org/10.1016/j.ejor.2012.06.044>.
- [38] E. Demir, T. Bektaş, et G. Laporte, « The bi-objective Pollution-Routing Problem », *European Journal of Operational Research*, vol. 232, n° 3, p. 464-478, févr. 2014. <https://doi.org/10.1016/j.ejor.2013.08.002>.
- [39] A. Franceschetti, D. Honhon, T. Van Woensel, T. Bektaş, et G. Laporte, « The time-dependent pollution-routing problem », *Transportation Research Part B: Methodological*, vol. 56, p. 265-293, oct. 2013. <https://doi.org/10.1016/j.trb.2013.08.008>.
- [40] R. Kramer, A. Subramanian, T. Vidal, et L. dos A. F. Cabral, « A matheuristic approach for the Pollution-Routing Problem », *European Journal of Operational Research*, vol. 243, n° 2, p. 523-539, juin 2015. <https://doi.org/10.1016/j.ejor.2014.12.009>.
- [41] Y. Xiao et A. Konak, « A simulating annealing algorithm to solve the green vehicle routing & scheduling problem with hierarchical objectives and weighted tardiness », *Applied Soft Computing*, vol. 34, p. 372-388, sept. 2015. <https://doi.org/10.1016/j.asoc.2015.04.054>.
- [42] P. Györgyi et T. Kis, « A probabilistic approach to pickup and delivery problems with time window uncertainty », *European Journal of Operational Research*, vol. 274, n° 3, p. 909-923, mai 2019. <https://doi.org/10.1016/j.ejor.2018.10.031>.
- [43] D. Goeke, « Granular tabu search for the pickup and delivery problem with time windows and electric vehicles », *European Journal of Operational Research*, vol. 278, n° 3, p. 821-836, nov. 2019. <https://doi.org/10.1016/j.ejor.2019.05.010>.
- [44] L. Dahle, H. Andersson, M. Christiansen, et M. G. Speranza, « The pickup and delivery problem with time windows and occasional drivers », *Computers & Operations Research*, vol. 109, p. 122-133, sept. 2019. <https://doi.org/10.1016/j.cor.2019.04.023>.
- [45] M. W. P. Savelsbergh et M. Sol, « The General Pickup and Delivery Problem », *Transportation Science*, vol. 29, n° 1, p. 17-29, 1995. <https://doi.org/10.1287/trsc.29.1.17>.
- [46] S. Ubeda, F. J. Arcelus, et J. Faulin, « Green logistics at Eroski: A case study », *International Journal of Production Economics*, vol. 131, n° 1, p. 44-51, mai 2011. <https://doi.org/10.1016/j.ijpe.2010.04.041>.
- [47] D. Karaboga, « An idea based on honey bee swarm for numerical optimization », *Technical Report-TR06*, Department of Computer Engineering, Erciyes University, 2005.
- [48] D. Karaboga et B. Basturk, « On the performance of artificial bee colony (ABC) algorithm », *Applied Soft Computing*, vol. 8, n° 1, p. 687-697, janv. 2008. <https://doi.org/10.1016/j.asoc.2007.05.007>.
- [49] D. Karaboga et B. Akay, « A comparative study of Artificial Bee Colony algorithm », *Applied Mathematics and Computation*, vol. 214, n° 1, p. 108-132, août 2009. <https://doi.org/10.1016/j.amc.2009.03.090>.
- [50] N. Karaboga, « A new design method based on artificial bee colony algorithm for digital IIR filters », *Journal of the Franklin Institute*, vol. 346, n° 4, p. 328-348, mai 2009. <https://doi.org/10.1016/j.jfranklin.2008.11.003>.
- [51] W. Y. Szeto, Y. Wu, et S. C. Ho, « An artificial bee colony algorithm for the capacitated vehicle routing problem », *European Journal of Operational Research*, vol. 215, n° 1, p. 126-135, nov. 2011. <https://doi.org/10.1016/j.ejor.2011.06.006>.
- [52] A. Brindle, « Genetic algorithms for function optimization », *Doctoral dissertation*, University of Alberta, Edmonton, Canada, 1980.
- [53] Q.-K. Pan, L. Wang, J.-Q. Li, et J.-H. Duan, « A novel discrete artificial bee colony algorithm for the hybrid flowshop scheduling problem with makespan minimisation », *Omega*, vol. 45, p. 42-56, juin 2014. <https://doi.org/10.1016/j.omega.2013.12.004>.

