Multi-Period Pickup and Delivery Problem with Time Windows and Paired Demands

Zaher Al Chami, Hervé Manier, Marie-Ange Manier

▶ To cite this version:

Zaher Al Chami, Hervé Manier, Marie-Ange Manier. Multi-Period Pickup and Delivery Problem with Time Windows and Paired Demands. International Conference on Control, Automation, Robotics and VIsion, Nov 2018, Singapore, Singapore. hal-02952672

HAL Id: hal-02952672

https://hal.science/hal-02952672

Submitted on 29 Sep 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Multi-Period Pickup and Delivery Problem with Time Windows and Paired Demands

Zaher Al Chami
Univ. Bourgogne
Franche-Comté FEMTO-ST
Institute/CNRS,
Rue Thierry-Mieg (UTBM),
90010 Belfort Cedex, France
zaher.al-chami@utbm.fr

Hervé Manier
Univ. Bourgogne
Franche-Comté FEMTO-ST
Institute/CNRS,
Rue Thierry-Mieg (UTBM),
90010 Belfort Cedex, France
herve.manier@utbm.fr

Marie-Ange Manier
Univ. Bourgogne
Franche-Comté FEMTO-ST
Institute/CNRS,
Rue Thierry-Mieg (UTBM),
90010 Belfort Cedex, France
marie-ange.manier@utbm.fr

***Abstract*— This paper addresses the Multi-Period Pickup and Delivery Problem with Time Windows and Paired Demands (Mu-PDPTWPD). A first strategy consists in dealing with the problem as a whole. A second one is to consider the studied problem as a sequence of mono-period selective PDPTWPDs. The Mu-PDPTWPD is an extension of well-known pick up and delivery problem where vehicles must satisfy a set of transportation requests under many constraints. The Multi-Period aspect is added to the problem to match real-world applications where sites must be visited one time during a set of periods. In this paper, we propose several methods to solve this problem. The efficiency of our approaches is validated by tests on two sets of new generated instances. A comparison between all proposed approaches is done in order to trace the advantages and drawbacks for each of them.**

I. INTRODUCTION AND PROBLEM STATEMENT

The Pickup and Delivery Problem (PDP) is a well-known combinatorial optimization problem with a wide range of real-world applications. It aims at routing a fleet of vehicles with a given capacity based at a depot to satisfy a set of transportation requests. The route of each vehicle starts and finishes at the depot. Furthermore, the demand of a given node must not exceed the maximal capacity of vehicles. Each request is represented by a known quantity to be loaded from a pickup location, then to be unloaded at a delivery location. Moreover, the depot can be treated as the place where the vehicles start and end theirs trips but also as a supplier to a set of customers and as a customer to a set of suppliers. Hence, we consider three types of flows: the first one is when vehicle picks up the goods demanded by a specific customer from its supplier. In this case, the depot is nor a customer, neither a supplier; the second one is when the depot plays the role of supplier so the vehicle must load goods from the depot to deliver its customer(s); and the last one is when it picks up the goods from supplier(s) and returns it back to the depot. Then it is considered as a customer. In our problem, the load of each vehicle when leaving the depot or when returning to the depot may be different from zero. In fact, the initial and final loads are two decision variables that should be calculated. The Mu-PDPTWPD is a NP-hard problem as it is an extension of the PDP [1]. It can be modeled by a graph where the edges are roads with assigned distances. Each node represents either a delivery location (customer) where a given demand must be unloaded, or a pickup location (supplier) where a given quantity must be loaded. In our variant, there may be several periods in which the vehicles can perform the delivery or pickup operations. The period can be defined as a day or any other time slot. Each site can accept to be visited in one or several periods before the planned period of its visit. Then, it proposes a time window per period and the vehicle visiting the node cannot arrive after the end of this time interval. But if it arrives earlier than the beginning of the associated time window, then it must wait for the earliest date of this time window. A paired demand links a specific origin to its destination. Hence, precedence constraints must be considered to ensure that each supplier is visited before its customer both by the same vehicle and during the same period. This last constraint means a strong but realistic hypothesis in the context of urban logistics in which we address the studied problem: each route must be performed within a given period.

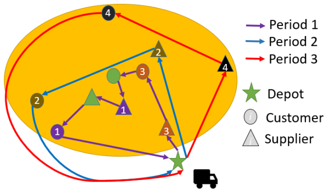


Fig. 1: Example of the Mu-PDPTWPD

Fig. 1 shows an example of the Mu-PDPTWPD with three different periods, a single depot represented by a green star, a single vehicle and a set of nodes composed of five customers (circles) and five suppliers (triangles). A given transportation demand is identified thanks to a color and a number associated with the corresponding customer and supplier nodes. In figure 1, we can identify nodes without number, which corresponds to suppliers or customers paired with the depot. The characteristics of the Mu-PDPTWPD studied in this paper can be summarized as follows :

(1) A fleet of heterogeneous vehicles based at the depot.
(2) A site can accept to be visited in one or several periods

after defining a time window per accepted period. It must be visited in one of it(s) accepted period(s).

