Optimization of m-MDPDPTW Using the Continuous and Discrete PSO*

Imen Harbaoui DRIDI[1]*, Essia Ben ALAÏA[2], Pierre BORNE[3], Hanen BOUCHRIHA[4]
[1]Laboratory of Analysis, Conception and Control of Systems (LACCS),
Ecole Nationale d'ingénieurs de Tunis (ENIT), University of Tunis El Manar,
BP 37, le Belvédère, 1002 Tunis, Tunisia
imen.harbaoui@issatkr.rnu.tn (*Corresponding author*)
[2]Jouf University, College of Computer and Information Sciences,
Laboratory of Analysis, Conception and Control of Systems (LACCS),
Ecole Nationale d'ingénieurs de Tunis (ENIT), University of Tunis El Manar
essia.alaia@enit.rnu.tn
[3]CRIStAL: Research Center in Computer Science, Signal and Automatics of Lille,
Ecole Centrale de Lille, Villeneuve d'Ascq, France
pierre.borne@centralelille.fr
[4]Laboratory of Analysis, Conception and Control of Systems (LACCS),
Ecole Nationale d'ingénieurs de Tunis (ENIT), University of Tunis El Manar,
BP 37, le Belvédère, 1002 Tunis, Tunisia
hanen.bouchriha@enit.rnu.tn

Abstract: This paper presents a variant of the vehicle routing problem (VRP) that combines several constraints. This variant addresses the pickup and delivery problem (PDP), the use of multiple vehicles (m), the multi- depots (MD) and the time constraint (TW): m-MDPDPTW. In the m-MDPDPTW, one should build a route made up of several pairs (customer / supplier), which starts and ends at the same depot and respects the precedence and capacity constraints. For solving this problem, two approaches based on particle swarms are proposed with a view to minimizing the total distance travelled by all vehicles. The results yielded by these two algorithms, the continuous PSO and the discrete PSO, are then compared by making use of the benchmarks generated by Li and Lim.

Keywords: Pick-up and delivery problem, Multi-depots, Multiple vehicles, Optimization, Particle Swarm.

1. Introduction and Literature Review

The topic that is going to be explored in this paper is related to the multi-vehicle pickup and delivery problem with time windows to multiple depots: m-MDPDPTW. The m-MDPDPTW includes an even set of origin (suppliers) and destination (customers) for the collection and delivery of the goods, and the same type of product to be transported to deliver any customer. Each customer must be served exactly once, by one vehicle, and must not be visited before its supplier. Each vehicle is assigned to a single depot, which is the origin and end of its route (fixed destination), and must transport the goods from suppliers to their respective customers while respecting their time windows and without exceeding its capacity.

For the efficient resolution of such a complex problem, bringing together several variants, it is essential to design a resolution tool for m-MDPDPTW, based on heuristic and meta-heuristic methods. These methods make it possible to find feasible solutions, close to the optimum, in a reasonable computing time. The tool developed must provide route planning for a fleet of vehicles, in order to serve a set of customers at lower costs while also taking into accountthe various constraints of the problem. This paper focuses on evolutionary algorithms which are optimization techniques based on population, in particular, on particle swarms (PSO: Particle Swarm Optimization). This approach provides coding that accurately represents the data of the problem presented. Therefore, two approaches that assign a higher importance to the cooperation of individuals for finding the best solution have been

* This research work is an extension of a former paper entitled "A Comparative Study of the PSO and GA for the m-MDPDPTW" published in the International Journal of Computers Communications & Control, ISSN 1841-9836, 13(1), 8-23, February 2018. The novelty and originality of this article lies in the fact that the current research uses discrete PSO. The research results of the study have been compared using continuous and discrete PSO.

https://doi.org/10.24846/v28i3y201905

developed. The algorithms the two approaches developed in this study are based on optimization by PSO. The continuous PSO and the discrete PSO were adapted to the problem presented, while developing new procedures for the generation of initial particles, as well as new heuristics for the coding and decoding for the representation of the solutions obtained.

Indeed, the PSO is based on a population of randomly initialized solutions, and the search of optimums is done by updating the involved generations. However, the PSO has no evolution operator such as crossover and mutation, and in its research strategy, it assigns a greater importance to cooperation rather than competition between individuals. Contrary to most meta-heuristics, the PSO is a method that was originally designed to solve continuous variable optimization problems. As a result, several improvements to the original PSO algorithm have been proposed in specialisedliterature, in order to improve its performance and solve various continuous and discrete problems.

