

An Adaptive Genetic Algorithm for a New Variant of the Gas Cylinders Open Split Delivery and Pickup with Two-dimensional Loading Constraints

Anouar Annouch¹

Information Technology and Management
ENSIAS-Mohammed V University in Rabat

Adil Bellabdaoui²

Information Technology and Management
ENSIAS-Mohammed V University in Rabat

Abstract—This paper studies a combination of two well-known problems in distribution logistics, which are the truck loading problem and the vehicle routing problem. In our context, a customer daily demand exceeds the truck capacity. As a result, the demand has to be split into several routes. In addition, it is required to assign customers to depots, which means that each customer is visited just once by any truck in the fleet. Moreover, we take into consideration a customer time windows. The studied problem can be defined as a Multi-depots open split delivery and pickup vehicle routing problem with two-dimensional loading constraints and time windows (2L-MD-OSPDTW). A mathematical formulation of the problem is proposed as a mixed-integer linear programming model. Then, a set of four class instances is used in a way that reflects the real-life case study. Furthermore, a genetic algorithm is proposed to solve a large scale dataset. Finally, preliminary results are reported and show that the MILP performs very well for small test instances while the genetic algorithm can be efficiently used to solve the problem for a wide-reaching test instances.

Keywords—Vehicle routing problem; split delivery and pickup; multi-depot; two-dimensional loading; genetic algorithm

I. INTRODUCTION

LPG logistics transportation, in particular the distribution of the Liquefied Petroleum Gas (LPG) cylinders is considered among the basic building blocks in the Oil and Gas downstream supply chain and also known to be a very complex supply chain [11]. A typical LPG downstream supply chain consists of filling plants, distribution locations, fleet of trucks and customer's depots. The filling plant is an industrial unit composed of several processes in order to fill a broad range of gas cylinders. It should also be noted that the production and storage capacity is different from one filling plant to another. Therefore, each filling plant has an attached distribution location which is a cylinder's storage unit supplied from the production unit, then serves a set of customers by using a fleet of trucks. Moreover, due to the flammable nature of the gas, it is usually stored in liquid form under a specific pressure, in steel or composite plastic gas cylinders carefully checked and protected against deterioration. In general, the companies do not sell the gas cylinders but just the gas contents inside and the packaging remains their property. This type of packaging management system is called consignment.

Motivated by a case of a Moroccan petroleum company, our paper considers a real-life application in the LPG distribution industry. The objective is to deal with a problem

which corresponds to current industry practice and also to give solution methods able to solve real large-scale instances in very fast computation times. The current distribution policy of our case study links two main entities: the gas filling plants and customers depots. Furthermore, each customer depot is served fully by only one truck. So, deliveries are very difficult to manage and generates a huge cost related to the production capacity and the use of trucks. Thus, the GPL Company faces a real problem which is serving customers in a way to optimize both loading and vehicle routing.

Through this document, we will propose a new distribution policy which will remove the dependency constraint between customer depots and filling plants. Those customers can be served by a fleet of heterogeneous trucks based in multiple filling plants. We note that all customers are visited just once by the truck in the same route. Every truck visits a number of customers in a certain order along its route. Thus, the new approach requires the assignment of customers to filling plants. In addition, a fleet of trucks is based at each filling plant, then, each truck originates from one filling plant serves the customers assigned to that plant. The objective of this policy is therefore to offer a global distribution model at minimal cost. This model concerns both the operation of loading cylinder racks into each truck and the construction of the trucks routes.

The remainder of this paper is organized as follows: Section 2 presents the loading and distribution problem considered in our case study. In Section 3, we explore a related literature review. In Section 4, we present a mathematical formulation of our problem based on a mixed integer programming model. Then, we present and discuss computational results. Section 5 presents a developed genetic algorithm (GA). Finally we conclude this paper by discussing results obtained by the GA and compare it with Cplex results, pointing out a few remarks of practical relevance and giving some perspectives of our further researches.

II. PROBLEM DESCRIPTION

The distribution problem studied in this paper includes two sub-problems: a two-dimensional loading vehicle problem and an open multi-depot split delivery and pickup routing problem with time windows (MD-OSPDTW).

A. Two-dimensional Loading Problem

As customer requests are in the form of a set of two-dimensional and rectangular racks, the goal is to find an optimal combination to load it into a set of dissimilar rectangular trailers of height H and width W , without overlapping. The two-loading vehicle problem can be described as a loading of a limited fleet of heterogeneous trucks of capacity P_k and a loading surface with length L_k and height H_k . During loading, racks have a prefixed orientation with respect to the trailers. In addition, all of them should occupy the entire surface width of the truck. We note that in our studied problem three types of trucks are considered.

We consider N racks of several types of gas cylinders to be shipped to r customers. Each rack c is characterized by a width $w_c < L_k$, a height $h_c < H_k$ and a weight q_c known. A problem instance includes a fleet of heterogeneous trucks, initially distributed among filling plants and may differ in term of their cylinders loading capacities. Furthermore, each truck can pull two types of trailers: flatbed trailer and lowboy trailer. Since, the order of each customer is a combination of different types of racks in which we put specific types of gas cylinder; we assume that the loading racks combinations of each truck are known. In addition, the loading surface in each truck can be represented on a rectangular coordinate system (x,y) with origin $(0,0)$, in which the height represents the y -axis and the length represents the x -axis. To summarize, the truck loading is feasible if and only if all racks are placed completely inside the tray surface, two different types of racks cannot overlap and all racks are placed vertically parallel to the truck head.

B. The MD-OSPDTW Problem

The problem considered in our paper can be defined on a directed graph $G = (V,A)$ in which $V = V_1, \dots, V_n, \dots, V_{n+r}$ represent the vertices and $A = (V_i, V_j)$; $i \neq j$ represent the arcs. The set of arcs comprises: $V_c = V_1, \dots, V_n$ as vertices corresponding to the filling plants and $V_e = V_{n+1}, \dots, V_{n+r}$ as vertices representing customer's depots. We indicate by c_{ij} the transportation cost between two vertices i and j . Moreover, customers have a daily requirement D_i of cylinders, a service time S_i and an associated time window. We have to take into account the trucks service time in filling plants $s = (t + w)$ which represent both, the duration t of the trucks processing and the loading or unloading time w . Furthermore, the requirement of each customer may exceed the capacity of the truck, so it has to be split into several routes and each customer can be visited several times by different routes. In addition, each depot has a specific time window and should not be visited outside of this interval time. The objective is to determine for each truck the loading combinations and the set of routes to achieve in order to minimize the total transportation costs.

The problem in our paper is illustrated in Figure 1 and can be considered as an open vehicle routing problem (OVRP). It is a generalization of VRP in which a truck may not return to its departure depot after shipping the last customer on a route. Moreover, instead of the single-depot, deliveries are made from several multi-depots. The considered transportation problems include also more constraints: The time window restriction (OVRPTW), the multi-depots constraints, the fleet of trucks

is assumed to be heterogeneous (HOVRP); the consideration of multi-types of products (MPOVRP). In this paper the transportation problem as a whole can be named as: open split delivery and pickup routing problem with multi-depot, multi-Product, time windows, using a heterogeneous fleet of trucks and obviously with consideration of Two-dimensional loading constraints. To the best of our knowledge, the OVRP as we present it has not been considered in the literature.

III. LITERATURE REVIEW

The vehicle routing problem (VRP) is the most studied combinatorial optimization problem in transport and logistics. It was firstly presented in the literature by [1] as the truck dispatching problem. It can be defined as the determination of an optimal set of routes for a fleet of vehicles which needs to serve a set of customers. Few years later, a considerable number of variants have been considered by researchers: Capacitated vehicles, multi-depots, time-windows, pickup and delivery, heterogeneous fleet, etc. In view of the problem statement, the main purpose of this paper is to focus on the combination of loading constraints and vehicle routing problems. Both problems belong to the NP-hard type of optimization problems. Thus, the combination of these two NP-hard problems gives us a more complicated problem.