(3) A set of transportation requests must be satisfied over all periods under condition that each request must be satisfied in a given period (we don't take into consideration the possibility of splitting a request over several periods).

(4) The depot is a logistics platform located in the periphery of the city where all vehicles start and end their trips. It receives goods from suppliers located outside the city which will be then delivered in the urban area. It also receives goods from suppliers located inside the city in order to send them to other cities. So the depot may represent the origin or the destination for one or several transportation requests. Otherwise, the origin will be any other supplier and the destination any other customer located both inside the city.

In terms of complexity, solving a multi-period variant with T periods is equivalent to solving T^n times the mono-period variant (if the n demands may be satisfied at any period). Indeed compared with the mono-period routing problem, the multi-period variant brings an additional challenge: the assignment of one period to each site to be visited. For this goal, we have considered several strategies:

- finding a global assignment for the whole problem. This involves to handle all assignment possibilities which allows to find the optimal solution. But this one is practically difficult to reach in reasonable time for big size instances;
- adopting a decomposition approach by partitioning the whole problem into several sequential sub-problems, each one being a mono-period selective PDPTWPD.

At a n^{th} period, the set of considered demands is composed of the requests not satisfied at the n-1 previous periods and which can be achieved within the current period. Then the current PDPTWPD to solve depends on the results of the PDPTWPDs solved at the n-1 previous periods. Besides, during this current period, the vehicles will not always be able to serve all the considered requests because of the capacity and/or the time window constraints. Then we must solve a selective problem. Finally, after solving sequentially all the mono-period selective problems, the final solution will be feasible if all the demands are satisfied.

Figure 2 illustrates the same problem as in figure 1, but studied as three sequential mono-period selective problems (Mo-SPDPTWPDs). In the first period, the nodes numbered 2 and 4 were not selected either because they did not accept to be visited in this period or because the vehicle cannot visit these nodes after serving the other ones. In the second period, we notice that the nodes served in the first period have disappeared and we can now choose between the nodes 2 and 4 and so on. In this example, we have solved the sequence and we got an optimal solution because we have the same routes as in figure 1, which is not always the case.

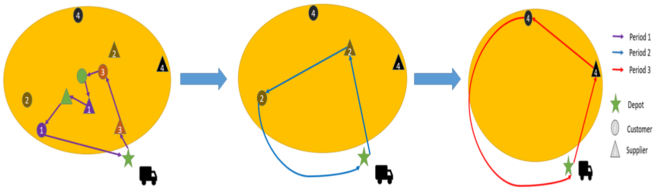


Fig. 2: Example of a sequence of the Mo-SPDPTWPD

II. Literature review

In this section, we present a brief state of the art firstly on the pickup and delivery problem and its variants, then on the multi-period aspect on different routing problems. Many reviews of the PDP literature have been presented in [2] and [3]. Hereafter, we focused on the studied variants.

In the selective PDP, vehicles must transport goods from suppliers to their corresponding customers in the most profitable way. For example, [4] has introduced a memetic algorithm to solve the profitable PDPTWPD. For this variant, the authors have considered that only a part of the available request can be served and the remaining ones are also served by subcontracted carriers. Moreover, only the flow between suppliers and customers is taken into account. In other words, the depot is only the starting and the ending point of each vehicle trip. Recently, we elaborated a MILP model in [5] that deals with the mono-objective Mo-SPDPTWPD, with the three types of flow defined in the first section.

The studies quoted above deal with PDP and its variants in a mono-period context, with a time window for each site. There are few studies that address the multi-period aspect or the multiple time windows possibility or both of them. Hence, we present some of these works which are not necessary dedicated to solve the PDP. The multiple time windows mean that a customer can has zero, one or several time windows per period within which loading or unloading of goods is allowed. Contrary to the multi-Period variant, in the periodic routing problem, each site has to be visited periodically (more than one time) during the periods ([6], [7]). The Mu-PDPTWPD is rarely studied in the literature. Table I gives an overview on the works addressing either the multi-period or the multiple time windows on different routing problems. The main contributions of this paper includes:

i) Introducing three different approaches based on a sequential strategy to solve the Mu-PDPTWPD.
ii) Generating of the first instances in the literature dealing with the considered variant to test our approaches.

In the next section, we propose three sequential approaches to solve the Mu-PDPTWPD.