Many researchers have been interested in the vehicle routing problems. (Gustavo et al., 2018) Some works introduce the characteristics that distinguish the PDP from the standard VRP and a state-of-the-art approach to this type of problem and proposed methods for its resolution (Berbeglia et al., 2007) and (Parragh et al., 2008).This problem is often coupled with time windows (Kumar & Panneerselvam, 2012). (Dridi, I. Het al., 2011) have developed an interesting genetic algorithm for the optimization of multicriteria m-PDPTW by using the aggregation method and minimizing the compromise between total travel cost and total tardiness time. This method then dealt with the dynamic case. (Dridi, I. Het al., 2015).

(Marinakis & Marinaki, 2010) propose a GA that changes the VRP solutions by using a PSO. This algorithm improves the performance of each individual of the population. Another approach that combines genetic algorithms (GA) with the clustering algorithm for the resolution of m-MDPDP was proposed by (Alaïa et al, 2015). The objective of this work is to insert new depots in order to obtain feasible solutions (routes) for the m-MDPDP. The same authors have developed a new algorithm based on the PSO for the resolution of m-MDPDPTW. (Alaïa et al, 2017). This work was extended by a comparative study between the genetic algorithms and the PSO for the resolution of m-MDPDPTW. (Alaïa et al, 2018). (Cabrera et al., 2012) proposed a hybrid algorithm that combines PSO and simulated annealing (SA) to solve The Probabilistic Traveling Salesman Problem (PTSP). These authors use the PSO for exploration and rapid convergence to the desired solution. As for the SA, this algorithm is used to improve particle diversity and prevent the proposed approach from being trapped in the local optimum.

(Ma et al., 2019) proposed a hybrid genetic algorithm based on priorities with a fuzzy logic controller and a random simulation for solving simultaneous pickup and delivery problems with time windows and multiple decision makers (SPDTW – MDM). The results obtained clearly showed the performance of the proposed optimization method. To solve Vehicle Routing Problem with Time Windows and Simultaneous Delivery and Pickup (VRPTWSDP), (Alinezhad et al., 2018) proposed an approach based on an improved PSO for the minimization of the total distance travelled. These authors combined PSO with Simulated Annealing (SA) to improve the PSO's research capacity and maintain the diversity of solutions.

For the optimization of the total cost, (Li et al., 2019) proposed an algorithm based on the PSO for solving the problem of ecological vehicle routing. The cost function integrated various costs, namely, penalty cost, energy cost, GHG emissions costs and others. In these works, a real case of vehicle routing has been studied. Another approach was proposed by (Shen et al., 2018), based on the PSO and tabu search (TS)for solving the multi-depot open vehicle routing problem with time windows (MDOVRPTW). The objective studied by these authors is the minimization of the total cost, by presenting a technique that provides better route planning under carbon emission constraints.

The PSO algorithm has been used in various areas, namely, supply problems, (Mousavi et al., 2017), quadratic allocation problems, (Pradeepmon et al., 2018), technical design issues, (Dhiman & Kaur, 2019), as well as for solving the problems of industrial and assembly workshops (Toader, 2015).

The use of PSO has been tested and validated in the literature to demonstrate its effectiveness in the domain of transport and industry. This has encouraged researchers to adopt it. Further on, the present PSO-based approach will be detailed for the optimization of m-MDPDPTW. The second section sets for the mathematical model adopted for solving the problem presented. The third section is dedicated to the different steps of algorithm development based on continuous and discrete PSO. The simulation results are presented in Section 4. Finally, Section 5 concludes this paper.

2. Mathematical Model

Our problem is characterized by the following variables:

L: All depots, $\{1,...,dep\}$;

H: All nodes (pick-up and delivery), $\{1,\ldots,n\}$;

$\boldsymbol{H^+}$: All pick-up nodes, $\{1,\ldots,n/2\}$;

$\boldsymbol{H^-}$: All delivery nodes, $\{1,\ldots,n/2\}$;

$\boldsymbol{H_c}$: All couples, $\{1,\ldots,n/2\}$;

$\boldsymbol{C_i}$: The couple (c_i,f_i), with c_i is the delivery node and f_i is its corresponding pick-up node, $\forall i \in \{1,\ldots,n/2\}$;

$\boldsymbol{V_m}$: the set of vehicles available at depot *m*;

$\boldsymbol{d_{ij}}$: Euclidean distance which separates node *i* and *j*;

K: Total number of vehicles available in all depots;

k: Vehicle index for each depot *m* $\{1,...,V_m\}$;

$\boldsymbol{q_i}$: The quantity of goods at node *i* (if q_i <0, it is a delivery node or if q_i> 0, it is a pickup node);

$\boldsymbol{t^k_{ij}}$: Time required for vehicle *k* to travel from node *i* to node *j*;