In this section, we review only the closely related works which deal with the multi-depots open split pickup and delivery routing problems with time windows and with two-dimensional loading constraints.

The variant of OVRP was firstly introduced in 1981 by [3]. Since then, it has gained much attention from researchers mostly in the studies addressing companies that avoiding being charged for the return trip of the vehicles to their departure depots. So, the important feature of the OVRP problem is that the vehicles are not required to return to the starting depot, but if they do, it must be through visiting customers in reverse order [10]. Authors in [6] presented the first heuristic algorithm to solve the OVRP. They developed a method based on a minimum spanning tree with penalties procedure. Afterwards, a tabu search algorithm was presented in [8] in which authors explore the structure of the OVRP problem and compare its performance with results obtained by other heuristics methods. One year later, the same heuristic was proposed by [10] for finding the routes that minimize both total travelling cost and operating cost while satisfying vehicle capacity and route length constraints. Many researchers have developed various heuristic approaches to efficiently solve the OVRP, such as ant colony optimisation by [19], simulated annealing by the researchers in [9] and genetic and evolutionary computing in [20].

To address the real-world business issues, the OVRP can be combined with more complicated constraints, such as split delivery (SDVRP). The SDVRP was introduced in the first time in [5]. The Authors studied this problem to prove that it can generate savings by allowing divisible deliveries. In [21] a study of the maximum possible savings achieved by enabling divisible deliveries is presented. Authors analyse the SDVRP algorithmic complexity with small capacity vehicles. Another work in which two hybrid genetic algorithms are used to solve a split delivery problem with respect to the total travel

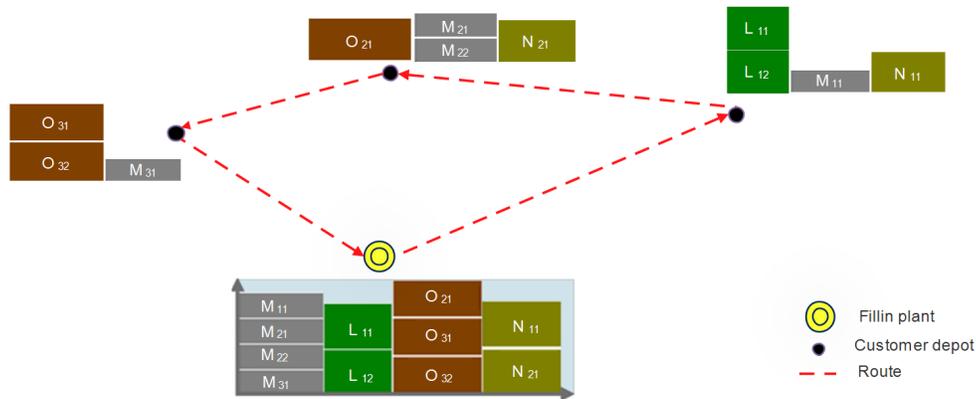


Fig. 1. Illustration of 2L-MD-OSPDTW.

distance is presented by [22]. In regards to the SDOVRP, we find only one interesting paper [30]. Authors developed a mathematical model of the OVRPSD that minimize both total fixed cost of vehicles and delivery workers and total transportation cost. Other features are also considered such as delivery and pickup (DPRP). This problem has received continual attention in transportation logistics field. The early work on the VRPPD was in [2], it was conducted for dial-a-ride scenarios. Later, researchers in [7] provided a lower limit where a demand of each customer does not exceed the vehicle capacity. In the VRPSPD, the vehicles are not only required to deliver goods to customer, but also pickup goods at customer locations. A general assumption in the VRPSPD is that all delivered goods must be originated from the depot and all pickup goods must be transported back to the depot. Very recently an iterated local search metaheuristic and a branch-and-price algorithm are proposed to solve the multi-vehicle one-to-one pickup and delivery problem with split loads [27]. Moreover, Authors in [31] studied a barge transportation of maritime containers between a dry port and a set of sea port terminals. They develop a hybrid local search metaheuristic algorithm, combined with a branch-and-cut method to solve the VRPSPD problem. The objective is to find the best allocation of containers to barges in order to guarantee on-time delivery and meet capacity restrictions. In a previous paper we have introduced the split delivery and pickup vehicle routing problem with two-dimensional loading constraints. We attempt to give solutions for small instances in order to validate the model [26].

Customer's time window is also an important characteristic of the studied OVRP. The modeling of such problems requires consideration of the arrival time of each customer, which requires the integration of time windows constraints of each customer. The first paper introduced the open vehicle routing problem (OVRPTW) was studied by [15] Authors studied a real-world cases such as the delivery of school meals, school bus routing, the plans of passing through tunnels of trains, etc. The problem variants are extended to include the case where more than one depot is considered. The multiple depots open routing problem (MD-OVRP) is a variant of the capacitated vehicle routing problem CVRP in which a fleet of vehicles is based at each depot and customers are assigned to depots.

Recently, [29] proposed a two-phase algorithm to solve a low-carbon multi-depot open vehicle routing problem with time windows (MDOVRPTW) with minimum total costs. In the literature, several papers have studied OVRP and VRPSPD with times windows and multi-depots constraints [8] [28] [29].

Another complicated problem presented in this paper which is the two-dimensional loading constraint. In recent years, a problem of loading customer items into vehicles has become a fertile ground for researchers in the field of combinatorial optimization. Nevertheless, the combination of the loading constraints and vehicle routing problems has received a little attention in the literature. The first work related to transportation problem with loading constraints was introduced by [4]. They address the Pickup and Delivery Traveling Salesman with LIFO Loading where a single rear-loaded vehicle serves a set of customer orders with LIFO policy. Next, [12] addressed the Two dimensional Capacitated Vehicle Routing Problem with Loading Constraints (2L-CVRP) using an Ant Colony Optimization (ACO). In another work they present an exact approach based on branch-and-cut for solving the same problem [14]. Researchers in [17] resolved a two dimensional Capacitated Vehicle Routing Problem using a tabu search algorithm. The 2L-VRP is combined to various VRP variants, such as time windows [16] [23][24], Pickup and delivery [13] [18] and heterogeneous fleet [25].

IV. MATHEMATICAL FORMULATIONS

Two virtual plants are considered to formulate our distribution problem: a departure plant and an arrival one. We suppose that the transportation cost is null between these plants and the other considered plants. We define our problem as an oriented graph $G = (V, A)$ with $V = V_0, V_1, \dots, V_{n+1}, \dots, V_{n+r}, V_{n+r+1}, \dots, V_{2n+r}, V_{n+r+1}$ and $A = (V_i, V_j) ; i \neq j$. The arcs A comprises: a virtual starting plant V_0 , a filling departure plants $V_c = V_1, \dots, V_{n+1}$, a set of customers to be visited $V_e = V_{n+1}, \dots, V_{n+r}$, the arrival filling plants $V_{c'} = V_{n+r+1}, \dots, V_{2n+r}$, and the arrival virtual plant V_{2n+r+1} (Figure 2).

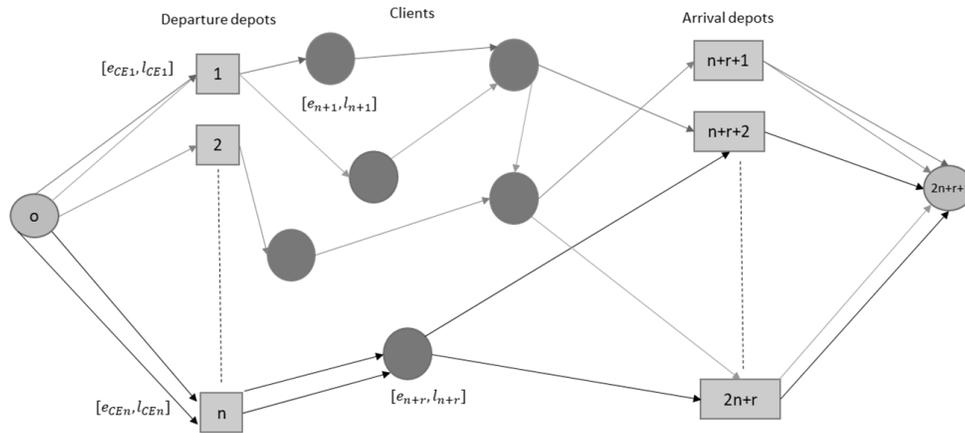


Fig. 2. Illustration of the Solution Method.

