Applying NSGA-II to a Multiple Objective Dial a Ride Problem

Pedro M.M. Guerreiro[1][0000−0001−7272−2000], Pedro J.S. Cardoso[1,2][0000−0003−4803−7964], and Hortênsio C. L. Fernandes[3]

[1] ISE, University of the Algarve, Faro, Portugal
`pmguerre@ualg.pt`
[2] LARSyS, University of the Algarve, Faro, Portugal
`pcardoso@ualg.pt`
[3] Yellowfish Travel, Lda
`hortensio@yellowfishtransfers.com`

Abstract. In Dial-a-Ride Problem (DARP) customers request from an operator a transportation service from a pick-up to a drop-off place. Depending on the formulation, the problem can address several constraints, being associated with problems such as door-to-door transportation for elderly/disabled people or occasional private drivers. This paper addresses the latter case where a private drivers company transports passengers in a heterogeneous fleet of saloons, estates, people carriers and minibuses. The problem is formulated as a multiple objective DARP which tries to minimize the total distances, the number of empty seats, and the wage differential between the drivers. To solve the problem a Non-dominated Sorting Genetic Algorithm-II is hybridized with a local search. Results for daily scheduling are shown.

Keywords: Dial a Ride · Non-dominated Sorting Genetic Algorithm-II (NSGA-II) · Multiple Objective Optimization · Private Drivers Scheduling.

1 Introduction

In recent years there has been a significant increase in the number of companies transporting tourists in the Algarve (Portugal), mainly between Faro's Airport and the various accommodation units in the region. Many companies, including Yellowfish Travel, Ltd. (YFT) provide a private transportation service into the region and surrounding areas (e.g., Lisbon, Alentejo and southern Spain). The company has a heterogeneous fleet of almost 100 vehicles, with capacities ranging from 4 to 8 passengers. The service philosophy of the company is based on a close relationship with each of the customers and business partners in order to meet all the requirements and needs in a personalized way, included in provision of services by private chauffeurs. This type of product distinguishes itself from the shuttle buses services as each request is carried out separated from the others.

Customers are currently ordering services through a web platform, by email, or via telephone, with advance appointments that can go from several days to a

few hours before the service. The requests are stored in a database and the daily scheduling of these, by vehicles/drivers, are made at the end of the previous day. In the present, the scheduling is built manually which implies a considerable human effort and a strict dependence on the capacity and availability of one or more operators. In addition, it is a process that has to be carried out by specialized people due to some required knowledge of key variables, such as the location of the accommodation or the time of transport/travel between pick-up and drop-off places, in order to articulate the various services. The set of variables to be taken into account for the distribution of services is, however, much broader, including the vehicles' capacities, vehicles' availabilities (e.g., more generally the availability of drivers and vehicles), traffic restrictions, the pick-up/drop-off time, the number and type of people to be transported, the type and size of the luggage, the number and age of the children, etc.

The aforementioned scheduling consists in the assignment of a set of transport services by drivers/vehicles, which must be fitted in such a way as to avoid large journeys and high waiting times between the end of a service and the beginning of the other, thus optimizing the resources and costs. The usage of "appropriate" vehicles is also an asset, avoiding, for instance, the usage of the vehicles with large capacities to transport a small number of passengers, as smaller vehicles generally provide more comfort and are cheaper to operate. Other factors, not directly linked to transport, are also taken into account, such as the balanced distribution of services by drivers, both for safety reasons (avoiding sharp wear and tear by providing rest to drivers) and for wage equity issues. The two aspects are related since driver earning come from service commissions, which depend on the number of services and drove distances.

Taking into account all of the above, the problem is formulated as a multiple objective DARP which tries to minimize the total distances, the number of empty seats, and the wage differential between the drivers.

DARP [5,22] consider transportation requests, each associated with an origin and a destination, resulting in paired pick-up and drop-off points, but with the constraint that passengers are transported service by service. DARP combines Scheduling and Vehicle Routing Problems (VRP) [17,26,28] as it tries to find the best set of routes that a fleet of vehicles must carry out in order to deliver goods to a certain set of customers from a certain set of depots, using a (possibly) heterogeneous set of vehicles. Variants of the VRP adapt it to fulfill constraints, such as: the Capacitated VRP (CVRP) where capacity constraints are added to the vehicles, the VRP with time windows (VRP-TW) where each delivery must be performed during a certain time interval, the VRP with Pick-Up and Delivering (VRP-PD) where it is admitted that not all services are delivery but also the pick-up of goods is considered, the VRP with Multi-Depot (VRP-MD) in which routes do not start in the same depot, the VRP with Last In First Out (VRP-LIFO) in which the last commodity to be loaded is the first one to be delivered, etc.