Q: Maximum capacity of vehicles;

$\boldsymbol{Y^k_i}$: Vehicle load *k* after leaving node *i*;

$\boldsymbol{ET_i}$: The start date of the time window of node *i*;

$\boldsymbol{LT_i}$: The end date of the time window of node *i*;

$\boldsymbol{S_i}$: Operating time of node *i*;

$\boldsymbol{A_i}$: Arrival time at node *i*;

$\boldsymbol{D_i}$: Departure time from node *i*;

$\boldsymbol{W_i}$: Waiting time at node *i*;

$\boldsymbol{T_i}$: Tardiness time at node *i*;

A decision variable, modelling the visit sequence for each vehicle, is defined as follows:

$$x_{ij}^{mk}\begin{cases}=1 & \text{if vehicle k originates from depot m travelling along arc (i, j)}\\ =0 & \text{Sinon}\end{cases}$$

The goal is to minimize the total distance travelled by all vehicles for the m-MDPDPTW. The objective function is formulated as follows:

$$Minimize\ f=\sum_{m\in L}\sum_{k\in V_m}\sum_{i\in(H\cup m)}\sum_{j\in(H\cup m)} d_{ij}\,x_{ij}^{mk} \quad (1)$$

Subject to:

Each node is visited by a single vehicle, this is guaranteed by equations 2 and 3:

$$\sum_{m\in L}\sum_{i\in H\cup L}\sum_{k\in V_m} x_{ij}^{mk}=1 \qquad (\forall j\in H\cup L) \quad (2)$$

$$\sum_{m\in L}\sum_{j\in H\cup L}\sum_{k\in V_m} x_{ij}^{mk}=1 \qquad (\forall i\in H\cup L) \quad (3)$$

Each vehicle starts and ends its tour at the same depot:

$$\sum_{j\in H} x_{ij}^{mk}=\sum_{j\in H} x_{ji}^{mk} \quad (\forall i=m\in L \,\text{et}\, k\in V_m) \quad (4)$$

All vehicles leave from and return to the depotempty:

$$x_{ij}^{mk}=1\Rightarrow y_i^k=0(\forall i\in L, j\in H\,\text{et}\,k\in V_m) \quad (5)$$

$$x_{ji}^{mk}=1\Rightarrow y_i^k=0(\forall i\in L, j\in H\,\text{et}\,k\in V_m) \quad (6)$$

The load of vehicle k leaving node j is expressed as follows:

$$x_{ij}^{mk}=1\Rightarrow y_j^k=y_i^k+q_j(\forall i,j\in H\,\text{et}\,k\in V_m) \quad (7)$$

Equation 7 applies to all vehicles in all depots.

The load of the vehicles must not exceed the maximum capacity:

$$0\le y_i^k\le Q \quad (\forall i\in H\,\text{et}\,k\in V_m) \quad (8)$$

Each node *i* admits a time interval [*ETi*, *LTi*] in which it must be served. The times defined in equations (9), (10) and (11) respectively represent: the arrival time, the departure time and the service time at each depot:

$$x_{ij}^{mk}=1 \quad \Rightarrow \quad A_j=D_i+t_{ij}^k \quad (\forall k\in V_m) \quad (9)$$

$$D_i=A_i+s_i \quad (\forall i\in H) \quad (10)$$

$$D_i=s_i=0 \quad (\forall i\in L) \quad (11)$$

To respect the time windows associated to each node, the arrival time at node *i* must not be lower than *ETi*, otherwise a waiting time will be calculated by equation (12). And if the departure time is higher than *LTi*, a tardiness time will be calculated by equation (13):

$$A_i<ET_i\Rightarrow W_i=ET_i-A_i(\forall i\in H) \quad (12)$$

$$T_i=\max(0,D_i-LT_i) \quad (\forall i\in H) \quad (13)$$

Equation (14) ensures that no customer node (*ci*) is served before its supplier (*fi*). This makes it possible to respect the precedence constraint between the nodes:

$$D_{f_i}<D_{c_i} \quad (\forall i\in H_c, f_i\in H^+\,\text{et}\,c_i\in H^-) \quad (14)$$

3. Particle Swarm Optimization for the m-MDPDPTW

Two new approaches based on particle swarm optimization were developed for the resolution of m-MDPDPTW. The first comes up with a solution based on a continuous PSO algorithm adapted to the discrete problem presented, while the second uses a discrete PSO algorithm. In order to explain the functioning of these two algorithms, the basic idea of the presented PSO algorithm is exposed, then the main techniques which lead to the final structure of the two above-mentioned elaborated algorithms are presented.

