

A New Metaheuristic to Solve a Selective Pickup and Delivery Problem

Zaher Al Chami, Hamza El Flity, Hervé Manier, Marie-Ange Manier

Univ. Bourgogne Franche-Comté FEMTO-ST Institute/CNRS,

Rue Thierry-Mieg (UTBM), 90 010 Belfort Cedex, France

(Zaher.al-chami@utbm.fr, hamza.el-flity@utbm.fr, herve.maniar@utbm.fr, marie-ange.maniar@utbm.fr)

Abstract – The pickup and delivery problem (PDP) is a problem of optimization of vehicles routes, in order to satisfy the demands for carriage of goods between suppliers (pickup locations) and customers (delivery locations) using a set of vehicles. In this paper, we discuss a variant of PDP which is the SPDPTWPD (Selective PDP with Time Windows and Paired Demands). In this type of problems, a set of constraints must be respected. Those constraints are related to the capacity of the vehicles, the opening and closing times of each site (occurrence of time windows), the precedence (paired demands), and the choice of sites to be visited (selective aspect). We proposed a new metaheuristic to solve the SPDPTWPD, then we tested our method on benchmark instances, and its efficiency is shown by obtained results.

Keywords – Pickup and Delivery Problem; Transportation; metaheuristic; algorithm

I. INTRODUCTION

The pickup and delivery problem (PDP) aims at constructing a set of routes in order of establishing a transportation network to satisfy a set of pickup and delivery requests under specified constraints. Each constructed route is traveled by a vehicle with limited capacity, which is, among other available vehicles, based at a starting depot. The vehicles must also return to depot at the end of their routes.

In our problem, each pickup or delivery site has a time window to be respected, which means that the site must be visited either between its opening time and its closing time or before its opening time. In this last case, the vehicle must wait until the beginning of service time. In addition, the paired demands constraints included in our problem ensure that each vehicle route satisfies the precedence constraints, so the customer must be served only after its supplier.

In the urban context, taking into account all the above constraints becomes more critical because of the applied policies by the public authorities in some cities. For example, limiting transportation hours per day to reduce the noise, or the regulation of parkings and street access.

The variant of the classic PDP studied in this paper is the selective PDP with time windows and paired demands (SPDPTWPD). This variant is a generalization of the standard PDP where in addition of constraints related to time windows and paired demands, it is not obligatory to visit all nodes.

In fact, sometimes it is impossible to visit all nodes in the transportation network in a given period, so we must choose nodes which will be served in this period. Hence, a profit is associated to each site to represent its priority and the goal of our variant is to maximize the total profit then to minimize the distance.

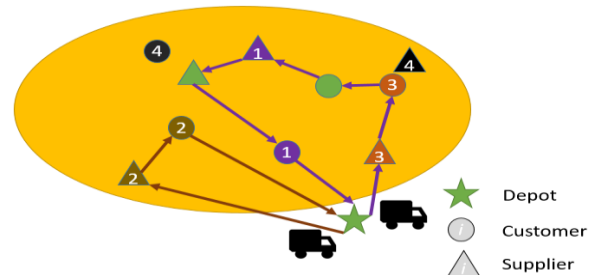


Figure 1. Example of the SPDPTWPD

Fig. 1 represents an example of our variant with 5 suppliers and 5 customers. The fleet is composed of 2 vehicles. Each paired demand (between a supplier and its customer) can be identified thanks to the same color and the same number. The green triangle represents a supplier paired with the depot (the goods picked up from this site are delivered to the depot). And the green circle represents a customer paired with the depot. In the shown solution, the nodes numbered 4 were not selected.

The next sections of this paper are organized as follows: a literature review is provided in section II. Then, the section III defines the SPDPTWPD with a mathematical formulation. In section IV, we explain our new metaheuristic to solve the SPDPTWPD. After that, section V is dedicated to the experimental results. Finally, section VI concludes this paper and gives direction for further research.

II. LITERATURE REVIEW

In this section, we provide a state of the art of various methods used to solve different variants of the PDP.

The PDP is considered as a variant of the combinatorial optimization problem VRP (Vehicle Routing Problem). It consists in establishing an optimal set of routes traveled by a fleet of vehicles in order to visit a specified set of customers. A taxonomic literature review of the VRP was made between 2009 and June 2015 and has been published in [4].

The PDP involves a collection of suppliers and a collection of customers geographically located. Over the few past decades, many studies concerning the variants of this problem have been achieved. For more details, see book [14] and a number of surveys [9][10][13].