Several adaptations and techniques were proposed over the last decades to solve the variants of DARP problem. For example, a branch-and-price-and-cut

algorithm for heterogeneous DARP with configurable vehicle capacity is proposed in [23] to solve a route planning problem arising at a senior activity center. In [4] is introduced a mixed-integer programming formulation of the problem and a branch-and-cut algorithm capable of solving small to medium-size instances of the formulated problem. The pickup and delivery problem with time windows (PDPTW) and a DARP along with some families of inequalities used within branch-and-cut algorithms were formulated in [25], being those inequalities tested on several instance sets of the PDPTW and the DARP. At the level of metaheuristics there is also a wide set of solutions that solve several variants of the problem in question, e.g., Genetic Algorithms (GA), Ant Colony Algorithms (ACO), or Tabu Search (TS) [9]. For example, in [29] an ACO is used to minimize the fleet size required to solve a DARP. In [6] is presented a study to develop and test different GAs in the aim of finding an appropriate encoding and configuration, specifically for the DARP problem with time windows. Parallel implementations of the TS are described and compared in [1] when applied to a static DARP. Other algorithmic solutions include the use of hybrid method such as in [15], where it is implemented a two-stage hybrid meta-heuristic method (uses ACO and TS) for vehicle routing problems with constraints of simultaneous pickups and deliveries and time windows (VRP-SPDTW). In [27] is proposed a parallel approach for solving the VRP-SPD. The parallel algorithm is embedded with a multi-start heuristic which consists of a variable neighborhood descent procedure, with a random neighborhood ordering, integrated in an iterated local search framework. The hybrid algorithm for the dynamic dial-a-ride problem in [2] combines an exact constraint programming algorithm and a TS heuristic. In [3] the authors address the problem of lack of transport service in sparsely inhabited areas as a Demand responsive transport problem (DRTP), comparing the Non-dominated Sorting Genetic Algorithm II (NSGA-II), the Strength Pareto Evolutionary Algorithm 2 (SPEA-2), and the Indicator Based Evolutionary Algorithm (IBEA). Furthermore, improvements using an iterative local search (ILS), added in the mutation operator, are used to select the best approach in a solution capable of producing answers in a short period of time. Another application of NSGA-II to a multiple objective variation of the DARP problem considering disruptive scenarios (e.g., accidents with the transporting vehicles, vehicle breakdown and traffic jams) is presented in [14]. NSGA-II is also used to solve another bi-objective formulation of DARP [10] consisting in the determination of routes to be performed by a fleet of vehicles available to serve a set of geographically dispersed customers (corresponding to patients). Please refer to [5,12] for two surveys on modeling and algorithms for the DARP.

From a modeling point of view, in addition to other constraints, the problem proposed in this paper ends up combining some of the above mentioned variants, since the vehicles have limited capacities (CVRP), involve the collection and delivery of customers (VRP-PD) with a LIFO of size 1 (VRP-LIFO), vehicles do not leave and have a final stop in the same place (VRP-MD), and there are temporary windows for pick-up and drop-off (VRPTW). The problem will

be formulated in Sec. 2 as a Multiple Objective DARP (MO-DARP), adapting the work presented in [19]. The difficulty of obtaining efficient methods, capable of giving the optimal or near-optimal scheduling in a timely manner, will be increased due to the dealing with problems with multiple objectives [8,20]. As stated, the objective function to be minimized in all its components, includes (a) the total distance run by the vehicles during a day work, (b) the total number of empty seats during a day work, and (c) the difference between the commissions earned by each driver and a reference value (promoting drivers working and earnings equity).

This paper shows an approach to solve the real problem proposed by the company. Solutions are computed based on a NSGA-II [8] hybridized with a local search (see Sec. 3). Furthermore, the implementation takes into account the restrictions imposed by the company, optimizing not only the operations cost but also the welfare of its workers. The prototype implemented in Python allows to obtain solution for a day's work in bearable times, provided the service request in proper advance. Solutions will be presented showing that distinct concerns from decision maker (relative to the objectives) are allowed.

The paper is structured as follows. The next section formulates the MO-DARP problem in study. Section 3 outlines the NSGA-II algorithm and how it was applied to solve the addressed MO-DARP problem, including the local search operator description and the system's architecture. Results are reported in Sec. 4 and a conclusion and future work is presented in the last section.

2   Problem formulation

Adapted from [19], the problem is formulated in a directed graph $G = (V, A)$, where V is the set of vertices/nodes, $A \subset V \times V$ is the set of edges/arcs, and an arc (g, h) has associated a distance $\delta_{g,h}$ and a transversing time $t_{g,h}$.

Each request $i \in R = \{1, 2, \ldots, n\}$ is characterized by a pick-up node $p_i^+ \in P^+ = \{1, 2, \ldots, n\}$, a drop-off node $p_i^- \in P^- = \{n + 1, \ldots, 2n\}$, a load $q_i = \sum_{l \in J} q_{i,l}$ where J is the type of passengers set (e.g., baby, child, adult) and $q_{i,l}$ is the number of passenger of type l in request i, and either a pick-up time e_i or a drop-off time d_i. The pick-up vs drop-off time specified in the request is related with its type, which is given by the binary parameter T_i ($T_i = 0$ if request i refers to a pick-up and 1 if it is a drop-off).