3.1 General principle of the proposed PSO algorithm for m-MDPDPTW

The basic principle of the PSO is to start from an initial swarm and to apply a research strategy based on the cooperation of its *Ne* particles.

The search for optimums is done by producing several generations. At each generation, a potential solution to the problem is created, then evaluated in order to record the best solutions found. The solution to the above-mentioned m-MDPDPTV represents the best solution found for all generations.

Begin
1. Choose the size of the swarm *Ne*;
2. Creation of the initial swarm size;
3. Initialize the speed $v_{i,j}$ of each particle;
4. Initialize the position $x_{i,j}$ of each particle;
5. Evaluation of each particle;
6. Initialize $Pbest_i$ (0);
7. Initialize *Gbest* (0) = Best *Pbest* (0); 8. **While** stopping criterion is not satisfied do
9. **For** *i* = *1* **to** *Ne* **do**
10. Update the speed then the positions of each particle *i*;
11. Decoding the new position of the particle *i*;
12. Generation of new solutions;
13. Apply corrections heuristics for generating viable solutions;
14. Evaluate new solutions;
15. **End for**;
16. Update *Pbest* and *Gbest*;
17. **End While**;
18. The best solution found *Gbest*.
End

Figure 1. PSO proposed for the m-MDPDPTW

In the framework of the PSO algorithm, speed is the basic mechanism that directs research in promising areas of the solution space. This speed allows the updating of the particles` positions. For most applications, the positions of the particles represent the solutions of the approached problem but in this case the solution to the presented m-MDPDPTW is decoded from the new positions of the particles, hence the addition of decoding heuristics. The proposed PSO algorithm for m-MDPDPTW is described by the steps in Figure 1,

with:

$$\mathrm{Pbest_i(t+1)} = \begin{cases} \mathrm{x_i(t+1)}, & \text{si } f(\mathrm{x_i(t+1)}) \text{ is better than } f(\mathrm{Pbest_i(t)}) \\ \mathrm{Pbest_i(t)}, & \text{else} \end{cases} \quad (15)$$

$$\mathrm{Gbest(t+1)} = \arg\min_{\mathrm{Pbest_i}} f(\mathrm{Pbest_i(t+1)}),\ 1 \le i \le N \quad (16)$$

$$\mathrm{v_{i,j}(t+1)} = \begin{cases} w\ \mathrm{v_{i,j}(t)} + \\ c_1 r_{1_{i,j}}\mathrm{(t)(pbest_{i,j}(t) - x_{ij}(t))} + \\ c_2 r_{2_{i,j}}\mathrm{(t)(gbest_j(t) - x_{i,j}(t))} \end{cases} \quad (17)$$

$$\mathrm{x_{i,j}(t+1)} = \mathrm{x_{i,j}(t)} + \mathrm{v_{i,j}(t+1)} \quad (18)$$

where w, c_1r_1 and c_2r_2 represent respectively, the inertia, cognitive and social components associated with particle displacement.

The elaborate approaches use new heuristics for the creation of the initial solution and the correction of non-viable particles. They associate roads with vehicles by applying an identical grouping phase and use the same evaluation function. Therefore, they share the same procedure, unlike some details related to the nature of the PSO version used (discrete or continuous). This difference lies in the coding and decoding of the solutions, the updating of the speed and the position of the particle, and the calculation of the best position obtained for the particle and the swarm.

3.1.1 Heuristic Method Developed for the Creation of the Initial Swarm $P_{node/depot}$

The first step is to assign the nodes to the depots containing the vehicles that will serve them. These nodes are assigned in pairs (customer / suplier) by applying a grouping phase that chooses the nearest depot. At the end of this first step each couple belongs to a single initial depot. In order to improve the quality of the solutions of the initial swarm, a heuristic that minimizes the total travelled distance, while taking into account the different characteristics of the problem, was developed for its construction. The principle is to randomly select a starting node, in each depot, to calculate the distance between this node and all the nodes belonging to the same depot, so that it can be followed by the one that is closest to it.

The initial above-mentioned swarm, denoted by $P_{node / depot}$, represents the visit order of the nodes, for assigning only one vehicle to each depot, and it is thus a solution to the problem of 1-MDPDPTW. Figure 2 shows an example of particles of this swarm with 20 nodes and 3 index depots that is 21, 22, and 23.