Many exact methods have been elaborated to solve the PDPTWPD. Among those methods, Ropke and Cordeau have accomplished a branch-and-cut-and-price algorithm in which the column generation algorithm is used to calculate the lower bounds [12]. Moreover, another exact algorithm based on a branch-and-cut-and-price approach has been proposed to solve the m-PDPTW (multi vehicles PDPTW) with two different objective functions: minimizing the total vehicle fixed costs, and minimizing the route costs [3]. Furthermore, researchers have elaborated a new formulation to solve the Mu-PDPTWPD (Multi-periods PDPTWPD) where each site can be visited in one or several periods [5]. The period can be represented as one day or any other time slot and their model is solved by using CPLEX.

Due to the combinatorial complexity of the PDPs, the efficiency of exact solving methods has been proved mainly for simple problems, but not for the complex problems with great size. For this reason, many studies applying metaheuristic algorithms have been carried out in the literature. Authors have developed a "Squeaky wheel" method which solves the m-PDPTW using a local search [8]. Another method to solve this problem has been presented in [11]. It is a construction heuristic method based on the integration principle. Moreover, a hybrid metaheuristic, which combines tabu search and simulated annealing, has been developed by researchers in [6]. A memetic algorithm has been introduced in [16] to solve the biobjective selective PDP (SPDP) where the demands are unpaired.

Concerning the selective aspect of the PDPTWPD, researchers have proposed a linear programming model, based on three-index formulation, to solve this problem [2]. Moreover, a lexicographic approach to solve the bi-objective SPDPTWPD has been presented in [1]. Recently, a hybrid genetic algorithm to solve the multi-objective SPDPTWPD has been published in [15] where the aim of our method was to maximize the profit and minimize the distance in the same time.

In this paper, we propose a new metaheuristic method to solve this variant.

III. MATHEMATICAL FORMULATION

A. Notations

In this part, we introduce a mathematical formulation for SPDPTWPD. In this purpose, we define the following parameters useful in our problem:

Data:

- V: Collection of available vehicles,
- C: Collection of all customers,

- B: Collection of all suppliers,
- Nodes: Collection of all suppliers and customers (Nodes = C ∪ B),
- W: Collection of all depots,
- N: Collection of all depots, suppliers, and customers (N = Nodes ∪ W),
- M: a great number
- Supplier_i: Collection of all suppliers related to customer i,
- Q^k: Capacity of vehicle k,
- Speed^k: Average speed of vehicle k,
- d_{ij}: Distance between site i and site j,
- P_i: Profit characterizing the site i,
- [e_i, l_i]: Time window of site i,
- S_i: Service time at site i,
- q_i: Goods quantity requested by site i
 - If i ∈ B then q_i > 0,
 - If i ∈ C then q_i < 0,

Variables:

- A_i^k: Starting service time of vehicle k in site i,
- D_i^k: Departure time of vehicle k from site i,
- Y_i^k: Goods quantity in vehicle k visiting site i,
- X_{ij}^k:
 - 1 if vehicle k moves from site i to site j,
 - 0 Otherwise,

B. Mixed Linear Program for the SPDPTWPD

A mathematical formulation for the SPDPTWPD could be as follows:

$$\text{Minimize/Maximize } F \quad (1)$$

Subject to:

$$\sum_{j \in N} \sum_{k \in V} X_{ij}^k \leq 1 \quad i \in \text{Nodes} \quad (2)$$

$$\sum_{i \in N} \sum_{k \in V} X_{ij}^k \leq 1 \quad j \in \text{Nodes} \quad (3)$$

$$\sum_{i \in N} X_{iu}^k - \sum_{j \in N} X_{uj}^k = 0 \quad k \in V; u \in \text{Nodes} \quad (4)$$

$$\sum_{j=1}^{\text{Nodes}} X_{jw}^k = 1 \quad k \in V; w \in W \quad (5)$$

$$\sum_{i=1}^{\text{Nodes}} X_{wi}^k = 1 \quad k \in V; w \in W \quad (6)$$

$$Q^k \geq Y_i^k \geq 0 \quad i \in N; k \in V \quad (7)$$

$$Y_j^k \geq Y_i^k + q_i - M(1 - X_{ij}^k) \quad i, j \in N; k \in V \quad (8)$$

$$Y_j^k \leq Y_i^k + q_i + M(1 - X_{ij}^k) \quad i, j \in N; k \in V \quad (8 \text{ bis})$$