I. Multi-period and Multiple Time windows studies

Reference	Routing problem	Multi-Period	Multiple Time Windows	Resolution method
[8]	OPTW	X	X	Variable neighborhood search
[9]	TOPTW		X	Hybrid ILS-GRASP
[10]	TOPTW		X	Simulated annealing
[11]	VRPTW		X	Ant colony
[12]	VRPTW		X	Hybrid tabu search
[13]	VRPTW		X	Quantum genetic algorithm
[14]	PDPTWPD	X		Linear programming MILP

OP : Orienteering Problem; TOP : Team Orienteering Problem

III. Sequential approaches for the Mu-PDPTWPD

A. Global-MILP approach

We first present a mathematical formulation for the Mu-PDPTWPD, characterized by the following parameters:

Data:

- P : set of all periods,
- C : set of all customers,
- S : set of all suppliers,
- Nodes : set of all suppliers and customers (C ∪ S),
- 0 : Id of the depot,
- N : set of all nodes and depot,
- $Supplier_i$: supplier associated with customer i,
- $Customer_i$: customer associated with supplier i,
- q_i : goods quantity requested by node i
  - If q_i >0 then i ∈ S, else i ∈ C
- $[e_{(i,t)}, l_{(i,t)}]$: time window interval associated to node i at period t,
- ST_i : time of service at node i,
- M : very big number,
- V : set of available vehicles,
- Q_k : maximal capacity of vehicle k,
- $Speed_k$: the average speed of vehicle k,
- d_{ij} : traveling distance from vertex i to vertex j.

Variables:

- $X_{i,j,k,t} = \begin{cases} 1 & \text{If node j is visited directly after node i} \\ & \text{by vehicle k during period t} \\ 0 & \text{Otherwise} \end{cases}$
- $A_{i,k,t}$: starting service time of vehicle k at node i during period t,
- $D_{i,k,t}$: departure time of vehicle k from node i during period t,
- $Y_{i,k,t}$: quantity presented in vehicle k visiting node i during period t,
- $E_{i,t}$ = 1 if node i is visited during period t and 0 otherwise.

Mixed Linear Program for the Mu-PDPTWPD:

$$Minimize \quad \sum_{i\in N}\sum_{j\in N}\sum_{k\in V}\sum_{t\in P} d_{ij}X_{i,j,k,t} \tag{1}$$

Subject to:

$$\sum_{i\in N}\sum_{k\in V}\sum_{t\in P} X_{i,j,k,t} = 1 \qquad \forall j \in Nodes \tag{2}$$

$$\sum_{j\in N}\sum_{k\in V}\sum_{t\in P} X_{i,j,k,t} = 1 \qquad \forall i \in Nodes \tag{3}$$

$$\sum_{i\in N} X_{i,u,k,t} - \sum_{i\in N} X_{u,i,k,t} = 0 \qquad \forall k \in V; \forall u \in Nodes; \forall t \in P \tag{4}$$

$$\sum_{i\in Nodes} X_{0,i,k,t} \leq 1 \qquad \forall k \in V; \forall t \in P \tag{5}$$

$$\sum_{i\in N}\sum_{j\in N} X_{i,j,k,t} \leq M * \sum_{i\in Nodes} X_{0,i,k,t} \qquad \forall k \in V; \forall t \in P \tag{6}$$

$$\sum_{j\in Nodes} X_{j,0,k,t} = \sum_{i\in Nodes} X_{0,i,k,t} \qquad \forall k \in V; \forall t \in P \tag{7}$$

$$D_{0,k,t} = 0 \qquad \forall k \in V; \forall t \in P \tag{8}$$

$$D_{i,k,t} \geq A_{i,k,t} + ST_i - M * (1 - \sum_{j\in N} X_{i,j,k,t}) \qquad \forall i \in Nodes; \forall k \in V; \forall t \in P \tag{9}$$

$$D_{i,k,t} - M * (1 - \sum_{j\in N} X_{i,j,k,t}) \leq A_{i,k,t} + ST_i \qquad \forall i \in Nodes; \forall k \in V; \forall t \in P \tag{10}$$

$$D_{i,k,t} + \frac{d_{ij}}{Speed_k} - M * (1 - X_{i,j,k,t}) \leq A_{j,k,t} \qquad \forall i, j \in N; \forall k \in V; \forall t \in P \tag{11}$$

$$D_{i,k,t} \leq M * \sum_{j\in N} X_{i,j,k,t} \qquad \forall i \in Nodes; \forall k \in V; t \in P \tag{12}$$

$$A_{i,k,t} \leq M * \sum_{j\in N} X_{j,i,k,t} \qquad \forall i \in Nodes; \forall k \in V; \forall t \in P \tag{13}$$

$$A_{0,k,t} \leq l_{(0,t)} \qquad \forall k \in V; \forall t \in P \tag{14}$$

$$\sum_{t\in P} E_{i,t} = 1 \qquad \forall i \in Nodes \tag{15}$$

$$E_{i,t} = \sum_{j\in N}\sum_{k\in V} X_{j,i,k,t} \qquad \forall i \in Nodes; \forall t \in P \tag{16}$$

$$\sum_{k\in V} A_{i,k,t} - M * (1 - E_{i,t}) \leq l_{(i,t)} \qquad \forall i \in Nodes; \forall t \in P \tag{17}$$

