

A Comparative Study of the PSO and GA for the m-MDPDPTW

E. Ben Alaïa, I. Harbaoui, P. Borne, H. Bouchriha

Essia Ben Alaïa*, **Imen Harbaoui Dridi**, **Hanan Bouchriha**

LACCS : Laboratoire d'Analyse, Conception et Commande des Systèmes

École Nationale des Ingénieurs de Tunis, LR11ES20, Tunisie

Université de Tunis El Manar

*Corresponding author: essia.benalaia@enit.rnu.tn

imen.harbaoui@issatkr.rnu.tn

hanen.bouchriha@enit.rnu.tn

Pierre Borne

CRIStAL : Centre de Recherche en Informatique Signal et Automatique de Lille

Ecole Centrale de Lille

Villeneuve d'Ascq, France

pierre.borne@centraledelille.fr

Abstract: The m-MDPDPTW is the multi-vehicles, multi-depots pick-up and delivery problem with time windows. It is an optimization vehicles routing problem which must meet requests for transport between suppliers and customers for the purpose of satisfying precedence, capacity and time constraints. This problem is a very important class of operational research, which is part of the category of NP-hard problems. Its resolution therefore requires the use of evolutionary algorithms such as Genetic Algorithms (GA) or Particle Swarm Optimization (PSO). We present, in this sense, a comparative study between two approaches based respectively on the GA and the PSO for the optimization of m-MDPDPTW. We propose, in this paper, a literature review of the Vehicle Routing Problem (VRP) and the Pick-up and Delivery Problem with Time Windows (PDPTW), present our approaches, whose objective is to give a satisfying solution to the m-MDPDPTW minimizing the total distance travelled. The performance of both approaches is evaluated using various sets instances from [10] PDPTW benchmark data problems. From our study, in the case of m-MDPDPTW problem, the proposed GA reached to better results compared with the PSO algorithm and can be considered the most appropriate model to solve our m-MDPDPTW problem.

Keywords: : Particle Swarm Optimization (PSO), Genetic Algorithm (GA), Vehicle Routing Problem (VRP), Pick-up and Delivery Problem with Time Windows (PDPTW), m-MDPDPTW, optimization.

1 Introduction

The multi-vehicles, multi-depots pick-up and delivery problem with time windows (m-MDPDPTW) combine several variant of the well-known Vehicle Routing Problem (VRP), which is part of the category of NP-hard problems. Even for a small problem size, the resolution of such complex problem requires to use heuristic and meta-heuristic methods, since these allow finding feasible solutions, in a reasonable calculation time.

The m-MDPDPTW principle is to construct a set of routes in order to pick up and to deliver goods between a set of suppliers (pickup nodes) and a set of customers (delivery nodes). We consider several depots which does not contain any goods and where is based a homogeneous fleet of vehicles. Every single vehicle has a limited capacity and must leave and return to its starting depot. And each load must be transported by one vehicle without any transshipment at

other locations. A time window is associated with each pick-up and delivery node, thus defining for each vehicle the earliest time to visit and the latest permitted time to leave each node.

We aim to minimize the sum of travel distance without violating the different problem constraints, which are: (1) the capacity constraint which ensures that at any time in the route, the load on the vehicle must not exceed its maximum capacity. And all the vehicles leave and return to depot unloaded. (2) The precedence constraint which ensures that for each request, the origin (suppliers) precedes the destination (customers). (3) The soft time constraint: we consider that if the vehicle arrives before the earliest permitted time to visit the node, then it must wait the beginning of the time window to serve it. But, if the end of service time, in the node, is after the latest permitted time to leave it, then a tardiness time is calculated and the solution is accepted but penalized.

In this context, we develop and compare two improving optimization approaches based on population search methods, which are Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) for solving our m-MDPDPTW.

This paper is organized as follows. The literature review of the Vehicle Routing Problem and the Pick-up and Delivery Problem is presented in section 2. The problem formulation and the mathematical model for the m-MDPDPTW is described in Section 3. Section 4, proposes our developed approach, based on GA and PSO, for solving a m-MDPDPTW to minimize the total travel distance. Section 5 validates and compares the proposed approaches by numerical example. Finally, the concluding remarks and further research are included in Section 6.

2 Related work

2.1 Genetic algorithm for the vehicle routing problem and the pick-up and delivery problem

Genetic algorithms are the most popular and most used evolutionary algorithms. They evolve a set of coded solutions called individuals populations. For each individual, the degree of adaptation to their local environment is measured by a predefined objective function to optimize, called fitness. From one generation to another, the best adapted individuals are selected for the reproduction by applying of genetic operations of crossover and mutation in order to produce better populations with better performing individuals. This process is repeated until a stop criterion is reached.