A. Assumptions.

- A truck cannot leave or arrive at customer depots outside a specific time window;
- Each filling plan has a determined service time corresponding to unload racks of the empty cylinders and to load racks of the filled cylinders;
- Each customer depot has a determined service time corresponding to unload racks of the filled cylinders and to load racks of the empty cylinders;
- The travel time and the distance between the gas filling plants and the customer depots are well known;
- At the beginning, all trucks are empty and supposed parked at the plants where it belongs;
- The starting location of each tour which is the filling plant can never be a pickup location;
- The last pickup location of the tour is the last served customer;
- Each customer may be served one or more times;
- We suppose the use of heterogeneous and limited fleet of trucks;
- The racks should occupy the entire surface width of the truck;
- The capacity of each truck is well known.

B. Parameters.

In order to formulate the above problem and solve it, we define the complete nomenclature and then state the model. The nomenclature is divided into parameters, sets and variables:

- o : Virtual starting plant;
- n : Number of Plants;
- r : Number of customers;
- $2n+r+1$: Virtual arrival plant;

- K : Number of trucks in the fleet;
- C_{ijk} : Truck k transport cost between vertex i and j ;
- P_k : Truck capacity;
- H_k : Maximum height of truck k ;
- L_k : Maximum length of truck k ;
- C : Number of racks types;
- Max_{ck} : Maximum number of racks with type c on truck k ;

$$\text{Max}_{ck} \leq \frac{H_k}{h_c} < \text{Max}_{c,k+1}, \forall k \in 1, \dots, K \forall c \in 1, \dots, C$$

- D_{ic} : Customer requirement of cylinders of the type of rack c .
- D_i : Customer requirement of cylinders:

$$D_i = D_{ic}, \forall c \in 1, \dots, C, i \in 1, \dots, r$$

- n_c : Max number of cylinders in rack c ;
- h_c : Rack c height;
- l_c : Rack c length;
- q_c : Rack c weight when it is full;
- t_{ij} : Travel time between vertex i and j ;
- d_{ij} : Distance time between vertex i and j ;
- M : Large number;
- (e_i, l_i) : Time windows
- w : Time to load / unload a rack in a truck
- s_i : Service time $s_i = 2w * \text{Number of racks}$.
- V_{ik} : Binary parameter

$$V_{ik} = \begin{cases} 1 & \text{if truck } k \text{ was parked in plant } i. \\ 0 & \text{otherwise} \end{cases}$$

- λ : Filling plant waiting time

$$\lambda_i = \begin{cases} \lambda & \text{if } i \in (1, \dots, n). \\ 0 & \text{if } i \in (n+1, \dots, n+r) \end{cases}$$

C. Decision Variables.

Five decision variables are used in our mathematical model as follow:

$$x_{ijk} = \begin{cases} 1 & \text{if truck } k \text{ visits } i \text{ from } j. \\ 0 & \text{otherwise} \end{cases}$$

$$i, j \in (0, \dots, 2n+r+1)$$

$$y_{ik} = \begin{cases} 1 & \text{if truck } k \text{ visits } i. \\ 0 & \text{otherwise} \end{cases} \quad i \in (0, \dots, 2n+r+1)$$

a_{ik} : The arrival time of truck k at customer i .

b_{cik} : Number of racks c delivered to customer i by truck k .

z_{ck} : Number of piles c occupied in the truck k .

$$z_{ck} = \sum_{i=n+1}^{n+r} \sup\left(\frac{b_{cik}}{Max_{ck}}\right)$$

D. Objective and Constraints.

Using the above notation, the problem can be presented as follows.

1)

$$\text{Minimize} = \sum_{i=1}^{2n+r+1} \sum_{j=1}^{2n+r+1} \sum_{k=1}^K C_{ijk} x_{ijk}$$

Subject to:

2)

$$\sum_{j=0}^{2n+r} x_{ijk} = \sum_{j=1}^{2n+r+1} x_{ijk} = y_{ik}$$

$$\forall i \in (0, 1, \dots, 2n+r+1), i \neq j, k \in (1, \dots, K)$$

3)

$$\sum_{j=1}^{2n+r+1} \sum_{k=1}^K x_{ijk} = 0 \quad \forall j \in (0, \dots, n)$$

4)

$$\sum_{i=0}^{2n+r} \sum_{k=1}^K x_{ijk} = 0 \quad \forall j \in (n+r+1, \dots, 2n+r+1)$$

5)

$$\sum_{j=n+1}^{n+r} x_{ojk} = \sum_{j=n+1}^{n+r} x_{j(2n+r+1)k} = 0 \quad \forall k \in (1, \dots, K)$$

6)

$$y_{ok} \leq \sum_{j=n+1}^{n+r} y_{jk} \leq M y_{ok} \quad \forall k \in (1, \dots, K)$$

7)

$$y_{(2n+r+1)k} \leq \sum_{j=n+1}^{n+r} y_{jk} \leq M y_{(2n+r+1)k} \quad \forall k \in (1, \dots, K)$$

8)

$$y_{ok} = \sum_{j=1}^n y_{jk} = y_{(2n+r+1)k} = \sum_{j=nr+1}^{2n+r} y_{jk} \quad \forall k \in (1, \dots, K)$$

9)

$$b_{cik} \leq \frac{D_{ic}}{n_c} y_{jk}$$

$$\forall i \in (n+1, \dots, n+r), c \in (1, \dots, C), k \in (1, \dots, K)$$

10)

$$\sum_{i=n+1}^{n+r} \sum_{c=1}^c (b_{cik} Q_c) \leq P_k \quad \forall k \in (1, \dots, K)$$

11)

$$\sum_{k=1}^K b_{cik} = \frac{D_{ic}}{n_c} \quad \forall i \in (n+1, \dots, n+r), c \in (1, \dots, C)$$

12)

$$a_{ik} + \lambda_i + S_i + t_{ij} \leq a_{jk} + M(1 - x_{ijk})$$

$$\forall i, j \in (1, \dots, 2n+r+1), i \neq j, \forall k \in (1, \dots, K)$$

13)

$$a_{ik} \geq e_i y_{ik} \quad \forall i \in (1, \dots, n+1, \dots, r+n), \forall k \in (1, \dots, K)$$

14)

$$a_{ik} \leq l_i + M(1 - y_{ik})$$

$$\forall i \in (n+1, \dots, r+n, \dots, 2n+r+1), \forall k \in (1, \dots, K)$$

15)

$$\sum_{c=1}^C \sum_{i=n+1}^{n+r} b_{cik} h_c l_c \leq H_k L_k + b_k c_k \quad \forall k \in (1, \dots, K)$$

16)

$$\sum_{c=1}^C z_{ck} l_c \leq L_k \quad \forall k \in (1, \dots, K)$$

17)

$$z_{ck} - 1 + \frac{1}{M} \leq \frac{\sum_{i=n+1}^{n+r} b_{cik}}{Max_{ck}} \leq z_{ck}$$

$$\forall k \in (1, \dots, K), \forall c \in (1, \dots, C)$$

18)

$$x_{ojk} \leq V_{jk} \quad \forall k \in (1, \dots, K), \forall j \in (1, \dots, n)$$