$$e_{(i,t)} \leq \sum_{k\in V} A_{i,k,t} + M * (1 - E_{i,t}) \qquad \forall i \in Nodes; \forall t \in P \tag{18}$$

$$Y_{0,k,t} = \sum_{i\in C/Supplier_i=0} (-q_i * \sum_{j\in N} X_{j,i,k,t}) \qquad \forall k \in V; \forall t \in P \tag{19}$$

$$Y_{i,k,t} + q_j - M * (1 - X_{i,j,k,t}) \leq Y_{j,k,t} \qquad \forall i \in N; \forall j \in Nodes; \forall k \in V; \forall t \in P \tag{20}$$

$$Y_{j,k,t} - q_j - M * (1 - X_{i,j,k,t}) \leq Y_{i,k,t} \qquad \forall i \in N; \forall j \in Nodes; \forall k \in V; \forall t \in P \tag{21}$$

$$Y_{i,k,t} \leq M * \sum_{j\in N} X_{j,i,k,t} \qquad \forall i \in N; \forall k \in V; \forall t \in P \tag{22}$$

$$0 \leq Y_{i,k,t} \leq Q_k \qquad \forall i \in N; \forall k \in V; \forall t \in P \tag{23}$$

$$\sum_{u\in N} X_{Supplier_i,u,k,t} = \sum_{u\in N} X_{u,i,k,t} \qquad \forall i \in C/Supplier_i \in S; \forall k \in V; \forall t \in P \tag{24}$$

$$D_{Supplier_i,k,t} \leq A_{i,k,t} \qquad \forall i \in C; \forall k \in V; \forall t \in P \tag{25}$$

The objective function (1) is the minimization of distance. Constraints (2) and (3) guarantee that each node is visited once during the set of periods. The routing continuity on each period for each vehicle is ensured by constraints (4). (5), (6) and (7) guarantee that each route start and finish at a depot. Constraints (8) ensure that each vehicle route begins on each period at time 0. (9), (10), (11), (12) and (13) update starting service and departure times. Constraints (14) ensure that all vehicles finish their trips before the closing time of the depot. (15) and (16) ensure that each node is visited during only one period. The compliance of time windows is ensured by constraints (17) and (18). (19), (20), (21), (22) and (23) allow to respect the capacity of each vehicle. Constraints (24) and (25) ensure that each vehicle visits the supplier before its customer. This MILP addresses the global problem and as a consequence, the complexity is very high. Hence, using the well-known solver CPLEX, we can solve only very small instances to the optimality in a reasonable time.

The advantage of this MILP (called Global-MILP to differ from other algorithms) is that it gives us the optimal solution of the Mu-PDPTWPD. But, the long solving time represents its big drawback. To deal with it, we present different approaches for the second considered strategy which tackles the Mu-PDPTWPD by considering it as a sequence of Mo-SPDPTWPDs. This strategy may not always give the optimal solution because the final solution is the sum of the solutions for each period. In addition, obtaining a feasible solution even if it exists is not guaranteed with such a strategy.

B. Sequential-MILP approach

Figure 4 represents our first sequential approach, denoted Sequential-MILP, which deals chronologically with each period. We start with the first period and run our MILP developed in [5] for the Mo-SPDPTWPD to get the optimal solution on this period. After that, we compute the distance obtained in order to add its value to the final distance and we insert the visited sites into a list of assigned sites (those nodes cannot be visited over the left periods). Finally, we solve the problem for the next period and we repeat this process to finish all periods, while all requests are not satisfied. In the global Mu-PDPTWPD, we aim at minimizing the total distance. But we cannot use this objective function on each period because of the selective aspect of each mono-period problem. Indeed, in a selective problem, the minimal distance is zero with no visited sites. Hence, we use the maximization of profit as our objective function. This choice is first justified as in the literature, the main objective for selective problems is to maximize the profit. Moreover, at equal profits for all the requests, maximizing the profit is the same as maximizing the number of visited sites. This may contribute to obtain a global feasible solution (with all requests satisfied).

After solving the problem, the total distance is evaluated for further comparisons. With our sequential MILP approach, the distance obtained may be not the optimal one, neither at each period nor globally. To deal with this drawback and improve the performance of our sequential strategy, we develop in the next subsections two bi-objective methods which aim

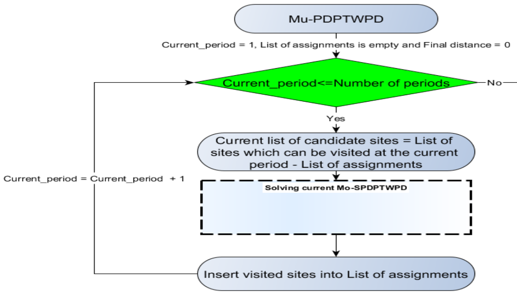

Mu-PDPTWPD
Current_period = 1, List of assignments is empty and Final distance = 0
Current_period<=Number of periods
No
END
Yes
Current list of candidate sites = List of sites which can be visited at the current period - List of assignments
Current_period = Current_period + 1
Solving current Mo-SPDPTWPD
Insert visited sites into List of assignments

Fig. 3: Sequential algorithm structure

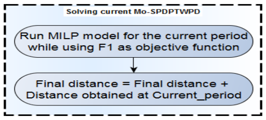

Solving current Mo-SPDPTWPD
Run MILP model for the current period while using F1 as objective function
Final distance = Final distance + Distance obtained at Current_period

Fig. 4: Sequential-MILP approach

at both minimizing the total distance and maximizing the profit for each period.