A literature review was proposed by [19], in which the authors detail 15 variants of the VRP and give a synthesis of the 48 heuristics for these problems. In [20], the authors propose an approach based on metaheuristic method, which combines a hybrid genetic algorithm for research and adaptive control for diversification to solve the multi-depot and periodic vehicle routing problems. In the work of [18], the authors present two metaheuristics for the resolution of the Multi depot Vehicle Routing Problem (MDVRP) which are based on a local search and a hybrid genetic algorithm.

A genetic algorithm was developed by [6] for the resolution of the multi-criteria PDPTW with multiple vehicles, based on the aggregation method and minimizing the compromise between the total travel cost, the total tardiness time and the vehicles number. This algorithm has been treated in the dynamic case in [5].

The authors in [1] proposed a multi-criteria approach based on GA for the optimization of the m-MDPDPTW. The aim is to discover a set of satisfying solutions (routes) minimizing total travel distance, total tardiness time and the total number of vehicles.

A tabu method, a genetic algorithm based on chromosome permutation, a "split" procedure and a local search are proposed by [9] to solve a particular problem of PDPTW. This Problem

considers several requests: the delivery of medicines and medical care pharmacy at home patients and the pick-up of biological samples and unused drugs at the patients.

The authors in [13] present a memetic algorithm for the resolution of PDPTW with an efficient crossover operator. Memetic algorithm is obtained by combining GA with local search.

A new genetic algorithm has been introduced in [14] for optimization of the MDVRP with capacity constraints and restrictions on the traveled distance. These authors use indirect coding and adaptive mutation operator inter-depots for the affectation of customers.

For the resolution of the MDVRP, [7] use a stochastic approach based on the GA and the fuzzy logic to adapt the crossover and mutation rates. They consider the total traveled cost and the time spent in the objective function. [2] propose an approach which is based on the combination of Genetic Algorithm (GA) with the clustering algorithm for the optimization of the m-MDPDP. The main contribution in this work is to find new depot locations in order to obtain feasible solution (routes) for the m-MDPDP.

The disadvantage of the GA is that it requires a high number of iterations. Particle Swarm Optimization (PSO) is a relatively recent heuristic algorithm which is based on the behavior of swarming characteristics of living organisms. These approaches are similar, nevertheless they each have its own particularities, its own research strategy and two different ways to develop the set of solutions.

2.2 PSO for the vehicle routing problem and the pick-up and delivery problem

The principle of the PSO is to start from an initial swarm (population) and apply a research strategy based on the cooperation of its N_p particles (individuals). In the PSO algorithm, the speed is the basic mechanism which drives the search in the promising areas of the solution space. This speed allows to put update the particles positions.

In [4], the authors present a solution approach based on particle swarm optimization (PSO) in which a local search is performed by variable neighborhood descent algorithm (VND). This approach was developed to solve the vehicle routing problem with simultaneous pickup and delivery (VRPSPD).

A hybrid particle swarm optimization (HPSO) is proposed by [16] to solve the multi-objective PDPTW. This algorithm adds particles neighbor information to diversify the particle swarm and use the variable neighborhood search (VNS) to enhance the convergence speed.

The PSO is also used to solve the vehicle routing problem with multi-depot. [8] propose an improved PSO for the multi-depot vehicle routing problem with time windows. Another PSO algorithm is proposed for solving the practical case of multi-depot vehicle routing problem with simultaneous pickup and delivery and time window [17].

A GA which evolves the VRP solutions using a PSO is proposed in [12]. This algorithm improves the performance of each individual of the population. We find other metaheuristic methods that have been developed for the resolution of the VRP who is one of the most famous combinatorial optimization problems [15], [21].

We choice to study two evolutionary algorithms GA and PSO. These are two optimization techniques based on the populations and have been widely compared in the literature. These two approaches provide a coding that perfectly represents the data of our problem. The difference is in their research strategy: The Genetic algorithm is categorizing as global research heuristics that uses crossover and mutation operators and a competition between individuals to find desired solutions, while the PSO has no evolutionary operator and in its research strategy, it gives more importance to cooperation between individuals.

Many researches in the literature showed that the PSO gives better results for the vehicle

routing problems. For our problem studied case, GAs gave better results compared to the PSO [11].