Depot1	21	13	4	8	17	10	5	12	20	21
Depot2	22	7	18	3	6	19	2	14	16	22
Depot3	23	1	15	11	9	23				

Figure 2. An example of Particles of the swarm $P_{node / depot}$

3.1.2 Decoding of Initial Solutions: Routing Phase for the Creation of the Swarm $P_{node / vehicle / depot}$

In order to decode the passage of vehicles from individuals in the $P_{node / depot}$ swarm, a new type of swarm $P_{vehicle / depot}$ of size *Ne* should be created. This will indicate, to each individual of the swarm $P_{node / depot}$, the number of nodes visited by each vehicle available in each depot. Once the swarm $P_{node / vehicle / depot}$ is generated, precedence and capacity correction heuristics are applied. Further on, a new correction heuristic is added to the former corrections in order to assign each couple to the same road. The swarm $P_{node / vehicle / depot}$ contains solutions to the m-MDPDPTW problem approached, as it represents for each depot the visit order of nodes for each vehicle. Figure 2 shows an example of particles of the swarm $P_{node / vehicle / depot}$.

Depot1	21	13	4	8	17	21	10	5	12	20	21
Depot2	22	7	18	3	6	22	19	2	14	16	22
Depot3	23	1	15	23	11	9	23				

Figure 3. An example of particles of the swarm $P_{node / vehicle / depot}$

The solutions to the m-MDPDPTW problem are represented as a direct permutation list containing a sequence of genes encoded in integers which represent the n nodes to be visited (customers and supplier). The start and end of each road is indicated by the depot index, from which the vehicle begins and ends its tour.

3.2 Adaptation of Continuous PSO for m-MDPDPTW Optimization

After updating the speed and calculating the new position of each particle, a decoding phase is required to obtain the new visit order of the nodes. Figure 4 shows the structure of the algorithm for adapting the continuous PSO developed for the optimization of m-MDPDPTW.

n: the number of nodes to serve, *f*: the function to be minimized, T: the maximum number of iterations, dep: number of depots;
1. Initialization
t = 0 ; $x_{max} = n$; $x_{min} = 0$;
2. Choose the size of the swarm *Ne*;
3. Generate Ne initial particles of $P_{node/depot}$;
4. Routing phase for creating initial solutions $P_{node/vehicle/depot}$
5. Application of correction heuristics: precedence, capacity and belonging to each couple on the same route;
6. Evaluate the initial particles of $P_{node/vehicle/deposit}$ (equation 1);
7. Initialize the speed of each particle by using random values between [0; $_n$];
8. Initialize the position of each particle x (0) = $P_{node/depot}$;
9. Initialize $Pbest_i$ (0) = $P_{node/vehicle/depot}$;
10. Initialize the best solution found by the swarm:
$Gbest(0) = \arg\min_{Pbest_i} f(Pbest_i(0))$
11. **While** (t<T) **do**
12. **For** *i* = 1 **to** *Ne* **do**
13. Update speed vi ($t + 1$) using equation (17);
14. Update of the new position x_i ($t + 1$) using equation (18);
15. Check if the new particle is coming out of the search space
— **if** ($x_i(t + 1)$ >xmax) **then** $x_i(t + 1) = x_{max}$;
— **else if** ($x_i(t + 1) < x_{min}$) **then** $xi(t + 1) = x_{min}$;
17. Generate new solutions: Routing phase;
18. Apply heuristics of corrections;
19. Evaluate the *Ne* particles of $P_{node/vehicle/depot}$ (equation 1);
20. Keep the best result found by the particle;
For *i* = 1 to *Ne* **do**
if $f(P_{node/vehicle/depot}(i)) <= f(Pbest_i(t))$
then $Pbest_i(t + 1) = f(x(t + 1))$ **else**
$Pbesti(t + 1) = Pbesti(t)$;
End For
21. Update the best solution found by the swarm:
$Gbest(t+1) = \arg\min_{Pbest_i} f(Pbest_i(t+1))$
22. **End For**;
23. **End While**;
24. The best solution found on all generations *Gbest*.

Figure 4. Algorithm for adapting the continuous PSO tothe m-MDPDPTW

3.3 Discrete PSO for the Optimization of m-MDPDPTW

The discrete PSO algorithm solves discrete optimization problems with variables between 0 and Z-1 (Z is a positive integer) (Osadciw & Veeramachaneni, 2009).