19)

$$a_{ik} \geq 0$$

20)

$$b_{cik}, z_{ck} \in \mathbb{N}$$

21)

$$x_{ijk}, y_{ik} = 0, 1$$

The objective function (1) minimizes the total transportation cost. Constraints (2) impose the continuity of a tour; $x_{ijk} = 1$ indicates that the truck k visits j after i and $y_{ik} = 1$ indicates that the truck k visits i . So if a truck k visits a vertex it should necessarily leave it except for the vertex $2n + r + 1$. Constraints (3) assure that the virtual departure plant 'O' can never be a destination, and the departure filling plant can only be a destination for this virtual plant. Constraints (4) enforce the virtual arrival plant $2n+r+1$ to never be a provenance and the arrival filling plant to be a provenance only for the virtual arrival plant $2n+r+1$. Constraints (5) ensure the passage by a departure filling plant and an arrival filling plant. Constraints (6) require the truck to leave the virtual departure plant if it is used. Constraints (7)-(8) enforce the truck to finish its tour at the virtual arrival plant and pass necessarily by the virtual departure plant. Constraints (9) assure that the quantity b_{cik} delivered to customer i cannot exceed the requirement of this customer. Constraints (10) limit the overall load to not exceed the truck capacity. Constraints (11) impose the satisfaction of customer's demands. Constraints (12) determine the arrival time and represent subtour elimination constraints. Constraints (13)-(14) enforce the respect of time windows. Constraints (15) ensure that the total surface occupied by racks on each truck is less than the total surface of the tray space. Constraints (16) ensure that the sum of the widths of racks loaded in the tray is less than the length of the truck. Constraint (17) determines the number of piles occupied by each type c of racks. Constraint (18) indicates that each truck can only be started at initial parked filling plant.

V. RESULTS

In this section we start by describing the problem instances used within the computational experiments. The test instances considered in our paper are derived from a real-life dataset provided by a major petroleum company in Morocco. For the exact resolution, the dataset consists of 4 problem classes; each problem class is divided into 3 dispersed configurations of different sizes and the model was tested on all 12 instances. In our problem settings we have at most fifteen customer's depots $C = C1...C15$ supplied from four gas filling plants $D = D1...D4$ and using at most forty trucks of three different configurations.

Exact solutions are obtained by implementing the model using the commercial solver IBM ILOG CPLEX 12.5 and tested on a workstation with Intel(R) Core(TM) i7 CPU @ 2.60Ghz (8 CPUs) and 16 GB RAM. Our goal is to determine how large an instance could be solved to optimality in a reasonable amount of time.

Furthermore, the requirement for each customer for three different types of gas cylinders (B03, B06 and B12), the transport costs between each vertex and the dimensions of the storage racks are all known

Furthermore, in the latest test class CL4 in which we have used the real volume of customers' demand, we use four filling plants and limiting the number of customers to be fifteen and the number of vehicles to be forty. Having discussed the savings generated by recovering the packaging, racks should return charged by empty cylinders. Thus, let us now consider the following decision rules: The delivered depot of order

D_i represent the pickup location of the same number and types of racks. In what follows, we detail our findings for each input setting.

Table I reports also the result for each instance. It shows the instances characteristics, the obtained objective value, the best bound, the real gap percentage and the CPU time. We observe that for instances of small size (CL1-01, CL1-02 and CL1-03) the resolution does not take too much time to obtain an optimal solution at less than the 107th iteration. Furthermore, when we increase the number of customers to be served, CPLEX reaches good feasible solutions for medium size instances (CL2-01 to CL3-01). After ten tests, we tried to solve problem instances (CL4-02 and CL4-03); the execution took more than one hour without obtaining results. So, the solver can be stopped sooner without significantly finding a feasible solution. Moreover, we observe that both, the number of iterations and the time of resolution increase extensively with the number of customers affected by the volume of requirement.

Summing up, the MILP model solves the small-size problems (1 gas filling plant and 1 customer) optimally within seconds. When the number of customers increases to 7 (instance CL3-02), MILP could not prove the optimality of solutions within time limits. We note also that the customers' demands used in our first test instances are in the order of a hundred cylinders only, while the actual demand varies between 2000 and 11000. By testing the model of distribution with actual customers' demands (instance CL4-02), the solver ran for a long time without generating any results. Solving our problem requires examining a very large number of combinations. If there are few customers to visit, the number of combinations to explore is low and the problem is resolved quickly. However, adding only a few clients can dramatically increase the number of combinations to explore so that the resolution time becomes excessively long; this phenomenon is called the combinatorial explosion.

An example of an optimal solution to the routing, as well as to the loading problem with five customers and two gas filling stations is depicted in Figure 3. It represents the test instance CL2-3 in which six vehicles spread over the two filling stations. As we have considered in the problem formulation two virtual stations (0 and 0') with transport cost between these virtual stations and the other stations is equal to zero. In the MILP formulation we consider that all vehicles are initially located at depot O.

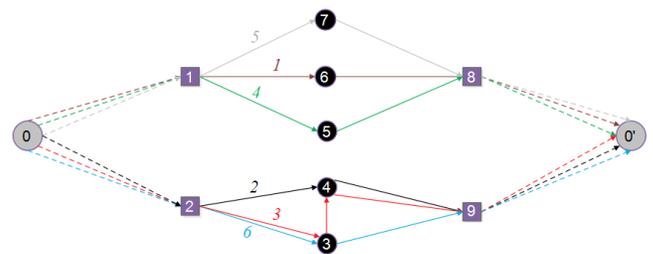


Fig. 3. Illustration of the CL2-03 Solution.

In Table II we report the visited sequence of each tour and the related loading characteristics. As customers' order in the

TABLE I. RESULTS USING ILOG IBM WITH CPLEX AS SOLVER.

I: Number of filling plant J: Number of customers K: Number of trucks R: Number of racks											
Instances	I	J	K	R	Variables	Constraints	Iterations	Objective	Bound	Rel. Gap	CPU (S)
CL1-01	1	1	2	1	75	90	107	1456	1456	0.00	0.02
CL1-02	1	2	3	2	163	208	2382	5983	5983	0.00	0.55
CL1-03	1	3	5	2	356	459	80	21403.47	21403.47	0.00	0.75
CL2-01	2	3	5	2	536	439	993	20045.57	18787.68	6.28	0.22
CL2-02	2	4	6	2	781	794	7175	10392.69	10123.23	2.59	0.42
CL2-03	2	5	6	3	967	1072	110786	14694.68	12435.76	15.37	1.86
CL3-01	3	5	7	2	1450	1293	113050	13224.71	13016.11	1.58	6.58
CL3-02	3	6	7	3	1716	1673	184771	16853.72	16424.73	2.55	105.92
CL3-03	3	10	12	3	5896	5961	17902486	30634.29	30609.29	0.08	1006.78
CL4-01	4	10	15	3	7096	6354	30136529	35215.265	32564.23	7.53	1869.53
CL4-02	4	13	20	3	14461	10980	52413896	-	-	-	>3600
CL4-03	4	15	40	3	25576	16866	52213264	-	-	-	>3600

test instance CL2-03 are small sizes in which total demand for each type of cylinder do not exceed 1000, the distribution of orders is balanced between all vehicles. Thus, we get the optimality for this test instance.

The addressed model is considered to be a NP-hard combinatorial problem. Thereby, the use of exact optimization method to solve this problem may be difficult in an acceptable CPU times, mostly when the problem involves large data sets. Therefore, the exact method is used only for small size problems.

TABLE II. CL2-03 SOLUTION WITH LOADED RACKS.

