A Data Driven Hybrid Heuristic for the Dial-A-Ride Problem with Time Windows

Slim Belhaiza

***Abstract*—The Dial-A-Ride Problem (DARP) consists of designing pick-up and delivery vehicle routing schedules for a set of customers. Commonly, it arises in door-to-door transportation for impaired or elderly people. The main objective is to accommodate as many users as possible with a minimum operation cost. It adds realistic precedence and transit time constraints on the pairing of vehicles and customers. This paper proposes a hybrid evolutionary heuristic for the dial-a-ride problem with time windows (DARPTW). It combines evolutionary crossover operators with a hybridized VNS algorithm. Data extracted from routes with DARP infeasibilities are used to guide customized local search operators. Using route duration minimization as its main objective, the competitiveness of the proposed heuristic is tested on benchmark instances.**

***Index Terms*—Evolutionary heuristic, Tabu Search, Variable Neighborhood Search, Dial-A-Ride Problem, Time Windows.**

I. Introduction

The transportation of passengers, furniture, electronics, food and other kinds of goods to public places, business malls, hyper-markets or grocery stores in populated areas plays a key role in modern cities' economy. Vehicle Routing Problems (VRP) address various aspects of these key activities. The dial-a-ride problem (DARP) defines a particular subclass of VRP where heterogeneous fleets of vehicles should be considered to meet customers' needs. This is mainly due to accessibility restrictions that forbid some vehicles from visiting some locations. Some destinations may not be equipped with adequate installations to process pick-up and delivery operations. For public and private transportation services, this may also happen when vehicles are not equipped to accommodate passengers with limited mobility. The most usual case arises in door-to-door transportation for senior or impaired people. The aim of the DARP is to design a set of minimum cost vehicle routes satisfying all requests, under a set of constraints. These constraints include vehicle maximum capacity, route maximum duration and passenger maximum transit time. The extension of the DARP to cases where service time for customers should be within a certain period of time is called DARP with time windows (DARPTW). The DARP is called static when all requests are known at the time of planning. Otherwise, it is called dynamic if these requests are gradually received during the execution of a primary routing plan.

This paper focuses on the DARP with Time Windows (DARPTW), where pick-ups and drops have to be initiated during specific time windows. The first innovation in our approach lies in the use of extracted DARP infeasibility data. These DARP infeasibilities are categorized into four different types. Our heuristic uses the information on the type of DARP infeasibilities and addresses it with the appropriate customized DARP operators. These operators are all part of an evolutionary hybridization of the variable neighborhood search. This hybrid heuristic uses the concept of adaptable memory of solutions, proper to tabu search, as well as evolutionary crossover operators. The second innovation in our approach lies in the choice of the main objective function and its components customized to fit the DARP requirements. In this paper, we aim to minimize the total duration of the designed routes. While this choice may not appear appropriate in the DARP context, it is supported by the computational results obtained with the proposed heuristic. These results show that the number of vehicles required to serve all customers is reduced for most of the benchmark instances tested. Section II describes the DARPTW. Section III provides some references on TS-based, VNS-based and GA-based heuristics used in the DARP context. Section IV presents the proposed heuristic and details its different steps. Section V presents the computational results on benchmark DARPTW instances. Finally, section VI summarizes the paper.

Department of Mathematics and Statistics, King Fahd University of Petroleum & Minerals, KSA. slimb@kfupm.edu.sa

II. Problem Description

The DARPTW could be described as a pick-up and delivery VRP with time windows, maximum duration and maximum transit-time conditions. The DARPTW could be defined on a graph $G = (V, A)$, where V is the set of vertices and A is the arc set. The DARPTW contains a set of n customers and a set of m vehicles. A customer i has two different stops: one for the pick-up denoted (i^+), and one for the drop denoted (i_-). Each stop, either pick-up or drop, has a non-negative service time s_i, generally assumed to be similar at corresponding pick-ups and drops, and two time windows: $[l_{i^+}, u_{i^+}]$ for the pick-up and $[l_{i_-}, u_{i_-}]$ for the drop. A pick-up stop is generally signaled by a positive demand d_i and a maximum transit time r_i between the departure from i^+ and the arrival at i_-. A drop stop is generally signaled by a negative demand $-d_i$. A vehicle k has a maximum capacity q_k, a maximum route duration t_k and work time window $[l_k, u_k]$. The DARPTW main goal is to design a set of m vehicle routes on G such that each route starts and ends at a depot D and satisfies all constraints. Each customer i has to be picked-up and dropped by the same vehicle, within its respective time windows. Each route performed by a vehicle k does not exceed the vehicle load capacity q_k and its total duration does not exceed the maximum duration t_k. One should know that if a vehicle arrives at a customer's pick-up or drop before the beginning of the corresponding time window, it should wait until then.

978-1-5386-2726-6/17/$31.00 ©2017 IEEE

The waiting times are considered as part of the route duration. The total route duration would be composed of the total travel, waiting and service times.

III. Literature Review

Cordeau and Laporte [8] present a survey on the DARP and its models and algorithms. Parragh and Schmid propose a hybrid large neighborhood heuristic for the DARP [23]. With a variety of neighborhoods used, they improve many best known benchmark instances' solutions. Madsen et al. [15] proposed an algorithm based on an insertion heuristic for the DARPTW with multiple capacities, and multiple objectives. The algorithm is implemented in a dynamic environment for on-line scheduling of new requests. Guerreiro *et al.* [11] proposed a bi-objective approach for the dial-a-ride public transportation problem. Under a constraint on the maximum duration of each route, the proposed algorithm minimizes the maximum total transit time and the total waiting time simultaneously. Masson *et al.* [17] propose an adaptive large neighborhood search (ALNS) algorithm for a variant of the DARP where users can change vehicles during their trip at predefined transfer points (DARPT). The proposed algorithm yielded 8 % of savings on real-life instances. While some of the literature proposes Branch-and-Cut [14], machine learning [18] or Simulated Annealing [24] methods for the DARP, Genetic algorithm (GA), tabu search (TS) and variable neighborhood search (VNS) are among the most successful approximate approaches for the DARP. This section reviews some of their most recent applications in the DARP context.

