J Intell Manuf (2013) 24:775–789
DOI 10.1007/s10845-012-0627-8

Particle swarm optimization algorithm for a vehicle routing problem with heterogeneous fleet, mixed backhauls, and time windows

Farah Belmecheri · Christian Prins · Farouk Yalaoui · Lionel Amodeo

Received: 29 June 2010 / Accepted: 24 January 2012 / Published online: 21 March 2012

Abstract Today, companies need to collect and to deliver goods from and to their depots and their customers. This problem is described as a Vehicle Routing Problem with Mixed Linehaul and Backhaul customers (VRPMB). The goods delivered from the depot to the customers can be alternated with the goods picked up. Other variants of VRP added to VRPMB are Heterogeneous fleet and Time Windows. This paper studies a complex VRP called HVRPMBTW which concerns a logistic/transport society, a problem rarely studied in literature. In this paper, we propose a Particle Swarm Optimization (PSO) with a local search. This approach has shown its effectiveness on several combinatorial problems. The adaptation of this approach to the problem studied is explained and tested on the benchmarks. The results are compared with our previous methods and they show that in several cases PSO improves the results.

Keywords Vehicle routing problem · Time windows · Mixed backhauls · Pick up and delivery · Heterogeneous fleet · Particle swarm optimization

F. Belmecheri (✉) · C. Prins · F. Yalaoui · L. Amodeo
Institut Charles Delaunay-OSI, University of Technology of Troyes, 12, rue Marie Curie, 10010 Troyes, France
e-mail: farah.belmecheri@utt.fr

C. Prins
e-mail: christian.prins@utt.fr

F. Yalaoui
e-mail: farouk.yalaoui@utt.fr

L. Amodeo
e-mail: lionel.amodeo@utt.fr

F. Belmecheri
TCP Distribution, 116 route d'Auxerre, 10120 Saint André les Vergers, France

Introduction

The Vehicle Routing Problem (VRP) consists in delivering goods from a single central depot to customers with identical vehicles of limited capacity. The problem studied has been identified thanks to our partnership with a French carrier, which picks up goods from the customers for its depot and delivers other goods to the customers with its vehicles. Customers are becoming increasingly exacting about delivery time. By combining all these constraints, the study of this VRP is recent and represents a step toward real problems and more realistic models. Our contribution is to develop a method to solve this industrial case.

In general, the VRP is less frequently present in the industrial domain; most companies have a heterogeneous fleet of vehicles. In our case, several vehicle types are considered; each type is defined by its capacity, a cost per distance unit, and a limited number of vehicles. A lot of customers require a specific time for delivering and picking up the goods. Thus, the VRP with Time Windows is an important problem occurring in many transport systems. We are interested to the VRP with Mixed Backhaul (VRPMB), where some customers need deliveries from a central depot (linehaul customers), while others have some goods to be picked up and centralized at the same depot (backhaul customers). In this paper, we study a VRPMB with Heterogeneous fleet (H), and with Time Windows (TW), giving a problem called HVRPMBTW.

The optimization approach proposed in this paper is the Particle Swarm Optimization (PSO). This has been successfully applied to several vehicle routing problems, but few on HVRPMBTW. This approach has proved its effectiveness with different combinatorial problems and VRP. For this reason, we are interested to adapt it to the problem. Our previous works, based on different strategies of PSO, can be found in Belmecheri et al. (2010a,b).

The remainder of the paper is organized as follows. Section "Literature review" presents some works on different vehicle routing problems, Section "Mathematical model" a model for the HVRPMBTW, and Sect. "Solution methods" the methods proposed. Sections "PSO approach", "ACO approach", "Local search", and "Exact Method" present the adapted PSO, the ACO of Belmecheri et al. (2009a,b), the local search, and the exact method, respectively. Section "Computational results" shows the results obtained for HVRPMBTW, and for specific cases VRP with Time Windows instances. Finally, Sect. is our "Conclusion".

Literature review

The following literature review focuses on the VRP with backhauls (VRPB) and with mixed backhauls, and/or Time Windows, and/or Heterogeneous fleet. Some applications of PSO on different variants of VRP are cited.

In VRPMB, we can observe the works of Süral and Bookbinder (2003). For the constructive heuristics, Casco et al. (1988) used the Clarke and Wright savings algorithm to build vehicle trips. Salhi and Nagy (1999) developed a similar insertion procedures. The works of Liu and Chung (2009) proposed a heuristic called variable neighborhood tabu search (VNTS) to solve the VRPB and VRPB Inventory (VRPBI). Landrieu et al. (2001) used the tabu search also to solve a single vehicle with time windows and backhauls.

For HVRP research, we have the tabu search heuristic of Gendreau et al. (1999), and Wassan and Osman (2002). In Prins (2009), Prins developed two memetic algorithms. In 2009, Yazgi Tütüncü described a decision support system (DSS) based on a Greedy Randomized Adaptive Memory Programming Search (GRAMPS). The classical VRP, the VRPMB, and the VRP with Clustered backhauls (VRPCB) are solved using an interactive environment. Tests on benchmarks from literature have proved the effectiveness of the proposed approach. In Yazgi Tütüncü (2010), the author extended these results to Heterogeneous Fixed Fleets, with and without backhauls (HFFVRP, HFFVRPB). Here again, a few manual corrections via a graphical user interface can quickly improve solution quality and robustness. Rieck and Zimmermann (2009) and Rieck et al. (2007) investigated a closely related problem, where each customer needed one delivery and one pickup (VRPSDP in the classification of Parragh et al. (2008)). Their problem is complicated by a heterogeneous fleet, time windows and the possibility for each vehicle to perform one or two trips.

Ant Colony Optimization (ACO) algorithms are much used in combinatorial problem as the works of Yalaoui et al. (2009). These algorithms are also available for VRP problems like the VRP with Time Windows (VRPTW), see for instance Ellabib et al. (2007). For the VRPTW with backhauls see Reimann and Ulrich (2006). The originality of this approach is that it uses the probabilities based on specific formulas to solve the VRPTW. We developed in our previous works several versions of ACO to solve the HVRPMBTW problem (Belmecheri et al. 2009a,b).

The particle swarm optimization is a search method first proposed by Kennedy and Eberhart (1995); Kennedy and Eberhart (1997) which is inspired by the social behavior of natural organisms such as bees (Clerc 2005), birds flocking and fish schooling to find a place with enough food. Their behavior consists of random movements without any central control (Talbi 2009). This approach is based on a population when agents, called particles, change their position and speed (velocity) in the multidimensional search space of the problem according to the experiments studied.

Although the PSO has developed initially for continuous optimization problems, there have been some works focusing on discrete problems (Clerc 2000; Kennedy and Eberhart 1997). The Traveling Salesman Problem (TSP) considers a set of interconnected cities with symmetric distances between two points. The problem is to find the shortest path for visiting all the cities passing only once at each point and returning to the initial city. There are many other variations of the problem, such as asymmetric distances between cities.

We can find some PSO approaches to solve TSP. Machado and Lopes (2005) have developed a model based on PSO using the characteristics of Genetic Algorithm (GA) (Goldberg and David 1989) and Fast Local Search (FLS) (Voudouris and Tsang 1999). In 2007, Shi et al. used a PSO for TSP using the concept of Wang et al. (2003). The PSO on VRP is still rare. We can find the works of Kim and Son (2010), and Vahdani et al. (2010). The authors Jiang et al. (2009) solved the VRP with Time Windows. In 2009a, Ai and Kachitvichyanukul developed PSO to solve CVRP (Capacitated Vehicle Routing Problem: VRP with fleet fixed of homogeneous capacity of vehicles). Their approach is based on two Solution Representations (SR-1, SR-2). The first is based on Ai and Kachitvichyanukul (2007) with a local search to improve solutions, the second SR-2 is the extension of the first, where some local searches are applied to further improve the results. Their approaches are based on the classical PSO using real values (vectors of positions and velocities). Chen et al. (2006) used integer values of the PSO vectors to solve the CVRP using PSO combined with Simulated Annealing (SA).

On VRP with simultaneous Pick up and Delivery, we can find the works of Ai and Kachitvichyanukul (2009b). In their definition, it is a problem where the goods need to be delivered from a depot to customers and other goods need to be collected for delivery to a depot. The authors proposed a PSO where the coding/encoding process is based on real values. They considered that a particle has a dimensional vector equal to $(n+2R)$. The first n values represent the positions of customers (*prior-*

ity), whereas the $2R$ values concern the positions of vehicles. The authors tested their results using the best known solutions in literature, and they proved that PSO results are very close or better than the previous solutions (Dethloff 2001; Tang Montané and Galvao 2006; Salhi and Nagy 1999).

As for the VRP with Time Windows (VRPTW), Dong et al. (2009) have solved an original problem using the time windows. We can find also the works of Zhu et al. (2006). They have used a simple PSO, whereas Lin (2008) used integer values of vectors positions and velocities including different strategies such as modulo operator on an original PSO.