$$X_w^k = \sum_{i \in C} (-q_i * \sum_{j \in N} X_{ji}^k) \quad k \in V; w \in W; j \neq I; \text{Supplier}_i = w \quad (9)$$

$$D_w^k = 0 \quad k \in V; w \in W \quad (10)$$

$$D_i^k \geq A_i^k + S_i - M(1 - X_{ij}^k) \quad i, j \in N; k \in V \quad (11)$$

$$D_i^k \leq A_i^k + S_i + M(1 - X_{ij}^k) \quad i, j \in N; k \in V \quad (11 \text{ bis})$$

$$e_i * \sum_{j=1}^N X_{ij}^k \leq A_i^k \leq l_i * \sum_{j=1}^N X_{ij}^k \quad i \in N; k \in V \quad (12)$$

$$A_i^k \geq D_i^k + \frac{d_{ij}}{\text{speed}^k} - M(1 - X_{ij}^k) \quad i, j \in N; k \in V \quad (13)$$

$$D_f^k \leq A_c^k \quad f \in B; c \in C \text{ such as } \text{Supplier}_c = f \quad (14)$$

Our objective function is represented by (1). The main objectives are expressed by the following functions:

- Maximizing profit: $F1 = \sum_{i \in N} \sum_{j \in N} \sum_{k \in V} P_i X_{ij}^k$

- Minimizing distance: $F2 = \sum_{i \in N} \sum_{j \in N} \sum_{k \in V} d_{ij} X_{ij}^k$

Constraints (2) and (3) ensure that each site is visited at most once by a single vehicle. Constraints (4) ensure the routing continuity by a vehicle. Constraints (5) and (6) guarantee that each vehicle begins its route and finishes it at the depot, even if they do not visit any site. The capacity constraints are ensured by (7), (8), (8bis) and (9). Moreover, the time windows constraints are guaranteed by (10), (11), (11bis), (12) and (13). Constraints (14) ensure that the supplier is visited before its customer.

In the next section, we present our approach to solve the considered problem.

IV. PROPOSED METAHEURISTIC

In this section, we introduce our new metaheuristic which solves the SPDPTWPD. Our approach could be considered as an extension of the tabu-embedded simulated annealing algorithm proposed by Li & Lim to solve the PDPTWPD [6].

A. Approach of Li and Lim

In their study, authors have considered an unlimited number of vehicles. The objective of their method is defined as follows, respecting the following priority order:

- 1- Minimization of number of vehicles
- 2- Minimization of total travel cost
- 3- Minimization of total schedule duration (the sum of waiting time, travel time and service time)
- 4- Minimization of total waiting time to start service

The metaheuristic developed by Li and Lim is principally based on three permutation operators used for several iterations allowing to restructure the routes in the purpose of reaching the objective earlier mentioned and obtaining the best possible solution. The first routes are obtained using the insertion heuristic proposed by Solomon [7]. An explanation for those operators is given by the following description:

1- PD-Shift Operator

This operator moves a pickup-delivery pair from a route to another. In fig. 2, the PD-Shift operator is used to move the pair P_i - D_i from route R_i to route R_j . P_i and D_i are originally in route R_i , so first the PD-Shift operator removes P_i and D_i from the route R_i , and then inserts them at a feasible position in route R_j , subject to the constraints imposed on PDPTWPD. Infeasible shifts are forbidden.