A. TS DARP Applications

Tabu search (TS) was proposed by Glover [10]. It is a meta-heuristic based on local search methods. Local search checks the closest neighbors to a solution using one or more similarity criteria. Local search methods usually gets stuck in suboptimal regions or plateaus where different solutions have equal costs. TS prohibits the search from going back to solutions which have been explored based on moves or patterns. Cordeau and Laporte [7] presented a TS heuristic for the DARP with capacity, duration and maximum user transit time constraints. The proposed algorithm proved to be efficient on real-life instances randomly generated. Attanasio *et al.* [1] proposed a parallel TS heuristic for the dynamic multi-vehicle dial-a-ride problem. In the dynamic DARP, the users' requests are continuously received through the day. While satisfying a number of operation constraints, the initial objective is to fill the routes with as many requests as possible. The authors compared different parallelizations of a TS heuristic previously developed for the static DARP and show that the algorithms were able to satisfy a high percentage of the dynamic users' requests. Kirchler and Calvo [13] proposed a Granular TS algorithm for the DARP. Their algorithm's objective was to produce good solutions in three minutes as time upper bound. The proposed Granular Tabu Search uses passengers clusters based on information on both pick-up and delivery locations, and time windows. Passengers within the same cluster are served by the same vehicle. The proposed algorithm produced good results when compared to regular TS algorithm, a GA, and a VNS algorithm. Paquette *et al.* [21] proposed a multi-criteria approach combined with Tabu Search. The three criteria are related to the cost and quality of service.

B. GA DARP Applications

Genetic Algorithms (GA) founding concepts were set by Holland [12] as an adaptive heuristic search method. They are mostly inspired from population genetics in biology. The generation of new individuals requires a representation scheme acting as the key for the selection, recombination and mutation. The representation scheme translates the most basic components of a solution to a gene. The selection is generally based on some criteria, which depend on the nature of problem to be solved. The recombination process is based on genes of selected parents to be inherited or not by their offspring. The mutation alters an offspring to preserve genetic diversity. Mutation usually happens with a small probability. A new generation is obtained by repeated selection, reproduction and mutation processes. Many papers studied the application of Genetic Algorithms (GA) to the DARP. Among these works, Chevrier *et al.* [9] proposed a multi-objective evolutionary hybrid heuristic applied to random and real-life DARP instances. They proposed local search operators specialized for the DARP. Masmoudi *et al.* [16] proposed a genetic algorithm for the heterogeneous DARP (H-DARP). They provide efficient construction, crossover and local search operators customized to the H-DARP specificities.

C. VNS DARP Applications

VNS was proposed by Mladenovic and Hansen [19]. It is famous for its easy implementation and limited number of parameters. It explores different neighborhoods in search for a global minimum. The neighborhoods usually have different central points. It intensifies the search by altering the best solution and diversifies it using different neighborhood types. Similarly to GA, VNS applications to the DARP are numerous. Muelas *et al.* [20] proposed a distributed VNS algorithm for a large scale DARP. Their algorithm used requests partition and routes' combination . Tested on a set of scenarios of a large-scale problem in San Francisco, the algorithm proved to be efficient. Parragh *et al.* [22] described a VNS-based heuristic using three kinds of neighborhoods. The first kind used simple swap operations customized to the dial-a-ride problem. The second used the ejection chain idea. The third was based on the existence of arcs where the vehicle is empty. They report peformant results on different benchmark instances.

IV. HG-DARP

Although it could be tuned to minimize the total travel time only, our HG-DARP heuristic uses the minimization of the total duration as its main objective. While minimizing the total travel time is commonly used in the DARP literature, it includes only the time traveled by every vehicle. Minimizing the total duration is commonly used for the VRP with Multiple

Time Windows [2], [5], [3], [4]. The total duration includes travel, service and waiting times. From a driver or customer point of view, minimizing the total duration could provide more interesting routes. Drivers may spend less time waiting at each stop and customers may spend less time traveling and being served. This point of view is confirmed by the computational results thoroughly detailed in section V.

The seven key core steps of our HG-DARP are inspired by those of HGVNS described in [4], but involve DARP customized operators in almost every one of them. These operators are based on the four different kinds of DARP infeasibilities. The first kind does not involve any route. It is the most basic form of infeasibility as it is due to a customer with both pick-up and drop stops left unserved. We here call it DARP infeasibility of **type** 1. The second kind called **type** 2 involves a single route where a pick-up, or a drop, is found without its corresponding drop, or pick-up, left unserved. The third kind **type** 3 also involves a single route where a drop is found before its corresponding pick-up. The fourth and last kind of DARP infeasibilities **type** 4 involves two different routes found to be sharing a pick-up and its corresponding drop.

- Step (1) addresses DARP infeasibilities of types 1 and 2. It inserts stops left unserved taking into account the relation between pick-ups and drops.
- Step (2) applies local search DARP improvement operators. It addresses DARP infeasibilities of types 3 and 4.
- Step (3) changes either the central point or the neighborhood of the search. It recenters the search around a new local minimum X', if it improves the cost function and the total number of DARP infeasibilities, and then reinitiates the local search. Otherwise, if a number of successive iterations without improvement is reached, it calls a restart procedure.
- Step (4) shakes the current solution X, using DARP shaking operators, to obtain a solution X" from a neighborhood κ of X randomly selected.
- Step (5) applies a reset procedure from a randomly chosen solution belonging to a pool of best DARP solutions.
- Step (6) applies DARP evolutionary operators to generate a new solution.
- Step (7) applies a ruin and recreate DARP procedure.