3 Mathematical model

Like the PDPTW problem, our m-MDPDPTW takes into account the following variables and parameters:

L : Set of depots;

H : Set of nodes (pick-up and delivery) $\{1, 2, \dots, n\}$;

H^+ : Sets of pick-up nodes;

H^- : Sets of delivery nodes;

H_c : Set of couples: delivery and pickup;

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

V_m : Set of available vehicles from depot $m, \{V_1, \dots, V_{dep}\}$;

d_{ij} : Euclidean distance between node i and j ;

K : The total number of vehicles available for all the depots;

q_i : Goods quantity request of the node i , (if $q_i < 0$ it is a delivery node else if $q_i > 00$ it is a pick-up node);

t_{ij}^k : Time taken by the vehicle k to travel from node i to node j ;

Q : The maximum capacity of a vehicle;

y_i^k : The load of vehicle k after leaving the node i ;

ET_i : The earliest time that node i can be serviced by a vehicle;

LT_i : The latest permitted time to leave node i ;

S_i : Service time at node i ;

A_i : Arrival time of the assigned vehicle at the node i ;

D_i : Departure time of the vehicle from the node i ;

W_i : Waiting time of the vehicle at node i ;

T_i : Tardiness time of the vehicle at node i .

We add the above sets

L : Set of depots $1, \dots, dep$;

H_c : Set of couples: delivery and pickup; C_i : The couple c_i, f_i : the pick-up node f_i with its corresponding delivery node c_i , $\forall i \in \{1, \dots, (n/2)\}$ V_m : Set of available vehicles from depot m V_1, \dots, V_{dep} .

Our decision variable is defined as follows:

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

The m-MDPDPTW considered in this study aims to minimize the total distance travelled (f_1). The objective function is formulated as:

$$f_1 = \sum_{m \in L} \sum_{k \in V_m} \sum_{i \in (H \cup m)} \sum_{j \in (H \cup m)} d_{ij} x_{ij}^{mk} \quad (1)$$

subject to:

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

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

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

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

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

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

$$0 < y_i^k \leq Q \quad (\forall i \in H \text{ and } k \in V_m) \quad (8)$$

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

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

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

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

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

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

In this formulation, constraints (2) and (3) ensure that each request is served once by the same vehicle, while constraint (4) guarantee that each vehicle starts and ends its route at the same depot. Constraints (5) and (6) impose that all the vehicles which leave and return to depot are unloaded. For each vehicle of each depot, the load of vehicle k leaving node i to j is defined in (7), while capacity constraint (8) guarantee that at any time the load, on the vehicle k , must not exceed the vehicle capacity. Each node i have time interval $[ET_i, LT_i]$ in which service at location i must take place. This time windows define in constraints (9) to (11) the arrival time, the departure time, service times at every depot, respectively. If the vehicle arrives at customer before the beginning of the applicable service time, a waiting time is calculated according the equation (12). And if the departure time from a node is later than its latest time of service, we calculate a tardiness time by equation (13). Finally, the precedence constraints (14) ensure that the pick-up node (f_i) of every couple i must be visited before the corresponding delivery node (c_i).

4 Optimization approach for solving the m-MDPDPTW

4.1 Solution representation

We have adopted a coding that is easy to use and to program, which fits well with the needs of our problem. Our choice was based on direct-type permutation list encoding to represent the solutions (individuals for the GA or particles for PSO) of the m-MDPDPTW. Our solution is a

sequence of genes encoded in integers. Each gene identifies a node and the order of the genes gives for each vehicle the order in which these nodes will be visited. Each coded individual contains both customers and suppliers. We chose to indicate the start and the end of each path by the depot indices. The index 0 is not used throughout the work. Figure 1 shows a solution encoded as a direct type permutation list. The example consists of twenty nodes numbered 1 to 20, which is ten pairs (customers/suppliers), three depots (index 21, 22 and 23) and two vehicles located in each depot. In the first depot the two vehicles are used, so we will have two routes that start at the depot (index 21) and ends at the same depot. The first vehicle visit two nodes in this order (13 before 4) and the second one visit three couple that is to say six nodes. On the other hand in the third depot just one vehicle is used to visits four nodes.

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

Figure 1: Solution representation

4.2 Genetic algorithm optimization for the m-MDPDPTW

Our developed approach based on GA, manipulates several types of populations and the solutions of the m-MDPDPTW are constructed using different heuristics which breaks down the problem into two major parts: regrouping then routing.

Structure of the initial population

The choice of the initial population is very important because it has an influence on the convergence speed of the genetic algorithm used. The initial population is constructed into two steps:

Step 1: The depot-grouping phase: creation of the population couple/depot

The m-MDPDPTW considers several depots in which a fixed number of vehicles are located. It is therefore necessary to determine from which depot the nodes will be served. The process rules of depot grouping phase is explained in details in the following reference [1]. At the end of this strategy, each couple (pick-up node and its associated delivery node) are assigned to the nearest depot.

Step 2: Generating the initial solution

A group of initials solutions is randomly generated, constructing the first population named Pcouple/depot containing N individuals. The chromosomes of the solution are encoded using path representation in which, for each depot, the couples are listed in the order in which they are visited [21].