Xu et al. (2008) have developed a PSO for solving the Open Vehicle Routing Problem (OVRP). The latter creates the routes for vehicles starting at the depot to deliver the goods for customers in using the Hamiltonian path (VRP: Hamiltonian cycle). This means that if the vehicle returns to a depot, it has to use exactly the same route in reverse order for the collection of goods such as a school bus. Their PSO was tested with PSO of Shi et al. (2007), and they improved two out of five total instances.

As we have seen before, several PSO approaches exist to solve different VRP. The approaches are based in the first instance on a coding process: integer, real, binary values. The existing different strategies allow the authors to expand the research space and to improve some of the different developed methods.

Mathematical model

This section records the model developed in our previous works (Belmecheri et al. 2009b). This model for the HVRPMBTW generalizes the one given by Parragh et al. (2008) for the VRPMB. The following symbols are used:

Data

n	number of customers (nodes)
$0, n+1$	start and end depot node
V	node set, $V = \{0, 1, 2, \ldots, n, n+1\}$
Dc	subset of delivery (linehaul) customers
Pc	subset of pickup (backhaul) customers
A	arc set $A = \{(i, j) : i, j \in V, i \neq n+1, j \neq 0, i \neq j\}$
a_i	earliest time to begin service at node i
b_i	latest time to begin service at node i
s_i	service time for customer i
d_i	amount delivered to customer $i \in Dc$, $d_i = 0$ if $i \in Pc \cup \{0\} \cup \{n+1\}$
p_i	amount picked up at customer $i \in Pc$, $p_i = 0$ if $i \in Dc \cup \{0\} \cup \{n+1\}$
t_{ij}	travel time from customer i to customer j
c_{ij}	cost of arc (i, j)
K	set of vehicles (already purchased)
Q_k	capacity of vehicle k
z_k	variable cost per distance of vehicle k
M	a large positive constant.

Variables

$T_i^k \geq 0$	start of service of vehicle k at customer i
$L_i^k \geq 0$	load of vehicle k when leaving customer i
$x_{ij}^k \in \{0, 1\}$	$x_{ij}^k = 1$ if and only if arc (i, j) is traveled by vehicle k.

Note that the depot is represented by two nodes 0 for the origin of routes and $n+1$ for their destinations. We consider a complete directed graph from which no used arcs are removed: loops, incoming arcs to node 0, and outgoing arcs from node $n+1$. Arc (i, j) models the shortest path with cost c_{ij} and duration t_{ij} in the network under consideration. Undirected graphs can be handled by setting $c_{ij} = c_{ji}$ and $t_{ij} = t_{ji}$ for each arc (i, j). Note that the distance between two nodes is Euclidean. The mathematical model is as follows:

$$\min \sum_{k \in K} \sum_{(i,j) \in A} c_{ij} x_{ij}^k z_k \tag{1}$$

Subject to:

$$\sum_{k \in K} \sum_{j:(i,j) \in A} x_{ij}^k = 1, \quad \forall i \in Dc \cup Pc \tag{2}$$

$$\sum_{i:(0,i) \in A} x_{0i}^k = 1, \quad \forall k \in K \tag{3}$$

$$\sum_{i:(i,n+1) \in A} x_{in+1}^k = 1, \quad \forall k \in K \tag{4}$$

$$\sum_{j:(j,i) \in A} x_{ji}^k = \sum_{j:(i,j) \in A} x_{ij}^k, \quad \forall i \in Pc \cup Dc, \forall k \in K \tag{5}$$

$$a_i \leq T_i^k \leq b_i, \quad \forall i \in V, \forall k \in K \tag{6}$$

$$T_i^k + s_i + t_{ij} \leq T_j^k + M(1 - x_{ij}^k), \quad \forall (i, j) \in A, \forall k \in K \tag{7}$$

$$p_i \leq L_i^k \leq Q_k, \quad \forall i \in Pc, \forall k \in K \tag{8}$$

$$0 \leq L_i^k \leq Q_k - d_i, \quad \forall i \in Dc, \forall k \in K \tag{9}$$

$$L_i^k + p_j - d_j \leq L_j^k + M(1 - x_{ij}^k), \quad \forall (i, j) \in A, \forall k \in K \tag{10}$$

$$L_0^k = \sum_{i \in Dc} d_i \sum_{j:(i,j) \in A} x_{ij}^k, \quad \forall k \in K \tag{11}$$

$$x_{ij}^k \in \{0, 1\}, \quad \forall (i, j) \in A, \forall k \in K \tag{12}$$

The objective function (1), to be minimized, is the total distance traveled, weighted by the variable costs of the vehicles used. Constraints (2) ensure that each customer is visited by one vehicle only. Constraints (3) and (4) guarantee that each vehicle begins and completes its route at the depot. Trip connectivity is modelled via constraints (5): if one vehicle k arrives at customer i, it must leave it.

Time windows are respected using constraints (6). Constraints (7) are used to link successive T_i^k along a route: if arc (i, j) is traveled by vehicle k, the constraint becomes $T_i^k + s_i + t_{ij} \leq T_j^k$, otherwise it is equivalent to $T_i^k + s_i + t_{ij} \leq T_j^k + M$ and is trivially verified. The role of these constraints is also to prevent subtours.

Constraints (8) and (9) are used to bound vehicle load after backhaul and linehaul customers, respectively. Constraints (10) are similar to (7) but they concern the successive vehicle loads along a route. The load of each vehicle when leaving the depot must be equal to the total demand of its linehaul customers, as specified in constraints (11). Finally, the binary variables are declared in (12). Note that variables L_i^k are defined in (8) and (9), and variables T_i^k in (6).

Solution methods

PSO approach

This part shows the PSO approach on continuous variables. This method is based on a population where agents, called particles, change their positions in a dimensional D research space of the problem according to the experiments under study. Each particle p has its vector position, its speed called vector velocity, and its best vector position ever visited (vector personal best position— $PBEST_p$). In the swarm of particles, there is only one best quality particle which is referred to as vector global best position ($GBEST$). Thus at each iteration, particles move (guided by $PBEST_p$ and $GBEST$) to a new vector position in the research space and the process is repeated until a stop condition is met, usually after a certain number of iterations.

It is important to note the difference between $PBEST_p$ and $GBEST$. The latter is used by the population (all particles); $PBEST_p$ is used only by an individual particle itself, and it is not used by other particles. When $PBEST_p$ is very close to $GBEST$, the particles cannot explore a large research space of the problem. This can be avoided with the parameters of velocity, $PBEST_p$, and $GBEST$ which as defined in Eqs. (13) and (14). Y_p^t is the next vector position of particle p which is updated using its current vector position Y_p^{t-1} and its next vector velocity V_p^t.

$$V_p^t = w \cdot V_p^{t-1} + C_p \cdot rnd_p() \cdot (PBEST_p - Y_p^{t-1}) \\ + C_g \cdot rnd_g() \cdot (GBEST - Y_p^{t-1}) \quad (13)$$

$$Y_p^t = Y_p^{t-1} + V_p^t \quad (14)$$

V_p^t is the new vector velocity of particle p modulated with an inertia weight w.

$rnd_p()$ and $rnd_g()$ are random values [0, 1] using a uniform distribution. C_p is a parameter to modulate ($PBEST_p - Y_p^{t-1}$) which denotes the distance between the current vector position and personal best position. The latter is calculated in using the best cost which corresponds to this position. The global best position $GBEST$ corresponds to the best cost found by a single particle in the population. C_g is a parameter for modulating the distance between the current vector position and global best position ($GBEST - Y_p^{t-1}$). These parameters are dedicated to an original PSO and they are used to diversify the solutions.

Figures 1, 2 show the behavior of a bird Bi. After some iterations, the particle p (called also bird Bi) can be attracted to its vector position, to $PBEST_p$, or to $GBEST$ using the parameters w, C_p, C_g respectively. Figure 1 shows that p is attracted to $GBEST$ and its vector position (Y_p^{t-1}) while parameters values are: $w = C_p = C_g$. In this case we assume that the vector position (see Fig. 1: in part "sum of vectors" the length of vectors) are as follows: $GBEST > Y_p^{t-1} > PBEST_p$. However, $PBEST_p > Y_p^{t-1} > GBEST$ (see Fig. 2: in part "sum of vectors"), so p is attracted between $PBEST_p$ and $GBEST$. These figures show the importance of choosing the right parameter values to explore a large research space so as to find correct solutions.

Section "PSO on HVRPMBTW" describes the PSO adapted to the discrete problem. Section "Example" shows in detail the PSO steps with an example.