This algorithm adapts well to the data of the above-mentioned problem by taking $Z = n$. Like in the continuous version, equation (17) is also used to update the particle speed. This speed, which is represented in real numbers, is subsequently transformed into a set of probabilities by equation (19). Then, the new position of the particle *i* is calculated by means of equations (20) and (21). The best positions obtained for the particle and the swarm are also calculated like in the continuous version using equations (15) and (16).

$$\tilde{v}_{i,j}(t+1) = \mathrm{sig}(v_{i,j}(t+1)) = \frac{Z}{1+e^{-v_{i,j}(t+1)}} \qquad (19)$$

$$\tilde{x}_{i,j}(t+1) = \mathrm{round}(\tilde{v}_{i,j}(t) + (Z-1)\times\sigma\times\beta) \qquad (20)$$

$$x_{i,j}(t+1) = \begin{cases} Z-1 & \text{si} \quad \tilde{x}_{i,j}(t+1) > Z-1 \\ 0 & \text{si} \quad \tilde{x}_{i,j}(t+1) < 0 \end{cases} \qquad (21)$$

where σ is a parameter and β is a random number of the interval [*0 1*]. Equation (19) is used to transform the speed into a continuous value between *0* and *Z*.

3.3.1 Coding Solutions for Discrete PSO

For the proposed discrete version of the PMP, the principle of permutation coding used in the continuous version was adopted. Each element of the particle represents a node to visit, and these indices must be integers belonging to the interval [*0 n-1*]. If one uses the example of particle *i* in figure 2, and keeps the same indices for depots in the discrete version, the particle will be coded as shown in Figure 5.

Depot1	21	12	3	7	16	9	4	11	19	21
Depot2	22	6	17	2	5	18	1	13	15	22
Depot3	23	0	14	10	8	23				

Figure 5. Particle i of the swarm $P_{node/depot}$ coded for the discrete PSO version

3.3.2 Decoding Solutions for Discrete PSO

The method of decoding solutions for the discrete PSO admits the same steps as those used in the continuous version. The values of the new position of the particle *i* are ordered in ascending order. These values represent the order of priority for the position of each node in the particle. The only difference for the discrete version is the use of *vi,j(t+1) = sig(vi,j(t+1))* instead of *vi (t + 1)* to order the elements of the particle that share the same values of position.

n: the number of nodes to serve, *f*: the function to be minimized, *T*: the maximum number of iterations, and *dep*: depot number;
1. Initialization
t = 0, *xmax* = n - 1, *xmin* = 0,σ, β;
2. Choose the swarm size *Ne*;
3.Generate Ne initial particles of $P_{node\,/\,depot}$
4. Routing phase for creating initial solutions $P_{node/vehicle/depot}$
5. Application of correction heuristics: precedence, capacity and belonging to each couple on the same road;
6. Evaluation of Ne initial particles of $P_{node/vehicle/depot}$ (equation 1);
7. Initialize the speed of each particle by using random values between [-4; 4];
8. Initialize the position of each particle *x*(*0*) = *Pnode/depot*;
9. Initialize
$Pbest_i$ (0) = *Pnode/vehicle/depot* [*i*];
10. Initialize the best solution found by the swarm;
$\mathrm{Gbest}(0) = \arg\min_{\mathrm{Pbest}_i} f(\mathrm{Pbest}_i(0))$
11.**While** (t<T) **do**
12. **For** *i* = *1* **to** *Ne* **do**
13. Update speed *vi* (*t* + *1*) using equation (17);
14. Update the new position *xi* (*t* + *1*) using equations (19), (20) and (21);
15. Check if the new particle is coming out of the search space:
if ($x_i(t+1) > x_{max}$) **then** $x_i(t+1) = x_{max}$;
else if ($x_i(t+1) < x_{min}$) **then** $x_i(t+1) = x_{min}$;
16.Decoding the new particle positions;
17. Generation of new solutions: Routing phase;
18. Apply heuristics of corrections;
19. Evaluation of the *Ne* particles of $P_{node/vehicle/depot}$(equation 1);
20. Keep the best result found for the particle;
21. **For** *i* = 1 **to** *Ne* **do**
if $f(P_{node/vehicle/depot}(i)) <= f(Pbest_i(t))$ **then** $Pbest_i(t+1) = Pbest_i(t)$;
22. **End For**;
$\mathrm{Gbest}(t+1) = \arg\min_{\mathrm{Pbest}_i} f(\mathrm{Pbest}_i(t+1))$
23. **End For**;
24. **End While**;
25. Show the best solution found for all generations Gbest;

Figure 6. Discreet PSO for the m-MDPDPTW

In Figure 6, the structure of the discrete PSO algorithm developed for the resolution of m-MDPDPTW is presented.

4. Simulation and Results

In this part, some simulation results obtained by using the Li and Lim instances are presented, by inserting depots nodes. The aim is to minimize the total travelled distance by observing all the constraints of the problem.