The routing phase

For each depot, the number of vehicles used and the order of delivery and pick-up within each route are specified by the population named $P_{node/vehicle/depot}$. To construct this population type we should follow three steps:

Step1: Creation of population $P_{vehicle/depot}$

Knowing the number of vehicles available in each depot, we start by creating $2N$ individuals of a new population named $P_{vehicle/depot}$. This population is generated at each iteration to indicate the new number of vehicles used and the new number of couples to be visited by each vehicle.

Step 2: Use of GA operators

The genetic operators are used to create new population $P_{couple/depot}$ containing $2N$ individuals. Its first part represents an exact copy of the N individuals of $P_{couple/depot}$ created in phase 2, while the remaining 50 percent are created by applying GA operators on the first part. In our case, we select two parent chromosomes from population of step 2 by using tournament selection. For recombination, it is difficult to determine the most effective crossover method in advance. Therefore, we have designed our recombination operator based on the one point crossover and uniform crossover. Our crossover algorithm is adapted for permutation coding with individuals containing multiple depots. The principle is to start by applying a binary mask of the same length as the number of depots. This mask will determine: (1) the depot that will be copied: the genes of the first parent contained in this depot will be copied in the second child and those of the second parent will be copied into the first child). (2) The depot that will be crossed: the same crossing point is chosen in each depot, for the two parent chromosomes. The first part of each child is copied, gene by gene, from its parent. The first child will be completed with genes that were not inherited from the first parent but rearranged according to their order of appearance in the second parent. And we apply the same procedure to complete the second child using the order of appearance in the first parent. This procedure is repeated until the crossing rate ($T_c = 0.8$) is reached (80% of children population size is reached). For diversification, what remains of the population (2 individuals) will be mutated with applying swap mutation according to a fixed probability ($T_m = 0.2$).

Step 3: Population $P_{node/vehicle/depot}$

For each depot, the number of vehicles used and the order of delivery and pick-up within each route are specified by the population named $P_{node/vehicle/depot}$. To construct this population type we should follow three steps: Knowing the number of vehicles available in each depot, we start by creating $2N$ individuals of a new population that indicates the number of nodes visited by each vehicle, named $P_{vehicle/depot}$. After, we affect visited couples to vehicles by coding each individual of the population $P_{couple/depot}$ created in phase 3 by an individual of the population $P_{vehicle/depot}$. This new population, named $P_{couple/vehicle/depot}$, verifies that each couple belongs exactly to same route. To find, for each depot the order in which all pick-up and delivery nodes are visited, we develop a heuristic algorithm. Its principle is to choose randomly, in each route, a starting node from the assigned couples. Then, we follow it by the closest node. Our heuristic minimizes the total distance traveled for each individual of $P_{node/vehicle/depot}$.

Figure 2 and Figure 3 show an example of the individual of $P_{vehicle/depot}$ and $P_{node/vehicle/depot}$.

	k1	k2	k3
Depot1	6	2	0
Depot2	4	4	
Depot3	4	0	

Figure 2: Example of individual of $P_{vehicle/depot}$

For this example we have a vehicles total number equal to 7 and 10 couples customer/supplier which are defined as follows:

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

Figure 3: Example of individual of $P_{node/vehicle/depot}$

$C_1(13, 8), C_2(10, 4), C_3(20, 12), C_4(5, 17), C_5(7, 19), C_6(18, 2), C_7(14, 3), C_8(16, 6), C_9(9, 1), C_{10}(11, 15)$.

Heuristics for creation of feasible solutions

Each individual of population $P_{node/vehicle/depot}$ must respect different constraints. The precedence constraint ensures that each delivery point, on the same route, is not visited before its supplier. The capacity correction constraint ensures that the total load of the vehicle must be smaller than or equal to the maximum capacity of the vehicle. The heuristics algorithms for precedence and capacity corrections procedures, to transform each individual into feasible solution, are explained in details in [1].

The structure of the genetic algorithm proposed for the m-MDPDPTW optimization is illustrated in Table 1. In our genetic algorithm we use the elitism strategy for each generation, the best N solutions in the current population are copied in the new initial population. The best solution found throughout the search is returned when a fixed number of iterations is reached. This number is determined after several experiments.