C. Sequential-Lexicographic approach

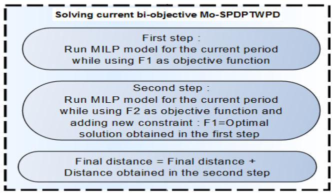

Solving current bi-objective Mo-SPDPTWPD
First step :
Run MILP model for the current period while using F1 as objective function
Second step :
Run MILP model for the current period while using F2 as objective function and adding new constraint : F1=Optimal solution obtained in the first step
Final distance = Final distance + Distance obtained in the second step

Fig. 5: Sequential-Lexicographic approach

The lexicographic approach represents one possible method to address multi-objective problems. It consists in establishing a pre-defined ordering between the two objective functions and then, each function is optimized one at a time in each period. We called this approach Sequential-Lexicographic and it is detailed in figure 5. In our case, the collected profit is first maximized and then the distance is minimized because of the selective characteristics of the problem. We have tested the efficiency of this approach by using it to solve the bi-objective Mo-SPDPTWPD [15].

D. Sequential Hybrid genetic algorithm

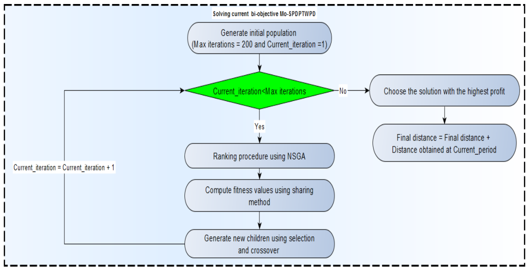

Solving current bi-objective Mo-SPDPTWPD
Generate initial population (Max iterations = 200 and Current_iteration =1)
Current_iteration<Max iterations
No
Choose the solution with the highest profit
Final distance = Final distance + Distance obtained at Current_period
Yes
Ranking procedure using NSGA
Current_iteration = Current_iteration + 1
Compute fitness values using sharing method
Generate new children using selection and crossover

Fig. 6: Sequential-HGA approach

The main limitation observed with exact methods is the high computational time needed to solve big size instances (table II). Then, we propose a Hybrid Genetic Algorithm (HGA) to address the studied problem. We call it Sequential-HGA (figure 6) and we still aim both to minimize the total distance and to maximize the collected profit for each period. It is an extension of a genetic algorithm developed in [16] to solve the Mo-SPDPTWPD, where each solution is represented by a chromosome composed of genes. Each gene is a route performed by one vehicle.

The initial population is composed of Max feasible solutions created randomly. Our genetic algorithm is hybridized with a local search procedure which improves each given route in terms of distance by testing CP permutations. The parameters tunning provided the values CP=15 and Max=200.

The evaluation of chromosomes is based on ranking procedure using Non-dominated Sorting Genetic Algorithm (NSGA) [17]. We use the obtained ranks and the sharing method to assign a fitness value for each individual [18]. The sharing method maintains the diversity of population. Thanks to a density parameter, This method allows us to distribute the population over the Pareto front by increasing the probability to select solutions in uncrowded areas [18].

For mating and reproduction, we introduce a new selection strategy, based on two consecutive roulette wheels. It aims at giving a bigger possibility to the best chromosomes to be selected while keeping the random aspect of the selection. The first wheel is based on the previous obtained ranks and is dedicated to determine the rank of future parent. The second wheel is based on the fitness. It chooses the final parent among all chromosomes associated with this computed rank. Then, a crossover operator is applied on the selected parents in order to get two children: child 1 by getting the genes with highest profit from its parents and child 2 by getting the genes with the lowest distance. A repairing procedure enables us to remove the duplicated sites. After being improved with our local search method, the children will replaced their parents if they dominate at least one of them.

IV. Experimental results

In this section, we give some experimental results obtained by running the four presented algorithms to solve the Mu-PDPTWPD: Global-MILP, Sequential-MILP, Sequential-Lexicographic and Sequential-HGA. For the three first ones, we use the commercial solver CPLEX 12.4. We coded our Sequential-HGA in C++. All tests have been done using an Intel core i7, 2.80 GHz CPU and 16 GB of RAM computer with Windows 10 as operating system. The tests were achieved on two sets of new instances we generated because of a lack of benchmark instances:

- Set 1 : we randomly generated 30 instances with a fixed number of periods equal to 10. A site can choose the period(s) in which it could be visited. It may be associated to several time windows but at most one per period. In this set, the same profit is associated with all the requests.
- Set 2 : 18 instances were generated from Li and Lim′s benchmark instances [19] which we have adapted to the Mu-PDPTWPD. All sites accept to be visited at any period, which means that for each site there is exactly one associated time window per period. Moreover two requests may be associated with different profits. For set 2, we determined the number of necessary periods for each instance by using the minimum number of periods required by our three sequential approaches (column 2 in table II), while repeating the solving procedure until all sites were visited.