In the continuous PSO version, particle movement is controlled by limiting the maximum distance they travel during iteration. Thus, to escape the problem of the particles exit from the search space, an interval confinement technique is used. The speed of each particle is initialized for values between *0* and *n* (where *n* is the number of nodes to visit). The position of each particle must contain integer values belonging to the interval [1; n]. To update the speed of each particle, equation 17, which contains an inertia factor, is used. Global exploration of the research space starts with a relatively high inertia factor value *w* = 0:8then the exploitation is intensified in order to refine the search on a small space by linearly decreasing *w* until it reaches 0.5. After several tests, the chosen parameter values are: *c1* = 0: 2, *r1* = 0: 3, *c2* = 0: 2, *r2* = 0: 5. The new position of the particle is calculated based on Equation 18.

Table 1. Location of the different depots

Instances	Coordinates of the 1st depot	Coordinates of the 2nd depot
LC	(40 , 50)	(34 , 32)
LR	(35 , 35)	(60 , 40)
LRC	(40 , 50)	(65 , 30)

Table 1 includes the location of the different depots considered in this simulation for each type of Li and Lim instances, that are available in the time window [*0 600*].

Table 2. Results of the simulations for the adaptation of the continuous PSO to m-MDPDPTW by using instances of Li and Lim for two depots

Instance	Vehicles number used	Better distance
LC101	14	1839,3962
LC102	12	1606.802
LC201	06	947,914
LC202	05	1472,292
LR101	09	2137.170
LRC202	07	2829,999

Table 2 presents some results of the simulations for the resolution of m-MDPDPTW by the continuous PSO approach.

For the discrete PSO version and in order to avoid the problem of the swarm's exit from its search space, a maximum speed is set at *4*. The initial speed of each particle is therefore initialized by random values in the interval [*-4 4*]. Each element of the particle is an integer belonging to the interval [*0 n-1*]. For updating the particle speed, equation 17 is also used. Equations 19, 20 and 21 are used for updating the particle's position. The values of the chosen parameters are as follows: σ = 0:5 and β takes on a random value between 0 and 1. Table 3 illustrates some solutions for adapting the discrete PSO to m-MDPDPTW by using Li and Lim instances for two depots.

Table 3. Simulation results for the adaptation of discrete PSO to m-MDPDPTW by using the instances of Li and Lim to two depots

Instance	Vehicles number used	Better distance
LC101	17	1940,205
LC102	09	1982,140
LC201	05	958,613
LC202	05	1563,263
LR101	07	2054,441
LRC202	07	2859,729

Table 4 illustrates the results of both continuous and discrete PSO approaches.

Table 4. Comparison of the results of the two PSO approaches

Instance	Better distance continuous PSO	Better distance discrete PSO
LC101	1839,3962	1940,205
LC102	1606,801	1982,140
LC201	947,914	958,613
LC202	1472,292	1563,263
LR101	2137,17	2054,441
LR102	2128,639	1999,699
LR201	2060,405	2077,726
LR202	2497,547	2568,205
LRC101	2513,376	2593,103
LRC102	2567,438	2530,135
LRC201	2937,202	2972,344
LRC202	2829,999	2859,729

When comparing the continuous PSO adapted to the m-MDPDPTW problem to the discrete PSO, one can notice that the values of the total travelled distance obtained by using the former approach are better than those obtained by using the latter approach.

The discrete PSO only triggered a slight improvement of the fitness for the LR101, LR102 and LRC102 instances.

5. Conclusion

This paper presents a new metaheuristic method based on particle swarms for the resolution of m-MDPDPTW.

The objective function analysed is the minimization of the total distance travelled by all vehicles. For this purpose, in the first part the mathematical model that represents the above-mentioned problem was formulated.

The second part of this paper has been devoted to the use of the continuous and discrete PSO and to outlining the different steps developed for finding the best solution for optimizing the objective function tackled in this paper.The above-mentioned solution represents the complete road of each vehicle, specifying the order of the nodes to be visited. The solutions found by these two approaches were compared using Li and Lim's instances. This comparison showed that the adaptation of the continuous PSO yields better results for the optimization of the m-MDPDPTW.

A future research work is planned to optimize the above-mentioned problem in its dynamic version.

REFERENCES

1. Alaïa, E. B., Dridi, I. H., Bouchriha, H. & Borne, P. (2015). Insertion of new depot locations for the optimization of multi-vehicles Multi-Depots Pickup and Delivery Problems using Genetic Algorithm. In *2015 International Conference on Industrial Engineering and Systems Management (IESM)* (pp. 695-701). IEEE.

2. Alaïa, E. B., Dridi, I. H., Bouchriha, H. & Borne, P. (2017). A Particle Swarm Optimization for the multi-depots pick-up and delivery problems with time windows and multi-vehicles. In *2017 International Conference on Industrial Engineering and Systems Management (IESM)* (pp. 623-628). IEEE.