Table 1: Structure of the genetic algorithm proposed for the m-MDPDPTW optimization

Begin
Step 1: Apply Depot-Grouping phase
Step 2: Generate randomly an initial population $P_{couple/depot}$ containing N individuals. Repeat until maximum of generation reached.
Step 3: Create a new $P_{couple/depot}$ containing $2 \times N$ individuals. The first part of this population represents one copy of the N individual $P_{Best-couple}$, while the remaining 50 percentage of this population are created by applying GA operators on population $P_{Best-couple}$. We select two parent chromosomes from population of step 2 by using tournament selection. For recombination, we apply our designed crossover and for diversification, we apply swap mutation according to a fixed probability.
Step 4: Generate $P_{vehicle/depot}$ containing $2 \times N$ individuals and respecting constraint vehicle numbers.
Step 5: Apply routing phase to create $P_{node/vehicle/depot}$. This population with size $[2 \times N \times m]$ specifies, for each depot, the number of routes (that are vehicles) and the order of delivery and pick-up within each route.
Step 6: Apply the precedence then the capacity correction procedure, to transform each individual into feasible solution.
Step 7: Calculate for every individual of $P_{node/vehicle/depot}$ fitness values (f_1, f_2) .
Step 8 : Elitism: Copy the N best $P_{couple/depot/vehicle}$ solution into the new initial population.
Increment the generation number
End

4.3 Particle Swarm Optimization for the m-MDPDPTW

General principle of the algorithm PSO proposed

PSO shares many similarities with GAs. All two techniques begin with a group of a randomly generated population; both utilize a fitness value to evaluate the population.

The 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 generating several generations. In every generation, a potential solution to the problem is generated, and then evaluated to record the best solutions found. The solution to our m-MDPDPTW represents the best solution found on all generations.

The main difference between the PSO approaches compared to GA, is that PSO does not have genetic operators such as crossover and mutation. Particles update themselves with the internal velocity. In PSO, only the best particle gives out the information to others. It is an on way information sharing mechanism, the evolution only looks for the best solution.

In most applications, the particles positions represent the solutions of the problem studied, but in our case the solution to m-MDPDPTW is decoded from the new particle position.

The travel strategy of a particle i is illustrated in Figure 4.

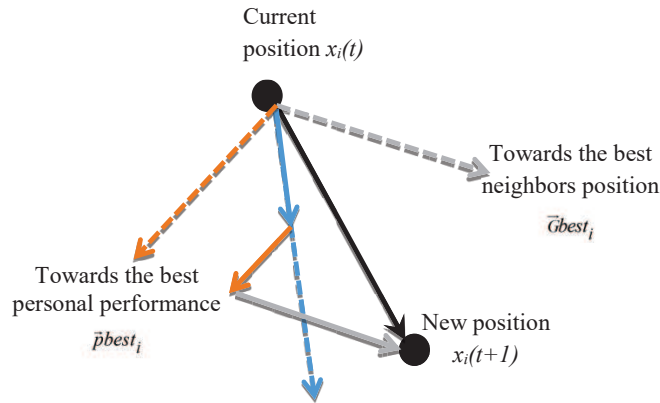


Figure 4: Analysis of the particle travel

For the creation of the initial swarm $P_{node/depot}$ and the Decoding of the initial solutions: routing phase for the creation of the swarm $P_{node/vehicle/depot}$ we follow the same steps as in the genetic algorithm.

Initializing and updating speed and position

We consider that the particles movement is controlled by the limitation of their traveled maximum distance during iteration. Thus, in order to escape the deviation problem of the particles from the search space, we use a technique of interval confinement. The speed of each particle is initialized by values between 0 and n (n is the nodes number to be visited).

The update of the speed and the best position is obtained by the particle ($\vec{P}best_i$: personal best) and by the swarm ($\vec{G}best_i$: global best), using equations (15), (16) and (17). The new position of the particle is calculated from equation (18).

$$\vec{P}best_i(t+1) = \begin{cases} \vec{x}_i(t+1), & \text{if } f(\vec{x}_i(t+1)) \text{ is better than } f(\vec{P}best_i(t)) \\ \vec{P}best_i(t), & \text{else} \end{cases} \quad (15)$$

$$\vec{G}best(t+1) = \arg \min_{\vec{P}best_i} f(\vec{P}best_i(t+1)), 1 \leq i \leq N \quad (16)$$

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

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

with :

$x_{i,j}(t)$: The position of the particle i in dimension j at time t ;

$v_{i,j}(t)$: The speed of the particle i in dimension j at time t ;

$pbest_{i,j}(t)$: The best position obtained by the particle i in dimension j at times t ;

$gbest_{i,j}(t)$: The best position known by the particle i at time t ;

c_1, c_2 : Acceleration coefficients;

r_1, r_2 : Random numbers drawn uniformly in $[0, 1]$, at each iteration t and for each dimension j ;

$w \vec{v}_i$: The inertia component of the movement;

$c_1 r_1 (pbest_i - \vec{x}_i)$: The cognitive component of the particle movement (moving to its best known position);

$c_2 r_2 (gbest_i - \vec{x}_i)$: The social component of the particle movement (closer to the best position of its neighbors).

Decoding the new particle position

The elements of the new particle position after its updating do not reveal directly the nodes index in the route. A decoding phase of the new visit order is therefore necessary in order to find the solutions adapted to m-MDPDPTW.