Moreover, the set of nodes is defined by $V = P^+ \cup P^- \cup D$, where $D = \{2n + 1, \ldots, 2n+m\}$ is the set of depot points. In the proposed model there are as many depots as drivers (a total of m), since it is considered that a vehicle starts and ends at the driver's home, being $d_k^+, d_k^- \in D$ the start and end nodes of vehicle k, respectively. Each vehicle $k \in K$ of capacity C_k is allocated to one driver, information given by the binary parameter $\omega_{j,k}$, where $j \in M = \{1, 2, \ldots, m\}$. Drivers' working time window $[\alpha_j, \beta_j]$ and maximum workload Max_j are also parameters of the model. Two binary variables

$$x_{g,h}^k = \begin{cases} 1 \text{ if arc } (g, h) \text{ is performed by the vehicle } k \in K \\ 0 \text{ otherwise} \end{cases}$$

and

$$y_{i,j}^k = \begin{cases} 1 \text{ if request } i \text{ immediately precedes request } j \text{ in vehicle } k \in K \\ 0 \text{ otherwise} \end{cases},$$

associate the vehicles to going through an arc and the vehicle's services order, respectively.

Regarding scheduling issues, υ_g^k indicates what time vehicle k starts serving node $g \in V$. In addiction, the model defines $\Delta_i^k = C_k - q_i$ as the number of unused seats on vehicle k while serving request i (Δ_i^k is equal to 0 when i is not served by k), S_j the cumulative salary of the driver j, and μ_S the drivers average salary. In order to calculate S_j the set of services for the vehicle k is needed, A_k.

The goal of the problem is expressed by a multi-objective function which comprises (a) the minimization of total distance made by the vehicles

$$f_1 \equiv \sum_{k \in K} \sum_{g \in V} \sum_{h \in V} \delta_{g,h} x_{g,h}^k \tag{1}$$

(b) the minimization of the drivers' wages difference

$$f_2 \equiv \sum_{j \in M} (S_j - \mu_S)^2, \tag{2}$$

and (c) the minimization of the total number of empty seats while satisfying all requests

$$f_3 \equiv \sum_{i \in R} \sum_{k \in K} \Delta_i^k. \tag{3}$$

It can be observed that f_1 and f_3 are linear functions, but f_2 is non-linear.

The problem constraints can be defined as follows. The first constrains ensure that requests are satisfied by a single vehicle

$$\sum_{k \in K} x_{p_i^+, p_i^-}^k = 1, i \in R$$

and that the vehicle which picks-up the customer is the same that drops him off

$$x_{p_i^-, p_j^+}^k = y_{i,j}^k, k \in K, i, j \in R.$$

The next equations ensure that a vehicle starts and ends its daily service at its depot

$$\begin{cases} \sum_{h \in P^+} x_{d_k^+, h}^k = 1, k \in K, d_k^+ \in D \\ \sum_{g \in P^-} x_{g, d_k^-}^k = 1, k \in K, d_k^- \in D. \end{cases}$$

The vehicles' capacity restrictions for each service are verified by satisfying

$$x_{p_i^+, p_i^-}^k \left(\sum_{l \in J} q_{i,j} + \Delta_i^k \right) = x_{p_i^+, p_i^-}^k C_k, k \in K, i \in R$$

Time specifications ensure that none of the requests of a vehicle overlap in time

$$y_{i,j}^k(v_{p_i^+}^k + t_{p_i^+,p_i^-} + t_{p_i^-,p_j^+}) \leq v_{p_j^+}^k, k \in K, i, j \in R.$$

The next restrictions address if a customer either defines a pick-up time or drop-off time from a certain place:

$$\begin{cases} v_{p_i^+}^k + t_{p_i^+,p_i^-} \leq d_i T_i, k \in K, i \in R \\ v_{p_i^+}^k \geq e_i(1 - T_i), k \in K, i \in R. \end{cases}$$

The last constraints ensure that a driver only works within a previously defined time window

$$\begin{cases} \omega_{j,k}(v_{d_k^-}^k - v_{d_k^+}^k) \leq Max_j, j \in M, k \in K, d_k^-, d_k^+ \in D \\ \omega_{j,k} \min(v_g^k) \geq \alpha_j, k \in K, g \in V, j \in M \\ \omega_{j,k} \max(v_g^k) \leq \beta_j, k \in K, g \in V, j \in M \end{cases}$$

and rev_i the drivers' revenue associated to each request

$$\omega_{j,k}S_j = \sum_{i \in A_k} rev_i, k \in K, j \in M.$$

3 Computational solution

3.1 General NSGA-II

Section 2 formulated MO-DARP, a nonlinear multi-objective optimization problem, derived from the single customer DARP. Non trivial multi-objective problems are characterized by having a conflicting set of objective functions, which implies that no single solution is capable of optimizing all objectives at once. In this sense, the solution of a MO problem is a set of trade-off solutions, called Pareto set [7,18], being its representation called Pareto Front (PF). In the Pareto (or efficiency) order relation, a solution R is said to dominate solution S, $R \prec S$, when R is not worse than S for all objectives and there is at least one on which it is strictly better, i.e., given an objective function with n objectives, $(f_1, f_2, \ldots, f_n)$, where all objectives are to be minimized, then

$$R \prec S \Leftrightarrow \begin{cases} \forall_{i \in \{1,2,\ldots,n\}} : f_i(R) \leq f_i(S) \\ \exists_{i \in \{1,2,\ldots,n\}} : f_i(R) < f_i(S). \end{cases} \tag{4}$$