PSO on HVRPMBTW

To demonstrate the implemented PSO, we use the following symbols:

Symbols

T	number of iterations
P	number of particles
D	dimension value, in our case $D = n$ (number of customers)
t	iteration index: $t = \{1, 2, ..T\}$
p	particle index: $p = \{1, 2, ..P\}$
d	dimension index: $d = \{1, 2, ..D\}$
$v_{dp}(t)$	velocity of particle p at dimension d in iteration t
$V_p(t)$	vector velocity of particle p in iteration t: $[v_{1p}(t)\ v_{2p}(t) \ldots v_{Dp}(t)]$
$y_{dp}(t)$	position of particle p at dimension d in iteration t
$Y_p(t)$	vector position of particle p in iteration t: $[y_{1p}(t)\ y_{2p}(t) \ldots y_{Dp}(t)]$
$pbest_{dp}$	personal best position of particle p at dimension d
$PBEST_p$	vector personal best position of particle p: $[pbest_{1p}\ pbest_{2p} \ldots pbest_{Dp}]$
$gbest_d$	global best position of all particles $p = \{1, 2, ..P\}$ at dimension d
$GBEST$	vector global best position of all particles $p = \{1, 2, ..P\}$: $[gbest_1 gbest_2 \ldots gbest_D]$

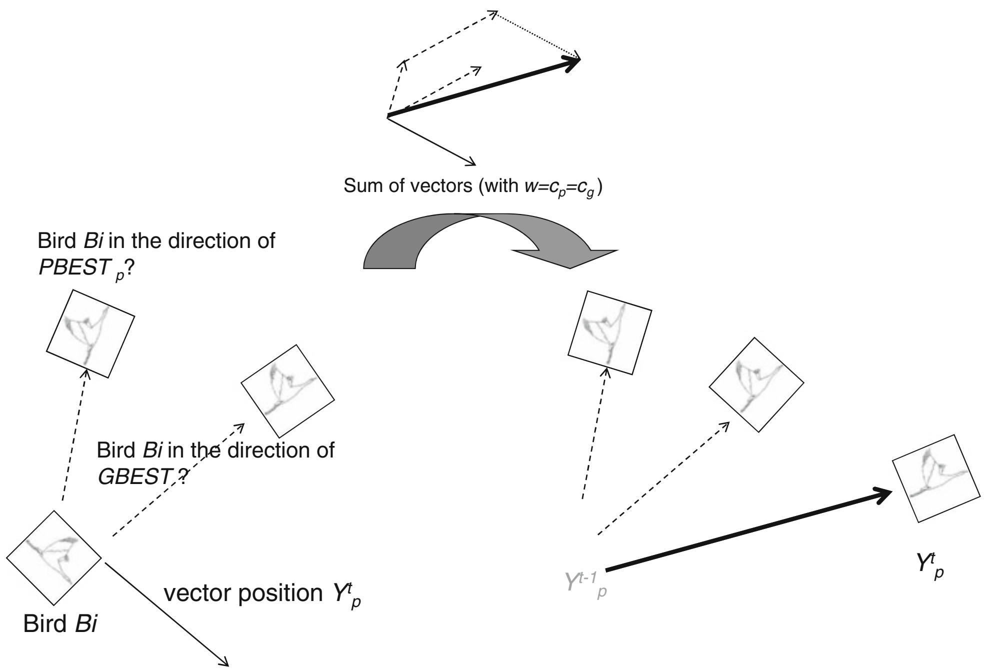

Fig. 1 Movement of bird *Bi* with $w = C_p = C_g$

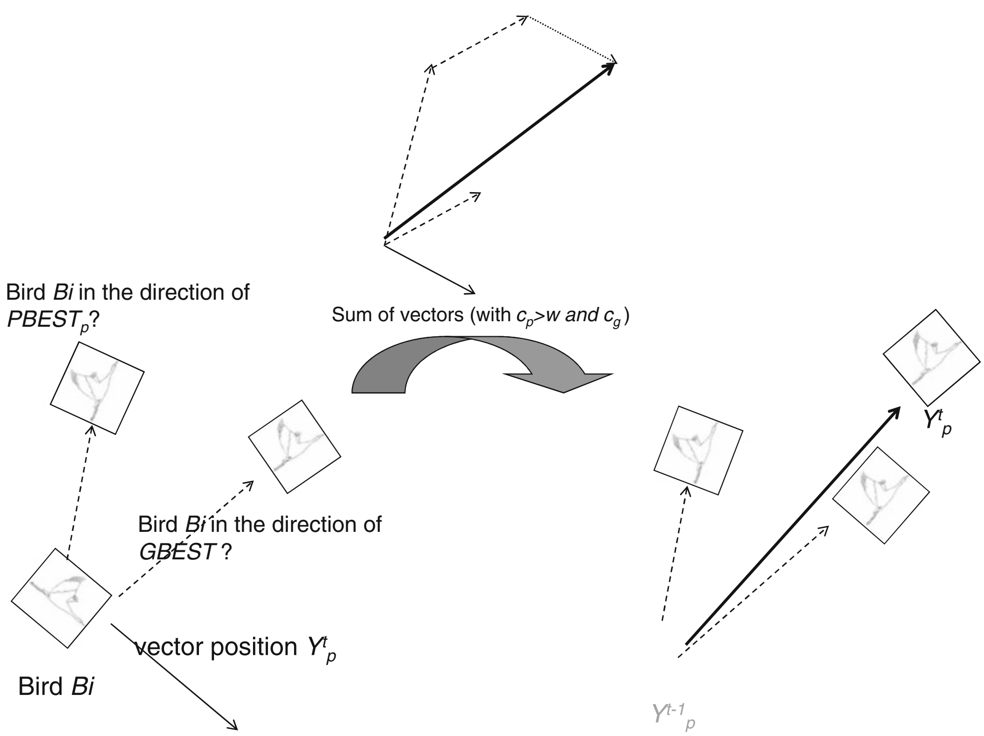

Fig. 2 Movement of bird *Bi* with $C_p > (w$ and $C_g)$

$Sol(Y_p(t))$	solution cost of vector position of particle p in iteration t
$Sol(PBEST_p)$	solution cost of vector personal best position of particle p
$Sol(GBEST)$	solution cost of vector global best position
w	inertia weight on particle p
C_g	parameter on vector global best position
C_p	parameter on vector personal best position
$ymin$	minimum position value, in our case it is equal to 1
$ymax$	maximum position value, in our case it is equal to $3 * D$
Err	percentage error on maximum and minimum position values, in our case it is equal to 5%
$rnd(RA, RB)$	randomly generated number between the numbers RA and RB using a uniform distribution.

In Algorithm 1, steps 1 and 2 initiate the parameters, and the cost solution of $PBEST_p$ and $GBEST$ respectively.

Step 3 creates a **L**ist of customers in **A**scending or **D**er (LAD) according to their order (d_i) and collect (p_i) requirements.

At each iteration t, one of the objectives of PSO is to make a **L**ist of **P**riority of **C**ustomers LPC (steps 8,9). The LAD and LPC are, in the first, the idea of Ai and Kachitvichyanukul (2007, 2009a). To construct the LPC, we use the value $y_{dp}(t)$ of each customer of the LAD. We rank the customers in decreasing order according to the position values $y_{dp}(t)$. This list assigns to each customer his priority to be inserted in the routes. So, at each iteration the priority varies. Steps 5 and 6 initiate the vector position and velocity which are assigned (step 7) to list LAD for initiating the list LPC.

Steps 9, 10, 11 create the solutions then they calculate the costs to update $PBEST_p$ and $GBEST$. Steps 15 and 16 update the vector velocity and position for every particles to create a new LPC for the next iteration $t+1$. Steps 17 and 18 enable position value $y_{dp}(t)$ to be in research space $[1, 3 * D]$. The $rnd(ymin, ymin + ymin.Err)$ is used to avoid that position value of some customers being equal to the same number $ymin$, similarly for $ymax$. Steps 20 to 22 create the solutions, then they update the $PBEST_p$ and $GBEST$. The process of constructing a solution to calculate its cost is explained in Algorithm 2.

As we have seen before, the coding of solution consists of building the vector position $Y_p(t)$ to create the list of priority of customer to be inserted (LPC) in the routes. This latter is used to encode a solution, in order to create the routes R. In this work, the number of routes is equal to the total number of vehicles in an available fleet for each vehicle type. Each route r ($\forall r \in R$) has its capacity Q_r and its variable cost per distance z_r (see Sect. "Example").