A. Initialization and Insertion

The construction of an initial solution randomly iterates through the m vehicles, which is slightly different from HGVNS as it iterates through vehicles in the order of their indices. Step (1) assigns as customers to vehicle k as long as the maximal total dial-a-ride trip duration t_k and vehicle's capacity q_k allow. The assignment phase addresses type 1 DARP infeasibilities as it chooses, among all unassigned customers, a customer's pick-up and drop stops that induce the least increase in total cost such that the total trip duration does not exceed t_k and the total load does not exceed the q_k. At this stage, all possible insertion positions within a route are tried to ensure that the stops to be added to the route are inserted with the best initial setting without increasing the total number of DARP infeasibilities of the route. This procedure usually generates a partial solution as it may leave some customers unassigned. In that case, insertion is repeatedly called at every iteration of HG-DARP. The insertion phase addresses DARP infeasibilities of type 2 differently depending on the stop left unassigned. If the type 2 infeasibility corresponds to a drop (i_-) stop left unserved, it searches for the route containing its corresponding pick-up i^+ and tries to insert (i_-) at every possible position after (i^+). If the type 2 infeasibility corresponds to a pick-up (i^+) stop left unserved, it searches for the route containing its corresponding drop (i_-) and tries to insert (i^+) at every possible position before (i_-). If the total number of DARP infeasibilities is reduced or maintained, this yields a successful insertion. If not, the insertion is left to be tried again at the next iteration.

B. DARP local search

While HG-DARP naturally inherits the local search operators of HGVNS, it uses a number of customized DARP improvement operators detailed as follows. Step (2) defines VNS-DARP steepest descent phase. This step addresses DARP infeasibilities of types 3 and 4. It starts from X and returns an improving solution X'. It mainly consists of two phases. The **first phase** is a single-route DARP improvement. The **second phase** is a multiple-route DARP improvement.

1) Single Route DARP operators: The first phase tries to improve every vehicle route by applying up to three DARP-exchange operators. The second operator is only applied when the first fails to find a solution with a lower cost of the route and a lower number of DARP infeasibilities. Similarly, the third operator is only applied when the second cannot decrease the cost or the number of DARP infeasibilities. In the current HG-DARP implementation, the selection of stops to be exchanged is based on the information provided by DARP infeasibilities of type 3.

- The DARP relocate operator moves a stop from its current position on the route to a different chosen position.
- The DARP stop exchange operator swaps the positions of two selected stops on the route.
- The DARP arc exchange operator swaps the order of the stops of a selected arc of the route.

(i) DARP relocate operator: The DARP relocate operator (DARP-relocate) moves a given stop from its current position on the route to a different chosen position if this maintains or reduces the number of DARP infeasibilities of the route and yields an expected cost saving. While the second condition of the move is common to all VRP operators, the first condition is proper to the DARP. The relocation may concern any stop on the route. It also addresses type 3 DARP infeasibility as it detects any pick-up appearing after its corresponding drop and moves it before if this maintains or lowers the solution cost. Figure 1 shows how (i^+) is moved to a position before (i_-) which reduces the number of type 3 DARP infeasibilities by one unit.

(ii) DARP stop exchange operator: The DARP stop exchange operator (DARP-swap) swaps the positions of two

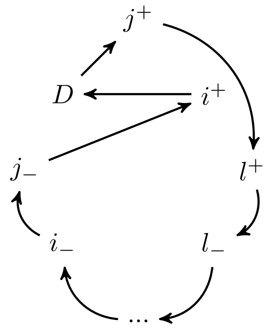

Fig. 1. DARP relocate.

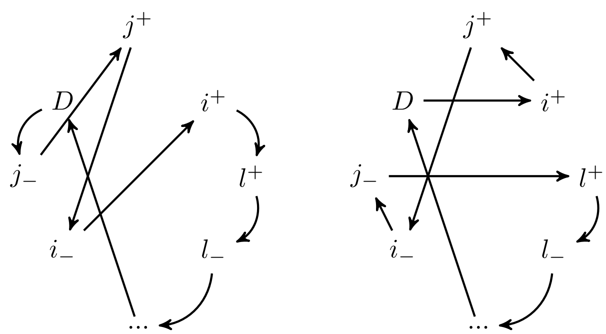

Fig. 2. DARP swap.

stops on a given route if this maintains or reduces the number of DARP infeasibilities and yields an expected cost saving. The relocation may concern any pair of stops on the route. It also addresses type 3 DARP infeasibility as it detects any pick-up appearing after its corresponding drop and swaps it with its drop, or any other stop (pick-up or drop) in a position before it, if this maintains or lowers the solution cost. Figure 2 shows how (i^+) is swapped with (j_-) which reduces the number of type 3 DARP infeasibilities by two units.

(iii) DARP arc exchange operator: The DARP arc exchange (DARP-2X-opt) operator swaps the positions of two selected arcs on the same route. The exchange may concern any pair of arcs on the route if its yields a cost saving and maintains the number of DARP infeasibilities. It also addresses type 3 DARP infeasibilities as it detects any pick-up appearing after its corresponding drop and exchanges one of the two arcs, ingoing or outgoing, with any other arc in the route with a position before its corresponding drop if this maintains or lowers the solution cost. Figure 3 shows how the arc (i^+, l^+) is swapped with the arc (j^+, i_-) which reduces the number of type 3 DARP infeasibilities by two units.

2) Multiple-Route operators: The second phase applies to two vehicles using three DARP multi-route exchange operators. Each operation is applied when the preceding one fails to lower the solution cost or the total number of DARP infeasibilities. This phase simultaneously addresses DARP infeasibilities of type 4 while seeking cost saving stops or arcs exchanges between different routes.

- The DARP inter-route relocate operator moves one stop from a route and places it in a different route.

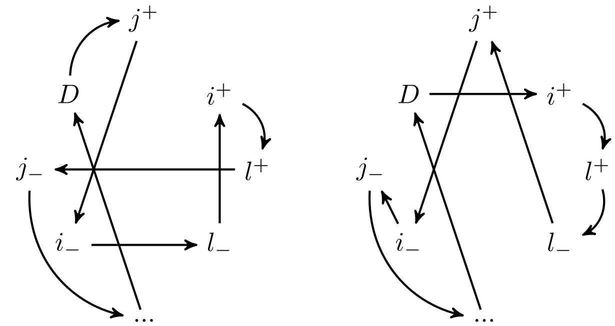

Fig. 3. DARP 2X-opt.