Using the dominance definition, the Pareto set is defined as $\mathcal{P} = \{S \in \Omega | \nexists_{R \in \Omega} : R \prec S\}$, where Ω is the problem's admissible solutions set. Without preferential information, all solution in the Pareto set are considered equally good, as they cannot be ordered completely, leaving the decision of which solution to use to a decision maker. Furthermore, as the Pareto set can be a continuous or discrete set (with possibly infinite cardinal), many times the goal is to find a representative set of optimal solutions. This can be even further relaxed as, under complex

problems, it might be acceptable a representative set of quasi-optimal solutions. These representative sets are called approximation sets.

To measure the approximation sets quality, several metrics where analyzed in the past [24]. This paper will use the Hypervolume (HV) [30] which uses a reference point to measure the size of the objective space covered by the approximation set. HV considers accuracy, diversity and cardinality, being the only unary metric with this capability.

The method elected to solve the proposed MO-DARP was NSGA-II [8]. As the name suggests, NSGA-II is a multiple objective Genetic Algorithm [13]. GAs evolve a population of candidate solutions throughout a number of generations, until a stopping criteria is reached (e.g, number of generations, number of evaluations of the objective, computational time, etc). In each generation, the candidate solutions are subjected to two genetic operators, namely: crossover and mutation. The crossover operator creates the offspring by combining randomly selected (according to some given crossover probability, p_c) candidate solutions, and the mutation operator applies a mutation to each offspring, again according to a given mutation probability, p_m. While the first operator retains the parents characteristics to the offspring, the second preserve and maintain the diversity of the population, preventing it from becoming trapped in some local minima. After applying the crossover and mutation operators, parents and offspring are joined and every solution is evaluated to determine their respective fitness. That fitness is used to select the best solutions to maintain for the next generation, discarding the others.

A few differences distinguish the NSGA-II procedure from the generic GA, such as the selection of the candidate solutions which are preserved for the next generation. Since in a non-dominated set of solutions of a MO optimization problem there is no "best" solution, but a set of trade-off solutions (approximation set), NSGA-II algorithm ranks each solution according to their respective layer in the approximation set, and uses the crowding distance to maintain the diversity, so that when needed, the solutions discarded are those with lower rank or in a more "crowded" space (refer to [8] for a detailed explanation of the process). Here, given an approximation set $\mathcal{AS}$, the referred solution's layer corresponds to a construction where the first layer is given by

$$\mathcal{L}_0 = \{S \in \mathcal{AS} | \nexists_{R \in \mathcal{AS}} : R \prec S\} \tag{5}$$

and the next layers are recursively given by

$$\mathcal{L}_{i+1} = \{S \in \mathcal{AS}_i | \nexists_{R \in \mathcal{AS}_i} : R \prec S\}, \tag{6}$$

where $\mathcal{AS}_i = \mathcal{AS} - \cup_{k=0}^{i}\mathcal{L}_k$ for $i = 0, 1, \ldots$.

3.2   Applying NSGA-II to MO-DARP

Returning to the MO-DARP, the solutions of the problem can be seen as assignments of drivers/vehicles to services. In this sense, to implement the NSGA-II method, vectors of integers were used to represent solutions, where each column

represents a service, and the value is the vehicle assigned to it, as the two parents sketched in Fig. 1 – top-left. For instance, Parent 1 represents a solution where service 1 is assigned to vehicle 3, service 2 is assigned to vehicle 2, etc.

The algorithm can be summarized as follows. (i) The initial population is generated by randomly attributing vehicles to services. (ii) Then the fitness of each individual in the population is computed, using Eqs. (1) – (3) and the number of violated restrictions, implementing a restriction-violation non-dominated sorting. In other words, solution which do not violate any restriction are ranked/sorted as in Eqs. (5) and (6). The ranking process is then repeated with the solutions which violate one restriction, then with two restrictions, etc. This means that the process is allowed to work with solutions which are not valid (e.g., a vehicle has two services which overlap in time). (iii) In the next step, pair of parents are selected proportionally to their rank (called fitness proportionate selection or roulette wheel selection) and the crossover operator is applied with probability, p_c. Figure 1 – top represents a crossover operation where Parent 1 and 2 are combined to generate Offspring 1 and 2. The represented combination is done by randomly selecting three cutting points followed by swap between the parents of the blocks defined by those cutting points. The number of cutting point, κ, is a parameter of the algorithm. (iv) Mutation is then randomly applied to each offspring vehicle/service assignment with probability p_m. In our case, the operator was implemented by randomly changing the vehicle assigned to the service, as the example sketched in Fig 1 – bottom, where Offspring 1 is mutated by changing the vehicles assigned to services S3 and S7, obtaining Offspring 1^*, which replaces Offspring 1. (v) The use of local search operators to improve the offspring is optional, but generally it allows to improve the algorithm performance in terms of solutions fitness for a given time/number of iterations/number of objective function evaluations (see Sec. 3.3 for the considered local search). Finally, (vi) the offspring are evaluated and from the set of parents and offspring, taking into consideration their fitness and the solutions crowding distance, a new population is selected, with the original size, returning to step (iii) if the stopping criteria is not met. When the stopping criteria is met, the feasible non-dominated solutions in the last population are returned as the proposed solution for the MO-DARP. An empty set is returned if no feasible solution was found.