Truck	Tour	Loaded racks
1	0→1→6→8→0'	Type 1 = 5 racks Type 2 = 8 racks Type 3 = 8 racks
2	0→2→4→9→0'	Type 1 = 7 racks Type 3 = 10 racks
3	0→2→3→4→9→0'	Type 1 = 3 racks Type 2 = 2 racks Type 3 = 10 racks
4	0→1→5→8→0'	Type 1 = 5 racks Type 2 = 8 racks Type 3 = 8 racks
5	0→1→7→8→0'	Type 1 = 7 racks Type 2 = 10 racks Type 3 = 12 racks
6	0→2→4→9→0'	Type 1 = 6 racks Type 2 = 7 racks Type 3 = 3 racks

VI. GENETIC ALGORITHM FOR THE DISTRIBUTION PROBLEM

By drawing inspiration from the mechanisms of the natural evolution of living beings, many optimization problems are resolved successfully by using genetic algorithms. The GA

is a metaheuristic, introduced for the first time by [32]. The main concern is to find approximate solutions for these problems in a reasonable computation time. To implement our GA, it is necessary to have a solution coding, parameters, a fitness function to evaluate the solutions and a mechanism to obtain the initial population. In addition, we need a method of selecting individuals to reproduce and genetic operators adapted to the problem.

In this section we detail our solution approach based on a genetic algorithm (GA) for the resolution of the gas cylinder distribution problem. We present the adopted coding, the initialization as well as the different genetic operators.

A. Solution Encoding

In our genetic algorithm we represent a coding based on a permutation representation for the adopted solution of the studied distribution problem. Each solution is represented in the form of a one-dimensional table in which a sequence represents the departure plant, the order of the visited customer's depots and the arrival plant. However, this representation poses a problem of delimitation between the filling plants and the customers. Therefore, we use a number $n-2$ of permutations in which n represents the number of customers in the tour, then we assume that the filling plants start from $c = a + 1$ with $n \leq a$ (in our study we take $a = 1000$). An example of this encoding for the vehicle routing problem is given in [33]. Authors propose a permutation which contains both customers and route separators with a set $[0, \dots, 9]$ represents the customers while $[10, \dots, 12]$ are the route separators.

Figure 4 represents an example of routes made by 3 trucks (K1, K2 and K3) in which a sequence of visited customers is assigned to them. In the first route made by truck K1, the departure plant represented by 1001, visited customers 4-1-6-2-3-5 (in that order), and the arrival plant as 1001. In the second route, truck K2 starts at same filling plant 1001 to serve customers 1-6-2-4 but this time it terminates at another filling center 1002. In the third route, truck 3 starts its route at plant 1002, visits customers 2-5 -3-1-6 and returns to the same departure plant. Furthermore, this representation makes it possible to represent trucks to which no route is assigned,

when the number of trucks in the problem is fixed and greater than 1. This case is represented by the last empty route of the truck K4.

Truck	Departure station	Visited customers						Arrival station
K1	1001	4	1	6	2	3	5	1001
K2	1001	1	6	2	4			1002
K3	1002	2	5	3	1	6		1002
K4	1002							

Fig. 4. Chromosome Representation.

B. Initial Population Construction

To generate an initial population of our distribution problem we decided to make a mix between two techniques, by generating in a completely random way a half of the population and the rest with a priori good solutions. To build a solution using the random method, we assign randomly selected customers to each truck until all trucks reach their maximum loading capacity. It should be noted that this assignment is made while respecting the various constraints of the problem (time windows, two-dimensional loading constraints, satisfaction of the customer's requirement, etc.). In addition, while building the solution, if the generated route cannot be added to the under construction solution since one of the constraints of the problem will be violated, the route will be rejected. On the other side, if no route can be added to the solution under construction or there is a risk that the random method will turn around the unfeasible routes, we risk that the algorithm will run infinitely without finding a feasible solution. Therefore, we introduce a stop criterion defined by the elapsed CPU time, so the GA stops when this variable reaches the defined limit. To add a randomly generated route to the solution under construction, we have to know the filling plants, the truck that will make the trip and its maximum capacity and the customers who will be visited. The rest of the tour characteristics, in particular the start time, end time and number of racks assigned to each customer, will be calculated based on the solution method. In addition, the calculation of the start and end time of each route must take into account that the truck cannot move for another route until after the end time of its last one.

To generate an initial population of our distribution problem we decided to make a mix between two techniques, by generating in a completely random way a half of the population and the rest with a priori good solutions. To build a solution using the random method, we assign randomly selected customers to each truck until all trucks reach their maximum loading capacity. It should be noted that this assignment is made while respecting the various constraints of the problem (time windows, two-dimensional loading constraints, satisfaction of the customer's requirement, etc.). In addition, while building the solution, if the generated route cannot be added to the under construction solution since one of the constraints of the problem will be violated, the route will be rejected. On the other side, if no route can be added to the

solution under construction or there is a risk that the random method will turn around the unfeasible routes, we risk that the algorithm will run infinitely without finding a feasible solution. Therefore, we introduce a stop criterion defined by the elapsed CPU time, so the GA stops when this variable reaches the defined limit.

Algorithm 1: Random Population Initialization

Input: Population size PS , Filling plants D , Customers C , Trucks K , Racks Cr , Maximal truck capacity CK_{max}

Output: Initial population P_i

```

1 while  $p \leq PS$  do
2   for  $d \in \{1, \dots, D\}$  do
3     Random select of truck  $k$  from  $K$ 
4     while Loaded capacity  $\leq CK_{max}$  do
5       Random select of customer  $c$  from  $C$ 
6       Random select of rack  $cr$  from  $Cr$ 
7       Add rack  $cr$  to  $k$ 
8       if All constraints are respected then
9         Add customer  $c$  to route
10    Add route of  $k$  to  $P_i$ 
11   $p \leftarrow p + 1$ 

```

To add a randomly generated route to the solution under construction, we have to know the filling plants, the truck that will make the trip and its maximum capacity and the customers who will be visited. The rest of the tour characteristics, in particular the start time, end time and number of racks assigned to each customer, will be calculated based on the solution method. In addition, the calculation of the start and end time of each route must take into account that the truck cannot move for another route until after the end time of its last one.

Algorithm 2: Greedy Randomized Algorithm

Input: Population size PS , Filling plants D , Trucks K

Output: Initial population P_i

```

1 while  $p \leq PS$  do
2   for  $d \in \{1, \dots, D\}$  do
3     Random select of truck  $k$  from  $K$ 
4      $V \leftarrow$  List of potential routes of  $k$ 
5     Sort  $V$  according to the Fitness
6     while  $V \neq \emptyset$  do
7       if All constraints are respected then
8         Add route of  $k$  to  $P_i$ 
9         Delete route from  $V$ 
10   $p \leftarrow p + 1$ 

```

The Random routes generator operates in a way that at each time it produces a random potential route that can be added to the solution under construction. On the other hand, if this route is not feasible the generator reproduces another one randomly which is different from the first until a feasible potential route is found. Otherwise, this mechanism cannot continue, so it has failed to find a feasible solution randomly. The greedy random method is similar to the random method except we

do not generate the routes randomly. In fact, we associate a fitness with each route in order to be able to compare them.

C. Mutation and Crossover Operators

In the case of our adopted genetic algorithm we will try to reproduce the same mechanism of changing in the sequence as in biology. As a result, the proposed mutation operators should allow us to guarantee the homogeneity of the population and thus avoid too rapid convergence towards a local optimum. In addition, the developed mutation operators in this paper do not apply to all individuals in the population and in each generation. Indeed, we will define a mutation rate P_m which indicates what average proportion of the population must undergo a mutation.

Moreover, the use of completely free mutations allows a better exploration of the solution space. However, as in the case of population initialization, if we have a priori information about our problem, we can restrict the possibilities in order to achieve faster convergence. In this case, there is always the risk, if the information is not relevant, of guiding the GA towards a local optimum. Furthermore, the mutation operator behaves in a completely random manner in choosing the individuals subject to the mutation. As a result, we have the guarantee of the quality of the solutions obtained. In this paper, we propose a mutation operator, based on the local improvements illustrated in Figure 5.