- The DARP inter-route swap operator exchanges two selected stops between two different routes.
- The DARP inter-route cross operator moves a selected arc from a route and inserts it in a different route.

(i) DARP Inter-Route relocate operator The DARP inter-route relocate operator (DARP-One-opt) moves a selected stop from a given route and places it in a selected chosen position of an adequately chosen route if this yields a lower solution cost and maintains or lowers the number of DARP infeasibilities. It also addresses type 4 DARP infeasibilities as it detects any pick-up appearing in a route different from its corresponding drop and tries two different one-opt moves. At first, it tries to insert the pick-up stop in its corresponding drop route. If it fails, it tries to insert the drop in its corresponding pick-up route. If the move is executed, one type 4 DARP infeasibility may yield zero or one type 3 DARP infeasibility. In other words, infeasible DARP insertion is tolerated if the solution cost is improved while transforming the DARP infeasibility type. Eventually, the new type 3 infeasibilities would be addressed by single-route DARP operators. Although DARP feasible moves are given priority over cost reduction, the move could also be executed if it yields a lower solution cost and maintains or lowers the total number of DARP infeasibilities. Hence, a DARP infeasibility of type 3 may be transformed to type 4. Figure 4 shows how the drop (i_-) from route k' is moved to route k after its corresponding drop (i^+) which reduces the number of type 4 infeasibilities by one unit.

(ii) DARP inter-Route swap operator: The DARP inter-route swap operator (DARP-two-opt) exchanges two stops from two different routes. It detects different pairs of pick-ups, drops or unrelated pick-up and drop, with DARP infeasibilities of type 4. At first, it tries to swap two pick-ups (i^+) and (j^+) if one's drop appears in the other's route, and vice versa. If it fails, it tries to swap two drops (i_-) and (j_-) if one's pick-up appears in the other's route, and vice versa. Finally, it tries to swap two unrelated pick-up (i^+) and drop (j_-) if (i_-) appears in (j_-)'s route and (j^+) appears in (i^+)'s route. The move is executed if it maintains or lowers the solution cost while maintaining or reducing the total number of DARP infeasibilities. If the move is executed, two type 4 DARP infeasibilities may yield zero, one or two type 3 infeasibilities. Again, infeasible DARP insertion is tolerated if the solution cost is improved while transforming the DARP infeasibility type. In that case, the new type 3 infeasibilities would be addressed by single-route

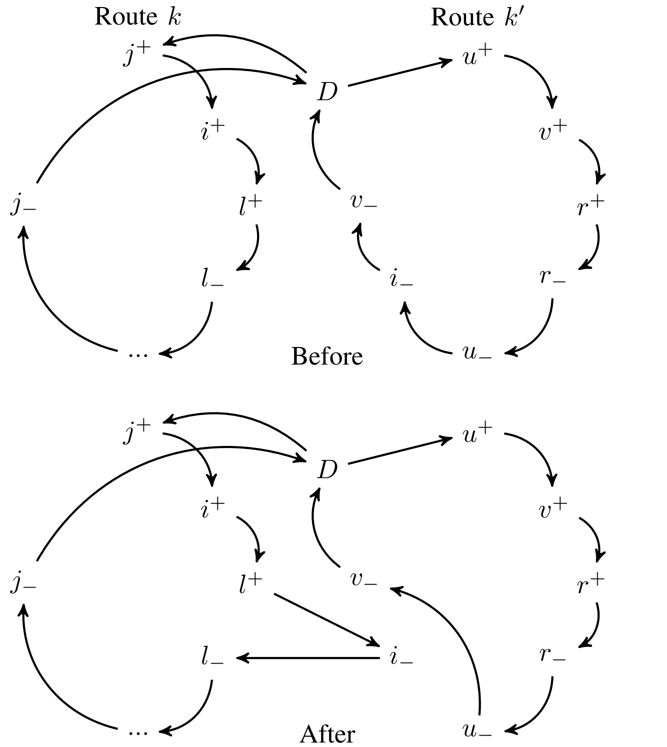

Fig. 4. DARP one-opt.

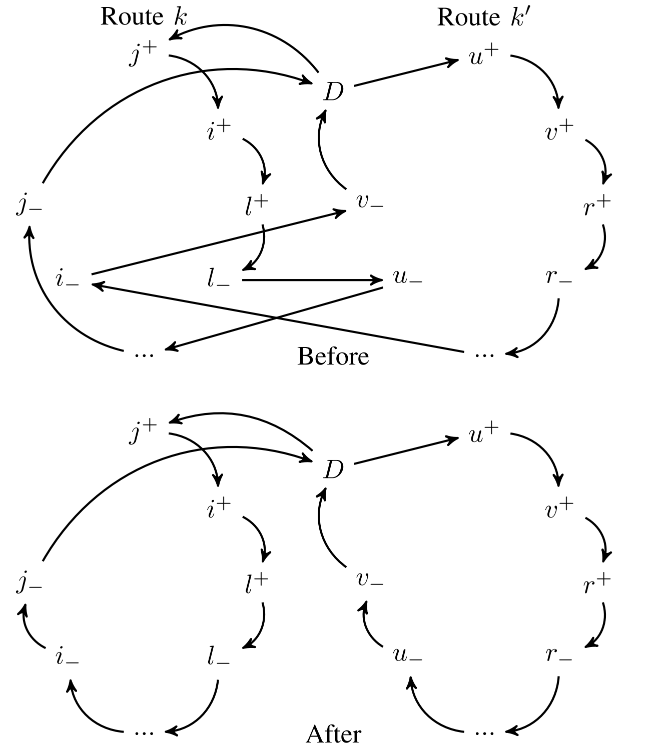

Fig. 5. DARP two-opt.