3.3   Local Search

In order to improve overall performance, a local search (LS) operator was implemented and applied to the offspring resulting from the NSGA-II's crossover and mutation operators as follows. Let us consider an offspring, such as the ones presented in Fig. 1. This offspring represents a solution by assigning services to vehicles. For each service, the local search operator starts by checking if there are vehicles satisfying the service's constraints (e.g., number of seats, luggage restrictions, etc.) – to speed up this step, a pre-computation associates admissible vehicles to the services. Secondly, if the set of admissible vehicles (except the assigned one) is not empty, for each of them, it is verified if assigning the service to the vehicles violates any of the time restrictions, namely if overlapping of

	S1	S2	S3	S4	S5	S6	S7	S8
Parent 1	3	2	1	1	4	4	3	2
Parent 2	2	1	4	2	3	1	4	3

Crossover $\Rightarrow$

	S1	S2	S3	S4	S5	S6	S7	S8
Offspring 1	3	2	4	2	4	4	3	3
Offspring 2	2	1	1	1	3	1	4	2

	S1	S2	S3	S4	S5	S6	S7	S8
Offspring 1	3	2	4	2	4	4	3	3

Mutation $\Rightarrow$

	S1	S2	S3	S4	S5	S6	S7	S8
Offspring 1*	3	2	2	2	4	4	1	3

Fig. 1. Solution representation using vectors of integers. Example of the crossover (top) and mutation (bottom) operators.

services occurs, if the vehicles' working hours are satisfied, if it is possible to go from the previous drop-off to the service's pick-up point in time and if after the service is still possible to reach the next pick-up location in time. If the previous conditions are all satisfied then it is computed the variation in the total distance that the change of the service from one vehicle to the other would induce. If the total distance would be improved in any of the admissible changes then the one with the greatest gain is executed.

3.4   System's architecture

The proposed architecture comprises 4 main components: database, routes server, the MO-DARP optimizer, and visualization. The (relational) database stores the data associated with the problem, such as: services (pick-up/drop-off places, pick-up/drop-off times, etc.), vehicles (availability, number of places, etc.), drivers, etc. To compute the routes between the places, an instance of the Open Source Routing Machine (OSRM) [16] was deployed in a local server. OSRM runs on OpenStreetMap data [21] and, among other things, furnishing an Web application programming interface (HTTP API) which replies to requests for nearest street matches (snaps coordinates to the street network and returns the nearest matches), route computation (finds the fastest route between coordinates), duration or distances tables (computes the duration or distances of the fastest route between all pairs of supplied coordinates), etc. The MO-DARP optimizer was prototyped over the Platypus (Multiobjective Optimization in Python) evolutionary computing framework [11]. Platypus provides optimization algorithms and analysis tools, allow to define constrained and unconstrained problems, and currently supports around ten MO algorithms.

4   Computational results

This section presents some results of the usage of the proposed algorithms. Two operation days were chosen: Day 1 with 54 services and 22 cars available, and Day 2 with 197 services and 38 vehicles. In both days it is considered that vehicles can do pick-ups between 00:00 and 24:00. It should be noticed that there might be more than one driver associated to a vehicle, i.e., it is not usual that the same driver does pick-ups early and late in the day – for the chosen days that data was not yet systematized in the database. The algorithm was run 10 times for each set of parameters and the results are summarized in Tab. 1, which

Table 1. Average of Hypervolume indicator and cardinality of the Pareto Front, for 10 executions of each set of parameters, ordered by the values of Day 2's Hypervolume.