3. Alaïa, E. B., Harbaoui, I., Borne, P. & Bouchriha, H. (2018). A Comparative Study of the PSO and GA for the m-MDPDPTW, *International Journal of Computers Communications & Control*, *13*(1), 8-23.

4. Alinezhad, H., Yaghoubi, Ù. S., Hoseini Motlagh, S. M. & Allahyari, S. (2018). An Improved Particle Swarm Optimization for a Class of Capacitated Vehicle Routing Problems, *International Journal of Transportation Engineering*, *5*(4), 331-347.

5. Berbeglia, G., Cordeau, J. F., Gribkovskaia, I. & Laporte, G. (2007). Static pickup and delivery problems: a classification scheme and survey, *Top*, *15*(1), 1-31.

6. Cabrera, G., Roncagliolo, S., Riquelme, J. P., Cubillos, C. & Soto, R. (2012). A Hybrid Particle Swarm Optimization – Simulated Annealing Algorithm for the Probabilistic Travelling Salesman Problem, *Studies in Informatics and Control*, *21*(1), 49-58.

7. Dhiman, G. & Kaur, A. (2019). A hybrid algorithm based on particle swarm and spotted hyena optimizer for global optimization, *Soft Computing for Problem Solving*, 599-615. Springer, Singapore.

8. Dridi, I. H., Kammarti, R., Ksouri, M. & Borne, P. (2011). Multi-objective optimization for the m-PDPTW: aggregation method with use of genetic algorithm and lower bounds, *International Journal of Computers Communications & Control, 6(2)*, 246-257.

9. Dridi, I. H., Alaïa, E. B. & Borne, P. (2015). Heuristic approach for the optimization of the dynamic multi-vehicle pickup and delivery problem with Time Windows. In *2015 International Conference on Industrial Engineering and Systems Management (IESM)* (pp. 733-739). IEEE.

10. Gatica, G., Ahumada, G., Escobar, J. W. & Linfati, R. (2018). Efficient Heuristic Algorithms for Location of Charging Stations in Electric Vehicle Routing Problems, *Studies in Informatics and Control*, vol. *27*(1), 73-82. DOI: 10.24846/v27i1y201808

11. Kumar, S. N. & Panneerselvam, R. (2012). A survey on the vehicle routing problem and its variants, *Intelligent Information Management*, *4*(3), 66-74.

12. Li, Y., Lim, M. K. & Tseng, M. L. (2019). A green vehicle routing model based on modified particle swarm optimization for cold chain logistics, *Industrial Management & Data Systems*, *119*(3), 473-494.

13. Ma, Y., Li, Z., Yan, F. & Feng, C. (2019). A hybrid priority-based genetic algorithm for simultaneous pickup and delivery problems in reverse logistics with time windows and multiple decision-makers, *Soft Computing, 23*(15), 6697-6714.

14. Marinakis, Y. & Marinaki, M. (2010). Ahybrid genetic-particle swarm optimization algorithm for the vehicle routing problem, *Expert Systems with Applications*, *37*(2), 1446–1455.

15. Mousavi, S. M., Bahreininejad, A., Musa, S. N. & Yusof, F. (2017). A modified particle swarm optimization for solving the integrated location and inventory control problems in a two-echelon supply chain network, *Journal of Intelligent Manufacturing*, *28*(1), 191-206.

16. Osadciw, L. & Veeramachaneni, K. (2009). Particle Swarm Optimization, chapter *Particle Swarms for Continuous, Binary, and Discrete Search Spaces*, 451-460.

17. Parragh, S. N., Doerner, K. F. & Hartl, R. F. (2008). A survey on pickup and delivery problems, part i: Transportation between customers and depot, *Journal für Betriebswirtschaft*, *58*(1), 21-51.

18. Parragh, S. N., Doerner, K. F. & Hartl, R. F. (2008). A survey on pickup and delivery problems, part ii: Transportation between pickup and delivery locations, *Journal für Betriebswirtschaft*, *58*(2), 81-117.

19. Pradeepmon, T., Sridharan, R. & Panicker, V. (2018). Development of modified discrete particle swarm optimization algorithm for quadratic assignment problems, *International Journal of Industrial Engineering Computations*, *9*(4), 491-508.

20. Shen, L., Tao, F. & Wang, S. (2018). Multi-Depot Open Vehicle Routing Problem with Time Windows Based on Carbon Trading, *International Journal of Environmental Research and Public Health*, *15*(9), 2025.

21. Toader, F. A. (2015). A Hybrid Algorithm for Job Shop Scheduling Problem, *Studies in Informatics and Control*, *24*(2), 172. DOI: 10.24846/v24i2y201505