DARP operators. It is worth mentioning that the two swapped stops do not necessarily exchange their position. While DARP feasible moves are given priority over cost reduction, the move is also executed if it yields a lower solution cost and maintains or lowers the total number of DARP infeasibilities. Eventually, type 3 DARP infeasibilities may be transformed to type 4 that could be addressed by the DARP-one opt operator. Figure 5 shows how the drop (i_-) from route k' is swapped with the drop (u_-) from route k, which reduces the number of type 4 DARP infeasibilities by two units.

(iii) DARP Inter-Route cross operator: The DARP inter-route cross operator (DARP-1X-opt) moves a selected arc from a route and inserts it in a different route. It detects a particular configuration of type 4 DARP infeasibilities as it may involve different pairs of pick-ups, drops or unrelated pick-up and drop required to be moved to the same route. In its first configuration, it tries to move the arc (i^+, j^+) to the route containing (i_-) or (j_-), or tries to move the arc (i_-, j_-) to the route containing i^+ or j^+. In its second configuration, it tries to move the arc (i^+, j_-) to the route containing (i_-) or (j^+), or tries to move the arc (i_-, j^+) to the route containing i^+ or j_-. Again, the move is executed if it maintains or lowers the solution cost while maintaining or reducing the total number of DARP infeasibilities. If the move is executed, one type 4 DARP infeasibility yields zero or one type 4 infeasibility. Infeasible DARP insertion is tolerated if the solution cost is improved while maintaining or transforming the DARP infeasibility type. The new type 4 infeasibility would be addressed by DARP-one-opt or DARP-Two-opt operators. Figure 6 shows how the arc (u_-, v_-) is moved from route k to route k', which reduces the number of type 4 DARP infeasibilities by two units.

3) Move-Or-Not: A routing solution X' improves X if it reduces the total number of DARP infeasibilities F or the cost function f (Step (3)). The total number of DARP infeasibilities F is obtained by scanning the whole routing solution at every iteration. This scanning is not only used to penalize the cost function f, but also to detect the infeasibility types to be addressed. The cost function f is a weighted sum of the total duration, the violation of the time window constraints, the overloading of the vehicles, and the non-satisfaction of the maximum duration constraints. It also adds a precedence penalty to each route's cost depending on the number of DARP infeasibilities of type 1, 2, 3 and 4. HG-DARP reduces the total cost by allowing moves to infeasible solutions that improve the overall value of the objective function and somehow the level of non-satisfaction of the different constraints. The DARP infeasibilities add violation penalties to the cost function f. If the number of infeasibilities of route k, denoted F_k, is larger than zero and less or equal to 3 the penalty ω^{F_k} is added to the route's cost. If F_k is larger than three, the penalty $\bar{\omega}$ is added to the route's cost. HG-DARP uses the backward slack time procedure [2] to minimize waiting times and delay departure times from the depot. The solution characteristics are regarded as genes. The most basic representation of a gene in our DARP implementation is simply the assignment of a stop to a given vehicle. If the routing solution X' has exactly the same genes as one of the routing solutions in the best solutions pool the move can only be approved if it improves the overall best solution in terms cost and number of

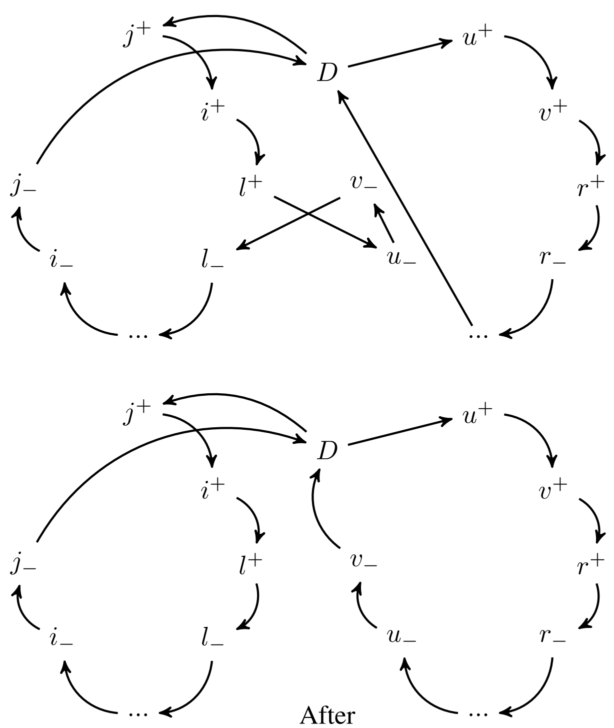

Fig. 6. DARP 1X-opt.

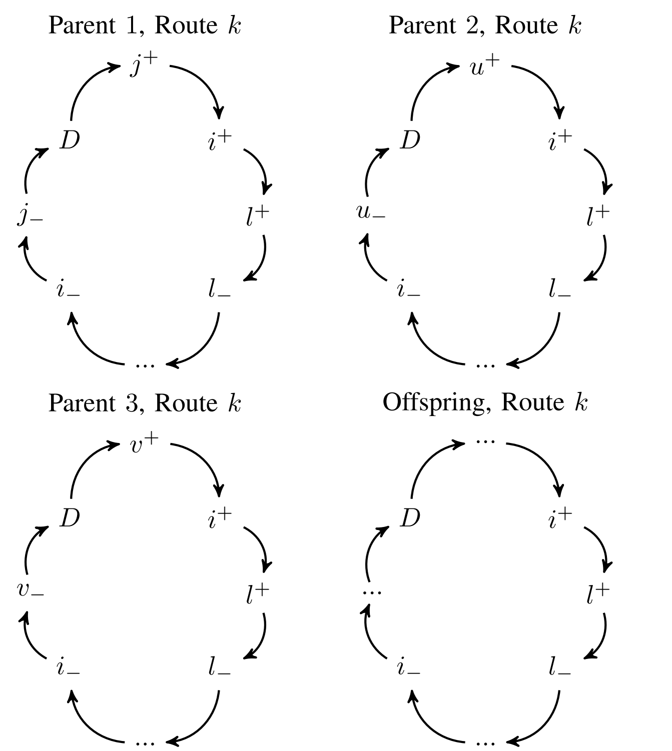

Fig. 7. Three parents crossover.