Algorithm 1 PSO-HVRPMBTW

1: Initiate the parameters $w, C_g, C_p, ymin, ymax$
2: Initiate $Sol(PBEST_p)$, $Sol(GBEST)$ to great value
3: Create the list LAD
4: **for** $t = 1$ **do**
5: for $d = \{1, 2, ..D\}$: initiate the vector velocity $V_p(t) = [v_{1p}(t)\ v_{2p}(t)..v_{Dp}(t)] = 0, \forall p \in P$
6: for $d = \{1, 2, ..D\}$: generate randomly in range $[D, 2 * D]$ the vector position $Y_p(t) = [y_{1p}(t)\ y_{2p}(t)..y_{Dp}(t)], \forall p \in P$
7: Assign to LAD the positions and velocities $Y_p(t)$, $V_p(t)$ (of steps 5 and 6)
8: Create the list LPC
9: Encode $Y_p(t)$ and calculate the solution cost $Sol(Y_p(t)), \forall p \in P$ (using Algorithm 2)
10: for $d = \{1, 2, ..D\}$: update $PBEST_p = Y_p(t), \forall p \in P$
11: for $d = \{1, 2, ..D\}$: update $GBEST$, if $Sol(GBEST) > Sol(Y_p(t))$ then $GBEST = Y_p(t), \forall p \in P$
12: **end for**
13: **for** $t = 2$ to T **do**
14: **for** $p = 1$ to P **do**
15: $V_p(t) = w.V_p(t-1) + C_p.rnd_p().PBEST_p + C_g.rnd_g().GBEST$
16: $Y_p(t) = Y_p(t-1) + V_p(t)$
17: for $d = \{1, 2, ..D\}$: if $y_{dp}(t) < ymin$ then $y_{dp}(t) = rnd(ymin, ymin + ymin.Err)$ and $v_{dp}(t) = 0$
18: for $d = \{1, 2, ..D\}$: if $y_{dp}(t) > ymax$ then $y_{dp}(t) = rnd(ymax - ymax.Err, ymax)$ and $v_{dp}(t) = 0$
19: Create the list LPC
20: Encode $Y_p(t)$ and calculate the solution cost $Sol(Y_p(t)), \forall p \in P$ (using Algorithm 2)
21: for $d = \{1, 2, ..D\}$: update $PBEST_p$, if $Sol(PBEST_p) > Sol(Y_p(t))$ then $PBEST_p = Y_p(t), \forall p \in P$
22: for $d = \{1, 2, ..D\}$: update $GBEST$, if $Sol(GBEST) > Sol(Y_p(t))$ then $GBEST = Y_p(t), \forall p \in P$
23: **end for**
24: **end for**
25: Store the best solution.

Algorithm 2 Process for encoding the solution

1: For each customer i of list LPC
2: In each route, check capacity and time windows constraints of linehaul/backhaul customers
3: Calculate the $valinsert$ (15) and take the best route r with the best insertion (after customer j)
4: Update route r

$$valinsert = \max_{j \in CR_r} \Big[\Big(\alpha \cdot c_{0i} - \beta \cdot (c_{ij} + c_{if} - c_{fj}) \cdot z_r + (1-\beta) \cdot (l_f^i - l_f) + \gamma \cdot type_i \Big)^{\delta} \Big] \quad \forall r \in R, \forall i \in N_u \tag{15}$$

$valinsert$ value calculates the best cost insertion of customer i in route r. The formula 15 is used in our previous works (defined in Sect. "ACO approach"). The list N_u denotes the list of customers not yet included in the routes. This formula calculates the cost insertion of i between customers j and f of route r.

f denotes the customer visited immediately after customer j, and CR_r is the list of customers of route r. l_f^i is a new arrival time at customer f, if customer i could be feasibly inserted. However, l_f is the arrival time at customer f before insertion of customer i.

Each customer is characterized by $type_i$ variable. This is equal to 0, if customer i is a linehaul, or to 1 if he is a backhaul. Parameter γ modulates the customer type (linehaul or backhaul).

Parameter α modulates the distance between depot 0 and customer i (c_{0i}). Parameter β modulates the distances c_{if}, c_{jf}, c_{ij} between customers i, j, f, and z_r is the variable cost per distance of route r used by one vehicle type k. The parameter δ modulates the impact after inserting customer i in route r.

Example

To explain the phases to code and encode one solution, we have chosen to apply the PSO on an example where the data are presented in Table 1. We assume 5 customers and one depot (0 and 6). Each customer has his geographic coordinates ($Coox$, $Cooy$), and information $type_i$. d_i, p_i, a_i, b_i, s_i (see Sect. "Mathematical model"). We assume also that the distance between nodes (customers and depot) is an Euclidean distance. Concerning the available fleet (see Table 2), we have 2 types of vehicles and 2 vehicles per type. Each type has its capacity and variable cost. So, is it interesting to serve the customers with 1, 2, 3, or 4 vehicles?

We start with 4 empty routes ($R = 4$), and each route has its vehicle (with capacity and variable cost defined).

- r = 1, $Q_1 = 50$, $z_1 = 1.2$,
- r = 2, $Q_2 = 50$, $z_2 = 1.2$,
- r = 3, $Q_3 = 40$, $z_3 = 1$,
- r = 4, $Q_4 = 40$, $z_4 = 1$.

Table 1 Data example

Customer	Coox	Cooy	$type_i$	d_i	p_i	a_i	b_i	s_i
0	40	50	–	0	0	0	3,390	–
1	52	75	0	10	0	311	471	90
2	45	70	0	30	0	213	373	90
3	62	69	0	10	0	1,167	1,327	90
4	60	66	1	0	20	1,261	1,421	90
5	42	65	1	0	10	25	185	90
6	40	50	–	0	0	0	3,390	–

Table 2 Fleet values of the example

Vehicle type	Number of vehicles	Variable cost	Capacity
Vty_a	2	1.2	50
Vty_b	2	1.0	40

To solve this small example, we assume 3 particles and 5 iterations. Figure 3 gives the process for constructing the solution of particle 1 at iteration 1. The dimension D of particles is equal to the number of customers ($n = 5$). First, the LAD list is created, then we assign it to the positions which are uniformly randomly generated in $[D, 2*D] = [5, 10]$ (at iteration $t = 1$). Secondly, the priority list (LPC) to insert the customers in routes is created. This uses the value position of customers from list LAD. For example, the customer 5 of list LAD has his position value $y_{31}(1) = 9.46$ which is the most important. Whereas, customer 4 has $y_{41}(1) = 5.17$ which is the least important. So, in decreasing order of the position values, we obtain the list priority of customers as follows: 5, 1, 3, 2, and finally 4. It means that the insertion of customer 4 depends to the insertion of the first customers (5, 1, 3, 2).

The parameter values (α, β, w, C_p, C_g, $\delta = 1$) are defined in Sect. "Computational results". For each customer, the variable $valinsert$ is calculated for all customers j of each route r and the maximum $valinsert$ (formula 15) is kept. For example, for the first customer (5) of list LPC, we calculate $valinsert$ between the depot 0 and 6 for each route r ($r = \{1, 2, 3, 4\}$). We can see that for $r = 1$ after depot 0, the value is -14.81 (similarly for $r = 2$). For $r = 3$, the value -9.09 (maximum value) corresponds to the insertion of customer 5 after depot 0. The chosen $r = 3$ is updated with customer 5 using a classical heuristic. At the beginning, the load amount of vehicle $r = 3$ is 0 (no customers to deliver). This load is updated with $p_5 = 10$ (quantity picked up at customer 5), so the new load of $r = 3$ is 10. Another updating is applied on time windows of customer 5, where the vehicle 3 arrives to 5 at $T_5^3 = 15.13$ (Euclidean distance 15.13). The last updating concerns the total distance traveled which is calculated as $c_{05} + c_{56} = 30.27$. We continue to calculate the $valinsert$ of customer 1 for each route r. The maximum value -3.69 corresponds to the best cost insertion to route $r = 3$ after customer 5. The process is repeated until all customers of LPC are inserted in routes. Note that customer 2 cannot be inserted into route $r = 3$ because the capacity of the vehicle is not verified.

When the routes are constructed, we calculate the solution cost. We can see that the solution of particle 1 at iteration 1 is equal to 111.39. In referring to Fig. 3, the updating $PBEST_1$ corresponds to initial position (see Algorithm 1—step 10) as follows:
$PBEST_1 = Y_1(1) = [8.39\ 7.32\ 9.46\ 5.17\ 6.29]$
Table 3 summarizes the $PBEST_p$ and $GBEST$ of 3 particles at iteration $t = 1$ and $t = 5$.

*Dimension (**D**)*	*d=1*	*d=2*	*d=3*	*d=4*	*d=5*
*List **LAD***	1	3	5	4	2
Vector of Positions ($Y_1(1)$)	$y_{11}(1)$=8.39	$y_{21}(1)$=7.32	$y_{31}(1)$= 9.46	$Y_{41}(1)$= 5.17	$Y_{51}(1)$= 6.29
Vector of Velocities ($V_1(1)$)	$v_{11}(1)$=0	$v_{21}(1)$=0	$v_{31}(1)$=0	$V_{41}(1)$=0	$v_{51}(1)$=0