We consider that the speed is defined as a probability vector, where the value of each element corresponds to the probability of its permutation in the route.

The decoding phase of the new particles gives us the new permutation of the nodes order in the route.

In the same route, nodes will be visited from the nearest to the furthest. If there are several nodes that are in the same position, then they will be visited according to the speed of their movement. The node with the highest speed will be visited first.

We therefore reorder in ascending order of the values of $\vec{x}_i(t+1)$.

These values are subsequently replaced by the corresponding node index to build the new visit order of the nodes.

The structure of the *adaptation algorithm of the PSO for the m-MDPDPTW optimization* is illustrated in the following.

n : nodes number to visit; f : function to minimize; T : maximum number of iterations; dep : number of depot; $t = 0$; $x_{max} = n$; $x_{min} = 0$;

1. Initialization : N_e : Swarm size ; c_1 ; c_2 ; r_1 ; r_2 ;
2. Generating N_e initial particles of $P_{node/depot}$;
3. Application of the routing phase for the creation of initial solutions $P_{node/vehicle/depot}$;

4. Application of the corrections heuristics of precedence, capacity and belonging of each couple to the same route;
5. Evaluate the N_e initial particles of $P_{node/vehicle/depot}$ (1);
6. Initialize the speed of each particle by random values between $[0; n]$;
7. Initialize the position of each particle

$$\vec{x}(0) = P_{node/depot}$$

8. Initialize

$$\vec{P}best_i(0) = P_{node/vehicle/depot}[i]$$

9. Initialize the best solution found by the swarm

$$\vec{G}best(0) = \arg \min_{\vec{P}best_i} f(\vec{P}best_i(0))$$

10. While ($t < T$) do;
11. For $i = 1$ to N_e do;
12. Speed Update (18);
13. Update of the new position (18);
14. Verify the new particle does not leave the search space:

$$\begin{aligned} &\text{if}(\vec{x}_i(t+1) > x_{\max}) \text{then} \vec{x}_i(t+1) = x_{\max}; \\ &\text{elseif}(\vec{x}_i(t+1) < x_{\min}) \text{then} \vec{x}_i(t+1) = x_{\min}; \end{aligned}$$

15. Decoding the new particles position;
16. Generating new solutions: Routing phase;
17. Apply corrections heuristics;
18. Evaluate the N_e particles of $P_{node/vehicle/depot}$ (1);
19. Save the best result found by the particle;
- For $i = 1$ to N_e do;

$$\begin{aligned} &\text{if} f(P_{node/vehicle/depot}(i)) \leq f(pbest_i(t)) \text{then} \\ &Pbest_i(t+1) = f(x(t+1)) \\ &\text{else} Pbest_i(t+1) = Pbest_i(t) \end{aligned}$$

End For;

20. Update the best solution found by the swarm

$$\vec{G}best(t+1) = \arg \min_{\vec{P}best_i} f(\vec{P}best_i(t+1))$$

21. End For;
22. End While;
Save the best solution found on all generation;

The same, to find the optimal solution, we browse the swarm $P_{node/vehicle/depot}$ particle by particle, to determine that which minimizes our objective function (1).

5 Computational results

This section describes computational results to assess and compare the performance of the two proposed algorithms. We programmed the algorithms in C language using Microsoft Visual Studio2010 and ran them on Mobile Intel Core i7, CPU 2.50 GHz and 6.00 Go memory (RAM) under the operating system Windows 8 Professional.

In literature, the existing benchmark instances do not combine all the characteristics of the m- MDPDPTW. So our simulation results are obtained using the problem instances generated by Li and Lim [10] which are created for the single depot pickup and delivery problem with time windows. We only added the additional information that can accommodate the m-MDPDPTW by generating multi depot locations using the algorithm described in our work [2]. The locations of the different depots considered in our simulations for each type of Li and Lim's problems are summarized in Table 2.

Table 2: Location of the different depots

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

The parameters of the two approaches are selected after many experiments.

The parameters values for PSO are: Number of swarms=1, swarm size=500 particles, the maximum number of generation=1000 iterations. The inertia weight linearly decreasing from 0.8 to 0.5, the acceleration constants, random numbers for personal best and global best are 0.2, 0.2, 0.3 and 0.5.

The parameters values for GA are: Population size equal to number of particles=500, maximum number of generations=1000. The crossover rate and mutation rate are 0.8 and 0.2, respectively.

The best results obtained from our two proposed GA and PSO approaches solving our m-MDPDPTW were compared as shows in Table 3.