DARP infeasibilities. Otherwise, the move is rejected. While this principle is proper to TS, it also accounts with the genetic components of the routing solutions. During the execution of HG-DARP, the pool of best solutions is also subject to an evolution as it is maintained in a way such that the newest best solution found takes the place of the oldest.

4) Restarts: If their respective conditions are met, four different restart strategies are applied. The first restart procedure (step (4)) is a regular VNS shaking phase which produces routing solutions possibly altering the current solution total cost, the number of DARP infeasibilities and their types. The shaking phase returns only routing solutions satisfying all the problem conditions except the so called DARP infeasibilities. The second restart procedure is a reset procedure (step (5)). It resets the current solution to one the previously obtained best solutions. The third restart procedure is a ruin-and-recreate phase (step (6)). It partially destroys a given route regardless of the number of DARP infeasibilities. The fourth and final restart procedure is the genetic crossover (step (7)). It produces solutions with the same genes shared by one, two or three drivers. At every restart, the DARP infeasibilities number and types may change. This information is later used during the descent phase to guide the DARP local search operators, as previously detailed. The shaking, reset and ruin-and-recreate phases share the same details as those of HGVNS. Therefore, we here only illustrate how the crossover between three different drivers operates. Figure 7 shows how the stops i^+, l^+, i_- and l_- are served by the same route k in parents 1, 2 and 3. Therefore, the generated offspring's route k shares this same gene with its parents.

V. Computational Results

This section presents our experimental results on 20 DARPTW benchmark instances proposed by Cordeau and Laporte [7]. These instances involve from $n = 24$ up to $n = 144$ customers. There are $2n$ stops in each instance. The first n stops are the pick-ups and the last n stops are the corresponding drops, by respective index order. The maximum transit time of each customer is of 90 time units and the service time at each stop is of 10 time units. The maximum route duration of each vehicle is of 480 time units. The maximum capacity of each vehicle is of 6 passengers. The computational results in Table I were obtained using a C++ implementation under MS Windows on workstations with 3.3GHz Intel Core i5 vPro processors, and 3.2 GB RAM. The column "Instance" indicates the name of the instance. The column "n" indicates the number of customers. The column "m" indicates the number of vehicles used by the state of the art heuristics for the DARPTW proposed by Parragh and Schmid [23] and Braekers *et al.* [6]. The column "m^*" indicates the number of vehicles used by our proposed HG-DARP heuristic. The column "$\Delta_t(m^*)$" indicates the average execution time, with m^*, in seconds. The column "$\Delta_f(m^*)$" indicates the average total route duration obtained with m^* over 10 randomly seeded runs and 2500 HG-DARP iterations. The column "$f^*(m^*)$" indicates the best route duration with m^*. The column "$f^*(m)$" indicates the best route duration with m. The column "$f(m)$" indicates the route duration of the best known solution with travel time minimization. The columns "$g(m)$" and "$g(m^*)$" indicate the gaps $\frac{f^*(m)-f(m)}{f(m)}$ and $\frac{f^*(m^*)-f(m)}{f(m)}$ in percentage.

Instance	n	m	m^*	$\Delta_t(m^*)$	$\Delta_f(m^*)$	$f^*(m^*)$	$f^*(m)$	$f(m)$	$g(m)$	$g(m^*)$
pr01	24	3	2	5.2	751.3	726.9	703.3	881.2	-20.2	-17.5
pr02	48	5	4	42.5	1666.8	1621.1	1564.2	1987.3	-21.3	-18.4
pr03	72	7	6	97.5	2360.6	2213.8	2309.9	2579.3	-10.4	-14.2
pr04	96	9	8	131.5	3015.7	2916.3	2970.0	3303.1	-10.1	-11.7
pr05	120	11	10	145.7	3652.3	3555.9	3564.6	3872.7	-7.9	-8.2
pr06	144	12	12	183.2	4430.6	4359.57	4303.5	4681.3	-8.1	-6.9
pr07	36	4	3	22.6	1169.4	1117.0	1133.3	1452.1	-21.9	-23.1
pr08	72	6	6	94	2214.7	2188.7	2188.7	2250.2	-2.7	-2.7
pr09	108	8	8	183.8	3355.4	3248.6	3248.6	3031.9	7.2	7.2
pr10	144	10	10	291.7	4673.4	4556.7	4556.7	4470.4	2.0	2.0
pr11	24	3	2	5.2	701.9	675.1	691.2	965.1	-28.4	-30.1
pr12	48	5	3	109.2	1419.2	1339.0	1350.5	1551.8	-13.0	-13.7
pr13	72	6	6	136.9	2165.9	2138.2	2138.2	2199.7	-2.8	-2.8
pr14	96	9	8	140.8	2744.4	2696.7	2820.3	2850.3	-1.1	-5.4
pr15	120	10	9	124.9	3448.8	3374.1	3493.4	3709.0	-5.8	-9.0
pr16	144	11	10	140	4131.9	4046.1	4100.1	4281.7	-4.2	-5.5
pr17	36	4	3	42.1	1123.4	1010.1	1062.1	1097.2	-3.2	-7.9
pr18	72	6	5	170.5	2140.5	2111.1	2144.7	2598.3	-17.5	-18.8
pr19	108	8	7	182.7	3140.8	3060.7	3157.0	3284.2	-3.8	-6.8
pr20	144	10	10	224.3	4280.8	4248.1	4248.1	3948.5	7.6	7.6