The proposed crossover operator in this paper ensures the mixing and recombination of parental genes to form descendants with new potential. It corresponds to form two new chromosomes (children) based on an exchange of genes between two reproducers (parents). However, we apply a single point crossover operator proposed by [34], which consists in randomly choosing a crossing point. The first child is obtained by taking the first sequence from the first parent, while the rest of the genes are taken from the second parent, so that the genes which have already been taken from the first parent cannot be considered again. It should be noted that the genes from the second parent remain in the same order (Figure 6). Since the crossover operator is random, it occurs with a probability P_c , where we set the rate at 75%.

D. Fitness Function

Because of the variation of trucks capacity in the fleet the crossover and mutation operators can build solutions that do not respect the constraint of satisfying customer needs. On the other hand, our problem is an OVRP where the trucks are not attached to any filling plant at the end of their route. This possibility allows them to end their route at any other plant. Therefore, we encounter a problem of balancing the number of trucks between filling plants at the end of the day which generates a cost that we must consider in our fitness function. Thus, to obtain solutions which respect all the constraints resulting from our problem and after having applied the crossover and mutation operators we are facing a development of very complex mechanisms. Therefore, the best solution is to evaluate each individual by construct a penalization function.

In order to balance the number of trucks between filling centers at the end of the day, we make the least expensive

routes, respecting the following criteria: A truck can move for the second time after all truck are used in the filling plant; a truck cannot move without a load.

Algorithm 3: Genetic Algorithm

Input: P_i : Initial population, P_c : Crossover rate,
 P_m : Mutation rate

Output: Best solution with minimum travelling cost

```
1
2  $P_i \leftarrow$  Generate 50% using the random algorithm
3  $P_i \leftarrow$  Generate 50% using the greedy random
  algorithm
4
5 Evaluate the population using fitness function
6
7 while  $s \leq MaxSteps$  do
8
9   Random select two parents from  $P_i$ 
10  Crossover operator(Parents,  $P_c$ )
11  Mutation operator(Parents,  $P_m$ )
12
13  Evaluate solution using the fitness function
14
15   $s \leftarrow s + 1$ 
```

We adopt the following indicator parameter: "The lower the fitness value, the better the individual". The fitness function is equal to the objective function plus the trucks balancing costs.

$$Fitness = \sum_{i=1}^n \sum_{j=1}^r \sum_{k=1}^K C_{ijk} x_{ijk}$$

A selection operator used in our paper is based on Tournament method in which we select two individuals from the population in a randomly way, we compare their fitness function and we select the one how has the lower value.

E. Computational Experiments of the AG

In this section, we present the statistics and numerical results from the developed genetic algorithm. A heuristic was implemented using the Java language which is an object oriented programming language. In addition, the tests are performed on a machine with an Intel (R) Core (TM) i7 processor at 2.60 GHz (8 processors) and 16 GB of RAM.

To test the performance of our genetic algorithm as well as the power of the mutation and crossover operators used, we present an illustration of the statistics obtained during the tests carried out. For this, we will use a larger instance than the one presented in the exact resolution. The test instance considered in this part is derived from actual data provided by the same oil company. It is made up of 120 customers $C = C1...C110$ supplied by 16 filling centers $D = D1...D16$ and using at most 50 trucks of three different configurations.

We associate with each filling plant a number of heterogeneous trucks in terms of capacity. It should be noted that the tests are carried out for customer needs for a single day, but these needs influence either the number of trucks used

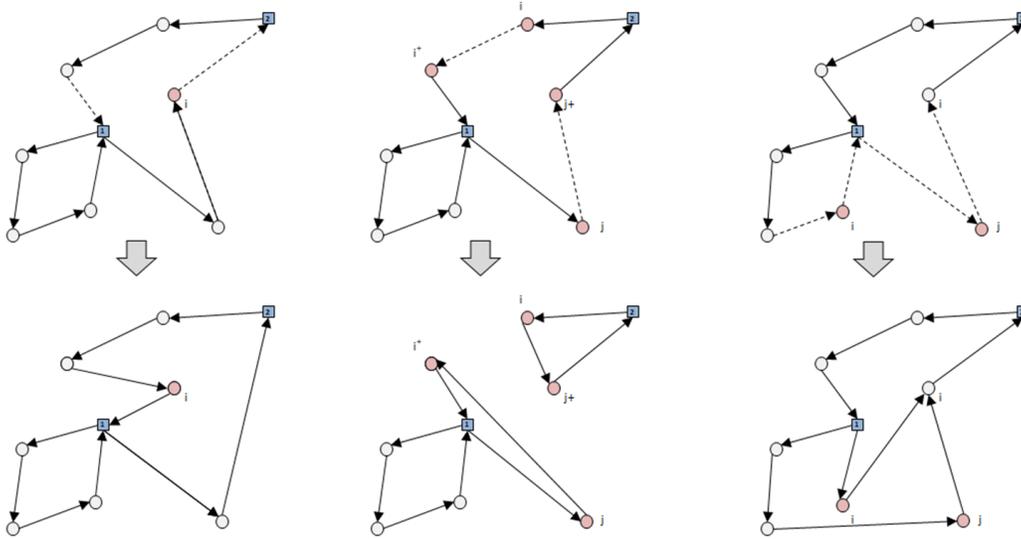


Fig. 5. Example of the Mutation Proposed Operators.

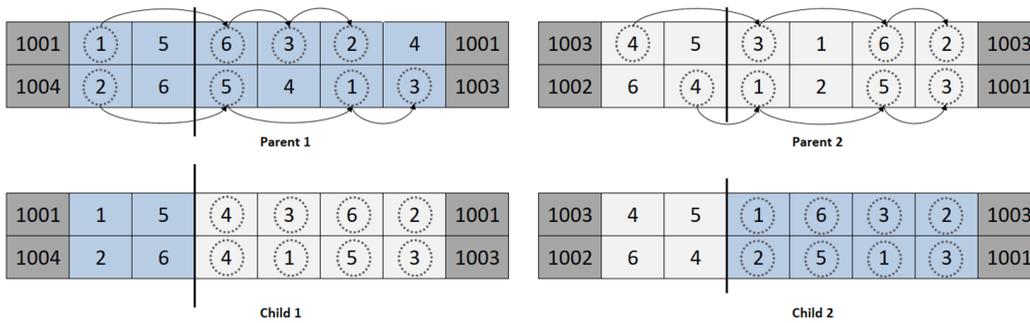


Fig. 6. An Example of the Single Point Crossover.

or the number of customers visited by the trucks or both. The solution obtained with the genetic algorithm depends on a number of parameters which are: population size, mutation rate and crossover rate. To study the influence of the population size we set the stop criterion at 100 iterations, crossover rate at $P_c = 0.85$ and mutation rate at $P_m = 0.50$. Figure 7 shows that the quality of the fitness function improves with the increase in the population size. In addition, we find a positive linear relationship between population size and the computation time.

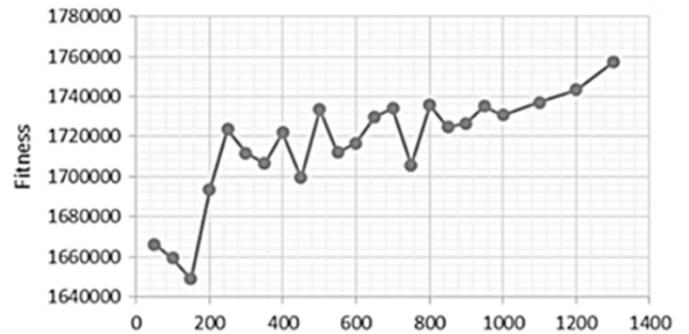


Fig. 7. Influence of the Population Size on the Fitness Value and CPU Time.

The graph illustrated in Figure 8 represents the distributions of the solution values over 37 iterations: the algorithm goes from a population very dispersed solution at the beginning to a population more centered on the optimum found at the end, thus the improvement of the population is very fast at the start (global search) and becomes slower and slower as time

passes (local search).