LPC		Update the routes
5	*valinsert*=max {-14.81$_{(r=1, j=0)}$, -14.81$_{(r=2, j=0)}$, -9,09 $_{(r=3,j=0)}$, -9,09 $_{(r=4,j=0)}$}= -9,09 $_{(r=3,j=0)}$	r=1→ 0-6 r=2→ 0-6 r=3→ 0-6 r=4→ 0-6
1	*valinsert*=max {-41.51$_{(r=1, j=0)}$, -41.51$_{(r=2, j=0)}$, -3.69 $_{(r=3,j=5)}$, -30.97 $_{(r=4,j=0)}$}= -3.69 $_{(r=3,j=5)}$	r=1→ 0-6 r=2→ 0-6 r=3→ 0- **5** - 6 r=4→ 0-6
3	*valinsert*=max {-1.68$_{(r=1, j=0)}$, -1.68$_{(r=2, j=0)}$, 52.24$_{(r=3,j=1)}$, 9.36$_{(r=4,j=0)}$}= 52.24$_{(r=3,j=1)}$	r=1→ 0-6 r=2→ 0-6 r=3→ 0- 5 **– 1** - 6 r=4→ 0-6
2	*valinsert*=max {-30.62$_{(r=1, j=0)}$, -30.62$_{(r=2, j=0)}$, NF$_{(r=-,j=-)}$, -22.78$_{(r=4,j=0)}$}= -22.78$_{(r=4,j=0)}$ NF: denotes that the insertion of customer 2 in route 3 is infeasible.	r=1→ 0-6 r=2→ 0-6 r=3→ 0- 5 – 1 **– 3** -6 r=4→ 0-6
4	*valinsert*=max {23.69$_{(r=1, j=0)}$, 23.69$_{(r=2, j=0)}$, 81.95$_{(r=3,j=3)}$, 62.59$_{(r=4,j=2)}$}= 81.95$_{(r=3,j=3)}$	r=1→ 0-6 r=2→ 0-6 r=3→ 0- 5 – 1 – 3 -6 r=4→ 0- **2** -6
The final solution		r=1→ 0-6 r=2→ 0-6 r=3→ 0- 5 -1 – 3 – **4** -6 r=4→ 0- 2 -6

Sol($Y_1(1)$) = $\sum_{r=1 \text{ to } 4}$ (traveled distance of route r * Variable cost(r)) = 0*1.2 + 0*1.2 + 70.15*1 + 41.23*1 = 111.39

Fig. 3 Coding/encoding particle. *NP* no feasible, - no value returned

Table 3 Evolution of $PBEST_p$ and $GBEST$ with the example

	$t = 1$			$t = 5$		
p	$Sol(Y_p(1))$	$Sol(PBEST_p)$	$Sol(GBEST)$	$Sol(Y_p(5))$	$Sol(PBEST_p)$	$Sol(GBEST)$
1	111.39	111.39	111.39	110.19	110.19	110.19
2	110.19	110.19	110.19	110.19	110.19	110.19
3	170.71	170.71	110.19	110.19	110.19	110.19

We can observe that the vehicle type used is Vty_b (Table 2), and the best solution called the optimal solution is 110.19 corresponding to the following solution:

– r = 1, $Q_1 = 50$, $z_1 = 1.2$: 0-6
– r = 2, $Q_2 = 50$, $z_2 = 1.2$: 0-6
– r = 3, $Q_3 = 40$, $z_3 = 1$: 0-1-3-4-6
– r = 4, $Q_4 = 40$, $z_4 = 1$: 0-5-2-6

ACO approach

In this section, we summarize the Ant Colony Optimization developed with HVRPMBTW in our previous works Belmecheri et al. (2009a,b). This approach is similar to the PSO approach. For more detail, we use Table 4. This shows the common and different points between both approaches.

A solution is created by inserting the customers of LAD into the routes. The list N_u denotes always the list of customers not yet included in the routes. An ant chooses randomly one unrouted customer to the best route r with the probability calculated by formula (16).

$$P_{if} = \begin{cases} \frac{\eta_i \cdot \tau_{if}}{\sum_{h/\eta_h>0} \eta_h \cdot \tau_{hk}} & \text{if } \eta_i > 0 \\ 0 & \text{otherwise} \end{cases} \quad (16)$$

Table 4 Comparison of PSO and ACO strategies

	PSO	ACO
LAD	Used	Used
Construct a solution	LPC, procedure encode	Probability of customer (formula 16)
Update	Position, velocity	Pheromones deposited on arcs (formulas 18,19)

P_{if} is the probability for choosing a single customer i. He will be included before customer f in current route r. Note, f represents the customer visited immediately after customer i (if his insertion is feasible). τ_{if} denotes the trail of pheromone on an arc (i, f) deposited by an ant a.

$$\eta_i = \max_{j \in CR_r} \Big[\Big(\alpha \cdot c_{0i} - \beta \cdot (c_{ij} + c_{if} - c_{fj}) \cdot z_r + (1-\beta) \cdot (l_f^i - l_f) + \gamma \cdot type_i \Big)^{\delta} \Big] \quad \forall r \in R, \forall i \in N_u \tag{17}$$

η_i is the visibility value, this information guides an ant for the insertion of customer i. It is the same formula used by the PSO approach. The decision to insert customer i between customers j and f is based on his attractiveness η_i. Customer f denotes always the customer visited immediately after j. CR_r is the list of customers of route r. l_f^j is a new arrival time at customer f. However, l_f is the arrival time at customer f before inserting customer i. The parameters α, β, γ are presented in formula 15. When the solution is created, we update the deposited pheromones on arcs:

- **Local updating**
  Every time an ant proposes a solution, an update of pheromone trail is applied before the next ant as follows:

$$\tau_{if} = (1-\rho) \cdot \tau_{if} + \rho \cdot \tau_0 \quad \text{if } (i, j) \in S_a \tag{18}$$

S_a denotes the solution of ant a (set of arcs chosen by an ant a). τ_0 is the initial value of pheromone deposited on arcs, which is equal to 1. The parameter ρ is called the trail persistence: it regulates the reduction of pheromone on the arcs. The aim of this local updating is to avoid that several ants choose the same arcs.

- **Global updating**
  Once all ants have proposed their solutions, the amount of pheromone deposited on each arc (i, k) is updated as follows:

$$\tau_{ik} = \begin{cases} (1-\rho) \cdot \tau_{if} + \rho \cdot (1/L^*) & \text{if } (i, f) \in S^* \\ (1-\rho) \cdot \tau_{if} & \text{otherwise} \end{cases} \tag{19}$$

L^* is the current best solution, and S^* is the set of arcs visited by the best ant a^*. This update favors the arcs of ant a^* to be chosen for the next population. This process allows the ants to obtain an approximate solution at S^*.

Local search

To improve the PSO and ACO methods, a Local Search (LS) is applied to each solution. It could be interesting to chose a stopping criterion such as the no improvement of the solution. But the running time could be very important. So, we assume that the stopping criterion of LS is the fixed running time defined for each instance size.

Note that, before each move, we check if the capacity and time windows constraints are verified. Figure 4 shows schematically the two in three moves used:

- 2-opt on each route,
- relocate one customer u from route T_2 to T_1,
- exchange two customers u, v of routes T_2 to T_1 respectively.

Exact method

We have developed a complete enumeration to demonstrate that PSO and ACO approaches obtain good solutions to small

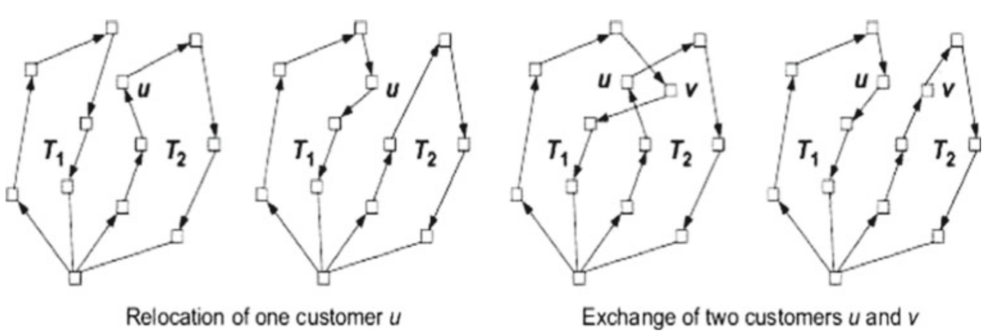

Fig. 4 Relocate and exchange moves (Prins 2009)

problems. This method built the solutions from a single vehicle to all available vehicles. The customers are served by vehicles in a defined sequence. The results on small problems show that the approaches adopted return high quality solutions, even optimal in many cases.

Computational results

Benchmark problem instances

To test the approaches on HVRPMBTW problem, we modify the original benchmarks VRPTW (Solomon 1987). We use six groups: R1, R2, C1, C2, RC1, and RC2. Groups R1 and R2 have the geographic coordinates of customers randomly generated according to a uniform distribution. While groups C1 and C2 have the customers clustered. Groups RC1 and RC2 are characterized by a semi-clustered customer distribution.

In each instance type (R1…RC2), a set of examples is proposed by Solomon (1987). Each example is defined by the localization of customers, quantities delivered (picked up) to (from) customers, and time windows $[a_i, b_i]$, service time s_i for every customer i.

The Solomon instances are modified to test our algorithm. In referring to a real case of a logistic/transport company, the number of linehaul and backhaul customers are considered as follows: the first customers are 80% linehaul and the remaider 20% backhaul. Three size instances are tested $n = \{5, 25, 100\}$, which are taken from the original instance (100 cutomers), starting with the first customer. To compare the approaches with the Exact Method (EM), we use 5 customers, the 4 first customers are linehaul (5*80%=4) and the last customer is backhaul. The second modification proposes a heterogeneous fixed fleet for $n = \{5\}$ (see Table 2), and for $n = \{25, 100\}$ (see Table 5).