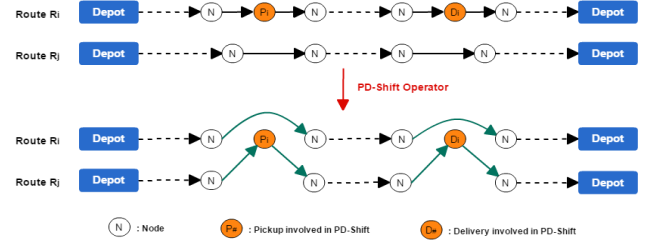


Figure 2. PD-Shift operator

2- PD-Exchange Operator

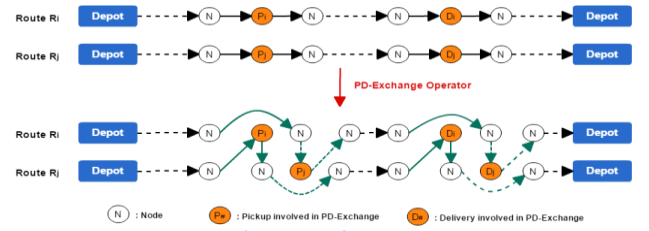


Figure 3. PD-Exchange operator

The PD-Exchange operator swaps pickup-delivery pairs of two routes. In other words, it moves one pair from each route to the other. For example, in fig. 3, the pair P_i - D_i is originally in route R_i , and P_j - D_j is originally in route R_j . The PD-Exchange operator first removes P_i - D_i from route R_i and P_j - D_j from route R_j , and then inserts P_i - D_i at a feasible position in route R_j , at the same time of inserting the locations P_j and D_j at a feasible position in route R_i . The infeasible insertions are not allowed.

3- PD-Rearrange Operator

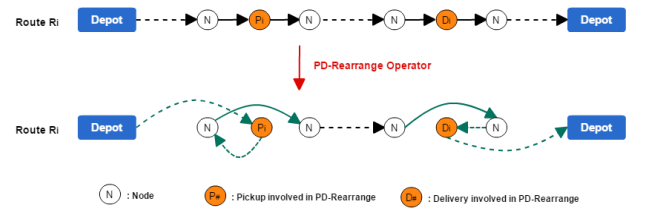


Figure 4. PD-Rearrange operator

The PD-Rearrange operator repositions the pickup-delivery pairs within the same route. Its objective is to move each pair in a route to the best position that maximally ameliorate the route in order to reach the objective early mentioned. In fig. 4, the PD-Rearrange operator removes the pair P_i - D_i which is in the route R_i , and then inserts them at another feasible position in the same route. Infeasible insertions are also not allowed.

B. New extension

As mentioned above, our study is principally based on the approach published in [6]. Due to the difference in objectives and criterions between our study and the one in [6], we adapted this method to the context of our problem, where the objective functions are the following:

- 1- Maximization of the total collected profit.
- 2- Minimization of the total traveled distance.

First, we added the profit attribute to each site, which doesn't exist in [6]. Then, as early mentioned, in [6] an unlimited number of vehicles is used to solve the problem, contrary to our case where we use a limited number of vehicles. Therefore, we added to the algorithm a new function which creates a solution with a limited number of vehicles from a solution with unlimited number of vehicles. In the purpose of respecting the objective functions, the solution generated must be the solution with maximal possible profit. To do so, the principle used is to keep removing the routes having the minimal profit among the set of routes in the solution one by one, until having a solution with the considered number of routes which is equivalent to the number of available vehicles.

In addition, we ameliorate the solution by trying to insert nodes from removed routes in the remaining routes. For each node, if the insertion is not feasible, with regard to the problem constraints, we try to exchange it with a node in the remaining routes having a lower profit. This explanation could be shown in figure 5.

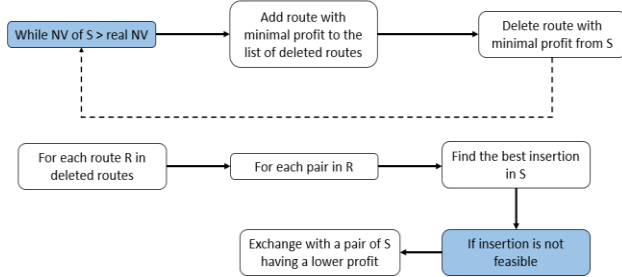


Figure 5. Solution optimization procedure

In Fig. 5, S is the current solution, NV is the number of vehicles used in S , and $real\ NV$ is the real number of available vehicles that we must have in the solution.

To compare two solutions into the algorithm, we used a comparison method that we called F described as follows:

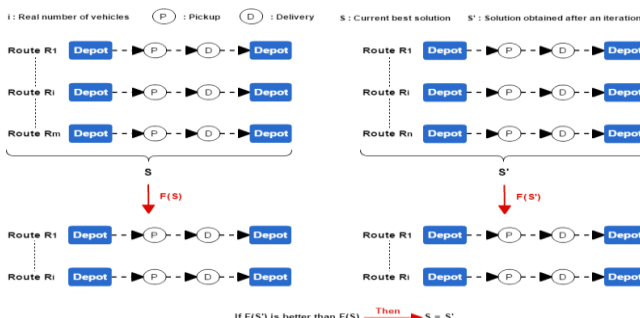


Figure 6. Comparing solutions

In Fig. 6, we apply the function F on each of the two solutions we want to compare, to have two solutions with the real number of vehicles. For the obtained solutions, the best is the one with the higher profit. If the two solutions have the same profit, the best is the one with the lower distance. If the two solutions have different number of routes after applying F , we decided to consider that the best is the one which needs the lower number of vehicles, ignoring the profit and the distance. Indeed, we remarked that this consideration helped us to get better solutions at the end. Finally, we choose among the solutions to compare initially, the one which created the best solution when applying F .