Appendix

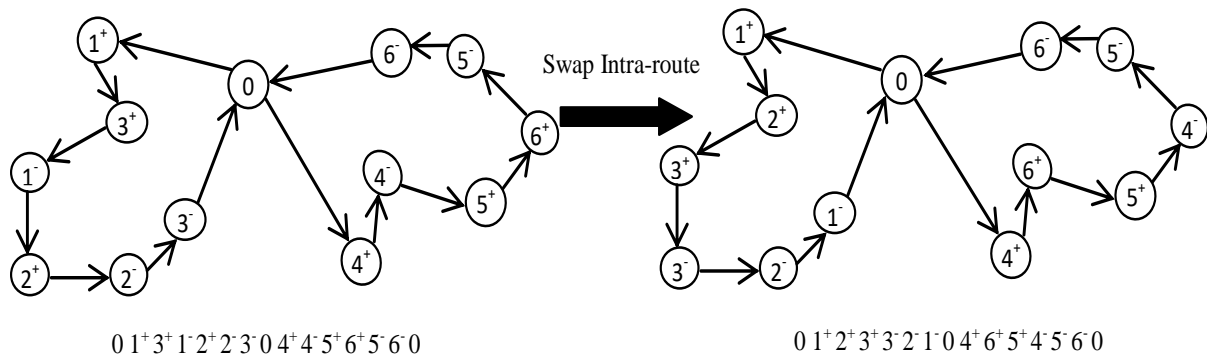


Figure 2: Swap Intra-route Operator.

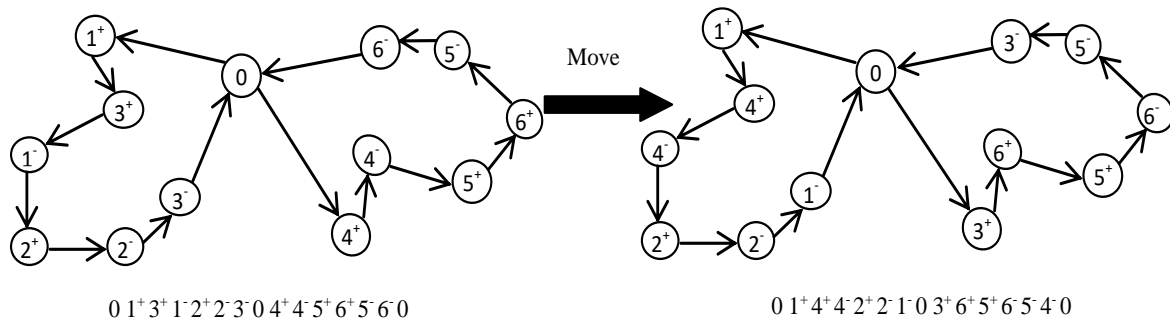


Figure 3: Move Operator.

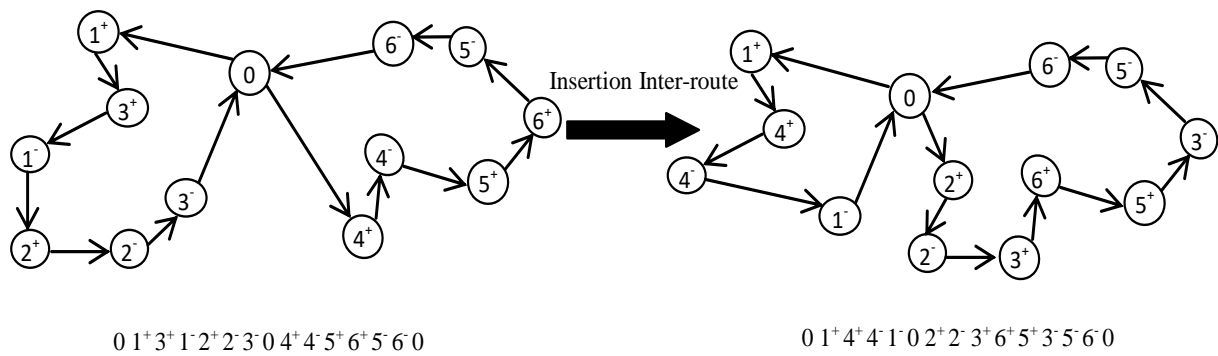


Figure 4: Insertion Inter-route Operator.