Parameter setting

PSO uses parameters C_p, C_g, w. We apply several tests by varying parameter values to obtain the right settings. The variation of values are based on our previous works (Belmecheri et al. 2010a,b). The C_p is fixed to 0.5, then C_g and w are varied in $\{0.01, 0.2, 0.5, 1, 1.5\}$. We take the best values $C_p = 0.5$, $C_g = 1.5$, $w = 0.1$. The dimensional value is fixed to $D = n$, and the number of particles is fixed as $P = \lceil n/10 \rceil$ for $n > 20$.

For the parameter values used in formulas (15, 17), they are the same as ACO of Belmecheri et al. (2009a,b). Parameters $\alpha = 0.01$, $\beta = 0.95$, and $\gamma = 13$. Parameter δ is varied in $[1, 2]$ for PSO. This is applied twice using parameter values $\delta = 1$ and $\delta = 2$. Then the best δ value corresponding to the best obtained solution is stored. Whereas for ACO, δ is equal to 2 because the probability values calculated have to be positive. The number of ants is also equal to $\lceil n/10 \rceil$ for $n > 20$, and $\rho = 0.95$. The parameter values assigned in both methods are the same. The aim is to compare their performance in the same conditions.

In this work, the technical idea of the following stopping criteria are inspired by the previous works (Reimann and Ulrich 2006; Yazgi Tütüncü 2010). We assume that the stopping criteria of both methods is the running time fixed for size 100 (Table 6). On size 5, it is the number of ants, particles, and iterations for ACO and PSO. But in LS it is the running time fixed. For the large size, the running time is fixed to 5 min for ACO and PSO.

Table 5 Fixed heterogeneous fleet

Vehicle type	Number of vehicles	Variable cost	Capacity
a	5	2	600
b	5	1.8	400
c	8	1.6	200
d	6	1.5	100
e	6	1.0	50

Comparison results of HVRPMBTW

To show that the approaches adopted give high quality solutions, we compare the results obtained by the Exact Method (EM). The gaps used in Table 7 are:

$$GAP_{\text{PSO}-\text{EM}} = \frac{(\text{Cost of PSO}) - (\text{Cost of EM})}{(\text{Cost of EM})} \times 100\% \tag{20}$$

$$GAP_{\text{ACO}-\text{EM}} = \frac{(\text{Cost of ACO}) - (\text{Cost of EM})}{(\text{Cost of EM})} \times 100\% \tag{21}$$

We record that the solution is the total distance traveled with variable cost.

In Table 7, for R1, R2, C1, C2 (40 instances), PSO obtain 39 optimal solutions out of 40. We observe that PSO uses more computing time than ACO. For PSO, the process for creating a solution is indirect, thus the processes coding and encoding of particles (solutions) have an important

Table 6 Stopping criteria of HVRPMBTW

size n	5	100 (s)
PSO	3 Particles, 4 iterations	300
ACO	3 Ants, 4 iterations	300
LS	0.05 s	2

Table 7 Average solution and gap of 5 customers of HVRPMBTW (R1, R2, C1, C2)

	Nbr	ACO		PSO		EM		%	
		Cost	cpu	Cost	cpu	Cost	cpu	$GAP_{\text{ACO-EM}}$	$GAP_{\text{PSO-EM}}$
R1-5	12	152.73	0.41	152.73	0.71	152.66	2.27	**0.04**	**0.04**
R2-5	11	145.19	0.41	145.19	0.71	145.19	2.17	**0.00**	**0.00**
C1-5	9	77.76	0.41	77.54	0.71	77.45	2.06	0.29	**0.00**
C2-5	8	110.19	0.41	110.19	0.71	110.19	2.08	**0.00**	**0.00**

Nbr Number of instances, *cpu* running time in "second" (s)
The values in boldface correspond to the improved results

computing time. However for ACO, the solution is constructed directly.

$$GAP_{\text{ACO-PSO}} = \frac{(\text{Cost of ACO}) - (\text{Cost of PSO})}{(\text{Cost of PSO})} \times 100\% \quad (22)$$

$$GAP_{\text{PSO-BKS}} = \frac{(\text{Cost of PSO}) - (\text{Cost of BKS})}{(\text{Cost of BKS})} \times 100\% \quad (23)$$

$$GAP_{\text{ACO-BKS}} = \frac{(\text{Cost of ACO}) - (\text{Cost of BKS})}{(\text{Cost of BKS})} \times 100\% \quad (24)$$

For 100 customers, in Tables 8, 9 and 10, PSO improves ACO from 0.19 to 7.31%. We can observe that the results obtained depend on the type of instance. For example, in Table 8, PSO has a difficulty in instances C1 and C2 than ACO. To create the routes by PSO, we use only formula 15. The latter is used also by ACO (formula 17) adding the pheromone value deposited on arcs (τ_{if}). ACO considers the cost to insert the customers in vehicles, and it uses pheromones to cluster the customers. Whereas, PSO focuses on cost insertion of customer only, without clustering customers.

Note that out of 56 instances, PSO improves 29 cases which represents 52% of total cases.

Table 8 Comparison of 100 customers of HVRPMBTW (C1, C2)

Instance	ACO	PSO		%
	Cost	Cost	δ	$GAP_{\text{ACO-PSO}}$
C101-100	2,892.29	2,560.02	2	**12.98**
C102-100	2,890.61	2,615.32	1	**10.53**
C103-100	2,502.23	2,405.3	1	**4.03**
C104-100	2,204.23	2,333.95	1	−5.56
C105-100	2,141.82	2,055.9	1	**4.18**
C106-100	2,298.4	2,366.05	2	−2.86
C107-100	2,035.59	1,992.83	1	**2.15**
C108-100	1,963.67	1,938.54	1	**1.30**
C109-100	2,048.73	2,234.79	1	−8.33
Average				**2.05**
C201-100	1,298.17	1,420.62	1	−8.62
C202-100	1,438.4	1,590.2	1	−9.55
C203-100	1,564.12	1,823.84	1	−14.24
C204-100	1,727.46	1,856.26	2	−6.94
C205-100	1,354.02	1,504.98	2	−10.03
C206-100	1,444.86	1,528.31	2	−5.46
C207-100	1,376.64	1,391.91	2	−1.10
C208-100	1,515.57	1,626.96	2	−6.85
Average				−7.85
Average of C1-C2				−2.90

In PSO: frequency for $\delta = 1$ is 10, and $\delta = 2$ is 7
The values in boldface correspond to the improved results

Validation process

Among the most commonly used statistical significance tests applied to data sets are the *T* and *Z*-test. Habitually, a *T*-test is appropriate to small samples (inferior to 30) while a *Z*-test is appropriate to large samples (superior to 30). These tests are often used for comparing two means, which is commonly applied in many cases. Here, we compare the analytical results obtained by two different methods ACO and PSO, in order to confirm if both methods provide similar analytical results or not.

We use an Excel Microsoft to apply this analysis. Here, we compare the ACO and PSO results of 100 customers of HVRPMBTW (56 observations : from C101-100 to RC207-100). The aim consists to confirm that PSO results improve ACO results. Before applying a *Z*-test, we need to know if the PSO and ACO samples are normally distributed, that is why the Kolmogorov–Smirnov (K–S) test is used. The Table 11 summarizes the necessary calculations for K–S and *Z*-test.

For K–S test, the Table 12 presents the statistics estimated on the input data and computed using the estimated parameters of the Normal distribution. So the first hypothesis H_0 and the alternative hypothesis H_a are presented as follows:

$\mathbf{H_0}$: The sample follows a Normal distribution.
$\mathbf{H_a}$: The sample does not follow a Normal distribution.

The used significance level (*alpha*) is equal to 5%. After applying the K-S test, the results are presented in Table 13. The risk to reject H_0 of ACO and PSO samples while they are true are 63.06 and 62.20% respectively. In concluding, we can see that the ACO and PSO samples are from a Normal distribution.