In the next section, we present the experimental results obtained by this study, comparing to the results of a lexicographic approach previously elaborated.

V. EXPERIMENTAL RESULTS

In this part, we test our new metaheuristic on benchmark instances for the SPDPTWPD (accessible via this link: <https://www.dropbox.com/sh/zpfyjo114etfuqe/AAAHUBMYfTfbKau0Nmc0yAOWa?dl=0>).

Each instance is labeled SPDPTWPDxyz, x 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), y is the number of nodes (20, 50 and 100) and i identifies the distribution of profit (1: nodes have the same profit and 2: each site have a different value of profit).

The hybrid genetic algorithm, presented in [15] to solve the same variant, provides a set of non-dominated solutions that maximize the profit and minimize the distance. The authors have proved that the gap between the obtained solutions and the optimal Pareto front was acceptable. On the other hand, our proposed metaheuristic focus on giving only one non-dominated solution for each tested instance (the one with the maximal profit). In this section, we will prove that the gap between the obtained solution and the optimal Pareto front is very small.

For that, we compared the results of our method with the results of a previous lexicographic approach solving the same variant in which the authors maximize the profit then minimize the total traveled distance [1]. The comparison also involves the CPU time to find the difference between the two approaches at the level of performance. Table 1 shows the results for each method, as well as the gap between them concerning each criterion.

As shown in table 1, we can find out 3 kinds of results:

Case 1: The results obtained by the new metaheuristic are the same as the results of the lexicographic approach, as in the case of the instance SPDPTWPD-C20. In this case, the metaheuristic gives the best possible solution and we can notice that this concerns the majority of tested instances with a percentage of 60.87%.

Instance	Number of vehicles	Lexicographic approach			Proposed metaheuristic			GAPs (%)	
		Profit	Distance	CPU Time (s)	Profit	Distance	CPU Time (s)	Profit	Distance
SPDPTWPD-C201	1	180	275,19	0,3	180	275,19	11	0,00	0,00
SPDPTWPD-C202	1	144	167,99	19,88	144	167,99	7	0,00	0,00
SPDPTWPD-C202	2	217	212,94	52,15	217	212,94	8	0,00	0,00
SPDPTWPD-R201	4	190	398,25	0,35	190	398,25	6	0,00	0,00
SPDPTWPD-R202	6	213	378,95	12,2	213	378,95	6	0,00	0,00
SPDPTWPD-RC201	2	160	233,2	0,33	160	233,2	6	0,00	0,00
SPDPTWPD-RC202	2	180	235,09	0,52	180	235,09	6	0,00	0,00
SPDPTWPD-C501	5	480	363,28	9,67	480	363,28	57	0,00	0,00
SPDPTWPD-C502	1	437	431,25	28,31	436	386,08	433	-0,23	-10,47
SPDPTWPD-C502	2	656	525,87	204,44	656	525,87	401	0,00	0,00
SPDPTWPD-R501	13	490	1075,77	14,86	490	1075,77	21	0,00	0,00
SPDPTWPD-R502	10	576	834,92	17,24	576	834,92	20	0,00	0,00
SPDPTWPD-RC501	4	320	481,37	1629,02	310	489,4	28	-3,13	1,67
SPDPTWPD-RC502	4	420	527,76	151,37	400	518,14	26	-4,76	-1,82
SPDPTWPD-RC502	5	500	686,69	1352,38	480	677,07	27	-4,00	-1,40
SPDPTWPD-RC502	6	562	758	4505,16	547	801,64	28	-2,67	5,76
SPDPTWPD-C1001	6	690	437,04	928,93	680	469,58	290	-1,45	7,45
SPDPTWPD-C1002	7	770	542,15	621,54	760	575,08	233	-1,30	6,07
SPDPTWPD-R1001	7	500	639,64	3540,04	500	639,64	138	0,00	0,00
SPDPTWPD-R1002	6	514	679,44	1391,96	512	687,81	157	-0,39	1,23
SPDPTWPD-RC1001	1	120	140,35	382,23	110	127,22	151	-8,33	-9,36
SPDPTWPD-RC1002	1	138	125,68	103	138	125,68	109	0,00	0,00
SPDPTWPD-RC1002	1	252	274,49	7000	252	274,49	145	0,00	0,00