We consider the depot locations already given in the Table 1 and we evaluate the total distance travelled, considered in equation 1, for some selected categories of problem instances with different sizes: clustered locations with short schedule horizon (LC101 and LC102), clustered locations with long schedule horizon (LC201 and LC202), randomly distributed locations with short schedule horizon (LR101 and LR102), randomly distributed locations with long schedule horizon (LR201 and LR202), and finally, problems category that have partially random and

partially clustered locations with a tight time window width (LRC101 and LRC102) and with a large time window width (LRC201 and LRC202).

Table 3: Comparison of the results of our GA and PSO approaches to solve our m-MDPDPTW problem

Instances	Best distance of GA approach	Best distance of PSO approach
LC101	916.078	1839.3962
LC102	910.598	1606.801
LC201	1081.395	947.914
LC202	1204.900	1472.292
LR101	1938.980	2137.17
LR102	1898.109	2128.639
LR201	1979.990	2060.405
LR202	1241.264	2497.547
LRC101	1079.137	2513.376
LRC102	1129.326	2567.438
LRC201	1453.306	2937.202
LRC202	1421.923	2829.999

From Table 3, we can observe that the values of the total distance traveled given by our approach based on the GA are better than those given by the PSO. This last only gave a low improvement of the fitness for clustered locations instance LC201.

However, these results show the strength of GAs that is in the parallel nature of their research and their ability to manage multiple large data sets.

Compared with GA, the advantages of PSO are that it is easy to implement and there are few parameters to adjust. A source of the AG's power is their used genetic operators: crossover and mutation. The crossover attempts to preserve the beneficial aspects of candidate solutions and to eliminate undesirable components, while the random nature of mutation is probably more likely to degrade a strong candidate solution than to improve it.

Through its genetic operators, even weak solutions may continue to be part of the makeup of future candidate solutions and thus allows the creation of new solutions that have, a higher probability of exhibiting a good performance. This tends to make the algorithm likely to converge towards high quality solutions within a few generations.

To validate and evaluate our proposed AG approach, we solve the single depot PDPTW

Table 4: Comparison of the results of our AG approaches using the Li and Lim instances of single depot problem

Case	Best known solutions of Li and Lim	Best known solutions of our GA approach
LC101	828.94	861.24
LC102	828.94	926.941
LC103	1035.35	959.303
LC104	860.01	940.05
LC105	828.94	867.609
LC106	828.94	955.339
LC107	828.94	922.66
LC108	826.44	914.211
LC109	1000.60	869.537

problem. The comparison with the best results published by Li and Lim with respect to the total traveled distance minimization is shown in Table 4.

The proposed GA approach shows the promising result in the clustered problem with short schedule horizon. Compared with the Lim and Lim's best solutions approach for the PDPTW problem with single depot, this algorithm produced good quality results that are sometimes even better than the results obtained, making it even more suitable for population-based solution algorithms.

Furthermore, the total traveled distances for the benchmark data sets are near the best result and are a little better than the best known distance in LC103 and LC109 category.

6 Conclusion

In this paper we have developed two meta-heuristic approaches based on the Genetic Algorithm (GA) and Particle swarm optimization (PSO), in order to compare and identify the best approach that can be used for solving a multi depots multi vehicles pickup and delivery problems with time windows (m-MPDPTW). We proposed a brief literature review on the VRP and the PDPTW. The mathematical formulation of our problem was subsequently presented. Then, we have detailed the use of GA and PSO algorithm to determine the solution which minimizes our

objective function. Simulation was presented in a last part by using benchmark's data. The experimental results on a large number of benchmark instances indicate that the use of GA seems to be the most favorable method to reach final best solutions for our m-MDPDPTW. For our future work, we propose to study the multi-objective optimization by comparing the genetic algorithms, the Particle Swarm Optimization and other metaheuristic methods.