κ	p_c	p_m	LS	Hypervolume Day 1	Hypervolume Day 2	# PF Day 1	# PF Day 2
4	0.9	0.01	✓	0.80	0.81	100	100
2	0.9	0.01	✓	0.80	0.80	100	100
4	0.7	0.01	✓	0.77	0.76	100	100
2	0.7	0.01	✓	0.77	0.73	100	100
4	0.9	0.01	–	0.81	0.66	100	100
2	0.9	0.01	–	0.80	0.65	100	100
4	0.7	0.01	–	0.80	0.65	100	100
2	0.7	0.01	–	0.80	0.64	100	100
2	0.9	0.05	✓	0.80	0.37	99.8	70.5
2	0.7	0.05	✓	0.80	0.36	100	76.9
4	0.9	0.05	✓	0.80	0.36	100	63.5
4	0.7	0.05	✓	0.79	0.35	100	59.5
2	0.9	0.05	–	0.76	0.30	99.9	71.7
2	0.7	0.05	–	0.75	0.29	100	81.2
4	0.9	0.05	–	0.75	0.29	99.6	64.4
4	0.7	0.05	–	0.75	0.27	100	79.2

presents for each set of parameters the Hypervolume values of the normalized (to the interval [0, 1]) objective function values and cardinality of the Pareto Front. Regarding the Hypervolume, in the results for Day 2 it can be observed that the most influential parameter is the mutation probability, and the lower its value the better are the results obtained. What this means is that as the algorithm is reaching good solutions, the mutation in reality is "destroying" those good solutions by randomly moving services to other vehicles – but remember that the purpose of the mutation is to prevent solutions from being trapped in some local minima, and its probability should not be null. As for the rest of the operators, not surprisingly, the local search allows for better results when compared with the same set of parameters – but cannot overwrite the effects of a badly chosen mutation – and an higher crossover probability also gives systematically better results than a lower one. The only parameter that is not "coherent" is κ, the number of points for the crossover. If for a lower mutation probability ($p_m = 0.01$), a 4-points crossover always gives better results, when this probability is higher ($p_m = 0.05$), using the classic 2-points crossover yields better results, when compared with using an higher crossover probability, as can be seen in the lower half of the table. The results for Day 1 are a little different, but it can be observed that all the values are very similar, indicating that this problem is "simple" to solve, and the values of the parameters do not influence much the results obtained. Similar results can be observed with respect to the cardinality of the PF, seeable from the fact that the maximum number of solution in the PF (equal to the defined population size) is always reached when $p_m = 0.01$, establishing again mutation as the most influential parameter.

In Fig. 2 are depicted the Pareto fronts for the best parameters for Day 2 ($\kappa = 4, p_c = 0.9$ and $p_m = 0.01$), with the local search (left) and without the local search (right). It is observable that, without the local search (right), the values for f_1 (distance) are centered around $14000 Km$, but using the local search
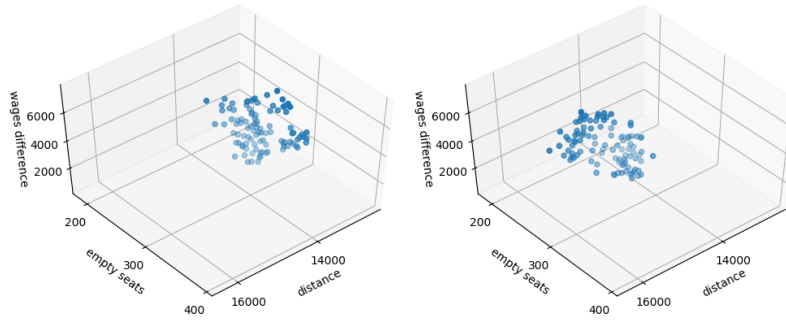
Fig. 2. Pareto fronts for $\kappa = 4$, $p_c = 0.9$ and $p_m = 0.01$, with (left column) and without (right column) local search operator.

(left), all solutions found are well below that value. Regarding f_2 (the number of empty seats) and f_3 (the difference of the commissions) the difference between using or not the local search is negligible. This is not a surprise, as the local search is about finding better solutions considering only the first objective (f_1).

Figure 3 plots two Gantt charts for the schedulings of Day 2, obtained from a single run of the proposed algorithm using as parameter $\kappa = 4$, $p_c = 0.9$, $p_m = 0.01$, and applying the local search. Each line of the plot is associated to a vehicle and the blue bars depicts connection times, i.e., the time to go from the drop-off place of one service to the pick-up place of the next, from the vehicles' starting place to the first pick-up place, or from the last vehicles' drop-off to its ending place. On the other hand, the red bars depict services times, i.e., the time each service takes from the pick-up to the drop-off place. The absence of blue bar between services (red bars) means that the drop-off place of the previous service coincides with the pick-up place of the next one. The scheduling represented in the top of the figure was chosen from the approximation set for being the one which minimizes the number of empty seats: the objective function values are $f_1 = 13782.37$ (kms), $f_2 = 190$ (empty seats) and $f_3 = 2762, 41$ (wage equity). On the other hand, the figure in the bottom corresponds to scheduling which minimizes the total distance: the objective function values are $f_1 = 12516.58$, $f_2 = 267$, and $f_3 = 2212.65$. Comparing plots it is observable that the second one has a smaller density of blue bars and, although possibly not so discernible, is less "dense". These observation follow the expected behavior since to diminish total distance vehicles should be allowed to have longer waiting times avoiding the movement between drop-off and pick-up points.