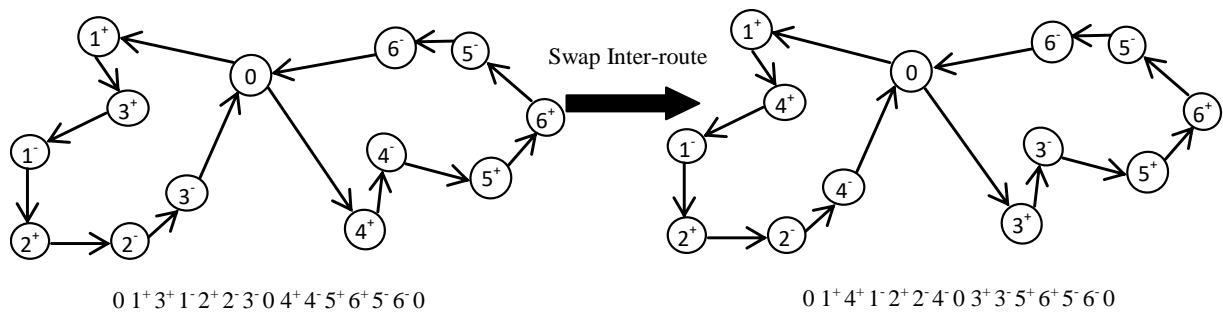


Figure 5: Swap Inter-route Operator.

Instance	Case 1-Minimum CO <sub>2</sub>			Case 2-Shortest distance			CO <sub>2</sub> gap(%)	distance gap(%)
	CO <sub>2</sub>	distance	vehicle	CO <sub>2</sub>	Distance	Vehicle		
lc101	498,31	686,63	4	527,43	643,49	5	-5,52	6,7
lc102	513,39	675,17	3	523,60	668,29	3	-1,95	1,03
lc103	410,91	532,75	2	411,01	523,14	2	-0,02	1,84
lc104	510,30	690,23	3	529,74	636,21	4	-3,67	8,49
lc105	487,41	699,78	4	540,43	627,90	5	-9,81	11,45
lc106	535,09	717,36	4	553,69	694,60	5	-3,36	3,28
lc107	522,01	690,23	4	529,74	675,09	4	-1,46	2,24
lc108	436,10	570,26	6	437,74	559,07	6	-0,37	2
lc109	413,46	563	4	433,77	539,64	5	-4,68	4,33
Avg	480,78	647,27	-	498,57	618,60	-	-3,43	4,6

Table 2: Comparative analysis between the shortest distance and the minimum CO<sub>2</sub> emissions.