Table 1. Experimental results

Case 2: For 17.39% of tested instances, the metaheuristic cannot give a solution with the same profit as the solution given by the lexicographic, but it gives a solution with a better travel distance, such as for example the instance SPDPTWPD-RC1001. In this case, it could be that the metaheuristic solution belongs to the Pareto front. To ensure that, we run our MILP to get the minimal distance for the obtained profit value. We remarked that 25% of obtained solutions in this case belongs to Pareto front.

Case 3: The solution given by the metaheuristic is worse than the one given by the lexicographic approach in terms of profit and distance, as for the instance SPDPTW-RC502. This case occurred in 21.74% of tested instances.

In average, the gap between our new metaheuristic and the lexicographic approach is 1,14 concerning the profit, and -0.04 concerning the distance. For the solving time, it could be noticed that our proposed metaheuristic is more performant for the majority of tested instances compared to the lexicographic approach. This time depends on several elements for example: the number of nodes, the time windows and the number of vehicles used. It can also depend on other characteristics like the distribution of the time intervals, the geographical distribution of the nodes...

VI. CONCLUSION

In this paper, we have presented a new metaheuristic to solve the SPDPTWPD. The results show that this new approach gives us very good solutions in a reasonable computational time.

In our future works, we will focus on improving our approach to achieve better results and we will add more constraints to our variant to match more real life cases.

ACKNOWLEDGMENT

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] 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*, pp. 1–19, 2017.
- [2] Z. Al Chami., H. Manier, and M.-A. Manier, "New model for a variant of Pickup and Delivery Problem", *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Budapest, Hungary, 1708-1713, 2016.
- [3] R. Baldacci, E. Bartolini, and A. Mingozzi, "An exact algorithm for the pickup and delivery problem with time windows," *Operations Research*, 59(2), 414-426, 2011.
- [4] K. Braekers, K. Ramaekers, and I.V. Nieuwenhuysse, "The vehicle routing problem: State of the art classification and review," *Computers Industrial Engineering*, 99, 300-313, 2016.
- [5] H. Manier, M.-A. Manier, and Z. Al Chami, "Shippers' collaboration in city logistics", In *IFAC-PapersOnLine*, Volume 49, Issue 12, 2016, Pages 1880-1885.
- [6] H. Li, and A. Lim, "A metaheuristic for the pickup and delivery problem with time windows", *International Journal on Artificial Intelligence Tools*, 12(02), 173-186, 2003.
- [7] M.M.Solomon, "Algorithms for the vehicle routing and scheduling problems with time window constrains", *Operations Research* 35, 254–264 (1987).
- [8] 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*, 319, 2002.
- [9] S.N. Parragh, K.F. Doerner, and R.F. Hartl, "A survey on pickup and delivery problems. part i: Transportation between customers and depot" (in German), *Journal für Betriebswirtschaft*, 58(1), 21-51, 2008a.
- [10] S.N. Parragh, K.F. Doerner, and R.F. Hartl, "A survey on pickup and delivery problems. part ii: Transportation between pickup and delivery locations" (in German), *Journal für Betriebswirtschaft*, 58(2), 81-117, 2008b.
- [11] L. Quan, and M. Maged, "A new insertion-based construction heuristic for solving the pickup and delivery problem with time windows", *European Journal of Operational Research*, 2003.
- [12] S. Ropke, and J.F. Cordeau, "Branch and cut and price for the pickup and delivery problem with time windows", *Transportation Science*, 43(3), 267-286, 2009.
- [13] M.W. Savelsbergh, and M. Sol, "The general pickup and delivery problem", *Transportation science*, 29(1), 17-29, 1995.
- [14] P. Toth, and D. Vigo, "The vehicle routing problem". Society for Industrial and Applied Mathematics; 2002.
- [15] 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-PapersOnLine*, Volume 50, Issue 1, 2017, Pages 14656-14661.
- [16] X. L. Liao, and C. K. Ting, "Solving the biobjective selective pickup and delivery problem with memetic algorithm." In *IEEE Workshop on Computational Intelligence In Production And Logistics Systems (CIPLS)*, 107-114, 2013.