TABLE I
DURATION MINIMIZATION RESULTS

It is interesting to notice that HG-DARP finds routes with a lower number of vehicles and less total duration than the state of the art heuristics. For example, for instance pr02, HG-DARP lowers the total duration by more than 21% when 5 vehicles are used and by more than 18 % when 4 vehicles are used. To the best of our knowledge, no results were reported with 4 vehicles before. For instance pr12, HG-DARP finds solutions with 3 vehicles instead of 5. HG-DARP is able to reduce the number of vehicles for all instances except five of them: pr08, pr09, pr10, pr13 and pr20. Our findings are justified by the varierty of restart strategies used within HG-DARP enhanced by the use of the backward slack time procedure [2] to minimize waiting times and delay departure times from the depot. The classification of DARP infeasibilities into fours different types and the use data of driven local search operators adapted to the problem have also certainly contributed to this. HG-DARP encountered some difficulties with instances pr09, pr10 and pr20 as it was not able to reach lower durations than the state of the art heuristics. These instances may have some particularities that reduce the efficiency of HG-DARP local search operators.

$k=1$	$f_1 = 295.6$							92.5981	
stop	x	y	wait	arrival	departure	transit	$[l,u]$	t_{next}	q
D	-1.044	2			185.36		[0,1440]	7.018	0
20	-4.094	8.321		192.383	202.383		[175,202]	2.479	1
8	-6.5	7.723		204.862	214.862		[0,1440]	8.919	2
7	0.524	2.226		223.781	233.781		[0,1440]	2.219	3
31	-1.678	1.954		236	246	2.219	[202,236]	0.949	2
32	-1.156	1.161		246.95	256.95	32.088	[225,252]	5.558	1
1	-2.973	6.414		262.51	272.51		[0,1440]	5.571	2
25	-5.476	1.437		278.079	288.079	5.571	[258,287]	4.292	1
44	-1.192	1.175		292.371	302.371	89.988	[0,1440]	4.895	0
12	-4.261	-2.639		307.266	317.266		[0,1440]	0.746	1
24	-3.53	-2.49	2.988	321	331		[321,346]	8.119	2
48	4.288	-0.297		339.12	349.12	8.12	[0,1440]	1.103	1
6	4.891	0.627		350.223	360.223		[0,1440]	8.849	2
4	-1.317	6.934		369.073	379.073		[0,1440]	4.195	3
36	-2.64	2.953		383.268	393.268	66.002	[381,397]	3.442	2
15	-5.204	0.657		396.71	406.71		[395,421]	2.702	3
18	-6.512	3.021		409.411	419.411		[409,426]	4.929	4
28	-2.275	5.541		424.341	434.341	45.268	[416,460]	6.119	3
30	-3.856	-0.37		440.46	450.46	80.237	[432,458]	1.687	2
39	-2.283	-0.981		452.147	462.147	45.437	[0,1440]	3.786	1
42	1.188	-2.493		465.933	475.933	46.522	[0,1440]	5.017	0
D	-1.044	2		480.95			[0,1440]		
$k=2$	$f_2 = 431.3$							144.482	
D	-1.044	2			83.130		[0,1440]	2.246	0
9	-0.417	-0.157		85.377	95.377		[0,1440]	9.248	1
17	-9.194	2.759		104.625	114.625		[86,114]	8.374	2
33	-4.655	9.797		123	133	27.623	[102,123]	4.771	1
14	-2.067	5.789		137.771	147.771		[111,152]	2.883	2
41	-0.785	3.207		150.654	160.654	36.029	[0,1440]	3.176	1
22	2.377	2.908		163.83	173.83		[147,177]	2.285	2
11	2.548	0.629		176.115	186.115		[0,1440]	2.421	3
35	0.129	0.735		188.537	198.537	2.422	[178,215]	6.023	2
38	-5.066	-2.313		204.56	214.56	56.789	[0,1440]	7.091	1
46	1.227	-5.581		221.651	231.651	47.821	[0,1440]	7.284	0
3	5.164	0.547		238.934	248.934		[0,1440]	1.923	1
27	5.74	2.382		250.858	260.858	1.924	[209,252]	3.646	0
10	2.303	1.164		264.504	274.504		[0,1440]	0.718	1
34	1.623	0.932		275.223	285.223	0.719	[260,276]	4.705	0
2	-3.066	0.546		289.927	299.927		[0,1440]	7.281	1
5	-6.741	6.832		307.209	317.209		[0,1440]	3.237	2
13	-7.667	9.934	4.554	325	335		[325,358]	3.283	3
29	-5.662	7.334		338.283	348.283	21.074	[305,349]	4.063	2
26	-4.933	3.337		352.346	362.346	52.419	[329,361]	5.684	1
37	0.435	1.469		368.03	378.03	33.03	[0,1440]	5.828	0
16	-4.138	5.082	2.142	386	396		[386,401]	7.553	1
40	-7.11	-1.862		403.553	413.553	7.553	[0,1440]	3.644	0
21	-3.776	-3.333		417.197	427.197		[416,453]	8.119	1
45	2.984	1.163		435.316	445.316	8.119	[0,1440]	8.583	0
19	1.86	9.672	0.101	454	464		[454,470]	9.806	1
23	-4.303	2.045		473.806	483.806		[471,499]	4.237	2
47	-3.793	-2.161		488.043	498.043	4.237	[0,1440]	1.912	1
43	-1.893	-2.373		499.954	509.954	35.954	[0,1440]	4.455	0
D	-1.044	2		514.409			[0,1440]		
Tot.	$f = 726.9$		wait $= 9.785$		transit $= 757.163$			237.08	
Avg.	$\Delta_f = 363.45$		$\Delta_{\text{wait}} = 0.204$		$\Delta_{\text{transit}} = 15.774$				