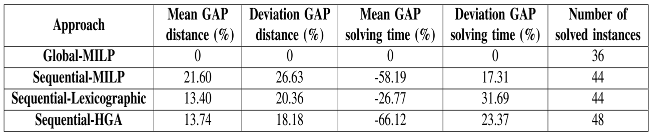
The instance files are available through the link `https://www.dropbox.com/sh/7h4xhoi69ilwxe4/AACa9ybtN_d58vRLVqHg_Cloa?dl=0`. Each instance is labeled **Sx-y-z-i**, x is the type of set, y is the type of instance (R: nodes are randomly distributed, C: nodes are totally clustered, and RC: nodes are partially clustered and partially randomly distributed), z is the number of nodes and i identifies the instance. Table II shows the results for the considered approaches in terms of total obtained distance and CPU time in seconds. The term ”Time limit” means that the approach has not solved to the optimality the corresponding instances after two hours of running. For instances with 50 nodes, the Global-MILP solved to the optimality only one instance over ten from set 1 and three instances over six from set 2.

II. Experimental results for different approaches

Instance	Number of periods	Global-MILP		Sequential-MILP		Sequential-Lexicographic		Sequential-HGA	
		Distance	CPU time (s)	Distance	CPU time (s)	Distance	CPU time (s)	Distance	CPU time (s)
S1-R-10-1	10	910.659	3.2	991.456	1.34	910.659	2.21	910.659	0.96
S1-R-10-2	10	657.043	5.67	930.628	2.21	930.628	3.45	930.628	0.46
S1-R-10-3	10	821.459	6.98	821.459	3.14	821.549	4.16	821.459	1.05
S1-R-10-4	10	912.905	6.45	1039.51	2.19	1039.51	3.23	1039.51	1.24
S1-R-10-5	10	843.879	6.37	914.891	2.13	914.891	3.15	914.891	1.21
S1-R-10-6	10	930.679	6.87	1011.39	2.57	1011.39	3.72	1011.39	1.45
S1-R-10-7	10	897.632	6.49	1031.62	2.25	1031.62	3.31	1031.62	1.34
S1-R-10-8	10	759.548	7.13	1026.88	2.36	955.985	3.4	955.985	1.51
S1-R-10-9	10	938.955	7.04	1015.72	2.33	1015.72	3.51	1015.72	1.57
S1-R-10-10	10	803.484	7.17	893.013	2.41	893.013	3.53	893.013	1.62
S1-R-20-1	10	1814.512	10.73	1986.65	7.23	1934.25	9.13	1903.12	9.53
S1-R-20-2	10	1706.712	8.95	2016.43	4.25	1855.51	7.21	1855.51	6.15
S1-R-20-3	10	1829.728	6.35	2137.83	4.23	2010.9	5.32	2010.9	4.12
S1-R-20-4	10	1493.822	11	1830.17	5.13	1573.07	8.97	1589.89	5.72
S1-R-20-5	10	1667.67	9.33	1781.38	8.76	1580.91	15.34	1784.01	7.29
S1-R-20-6	10	1643.006	11.64	1993.01	6.21	1724.93	9.32	1724.93	8.2
S1-R-20-7	10	1547.684	6.76	1929.81	6.12	1870.6	14.52	1889.09	5.27
S1-R-20-8	10	1635.755	5.82	1857.75	3.71	1672.39	5.03	1672.39	3.52
S1-R-20-9	10	1605.324	6.74	1938.1	3.51	1807.07	5.4	1815.19	4.16
S1-R-20-10	10	1524.8	6.1	1719.11	3.2	1696.7	5.7	1696.7	2.23
S1-R-50-1	10	Time limit	Time limit	4967.02	173.122	3814.23	612.84	3834.33	76.249
S1-R-50-2	10	Time limit	Time limit	4497.41	250.266	3626.87	701.44	3672.24	85.661
S1-R-50-3	10	Time limit	Time limit	4612.08	117.089	4029.62	436.43	4029.62	58.269
S1-R-50-4	10	Time limit	Time limit	4262.57	135.515	3734.91	472.85	3758.32	60.311
S1-R-50-5	10	Time limit	Time limit	4589.05	164.35	3748.56	542.24	3748.56	70.534
S1-R-50-6	10	Time limit	Time limit	4528.73	128.641	3410.08	452.23	3426.08	56.874
S1-R-50-7	10	Time limit	Time limit	4516.64	180.295	3474.27	637.15	3520.86	78.214
S1-R-50-8	10	Time limit	Time limit	4539.33	107.972	3474.69	422.72	3494.81	80.561
S1-R-50-9	10	2996.748	4259.32	4267.47	129.265	3627.01	467.68	3686.43	58.721
S1-R-50-10	10	Time limit	Time limit	4288.83	114.74	3783.69	435.29	3796.12	62.968
S2-C-10-1	1	58.7243	3.16	58.7243	1.19	58.7243	2.53	58.7243	1.1
S2-C-10-2	1	59.4222	17.18	63.8216	10.12	59.4222	16.42	59.4222	4.6
S2-R-10-1	5	270.925	2.13	290.643	0.87	290.643	1.58	290.643	0.09
S2-R-10-2	3	226.993	13.42	294.437	5.13	294.437	11.71	294.437	3.14
S2-RC-10-1	2	193.085	2.04	193.085	0.45	193.085	1.65	193.085	0.23
S2-RC-10-2	2	195.147	3.21	195.147	0.93	195.147	1.97	195.147	0.76
S2-C-20-1	2	204.177	5.13	484.5	2.3	356.868	3.7	347.429	2.15
S2-C-20-2	2	212.947	42.12	232.838	21.88	221.31	41.75	249.7233	16.23
S2-R-20-1	2	449.999	4.81	548.814	1.9	479.247	2.74	455.168	2.15
S2-R-20-2	2	400.301	140.51	491.133	14.53	451.07	29.27	406.317	13.89
S2-RC-20-1	2	323.981	3.63	399.931	1.31	323.981	2.59	323.981	2.21
S2-RC-20-2	2	327.786	5.12	405.616	1.87	327.786	3.54	327.786	2.54
S2-C-50-1	2	435.339	192.13	523.752	55.452	435.339	167.458	610.244	14.78
S2-C-50-2	2	525.871	305.49	1034.05	206.12	1025.55	258.889	909.847	26.58
S2-R-50-1	2	1125.31	1024.17	1158.24	131.505	1156.77	864.67	1138.93	181.328
S2-R-50-2	2	Time limit	Time limit	Time limit	Time limit	Time limit	Time limit	1208.04	81.08
S2-RC-50-1	2	Time limit	Time limit	Time limit	Time limit	Time limit	Time limit	920.245	71.372
S2-RC-50-2	3	Time limit	Time limit	Time limit	Time limit	Time limit	Time limit	969.247	163.378