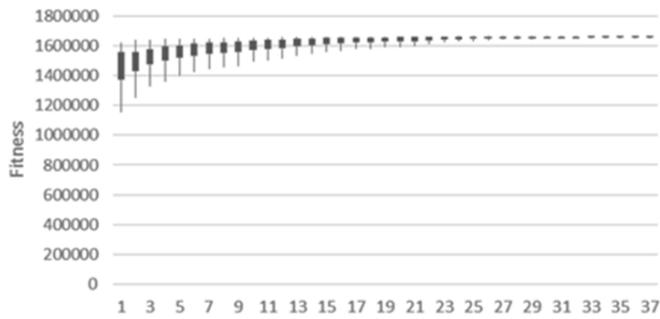


Fig. 8. Variation and Dispersion Fitness Value.

Throughout the evolution of the genetic algorithm the crossover and mutation operators can build solutions that do not respect the problem constraints. However, in most cases these solutions are rejected by the algorithm mechanisms. With the aim of observing this behavior we define a reproduction error rate according to the number of iterations.

$$\text{Error rate} = \frac{\text{Number of invalidate individuals}}{\text{Total number of individuals}}$$

The point cloud in Figure 9 shows an absence of connection between reproduction error and number of iterations. Moreover, the percentage of reproduction error varies between 2% and 8%, so the choice of the crossover and mutation operators is good depending on the probability of reproducing valid individuals. On the other hand, a good choice of the parameters of a method is very important, the adjustment of the parameters is crucial and conditions the quality of the results obtained. However, optimal tuning remains a difficult task.

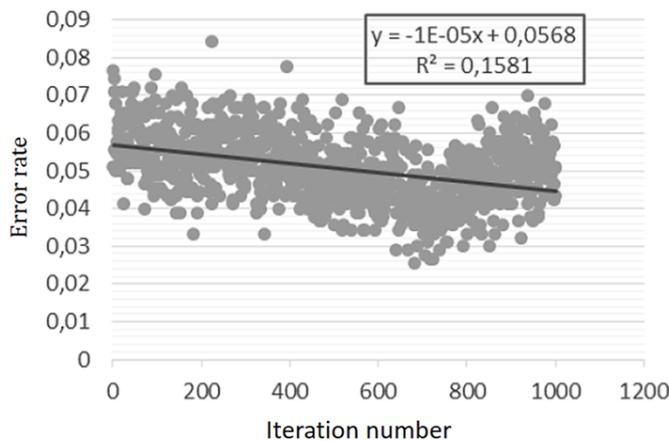


Fig. 9. Error Rate According to the Number of Iterations.

As described above, appropriate adjustment of parameters in genetic algorithms can make a significant difference in

terms of performance. Some values can provide very high performance in specific instances while giving premature convergence in others, even over the same kind of problems. The parameters used in the comparison tests between the proposed GA and the exact method are listed as flow:

- The population size equal to 1400
- The number of iteration equal to 100
- The crossover rate $P_c = 0.85$
- The mutation rate $P_m = 0.50$

Table III shows computational results of the developed genetic algorithm, which also makes comparisons with the exact method illustrated in table I. This table reports the configuration of each instance, the objective and CPU obtained from CPLEX running, solution and CPU obtained with GA and percentage Gaps. Based on results reported in table III we remark that both methods could find optimal solution for the four first test instances. The CPU time performs better in the case of GA for all test instances. In addition, the results obtained from the GA are not too far from those obtained by Cplex. This is justified by the choice of the crossover and mutation operators. Unlike the exact resolution of the studied open split delivery and pick up problem, the developed genetic algorithm gives ambitious results. We have performed tests for real instances of a large scale and in each time we obtain good solutions in a good CPU time. An example of a solution with five customers and two gas filling plants is depicted in Figure 10. It represents the test instance CL2-3 in which six trucks spread over the two filling plants. For example, we have the first truck which leaves from station 1, serves customers 3 and 4 and returns to the station 1 while the second truck which leaves from the same center, serves customer 5 and finishes its trip in the station 2. This is explained by the fact that the trip is open. On the other hand, the first truck which leaves from station 2, serves customers 1 and 3 and returns to the station 2 while the second truck which leaves from the same center, serves customer 1 then customer 2 and finishes its trip in the station 1.

Regarding the loading of each truck, it is clear that the combination is in the form of stacks of n racks of type L, M and N. For example, the truck which leaves from station 1 to serve customer 4 is loaded with five racks of type L, eight racks of type M and eight racks of type N.

VII. CONCLUSIONS

In this paper, we have addressed the open multi-depot split delivery and pickup routing problem with two-dimensional loading constraints in the case of a real-world problem of an LPG distribution company with its real-life assumptions and data. First, we have formulated our problem as a mixed integer linear programming (MILP), then we have tested the problem for 12 tests instances grouped in 4 classes in which we have proved optimality for 10 instances. Also, we have discussed the effect of changing the setting of the input data on the quality of the solutions obtained and on the running time.

TABLE III. COMPARISON BETWEEN RESULTS FROM CPLEX AND FROM THE PROPOSED GA.

Instances	I: Number of filling plant J: Number of customers K: Number of trucks R: Number of racks				CPLEX		GA		Gaps(%)	
	Configuration				Objective	CPU(S)	Solution	CPU(S)	Obj. Gap(%)	CPU Gap (%)
	I	J	K	R						
CL1-01	1	1	2	1	1456	0.02	1456	0.01	0	50
CL1-02	1	2	3	2	5983	0.55	5983	0.01	0	98.18
CL1-03	1	3	5	2	21403.47	0.75	21403.47	0.01	0	98.67
CL2-01	2	3	5	2	20045.57	0.22	20045.57	0.04	0	81.82
CL2-02	2	4	6	2	10392.69	0.42	11023.92	0.13	6.07	69.05
CL2-03	2	5	6	3	14694.68	1.86	14918.11	0.08	1.52	95.7
CL3-01	3	5	7	2	13224.71	6.58	14036.06	1.03	6.14	99.54
CL3-02	3	6	7	3	16853.72	105.92	14564.65	1.56	13.58	98.53
CL3-03	3	10	12	3	30634.29	1006.78	31256.36	3.09	2.03	99.69
CL4-01	4	10	15	3	35215.265	1869.53	36952.47	4.7	4.93	99.75
CL4-02	4	13	20	3	-	>3600	43625.24	4.33	-	-
CL4-03	4	15	40	3	-	>3600	61452.87	6.87	-	-

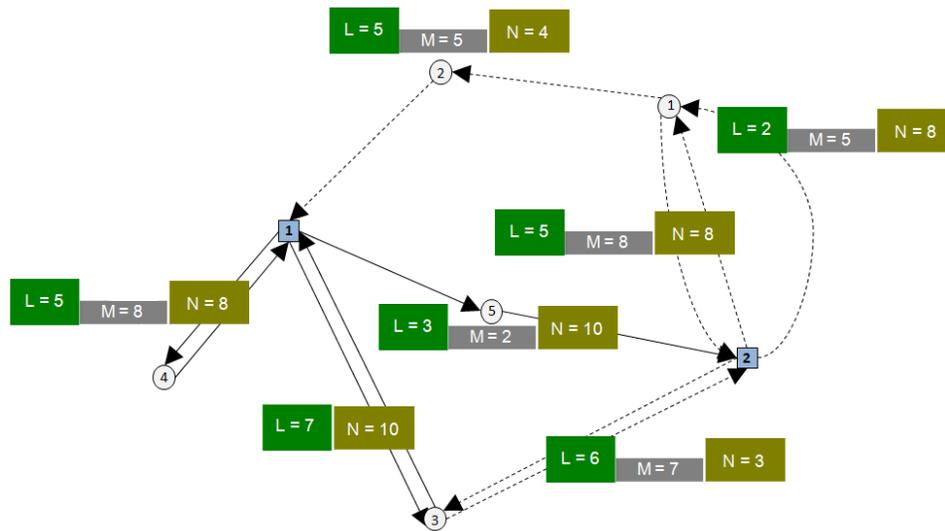


Fig. 10. An Example of Solution.