Table 9 Comparison on 100 customers of HVRPMBTW(R1, R2)

Instance	ACO	PSO		%
	Cost	Cost	δ	$GAP_{\text{ACO}-\text{PSO}}$
R101-100	2,854.82	2,632.13	1	**8.46**
R102-100	2,671.04	2,375.41	1	**12.45**
R103-100	2,124.03	2,006.8	1	**5.84**
R104-100	1,863.78	1,853.21	1	**0.57**
R105-100	2,379.08	2,253.42	1	**5.58**
R106-100	2,063.28	2,031.19	1	**1.58**
R107-100	1,905.43	1,928.9	1	−1.22
R108-100	1,918.02	1,877.52	1	**2.16**
R109-100	2,133.42	2,001.56	1	**6.59**
R110-100	1,979.53	1,983.98	1	−0.22
R111-100	1,889.19	1,896.7	1	−0.40
R112-100	1,936.93	1,895.77	1	**2.17**
Average				**3.63**
R201-100	2,012.12	1,990.47	1	**1.09**
R202-100	1,922.72	1,932.74	1	−0.52
R203-100	1,736.2	1,745.37	1	−0.53
R204-100	1,584.13	1,522.5	2	**4.05**
R205-100	1,848.66	1,885.75	2	−1.97
R206-100	1,758.78	1,813.48	2	−3.02
R207-100	1,650.12	1,654.84	2	−0.29
R208-100	1,536.68	1,589.42	2	−3.32
R209-100	1,777.49	1,729.58	2	**2.77**
R210-100	1,793.64	1,754.44	2	**2.23**
R211-100	1,615.85	1,699.39	2	−4.92
Average				−0.40
Average of R1-R2				**1.61**

In PSO: frequency for $\delta = 1$ is 15, and $\delta = 2$ is 8
The values in boldface correspond to the improved results

Table 10 Comparison of 100 customers of HVRPMBTW (RC1, RC2)

Instance	ACO	PSO		%
	Cost	Cost	δ	$GAP_{\text{ACO}-\text{PSO}}$
RC101-100	3,348.18	2,957.49	2	**13.21**
RC102-100	2,867.89	2,464.51	2	**16.37**
RC103-100	2,432.26	2,426.88	2	**0.22**
RC104-100	2,400.72	2,244.58	1	**6.96**
RC105-100	3,005.21	2,711.05	1	**10.85**
RC106-100	2,706.68	2,495.57	1	**8.46**
RC107-100	2,414.86	2,420.42	1	−0.23
RC108-100	2,444.32	2,381.45	1	**2.64**
Average				**7.31**
RC201-100	2,484.64	2,401.11	1	**3.48**
RC202-100	2,190.48	2,251.39	1	−2.71
RC203-100	1,931.69	2,022.9	1	−4.51
RC204-100	1,750.12	1,827.48	1	−4.23
RC205-100	2,226.16	2,274.91	1	−2.14
RC205-100	2,196.88	2,123.08	1	**3.48**
RC206-100	2,054.41	2,084.5	2	−1.44
RC207-100	2,012.44	1,836.63	2	**9.57**
Average				**0.19**
Average of RC1-RC2				**3.75**

In PSO: frequency for $\delta = 1$ is 11, and $\delta = 2$ is 5
The values in boldface correspond to the improved results

Table 11 Summary statistics

Sum of difference of 56 observations	1,894.39
Mean of difference of 56 observations	33.82
Mean of ACO sample	2,073.54
Mean of PSO sample	2,039.71
SD of ACO sample	460.50
SD of PSO sample	355.74

Table 12 Statistics and parameters to Kolmogorov–Smirnov test

	ACO	PSO
μ	2,073.54	2,039.71
σ	460.50	355.74
Mean	2,073.54	2,039.71
Variance	212,066.58	126,553.97
Skewness (Pearson)	0.57	0.29
Kurtosis (Pearson)	−0,17	−0.59

After confirming that ACO and PSO samples follow the Normal distribution. The *Z*-test could be applied to analyze the results of Sect. "Comparison results of HVRPMBTW". The aim is to demonstrate that ACO results are improved by PSO results, ACO mean has to be superior to PSO mean. For *Z*-test, several parameters are chosen as follows:

$\mathbf{H_0}$: The difference between the means is equal to 0.
$\mathbf{H_a}$: The difference between the means (Mean ACO-Mean PSO) is greater than 0.

Significance level is equal to 5% $\longrightarrow$ 95% confidence interval on the difference between the means. Note that the samples are dependent because we should to prove that first sample is improved by the second one (unilateral test).

In Table 14, we can see that P-value is lower than $alpha$. One should reject the null hypothesis H_0, and accept the alternative hypothesis H_a because the risk to reject H_0 is only 3%. These observations confirm the positive value of the difference (see Table 14) between ACO and PSO means which is equal to 33.82.

To validate more the results, we apply a new *Z*-test with the same parameters (Significance level, …) but on different hypothesis which are:

$\mathbf{H_0}$: The difference between the means is equal to −30.
$\mathbf{H_a}$: The difference between the means (Mean ACO-Mean PSO) is greater than −30.

Table 13 Kolmogorov–Smirnov test

	ACO	PSO
$D_s statistic$	0.098	0.099
p-value	0.631	0.622
$Alpha$	0.05	0.05
Test interpretation	p-value is greater than $alpha$	p-value is greater than $alpha$

Table 14 Z-test for difference between means is 0

Difference	33.82
Df (degree of freedom)	55
z (Observed value)	1.78
z (Critical value)	1.64
p-value	0.03
$alpha$	0.05

Table 15 Z-test for difference between means is −30

Difference	33.82
Df (degree of freedom)	55
z (Observed value)	3.35
z (Critical value)	1.64
p-value	0.0004
$alpha$	0.05

Table 16 Stopping criterion for VRPTW

size n	25 (s)
PSO	150
ACO	150
LS	1

Here, the new H_0 supposes that PSO mean is greater than ACO mean. The results are in Table 15.

We can see that the risk to reject new H_0 is 0.04%, so the new alternative hypothesis H_a is accepted. This new test confirms the first one, where ACO mean is greater than PSO mean.

In concluding, the results of HVRPMBTW obtained by ACO and PSO are analyzed in using the validation process. The PSO results improve the ACO results.

Application on VRPTW

Other tests are applied on specific cases where the constraints of Heterogeneous fleet and mixed backhauls are relaxed for studying only the VRPTW. The interest is for demonstrating that if the problem studied is relaxed, then the approach proposed can return good results. The instances used are the standard benchmark VRPTW of Solomon with size instance $n = \{25\}$ (the first 25 customers of original benchmark). The stopping criteria of the approaches is presented in Table 16.

The results obtained by PSO are compared with ACO, and Best Known Solution [33] (BKS) of the literature. In referring to Table 17, we can observe that PSO results are very close to BKS. Out of 56 instances, the $GAP_{\text{ACO−BKS}}$ is 8.03%, $GAP_{\text{PSO−BKS}}$ is 3.51%. So, for $n = 25$ of VRPTW, PSO improves ACO with 4.12%. The algorithms of BKS are more important than our algorithm. This latter dedicated to a complex problem can return good adequate solutions but the reverse (BKS algorithms) cannot solve the studied problem. The PSO is encouraging because it is designed to tackle heterogeneous fleet and mixed backhauls.

To validate the results, we have applied the validation process on ACO and PSO results obtained on the VRPTW. The hypothesis H_a is accepted because the p value is equal to 0.04 (lower than significance level 0.05). So with this process, we confirm that PSO improves again ACO.

Table 17 Average solution and gap of 25 customers of VRPTW (R1, R2, C1, C2, RC1, RC2)

Instance	Nbr	ACO	PSO	BKS	%		
		Cost	Cost	Cost	$GAP_{\text{ACO−BKS}}$	$GAP_{\text{PSO−BKS}}$	$GAP_{\text{ACO−PSO}}$
R1-25	12	474.09	468.54	463.37	2.31	**1.12**	**1.18**
R2-25	11	390.30	385.86	382.15	2.13	**0.97**	**1.15**
C1-25	9	251.87	216.01	190.59	32.15	**13.34**	**16.60**
C2-25	8	216.15	216.98	214.45	**0.79**	1.18	−0.38
RC1-25	8	376.18	359.87	350.24	6.89	**2.66**	**4.14**
RC2-25	7	330.06	324.75	319.28	3.13	**1.89**	**1.32**
Average:					8.03	**3.51**	**4.12**

Nbr Number of instances
In PSO, in R1 and R2: frequency for $\delta = 1$ is 16, and $\delta = 2$ is 7
In PSO, in C1 and C2: frequency for $\delta = 1$ is 15, and $\delta = 2$ is 2
In PSO, in RC1 and RC2: frequency for $\delta = 1$ is 9, and $\delta = 2$ is 7
The values in boldface correspond to the improved results

Conclusion

This work proposes a Particle Swarm Optimization (PSO) approach with Local Search (LS) to solve the Vehicle Routing Problem with Heterogeneous fleet, mixed backhauls, and Time Windows called HVRPMBTW. This problem has been rarely studied in literature, but it has been raised by a logistic/transport company. When the PSO has been implemented, we have compared the results with an Exact Method and Best Known Solutions (BKS) of literature on specific cases (VRPTW). In HVRPMBTW, on small problems, we have observed that PSO gives high quality solutions, and in several instances the optimal solutions. For large problems, PSO has improved our previous works in 29 cases out of 56 by 5.62%. The test has been applied to specific cases (VRPTW) to show that an approach which is not dedicated to this problem can return good solutions. The comparison shows that PSO outperforms our previous methods in 34 cases out of 56 by 5.63%. It will be interesting in future works to develop a new method to solve more complex problems for example by adding a new specificity to the fleet.

Acknowledgments This project is supported by TCP company (France) and ANRT french association.

References

Ai, T. J., & Kachitvichyanukul, V. (2007). A particle swarm optimization for the capacitated vehicle routing problem. *Logistics and Supply Chain Management Systems, 2*(1), 50–55.