The difference between the number of solved instances can be explained by the fact that the number of periods in set 2 may be smaller than the one in set 1 (which is fixed to 10). For the instance S1-R-20-5, the sequential-MILP and Sequential-Lexicographic give us unacceptable solutions which consist in visiting only 19 sites over the 20 (in grey in table II). Except this case, all the reported results refer to feasible solutions. To compare our different approaches, table III synthesizes the average gap for distance and solving time between our three sequential approaches and the global approach. It also provides the associated standard deviations. These gaps and deviations are calculated based on the 36 instances solved to optimality using the global approach and we did not take into account the unfeasible solutions. Table III also gives the total number of solved instances for each approach. It provides a clear idea of the advantages and drawbacks of each approach. The Global-MILP can be considered as the best approach because it provides optimal solution for the Mu-PDPTWPD but the number of solved instances highlights its limitation to solve small instances (with less than 50 nodes). The three sequential approaches do not provide optimal solutions and this can be justified by the fact that the sum of optimal local solutions cannot give the optimal global solution. But we can observe that we minimize the solving time and we increase the number of solved instances. We also remark that the mean gap obtained using the Sequential-Lexicographic and the Sequential-HGA is almost equal for the distance. Furthermore, the standard deviation using Sequential-HGA is smaller than the one using Sequential-Lexicographic this means that the distance values are scattered around the mean. In addition, the Sequential-HGA can solve all the instances in very small time. So, we can consider it better than the other sequential approaches.

III. Analysis of obtained results



Approach	Mean GAP distance (%)	Deviation GAP distance (%)	Mean GAP solving time (%)	Deviation GAP solving time (%)	Number of solved instances
Global-MILP	0	0	0	0	36
Sequential-MILP	21.60	26.63	-58.19	17.31	44
Sequential-Lexicographic	13.40	20.36	-26.77	31.69	44
Sequential-HGA	13.74	18.18	-66.12	23.37	48

V. Conclusions and Future Works

Among the four proposed approaches to solve the Mu-PDPTWPD, three of them address the problem as a sequence of Mo-SPDPTWPDs. We have generated the first instances dedicated to the Mu-PDPTWPD. Experimental results show the strength and weakness of each approach. The gap between the solutions provided by our sequential approaches and the optimal ones is caused by the strategy itself which consists in solving each period without having any idea about the global problem. In future works, we will develop heuristic methods to address the global problem and we will introduce additional constraints to match better real life cases.