5 Conclusion and future work

The number of companies transporting tourists in the Algarve and in many touristic regions has recently increased dramatically. Between those companies are the ones which provide private transportation services in a heterogeneous fleet of vehicles. These companies have to schedule the daily services taking into consideration constraints (e.g., vehicles and divers availability, vehicles vs.
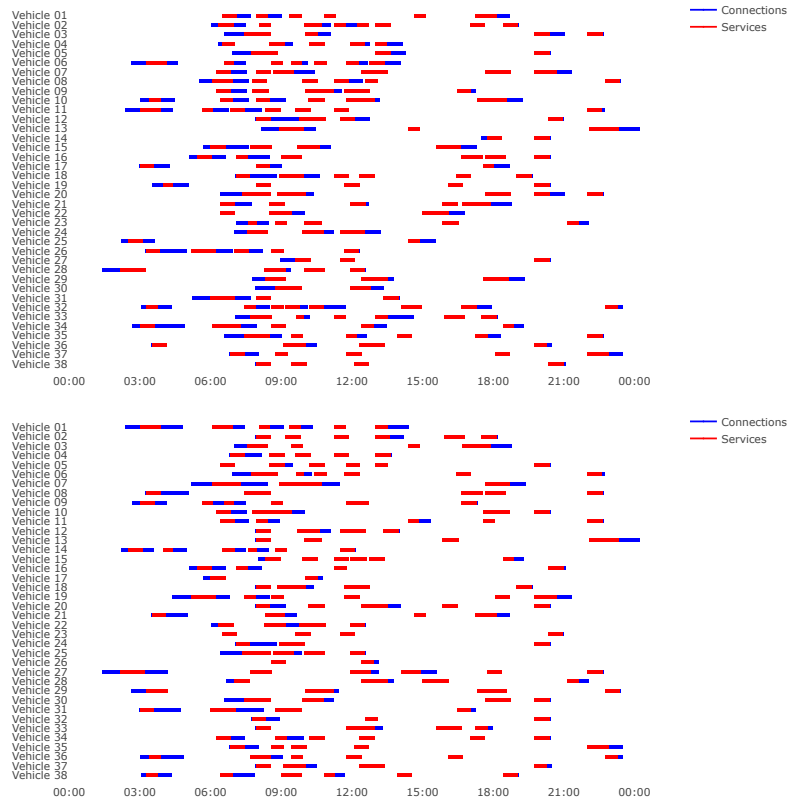
Fig. 3. Two examples of scheduling proposed from a single run of the algorithm using parameter $\kappa = 4$, $p_c = 0.9$, $p_m = 0.01$, and local search. On top the scheduling which minimizes the number of empty seats (objective function values $f_1 = 13782.37$, $f_2 = 190$ and $f_3 = 2762, 41$ and on the bottom the scheduling which minimizes the total distance (objective function values: $f_1 = 12516.58$, $f_2 = 267$, and $f_3 = 2212.65$).

services characteristics, etc) and objectives. Among these objectives are the operational costs, the appropriate usage of the vehicles fleet, and welfare of the drivers. The problem can be formulated as a multiple objective DARP.

This paper presents the usage of a NSGA-II with a local search to solve the MO-DARP problem originated by those transportations. The proposed system uses data collected by a company (e.g., services, vehicles, drivers, etc.) and a geographical information system (namely, OSRM) to compute the routes between the pick-up and drop-off points. The returned approximation set gives the company a set of scheduling from which a decision maker can choose one, and do the arrangements for the next day.

In the future, the authors intend to improve the algorithm by including other pre and post operators. The comparison with other heuristics is also an objective, looking for opportunities to further hybridize and consequently improve

the computational process. Another major asset will be the adaptation of the algorithmic solution to allow a dynamic environment where new services, delays, vehicles/drivers unavailabilities, traffic, etc. are taken into consideration.