Due to the difficulty in solving the combination of two NP-hard problems, such as vehicle routing and two-dimensional loading; efficiently, feasible solutions for large instances can be obtained only by developing a metaheuristic method like genetic algorithm. We developed a genetic algorithm and execute it for large data by increasing the size of instances which we are not capable of solving exactly within reasonable computational times. Finally, we gave some statistics and results for the developed GA and discuss obtained results.

REFERENCES

[1] G. B. Dantzig and J. H. Ramser, *The Truck Dispatching Problem*, Management Science, vol. 6, pp 80-91, 1959.

[2] H. Wilson, J. Sussman, H. Wang, B. Higonnet, *Scheduling algorithms for dial-a-ride systems.*, Massachusetts Institute of Technology, Urban Systems Laboratory, Technical Report USL TR-70-13, 1971.

[3] L. Schrage, *Formulation and Structure of More Complex-Realistic Routing and Scheduling Problems.*, Networks, vol. 11, pp 229–232, 1981.

[4] S. Ladany, A. Mehrez, *Optimal routing of a single vehicle with loading and unloading constraints.*, Transportation Planning and Technology, vol 8, pp 301–306, 1984.

[5] M. Dror, G. Laporte, P. Trudeau, *Vehicle Routing with Split Deliveries.*, Discrete Applied Mathematics, vol. 50, pp 239-254, 1994.

[6] D. Sariklis, S. Powell, *A heuristic method for the open vehicle routing problem.*, Journal of the Operational Research Society, vol. 51, pp 564–573, 2000.

[7] J. M. Belenguer, M. C. Martinez, E. A. Mota, *Lower Bound for the Split Delivery VRP.*, Operational Research, vol. 48, pp 801-810, 2000.

[8] J. A. Brandão, *tabu search algorithm for the open vehicle routing problem.*, European Journal of Operational Research, vol. 157, pp 552-564, 2004.

[9] C. D. Tarantilis, G. Ioannou, C. T. Kiranoudis, G. P. Prastacos, *A threshold accepting approach to the open vehicle routing problem.*, RAIRO Operations Research, vol. 38, pp 345-360, 2004.

[10] R.W. Eglese, Z. Fu, L. Li, *A tabu search heuristic for the open vehicle routing problem.*, Journal of the Operational Research Society, vol. 56, pp 267–274, 2005.

- [11] R. Hussain, T. Assavapokee, B. Khumawala, *Supply Chain Management in the Petroleum Industry: Challenges and Opportunities.*, International Journal of Global Logistic Supply Chain Management, vol. 1, pp 90-97, 2006.
- [12] F. Guenther, K. F. Doerner, R. F. Hartl, M. Iori, *Ant colony optimization for the two-dimensional loading vehicle routing problems.*, Computers and Operations Research, vol. 36, pp 655-673, 2007.
- [13] F. Carrabs, J. Cordeau, G. Laporte, *Variable neighborhood search for the pickup and delivery traveling salesman problem with LIFO loading.*, INFORMS Journal on Computing, vol. 19, pp 618-632, 2007.
- [14] M. Iori, J. Salazar-González, D. Vigo, *An exact approach for the vehicle routing problem with two dimensional loading constraints.*, Transportation Science, vol. 41, pp 253-264, 2007.
- [15] P. P. Repoussis, C. D. Tarantilis, G. Ioannou, *The open vehicle routing problem with time windows.*, Journal of the Operational Research Society, vol. 58, pp 355-367, 2007.
- [16] A. Attanasio, A. Fuduli, G. Ghiani, C. Triki, *Integrated shipment dispatching and packing problems: a case study.*, Journal of Mathematical Modelling and Algorithms, vol. 6, pp 77-85, 2007.
- [17] M. Gendreau, M. Iori, G. Laporte, S. Martello, *A tabu search heuristic for the vehicle routing problem with two-dimensional loading constraints.*, Networks, vol. 51, pp 4-18, 2008.
- [18] G. Erdogan, J. Cordeau, G. Laporte, *The pickup and delivery traveling salesman problem with first-in-first-out loading.*, Computers and Operations Research, vol. 36, pp 1800-1808, 2009.
- [19] X. Li, P. Tian, S. C. Leung, *An ant colony optimization metaheuristic hybridized with tabu search for open vehicle routing problems.*, Journal of the Operational Research Society, vol. 60, pp 1012-1025, 2009.
- [20] S. Yu, C. Ding, K. Zhu, *A hybrid GA-TS algorithm for open vehicle routing optimization of coal mines material.*, Expert Systems With Applications, vol. 38, pp 10568-10573, 2011.
- [21] F. Archetti, I. Giordani, G. Mauri, E. Messina, E. *A new clustering approach for learning transcriptional modules.*, International Journal of Data Mining and Bioinformatics, vol. 6, pp 304-23, 2012.
- [22] J. H. Wilck, T. M. Cavalier, *A genetic algorithm for the split delivery vehicle routing problem.*, American Journal of Operational Research, pp 207-216, 2012.
- [23] L. Martinez, C. Amaya, *A vehicle routing problem with multi-trips and time windows for circular items.*, Journal of the Operational Research Society, vol. 64, pp 1630-1643, 2012.
- [24] S. Khebbache-Hadji, C. Prins, A. Yalaoui, M. Reghioui, *Heuristics and memetic algorithm for the two-dimensional loading capacitated vehicle routing problem with time windows.*, Central European Journal of Operations Research, pp 307-336, 2013.
- [25] S. Leung, Z. Zhang, D. Zhang, X. Hua, M. K. Lim, *A meta-heuristic algorithm for heterogeneous fleet vehicle routing problems with two-dimensional loading constraints.*, European Journal of Operational Research, pp 199-210, 2013.
- [26] A. Annouch, A. Bellabdaoui, J. Minkhar, *Split delivery and pickup vehicle routing problem with two-dimensional loading constraints.*, 11th International Conference on Intelligent Systems: Theories and Applications, 7772277, 2016.
- [27] M.N. Haddad, R. Martinelli, T. Vidal, S. Martins, L. S. Ochi, M.J.F. Souza, R. Hartl, *Large neighborhood-based metaheuristic and branch-and-price for the pickup and delivery problem with split loads.*, European Journal of Operational Research, vol. 270, pp 1014-1027, 2018.
- [28] Y. Niu, Z. Yang, P. Chen, J. Xiao, *Optimizing the green open vehicle routing problem with time windows by minimizing comprehensive routing cost.*, Journal of Cleaner Production, vol. 171, pp 962-971, 2018.
- [29] S. Ling, T. Fengming, W. Songyi, *Multi-depot open vehicle routing problem with time windows based on carbon trading.*, International Journal of Environmental Research and Public Health, vol. 15, 2018.
- [30] R. Tarit, C. Phanchita, P. Panurut, V. Rapeeporn, *An Optimization for Split Delivery Open Vehicle Routing Problem with Manual Materials Handling.*, 2nd International Conference on Engineering Innovation (ICEI), 2018.
- [31] F. Stefano, J. C. Fransoo, V. W. Tom, V. Woensel, D. Jing-Xin, *A variant of the split vehicle routing problem with simultaneous deliveries and pickups for inland container shipping in dry-port based systems.*, Transportation Research Part E: Logistics and Transportation Review, vol. 142, 102057, 2020.
- [32] J. H. Holland, *Adaptation in natural and artificial systems.*, Ann Arbor: The University of Michigan Press, 1975.
- [33] E. Alba and B. Dorronsoro, *The exploration/exploitation tradeoff in dynamic cellular evolutionary algorithms.*, IEEE Transactions on Evolutionary Computation, vol. 9, pp 126-142, 2005.
- [34] S. Hartmann, *A competitive genetic algorithm for resource-constrained project scheduling.*, Naval Research Logistics, vol. 45, pp 733750, 1998.