A comparison of multiple objective algorithms in the context of a 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, Universidade do Algarve, Faro, Portugal
pmguerre@ualg.pt
[2] LARSyS, Universidade do Algarve, Faro, Portugal
pcardoso@ualg.pt
[3] Yellowfish Travel, Lda
hortensio@yellowfishtransfers.com

Abstract. In their operations private chauffeur companies have to solve variations of the multiple objective dial a ride problem. The number and type of restrictions make the problem extremely intricate and, when manually done, requires specialized people with a deep knowledge of the *modus operandi* of the company and of the environment in which the procedure takes place. Nevertheless, the scheduling can be automated through mean of computational methods, allowing to deliver solutions faster and, possible, optimized. In this context, this paper compares six algorithms applied to solving a multiple objective dial a ride problem, using data from a company mainly working in the Algarve, Portugal. The achieved results show that ϵ-MOEA overcomes the other algorithms tested, namely the NSGA-II, NSGA-III, ϵ-NSGA-II, SPEA2, and PESA2.

Keywords: DARP · Evolutionary Algorithms · Multiple Objective Optimization · Private Drivers Scheduling

1 Introduction

One of the problems addressed by private chauffeur companies is the design of optimal routes and schedules to pick up, drive, and deliver clients to drop off locations previously scheduled. The optimization has to attend to a large number of parameters and restrictions, e.g.: capacity demands (number of persons, volume of the luggage, large/special sport items etc.) or service restrictions (transportation of children, disabled people, drivers availability, drivers working hours, vehicles availability, etc.). The problem is even more intricate and dynamic when issues like delayed services, absence of workers, traffic congestion,

* This work was supported by the Portuguese Foundation for Science and Technology (FCT), project LARSyS - FCT Plurianual funding 2020-2023. We also thank Yellowfish Travel, Lda for the valuable data, participation in the problem definition and solution development.

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50436-6_28

vehicle breakdowns or service cancellations (just to give some examples) are taken into account.

In its general form, the described problem is a variation of the dial a ride problem (DARP), where clients obligatorily provide the pick up and drop off locations and time, and other parameters such as number of passengers and their ages or luggage volume (e.g., see [6, 26] for formal definition of the problem). DARP consists in, given a characterized fleet of vehicles, designing (optimal) routes and schedules to satisfy clients requests of pick up and deliver in certain time windows, not necessarily using all available vehicles. DARP is known to be NP-hard and has been addressed in many ways including exact and meta-heuristic solutions [14]. For example, to solve a route planning problem at a senior activity center, a heterogeneous DARP with configurable vehicle capacity was solved by Qu and Bard [27], using a branch and price and cut algorithm. A branch and cut algorithm capable of solving small to medium-size instances was developed by Cordeau [5] along with a mixed-integer programming formulation. In [29] a pick up and delivery problem with time windows and a DARP tested a branch and cut algorithms with families of inequalities on several instance sets of the problems. In [31] an ant colony optimization algorithm was applied to minimize the fleet size required to solve a DARP. The study in [9] aimed to develop and test different genetic algorithms to find appropriate encodings and configurations, specifically for the case with time windows. Parallel implementations were also studied as the tabu search variants described and compared in [1] (applied to a static DARP). A two-stage hybrid meta-heuristic method using ant colony optimization and tabu search for the vehicle routing problem with constraints of simultaneous pick up and delivery, and time windows was presented by Lai and Tong [21]. Another parallel approach embedded with a multi-start heuristic for solving the vehicle routing problem with simultaneous pick up and delivery was proposed in [30]. Berbeglia, Cordeau and Laporte presented an hybrid algorithm for the dynamic DARP [2], which combines an exact constraint programming algorithm and a tabu search heuristic.

The DARP was formulated with distinct objectives such as the minimization of the routes' duration, ride time, waiting time, number of vehicles used, operational cost, etc. (for a more complete list see [7]). In the criteria context, the problem has been addressed in several multiple objective (MO) formulations. In the MO case, several possibly antagonistic objectives are evaluated producing a set of solutions over a partially ordered objective space [12]. For instance, Chevrier *et al.* [4] addressed the problem of lack of transport service in sparsely inhabited areas as a demand responsive transport problem, comparing the non-dominated sorting genetic algorithm II (NSGA-II), the strength pareto evolutionary algorithm 2 (SPEA2), and the indicator based evolutionary algorithm (IBEA). In the same work, 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,

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50436-6_28

vehicle breakdown, and traffic jams) was presented by Issaoui *et al.* [19]. A bi-objective formulation of DARP is solved using NSGA-II in [16]. The formulation consists in the determination of routes to be performed by a fleet of vehicles available to serve geographically dispersed customers. For more in-depth surveys on the subject, please refer to the works of Cordeau and Laporte [6] and Ho *et al.* [18].

This paper compares six multiple objective evolutionary algorithms (MOEA) applied to solving an instance of the multiple objective dial a ride problem (MO-DARP), namely: Multi-Objective Evolutionary Algorithm based on ϵ-dominance (ϵ-MOEA), Non-dominated Sorting Genetic Algorithm II (NSGA-II), NSGA-III, ϵ-dominance Non-dominated Sorting Genetic Algorithm II (ϵ-NSGAII), improved Strength Pareto Evolutionary Algorithm (SPEA2) and Pareto Envelope-based Selection Algorithm 2 (PESA2). Data used to produce the experimental results was retrieved from the working flow of a private chauffeur company with a heterogeneous fleet, which mainly works in the Algarve, Portugal. The objectives were discussed with the company and set to be: (a) the minimization of total distance traveled by the vehicles to serve all requests, (b) the minimization of the drivers' wages difference, and (c) the minimization of the total number of empty seats while satisfying all requests. Results show that the algorithms can return a very large set of valid solution but, for the tested instances and in general, ϵ-MOEA overcomes the other algorithms in indicators such as hypervolume, generational distance, and inverted generational distance.