TABLE II
DETAILED BEST SOLUTION FOR PR01

For instance pr01, we report the solution with the least total duration. For each of the 2 vehicles in this solution, Table II provides the route's order of visit, the stop indices, coordinates x and y, waiting, arrival and departure times, the transit time (only at drops), the time windows, the travel times to the next stop, and finally the current vehicle used capacity q after serving each stop. One can verify that all arrival times are within their respective time windows, transit times never exceed 90 time units, vehicles used capacities never exceed 6 passengers and route durations does not exceed 480 time units. Our solution's total waiting is of 9.785 time units with an average of only 0.204 time units per stop, which is much lower than the average waiting of 4.40 time units per stop provided by the best known solution with travel time minimization, which uses 3 vehicles [7]. The total transit time is of 757.163 time units with an average of only 15.774 time units per customer. The best known solution with travel time minimization for this instance totals 1094.99 of transit time units, with an average of 45.62 time units per customer. From the drivers and the customers point of view, one can argue that the best solution with total duration minimization provides more satisfaction than the solution with total travel time minimization. Indeed, drivers spend less time at each stop and customers spend less time to reach their destinations. Finally, the least total travel time found is of 194.63 time units (2 vehicles) for pr01. While the best known solution with travel time minimization yields 190.02 time units (3 vehicles) [7].

VI. CONCLUSION

This paper proposed a data driven evolutionary heuristic for the dial-a-ride problem with Time Windows (DARPTW).

Four types of DARP infeasibilities were categorized and used by customized local search operators within a hybrid genetic heuristic (HG-DARP). The heuristic uses an evolutionary pool of best solutions, as well as a set of crossover operators, in addition to shaking and reset restart strategies, to minimize the total duration of all routes. The computational results on a set of benchmark instances show that the number of vehicles required to serve all customers is reduced for most of the benchmark instances tested. Moreover, these results show that drivers and customers would have keen interest in the solutions provided by HG-DARP.

References

[1] Attanasio A, Cordeau JF, Ghiani G, Laporte G. Parallel tabu search heuristics for the dynamic multi-vehicle dial-a-ride problem. Parallel Computing 2004; 30(3):377–387.

[2] Belhaiza S, Hansen P, Laporte G. Hybrid variable neighborhood tabu Search heuristic for the vehicle routing problem with multiple time windows. Computers & Operations Research 2014; 52:269-281.

[3] Belhaiza S, M'Hallah R. A Pareto non-dominated solution approach for the vehicle routing Problem with multiple time windows. Proceedings of the 2016 IEEE Congress on Evolutionary Computation CEC 2016; 3515-3524.

[4] Belhaiza S, M'Hallah R, Ben Brahim G. A new hybrid genetic variable neighborhood search heuristic for the vehicle routing problem with multiple time windows. Proceedings of the 2017 IEEE Congress on Evolutionary Computation CEC 2017; 1319-1326.

[5] Belhaiza S. A Game Theoretic Approach for the Real-Life Multiple-Criterion Vehicle Routing Problem With Multiple Time Windows. IEEE Systems Journal 2017; DOI:10.1109/JSYST.2016.2601058.

[6] Braekers K , Caris A , Janssens GK . Exact and meta-heuristic approach for a general heterogeneous dial-a-ride problem with multiple depots. Transport Research Part B: Methodological 2014;67:16686.

[7] Cordeau JF, Laporte G. A tabu search heuristic for the static multi-vehicle dial-a-ride problem. Transportation Research Part B 2003; 37:579–594.

[8] Cordeau JF, Laporte G. The dial-a-ride Problem: models and algorithms. Annals of Operations Research 2007; 153:29–46.

[9] Chevrier R, Liefooghe A, Jourdan L,, Dhaenens C. Solving a dial-a-ride problem with a hybrid evolutionary multi-objective approach: application to demand responsive transport. Applied Soft Computing 2012; 12:1247–1258.

[10] Glover F. Future paths for integer programming and links to artificial intelligence. Computers & Operations Research 1986; 13:533-549.

[11] Guerreiro F, Pezzella F, Pisacane O, Trollini L. Multi-objective optimization in dial-a-ride public transportation. Transportation Research Procedia 2014; 3:299–308.

[12] Holland JH. Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor, 1975.

[13] Kirchner D, Calvo RW. A granular tabu search algorithm for the dial-a-ride problem. Transportation Research Part B: Methodological 2013; 56:120–135.

[14] Liu M, Luo Z, Lim A. Hybridization of harmony search with variable neighborhood search for restrictive single-machine earliness/tardiness problem. Information Sciences 2013; 226:68–92.

[15] Madsen OBG, Raven HF, Rygaard JM. A branch-and-cut algorithm for a realistic dial-a-ride problem. Transportation Research Part B 2015; 81:267–288.

[16] Masmoudi MA, Braekers K, Masmoudi M, Dammak A. A hybrid genetic algorithm for the heterogeneous dial-a-ride problem. Computers & Operations Research 2017; 81:1–13.

[17] Masson R, Léhudé F, Péton O. The dial-a-ride problem with transfer. Computers & Operations Research 2013; 41:12-23.

[18] Markovic N, Kim M, Schonfeld P. Statistical and machine learning approach for planning dial-a-ride systems. Transportation Research Part A 2016; 89:41-55.

[19] Mladenovic N, Hansen P. Variable neighborhood search. Computers & Operations Research 1997; 24:1097-1100.

[20] Muelas S, LaTorre A, Pena J-M. A distributed VNS algorithm for optimizing dial-a-ride problems in large-scale scenarios. Transportation Research Part C 2015; 54:110–130.

[21] Paquette J, Cordeau J F, Laporte G. Combining multicriteria analysis and tabu search for dial-a-ride problems. Transportation Research B 2013, 52:1-16.

[22] Parragh S, Doerner K, Hartl R. Variable neighborhood search for the dial-a-ride problem. Computers & Operations Research 2010;37:1129–1138.

[23] Parragh S, Schmid V. Hybrid large neighborhood search for the dial-a-ride problem. Computers & Operations Research 2013;40:490–497.

[24] Zidi I, Mesghouni K, Zidi K, Ghedira K. A multi-objective simulated annealing for the multi-criteria dial a ride problem. Engineering Applications of Artificial Intelligence 2012; 25:1121–1131.