Ai, T. J., & Kachitvichyanukul, V. (2009a). Particle swarm optimization and two solution representations for solving the capacitated vehicle routing problem. *Computers and Industrial Engineering, 56*(1), 380–387.

Ai, T. J., & Kachitvichyanukul, V. (2009b). A particle swarm optimization for the vehicle routing problem with simultaneous pickup and delivery. *Computers and Operations Research, 5*(4), 1693–1702.

Belmecheri, F., Prins, C., Yalaoui, F., & Amodeo, L. (2009a). An ant colony optimization algorithm for a vehicle routing problem with heterogeneous fleet, mixed backhauls, and time windows. In *13th IFAC Symposium on INformation COntrol problems in Manufacturing* (Vol. 13, pp. 1533–1538), Moscow, Russia.

Belmecheri, F., Prins, C., Yalaoui, F., & Amodeo, L. (2010a). A new particle swarm optimization on vehicle routing problem with heterogeneous fleet, mixed backhauls, and time windows. In *5th International Symposium on Hydrocarbons and Chemistry-Session: Optimizations and Logistics (Layout, Transportation, Scheduling)*, Algiers, Algeria.

Belmecheri, F., Prins, C., Yalaoui, F., & Amodeo, L. (2010b). Particle swarm optimization to solve the vehicle routing problem with heterogeneous fleet, mixed backhauls, and time windows. In *24th IEEE International Parallel and Distributed Processing Symposium*, Atlanta, GA, USA.

Belmecheri, F., Yalaoui, F., Prins C., & Amodeo, L. (2009b). A metaheuristic approach for solving the vehicle routing problem with heterogeneous fleet, mixed backhauls, time windows. In *40th Annual Conference of the Italian Operational Research Society*, Siena, Italy.

Casco, D. O., Golden, B. L., & Wasil, E. A. (1988). *Vehicle routing with backhauls: Models, algorithms and case studies* (pp. 127–147), North-Holland, Amsterdam

Chen, A. L., Yang, G. K., & Wu, Z. M. (2006). Hybrid discrete particle swarm optimization algorithm for capacitated vehicle routing problem. *Journal of Zhejiang University-Science A, 7*(4), 607–614.

Clerc, M. (2000). Discrete particle swarm optimization for traveling salesman problem. http://clerc.maurice.free.fr/pso/pso_tsp/Discrete_PSO_TSP.htm.

Clerc, M. (2005). *L'optimisation par essaim prticulaire*. France: Lavoisier.

Dethloff, J. (2001). Vehicle routing and reverse logistics: The vehicle routing problem with simultaneous delivery and pick-up. *Operations Research Spectrum, 23*, 79–96.

Dong, G., Tang, J., Lai, K. K., & Kong, Y. (2009). An exact algorithm for vehicle routing and scheduling problem of free pickup and delivery service in flight ticket sales companies based on set-partitioning model. *Journal of Intelligent Manufacturing*. doi:10.1007/s10845-009-0311-9.

Ellabib, I., Calamai, P., & Basir, O. (2007). Exchange strategy for multiple ant colony system. *Information Sciences, 177*(5), 1248–1264.

Gendreau, M., Laporte, G., Musaraganyi, C., & Taillard, E. D. (1999). A tabu search heuristic for the heterogeneous fleet vehicle routing problem. *Computers and Operations Research, 26*, 1153–1173.

Goldberg, D. E., & David, E. (1989). *Genetic algorithms in search, optimization and machine learning*. Boston, MA: Addison-Wesley Longman Publishing.

Jiang, W., Zhang, Y., & Xie, J. (2009). A particle swarm optimization algorithm with crossover for vehicle routing problem with time windows. In *IEEE Symposium on Computational Intelligence in Scheduling* (pp. 103–106), Nashville, TN.

Kennedy, J., & Eberhart, R. C. (1995). *Particle swarm optimization* (pp. 1942–1948). Perth: Perth-Australie.

Kennedy, J., & Eberhart, R. C. (1997). A discrete binary version of the particle swarm algorithm. In *IEEE International Conference on Computational Cybernetics and Simulation* (Vol. 5, pp. 4104–4108), Orlando, FL, USA.

Kim, B. T., & Son S. J. (2010). A probability matrix based particle swarm optimization for the capacitated vehicle routing problem. *Journal of Intelligent Manufacturing*, (Online). doi:10.1007/s10845-010-0455-7.

Landrieu, A., Mati, Y., & Binder, Z. (2001). A tabu search heuristic for the single vehicle pickup and delivery problem with time windows. *Journal of Intelligent Manufacturing, 12*(5–6), 497–508.

Lin, C. T. (2008). Using predicting particle swarm optimization to solve the vehicle routing problem with time windows. In: *Industrial Engineering and Engineering Management*, (pp. 810–814)

Liu, S. C., & Chung, C. H. (2009). A heuristic method for the vehicle routing problem with backhauls and inventory. *Journal of Intelligent Manufacturing, 20*(1), 29–42.

Machado, T. R., & Lopes, H. S. (2005). *A hybrid particle swarm optimization model for the traveling salesman problem* (pp. 255–258). Vienna: Springer.

Parragh, S. N., Doerner, K. F., & Hartl, R. F. (2008). A survey on pick up and delivery problems: Part I: Transportation between customers and depot. *Journal Fur Betriebswirtschaft, 58*(1), 21–51.

Prins, C. (2009). Two memetic algorithms for heterogeneous fleet vehicle routing problems. *Engineering Applications of Artificial Intelligence, 22*(6), 916–928.

Reimann, M., & Ulrich, H. (2006). Comparing backhauling strategies in vehicle routing using ant colony optimization. *Central European Journal of Operations Research, 14*(2), 105–123.

Rieck, J., & Zimmermann, J. (2009). *A hybrid algorithm for vehicle routing of less-than-truckload carriers* (Vol. 624, pp. 155–171). Berlin Heidelberg: Springer.

Rieck, J., Zimmermann, J., & Glagow, M. (2007). Tourenplanung mittelständischer speditionsunternehmen in stückgutkooperationen: Modellierung und heuristische lösungsverfahren. *Zeitschrift Planung and Unternehmenssteuerung, 17*, 365–388.

Salhi, S., & Nagy, G. (1999). A cluster insertion heuristic for single and multiple depot vehicle routing problems with backhauling. *Journal of the Operational Research Society, 50*(10), 1034–1042.

Shi, X. H., Liang, Y. C., Lee, H. P., Lu, C., & Wang, Q. X. (2007). Particle swarm optimization-based algorithms for tsp and generalized TSP. *Information Processing Letters, 103*, 169–176.

Solomon, M. M. http://w.cba.neu.edu/~msolomon/problems.htm.

Solomon, M. M. (1987). Algorithms for the vehicle routing and scheduling problem with time windows constraints. *Operation Research, 35*, 254–265.

Süral, H., & Bookbinder, J. H. (2003). The single-vehicle routing problem with unrestricted backhauls. *Networks, 41*(3), 127–136.

Talbi, E. G. (2009). *Metaheuristics: From design to implementation*. New York: Wiley.

Tang Montané, F. A., & Galvao, R. D. (2006). A tabu search algorithm for the vehicle routing problem with simultaneous pick-up and delivery service. *Computers and Operations Research, 33*(3), 595–619.

Vahdani, B., Tavakkoli-Moghaddam, R., Zandieh, M., & Razmi, J. (2010). Vehicle routing scheduling using an enhanced hybrid optimization approach. *Journal of Intelligent Manufacturing*. doi:10.1007/s10845-010-0427-y.

Voudouris, C., & Tsang, E. P. K. (1999). Guided local search and its application to the traveling salesman problem. *European Journal of Operational Research, 113*, 469–499.

Wang, K. P., Huang, L., Zhou, C. G., & Pang, W. (2003). Particle swarm optimization for traveling salesman problem. In *International Conference on Machine Learning and Cybernetics* (Vol. 3, pp. 1583–1585).

Wassan, N. A., & Osman, I. H. (2002). Tabu search variants for the mix fleet vehicle routing problem. *Operational Research Society, 53*(7), 768–782.

Xu, Y., Wang, Q., & Hu, J. (2008). An improved discrete particle swarm optimization based on cooperative swarms. In *IEEE/WIC/ACM International Conference on Intelligent Agent Technology* (pp. 79–82).

Yalaoui, N., Mahdi, H., Amodeo, L., & Yalaoui, F. (2009). *A new approach for workshop design*. doi:10.1007/s10845-009-0368-5.

Yazgi Tütüncü, G. (2010). An interactive GRAMPS algorithm for the heterogeneous fixed fleet vehicle routing problem with and without backhauls. *European Journal of Operational Research, 201*(2), 593–600.

Yazgi Tütüncü, G., Carreto, C. A. C., & Baker, B. M. (2009). A visual interactive approach to classical and mixed vehicle routing problems with backhauls. *Omega, 37*(1), 138–154.

Zhu, Q., Qian, L., Li, Y., & Zhu, S. (2006). An improved particle swarm optimization algorithm for vehicle routing problem with time windows. In *IEE Congress on Evolutionary Computation* (pp. 1386–1390).