Bibliography

- [1] Ben Alaia, E.; Harbaoui, D.I.; Bouchriha, H.; Borne, P. (2015); Genetic Algorithm for Multi-Criteria Optimization of Multi-Depots Pickup and Delivery Problem with Time Windows and multi vehicles, *Acta Polytechnica Hungarica*, 12(8), 155–174, 2015.
- [2] Ben Alaia, E.; Harbaoui, D.I.; 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, *IEEE International Conference on Industrial Engineering and Systems Management*, Seville, Spain, 695-701, 2015.
- [3] Ben Alaia, E.; Harbaoui, D.I.; Bouchriha, H.; Borne, P. (2015); Genetic Algorithm with Pareto Front selection for Multi-Criteria Optimization of Multi-Depots and multi- vehicle Pickup and Delivery Problems with Time Windows, *IEEE International Conference on Sciences and Techniques of Automatic control and computer engineering*, 21-23 December, Hammamet, Tunisie, 488-493, 2014.
- [4] Fatma, P.G.; Fulya, A.; Ismail, K. (2012); A Hybrid Particle Swarm Optimization for Vehicle Routing Problem with Simultaneous Pickup and Delivery, *Computers and Industrial Engineering*, 1-6, 2012.
- [5] Harbaoui, D.I., Ben Alaia, E.; Borne, P. (2015); Heuristic Approach for The Optimization of The Dynamic Multi-Vehicle Pickup and Delivery Problem with Time Windows, *IEEE International Conference on Industrial Engineering and Systems Management*, 21–23 October, Seville, Spain, 488-493, 2015.
- [6] Harbaoui, D.I.; Kammarti, R., Ksouri, M.; Borne, P. (2011); Multi-objective optimization for the mpdptw : Aggregation methode with use of genetic algorithm and lower bounds, *International Journal of Computers Communications & Control*, 6(2), 246–257, 2011.
- [7] Lau, H.C.W.; Chan, T.M.; Tsui, W.T.; Pang, W.K. (2010); Application of genetic algorithms to solve the multidepot vehicle routing problem, *IEEE Transactions on Automation Science and Engineering*, 7(2), 383–392, 2010.
- [8] Lei, W.; Fanhua, M. (2008); An Improved PSO for the Multi-Depot Vehicle Routing Problem with Time Windows, *Pacific-Asia Workshop on Computational Intelligence and Industrial Application*, 852-856, 2008.
- [9] Liu, R.; Xi, X.; Augusto, V.; Rodriguez, C. (2013); Heuristic algorithms for a vehicle routing problem with simultaneous delivery and pickup and time windows in home health care, *European Journal of Operational Research*, 230(3), 475-486, 2013.
- [10] Li, H.; Lim, A. (2001); A metaheuristic for the pickup and delivery problem with time windows, *In IEEE International Conference on Tools with Artificial Intelligence*, 13, 160–167, 2001.

-
- [11] Marinakis, Y.; Iordanidou, G.R.; Marinaki, M (2013); Particle Swarm Optimization for the Vehicle Routing Problem with Stochastic Demands, *Applied Soft Computing*, 13 (4), 1693–1704, 2013.
- [12] Marinakis, Y.; Marinaki, M. (2010); A hybrid genetic - particle swarm optimization algorithm for the vehicle routing problem, *Expert Systems with Applications*, 37(2), 1446–1455, 2010.
- [13] Nagata, Y.; Kobayashi, S. (2011); A memetic algorithm for the pickup and delivery problem with time windows using selective route exchange crossover, In: *Schaefer R., Cotta C., Kolodziej J., Rudolph G. (eds), Parallel Problem Solving from Nature, PPSN XI. PPSN 2010. Lecture Notes in Computer Science*, Springer, Berlin, Heidelberg, 6238, 536–545, 2011.
- [14] Ombuki, B. B.; Hansharin, F. (2009); Using genetic algorithms for multi depot vehicle routing, *Studies in Computational Intelligence*, 161, 77–99, Springer Berlin Heidelberg, 2009.
- [15] Pop, P.C.; Pop Sitar, C.; Zelina, I.; Lupse, V., Chira, C. (2011); Heuristic Algorithms for Solving the Generalized Vehicle Routing Problem, *International Journal of Computers Communications & Control*, 2011(1), 158–165, 2011.
- [16] Shuilong, Z.; Jin, L.; Xueqian, L. (2013); A Hybrid Particle Swarm Optimization Algorithm for Multi-Objective Pickup and Delivery Problem with Time Windows, *Journal of Computers*, 8(10), 2583–2589, 2013.
- [17] Sombuntham, P., Kachitvichayanukul, V. (2010); A Particle Swarm Optimization Algorithm for Multi-depot Vehicle Routing problem with Pickup and Delivery Requests, *The International MultiConference of Engineers and Computer Scientists*, 1-6, 2010.
- [18] Vidal, T.; Crainic, T.G.; Gendreau, M., Prins, C. (2014); Implicit depot assignments and rotations in vehicle routing heuristics, *European Journal of Operational Research*, 237(1), 15–28, 2014.
- [19] Vidal, T.; Crainic, T G.; Gendreau, M.; Prins, C. (2013); Heuristics for multiattribute vehicle routing problems : a survey and synthesis, *European Journal of Operational Research*, 231(1), 1–21, 2013.
- [20] Vidal, T.; Crainic, T.G.; Gendreau, M.; Lahrichi, N.; Rei, W. (2012); A hybrid genetic algorithm for multi-depot and periodic vehicle routing problems, *Operations Research*, 60(3), 611–624, 2012.
- [21] Yousefikhoshbakht, M.; Didehvar, F.; Rahmati, F. (2014); An Efficient Solution for the VRP by Using a Hybrid Elite Ant System, *International Journal of Computers Communications & Control*, 9(3), 340–347, 2014.