VI. Acknowledgments

This work is supported by the ANR (French National Research Agency) in the framework of the project TCDU (Collaborative Transportation in Urban Distribution). This project ANR-14-CE22-0017 is labelled by the Pôle Véhicule du Futur, and is jointly performed by four partners, the three french universities of technology (UTT, UTBM, UTC) and the society Share And Move Solutions.

References

[1] J. Desrosiers, Y. Dumas, M. M. Solomon, and F. Soumis, "Time constrained routing and scheduling," *Handbooks in operations research and management science*, vol. 8, pp. 35–139, 1995.

[2] S. N. Parragh, K. F. Doerner, and R. F. Hartl, "A survey on pickup and delivery problems. part i: Transportation between customers and depot," *Journal für Betriebswirtschaft*, vol. 58, no. 1, pp. 21–51, 2008.

[3] ——, "A survey on pickup and delivery problems. part ii: Transportation between pickup and delivery locations," *Journal für Betriebswirtschaft*, vol. 58, no. 2, pp. 81–117, 2008.

[4] J. Schönberger and H. Kopfer, "Planning the incorporation of logistics service providers to fulfill precedence-and time window-constrained transport requests in a most profitable way," in *Distribution Logistics*. Springer, 2005, pp. 141–156.

[5] Z. Al Chami, H. Manier, and M.-A. Manier, "New model for a variant of pick up and delivery problem," in *IEEE International Conference on Systems, Man, and Cybernetics (SMC), Budapest, Hungary*, 2016, pp. 1708–1713.

[6] S. Pirkwieser and G. R. Raidl, "A variable neighborhood search for the periodic vehicle routing problem with time windows," in *Proceedings of the 9th EU/meeting on metaheuristics for logistics and vehicle routing, Troyes, France*, 2008, pp. 23–24.

[7] K. Braekers, K. Ramaekers, and I. Van Nieuwenhuyse, "The vehicle routing problem: State of the art classification and review," *Computers & Industrial Engineering*, vol. 99, pp. 300–313, 2016.

[8] F. Tricoire, M. Romauch, K. F. Doerner, and R. F. Hartl, "Heuristics for the multi-period orienteering problem with multiple time windows," *Computers & Operations Research*, vol. 37, no. 2, pp. 351–367, 2010.

[9] W. Souffriau, P. Vansteenwegen, G. Vanden Berghe, and D. Van Oudheusden, "The multiconstraint team orienteering problem with multiple time windows," *Transportation Science*, vol. 47, no. 1, pp. 53–63, 2013.

[10] S.-W. Lin and F. Y. Vincent, "A simulated annealing heuristic for the multiconstraint team orienteering problem with multiple time windows," *Applied Soft Computing*, vol. 37, pp. 632–642, 2015.

[11] D. Favaretto, E. Moretti, and P. Pellegrini, "Ant colony system for a vrp with multiple time windows and multiple visits," *Journal of Interdisciplinary Mathematics*, vol. 10, no. 2, pp. 263–284, 2007.

[12] S. Belhaiza, P. Hansen, and G. Laporte, "A hybrid variable neighborhood tabu search heuristic for the vehicle routing problem with multiple time windows," *Computers & Operations Research*, vol. 52, pp. 269–281, 2014.

[13] A. K. Beheshti, S. R. Hejazi, and M. Alinaghian, "The vehicle routing problem with multiple prioritized time windows: A case study," *Computers & Industrial Engineering*, vol. 90, pp. 402–413, 2015.

[14] H. Manier, M.-A. Manier, and Z. Al Chami, "Shippers' collaboration in city logistics," *IFAC MIM (Troyes, France). IFAC-PapersOnLine*, vol. 49, no. 12, pp. 1880–1885, 2016.

[15] Z. Al Chami, H. Manier, and M.-A. Manier, "A lexicographic approach for the bi-objective selective pickup and delivery problem with time windows and paired demands," *Annals of Operations Research*, Apr 2017. [Online]. Available: https://doi.org/10.1007/s10479-017-2500-9

[16] Z. Al Chami, H. Manier, M.-A. Manier, and C. FITOURI, "A hybrid genetic algorithm to solve a multi-objective pickup and delivery problem," in *IFAC World Congress (Toulouse, France)*, 2017.

[17] N. Srinivas and K. Deb, "Multiobjective optimization using nondominated sorting in genetic algorithms," *Evolutionary computation*, vol. 2, no. 3, pp. 221–248, 1994.

[18] D. E. Goldberg and J. Richardson, "Genetic algorithms with sharing for multimodal function optimization," in *Genetic algorithms and their applications. In: proceedings of the Second International Conference on Genetic Algorithms*. Hillsdale, NJ: Lawrence Erlbaum, 1987, pp. 41–49.

[19] H. Lim, A. Lim, and B. Rodrigues, "Solving the pickup and delivery problem with time windows using squeaky wheel optimization with local search," *AMCIS 2002 Proceedings*, p. 319, 2002.