Instance	Distance	vehicle	CO <sub>2</sub> emissions	Visited nodes
lc101	643,49	5	527,34	0 20 24 32 31 18 19 8 10 15 16 14 12 6 2 0 0 33 37 30 11 28 9 4 22 1 21 0 0 3 13 17 35 39 23 0 0 38 34 0 0 5 7 25 27 40 29 26 36 0
lc102	668,29	3	523,60	0 24 32 33 25 27 40 29 26 1 7 11 30 38 36 37 34 23 21 0 0 13 18 31 35 8 2 3 10 15 12 5 28 39 9 6 4 0 0 20 17 19 16 14 22 0
lc103	532,14	2	411,01	0 32 33 18 17 19 16 25 40 35 27 30 31 38 37 39 36 34 23 4 2 1 26 29 28 22 21 20 24 0 0 15 14 13 3 8 12 10 5 11 7 9 6 0
lc104	636,21	4	529,74	0 17 19 40 35 5 15 14 11 31 38 0 0 37 39 36 34 4 2 36 34 23 4 2 1 0 0 26 29 28 22 21 20 24 0 0 15 14 13 3 8 12 10 5 11 7 9 6 0
lc105	627,90	5	540,43	0 5 3 19 15 37 39 26 28 22 21 0 0 7 29 30 9 16 14 23 6 2 1 0 0 25 27 40 10 11 34 0 0 20 24 17 13 18 32 33 31 35 38 36 12 0 0 8 4 0
lc106	694,60	5	553,69	0 20 25 27 29 30 38 39 28 9 2 0 0 3 31 40 8 4 36 34 22 0 0 24 18 19 15 14 23 0 0 7 5 13 17 33 32 35 37 10 11 0 0 16 12 26 6 1 21 0
lc107	675,09	4	529,74	0 5 7 3 10 11 9 16 12 28 21 0 0 40 8 2 23 0 0 33 36 0 0 20 24 32 31 25 13 17 18 19 15 27 35 37 29 14 30 39 38 34 6 26 22 4 1 0
lc108	559,07	6	437,74	0 24 31 35 29 38 36 0 0 20 32 40 15 12 39 34 22 0 0 13 17 18 19 16 14 26 23 0 0 25 33 37 30 28 21 0 0 27 2 0 0 5 3 7 8 10 11 9 6 4 1 0
lc109	539,64	5	433,77	0 32 33 31 37 38 34 0 0 20 18 15 22 0 0 29 26 13 12 24 25 40 27 0 0 8 3 10 2 30 7 11 5 9 28 0 0 39 37 34 6 4 23 21 1 0

Table 3: Results concerning shortest distance.

Instance	Distance	vehicle	CO <sub>2</sub> emissions	Visited nodes
lc101	686,63	4	498,31	0 18 29 26 12 6 2 0 0 3 13 17 33 37 38 34 23 0 0 30 28 22 21 5 7 40 35 39 36 0 0 20 24 32 31 25 27 19 8 10 15 11 16 14 9 4 1 0
lc102	675,17	3	513,39	0 31 32 33 17 19 35 38 16 14 36 0 0 20 18 15 22 0 0 29 26 13 12 24 25 40 27 8 3 10 2 30 7 11 5 9 28 39 37 34 6 4 23 21 1 0
lc103	532,75	2	410,91	0 32 33 17 18 16 19 25 40 35 27 30 31 38 37 39 36 34 23 4 2 22 21 1 26 29 28 20 24 0 0 15 14 13 3 8 12 10 5 11 7 9 6 0
lc104	690,23	3	510,30	0 17 32 31 18 13 19 35 37 27 29 15 14 16 12 6 4 26 22 0 0 5 3 7 20 30 11 10 9 28 38 34 21 0 0 8 40 23 2 24 25 33 36 39 1 0
lc105	699,78	4	487,41	0 20 24 17 25 31 35 27 19 29 30 15 16 14 12 23 6 2 1 0 0 40 28 22 34 0 0 32 33 38 36 0 0 5 3 13 18 8 7 10 11 37 39 9 4 26 21 0
lc106	717,36	4	535,09	0 33 37 26 21 18 19 15 14 6 1 0 0 20 25 7 11 9 16 12 2 0 0 24 32 35 23 8 4 5 10 0 0 3 13 17 31 40 27 29 30 38 39 28 36 34 22 0
lc107	690,23	4	522,01	0 17 32 31 18 13 19 35 37 27 29 15 14 16 12 6 4 26 22 0 0 5 3 7 20 30 11 10 9 28 38 34 21 0 0 8 40 23 2 0 0 24 25 33 36 39 1 0
lc108	570,26	6	436,10	0 33 31 35 37 11 9 6 4 0 0 27 40 39 38 36 2 0 0 3 24 29 10 28 21 0 0 25 7 8 30 0 0 5 32 15 12 34 1 0 0 20 13 17 18 19 16 14 26 23 22 0
lc109	563	4	413,46	0 20 24 8 10 29 26 9 2 23 21 7 19 12 4 0 0 25 5 3 30 27 11 6 1 32 37 0 0 33 31 40 35 39 36 38 34 0 0 13 17 18 16 15 14 28 22 0

Table 4: Results with minimum of CO<sub>2</sub> emissions.