Acknowledgments

References

1. Attanasio, A., Cordeau, J.F., Ghiani, G., Laporte, G.: Parallel tabu search heuristics for the dynamic multi-vehicle dial-a-ride problem. Parallel Computing **30**(3), 377–387 (2004)
2. Berbeglia, G., Cordeau, J.F., Laporte, G.: A hybrid tabu search and constraint programming algorithm for the dynamic dial-a-ride problem. INFORMS Journal on Computing **24**(3), 343–355 (aug 2012). https://doi.org/10.1287/ijoc.1110.0454
3. 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. Appl. Soft Comput. **12**(4), 1247–1258 (Apr 2012). https://doi.org/10.1016/j.asoc.2011.12.014
4. Cordeau, J.F.: A branch-and-cut algorithm for the dial-a-ride problem. Operations Research **54**(3), 573–586 (2006)
5. Cordeau, J.F., Laporte, G.: The dial-a-ride problem: models and algorithms. Annals of Operations Research **153**(1), 29–46 (2007)
6. Cubillos, C., Rodriguez, N., Crawford, B.: A study on genetic algorithms for the DARP problem. In: Mira, J., Álvarez, J.R. (eds.) Bio-inspired Modeling of Cognitive Tasks. pp. 498–507. Springer Berlin Heidelberg, Berlin, Heidelberg (2007). https://doi.org/10.1007/978-3-540-73053-8_50
7. Deb, K.: Multi-objective Optimization using Evolutionary Algorithms. John Wiley & Sons (2001)
8. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE transactions on evolutionary computation **6**(2), 182–197 (2002)
9. Gendreau, M., Potvin, J.Y.: Handbook of metaheuristics, vol. 2. Springer (2010)
10. Haddadene, S.A., Labadie, N., Prodhon, C.: NSGA-II enhanced with a local search for the vehicle routing problem with time windows and synchronization constraints. IFAC-PapersOnLine **49**(12), 1198–1203 (2016)
11. Hadka, D.: Platypus - Multiobjective Optimization in Python (2015), `https://platypus.readthedocs.io`, accessed in 08-Fev-2019
12. Ho, S.C., Szeto, W., Kuo, Y.H., Leung, J.M., Petering, M., Tou, T.W.: A survey of dial-a-ride problems: Literature review and recent developments. Transportation Research Part B: Methodological **111**, 395 – 421 (2018). https://doi.org//10.1016/j.trb.2018.02.001
13. Holland, J.: Adaptation in natural and artificial systems. University of Michigan Press, Ann Arbor (1975)
14. Issaoui, B., Khelifi, L., Zidi, I., Zidi, K., Ghédira, K.: A contribution to the resolution of stochastic dynamic dial a ride problem with NSGA-II. 13th International Conference on Hybrid Intelligent Systems (HIS 2013) pp. 54–59 (2013)

15. Lai, M., Tong, X.: A metaheuristic method for vehicle routing problem based on improved ant colony optimization and tabu search. Journal of Industrial and Management Optimization **8**(2), 469–484 (apr 2012). https://doi.org/10.3934/jimo.2012.8.469
16. Luxen, D., Vetter, C.: Real-time routing with openstreetmap data. In: Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems. pp. 513–516. GIS '11, ACM, New York, NY, USA (2011). https://doi.org/10.1145/2093973.2094062
17. Männel, D., Bortfeldt, A.: A hybrid algorithm for the vehicle routing problem with pickup and delivery and three-dimensional loading constraints. European Journal of Operational Research **254**(3), 840–858 (2016)
18. Miettinen, K.: Nonlinear Multiobjective Optimization. Kluwer Academic Publishers (1999)
19. Morais, A.C., Torres, L., Dias, T.G., Cardoso, P.J.S., Fernandes, H.: A combined data mining and tabu search approach for single customer dial-a-ride problem. In: 7th International Conference on Metaheuristics and Nature Inspired Computing. pp. 121–123. Marrakech, Morocco (Oct 2018)
20. Murata, T., Itai, R.: Multi-objective vehicle routing problems using two-fold EMO algorithms to enhance solution similarity on non-dominated solutions. In: International Conference on Evolutionary Multi-Criterion Optimization. pp. 885–896. Springer (2005)
21. OpenStreetMap contributors: Planet dump retrieved from https://planet.osm.org. https://www.openstreetmap.org (2017)
22. Parragh, S.N., Doerner, K.F., Hartl, R.F.: A survey on pickup and delivery problems. Journal für Betriebswirtschaft **58**(1), 21–51 (2008)
23. Qu, Y., Bard, J.F.: A branch-and-price-and-cut algorithm for heterogeneous pickup and delivery problems with configurable vehicle capacity. Transportation Science **49**(2), 254–270 (2014)
24. Riquelme, N., Lücken, C.V., Baran, B.: Performance metrics in multi-objective optimization. In: 2015 Latin American Computing Conference (CLEI). pp. 1–11 (Oct 2015). https://doi.org/10.1109/CLEI.2015.7360024
25. Ropke, S., Cordeau, J.F., Laporte, G.: Models and branch-and-cut algorithms for pickup and delivery problems with time windows. Networks: An International Journal **49**(4), 258–272 (2007)
26. Salhi, S., Imran, A., Wassan, N.A.: The multi-depot vehicle routing problem with heterogeneous vehicle fleet: Formulation and a variable neighborhood search implementation. Computers & Operations Research **52**, 315–325 (2014)
27. Subramanian, A., Drummond, L., Bentes, C., Ochi, L., Farias, R.: A parallel heuristic for the vehicle routing problem with simultaneous pickup and delivery. Computers & Operations Research **37**(11), 1899 – 1911 (2010). https://doi.org/10.1016/j.cor.2009.10.011
28. Toth, P., Vigo, D.: Vehicle routing: problems, methods, and applications. SIAM (2014)
29. Tripathy, T., Nagavarapu, S.C., Azizian, K., Pandi, R.R., Dauwels, J.: Solving dial-a-ride problems using multiple ant colony system with fleet size minimisation. In: UK Workshop on Computational Intelligence. pp. 325–336. Springer (2017)
30. Zitzler, E., Thiele, L.: Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. IEEE transactions on Evolutionary Computation **3**(4), 257–271 (1999)