The added value of this paper is the comparison of the computational performance of the six mentioned state-of-the-art algorithms over a set of DARP instances with different sizes, allowing us to propose ϵ-MOEA as a good starting point (since, e.g., computational time was not a major concern) to solve the private chauffeur company's problem.

The remaining document is structured as follows. The next section deals with the problem's formulation and multiple objective optimization issues. Section 3 briefly describes the algorithmic solutions tested and discusses the problem's codification and chosen operators. The next section, Sec. 4, presents and discusses results obtained with the experimented methods. The last section presents a conclusion and future work.

2 Problem formulation

The MO-DARP addressed in this paper was formulated in [24] and took up again in [15]. The problem is formulated over a directed network where nodes are pick up, delivery or shift start/end locations, and edges have associated to them distance and traversing time. A request is characterized by a pick up node, a drop off node, the number and type of passengers (load), and pick up (or drop off) time. Driver and vehicles are connected as there are as many depots as drivers, since it is considered that a vehicle starts and ends at the driver's homes (start/end location). Each vehicle has a capacity and is allocated to one driver. Drivers' working time window and maximum workload are also parameters of

DOI: 10.1007/978-3-030-50436-6_28

the model. The formulation allows for a vehicle to be shared by more than one driver, given that the driver's time windows do not overlap with each other, and the end point of one shift coincides with the start point of the next shift. In the algorithm, these cases are solved considering dummy-duplicated cars with the same characteristics of the original ones.

Regarding scheduling issues, the model takes into consideration operational costs, the number of unused vehicle seats while serving a request, and the cumulative salary of the driver regarding the drivers' average salary (part of the salary of the drivers depends on the number and type of services). Therefore, for a given set of services, the goal of the problem is expressed by a multiple objective function which comprises (a) the minimization of total distance made by the vehicles to serve all services, (b) the minimization of the drivers' wages difference, and (c) the minimization of the total number of empty seats while satisfying all requests. Being obvious that the first objective minimizes the operational costs, the third one also has the same effect as optimizing the number of seats to the request will reduce the use of larger vehicles, which usually have associated higher costs (e.g., larger consumption) and might be less comfortable for the passengers. The second objective reduces the overload of some of the drivers/vehicles which might represent a security issue, promoting, at the same time, more balanced wages among drivers. Most requests solving issues are more or less common sense as, all requests are satisfied by a single vehicle, the vehicle which picks up the customer is the same that drops him off, a vehicle starts and ends its daily service at its depot, vehicles' capacity restrictions for each service are verified, none of the requests of a vehicle overlap in time, and workers only works within the defined time window.

As these different objectives are conflicting, when a solution is better in one objective is normally worse in other(s), e.g., a solution will not improve/decrease the total distance traveled by the vehicles to serve all services (objective (a)) without increasing at least one of the other two objectives (drivers' wages difference (b) or total number of empty seats while satisfying all requests (c)). This means that the problem will not have one best solution, but rather a collection of trade-off solutions, each better in one objective, but worse in other(s). If no decision formula is known (e.g., priority on the objectives), a human decision maker should be presented with this collection of "best solutions", to decide which one should be used, based on the company's politics.

To determine these "best solutions", the Pareto dominance is used: suppose an objective function $F = (f_1, f_2, \ldots, f_m)$ such that $f_i : \Omega \rightarrow R$ ($i \in \{1, 2, \ldots, m\}$, Ω is the solutions/search space, and R is an ordered set), and, without loss generality, all objectives of our MO problem are to be minimized. The solution $X \in \Omega$ dominates solution $Y \in \Omega$, $X \prec Y$, if the following pair of conditions are verified: $f_i(X) \leq f_i(Y)$ for all $i \in \{1, 2, \ldots, m\}$ and exists $j \in \{1, 2, \ldots, m\}$ such that $f_j(X) < f_j(Y)$. The solution of the MO problem is the set of solutions which are not dominated by any other solution in the search space, i.e., $\mathcal{P} = \{X \in \Omega | \nexists_{Y \in \Omega} : Y \prec X\}$, called Pareto set. The set of images of the solutions in the Pareto set is called Pareto front, $\mathcal{F} = \{y \in R^m | y = F(X), X \in \mathcal{P}\}$.

However, it is not always possible to compute the Pareto set as, for instance, it can be very large or there are no "efficient" algorithms to compute it. In this case, decision makers might be satisfied when a good approximation set is returned. To measure the quality of approximation set, many indicators can be found in literature [3, 28, 33]. Desirable features of those indicators include convergence (toward the Pareto front), spread (extent of the front, i.e., distance between the front's extreme solutions), distribution (evenness of the front, i.e., uniformity of the objective distances between pairs of "adjacent" solutions), and Pareto compliance (the ranking established by the indicator does not contradict Pareto optimality, i.e., a front ranked better by one metric must be Pareto preferred over the ones which are ranked worst).

In this work, we will compare the results of the algorithms using the hypervolume (HV), the generational distance (GD) and the inverted generational distance (IGD) indicators. The (i) hypervolume, HV, uses a reference point to measure the size of the objective space covered by the approximation front [35]. HV considers accuracy, diversity and cardinality, being the only known unary metric with this capability. In our case, the hypervolume ratio is computed has the ratio between the hypervolume of the approximation front and the hypervolume of a reference front (i.e., best know approximation set, computed as the non-dominated elements of all known solutions) will be used. The (ii) generational distance, GD, of an approximation front A is computed as the sum of the distance between each solution in A to the closest objective vector in a reference front, averaged over the size of A. The reference front can be the Pareto front or the best known approximation of it. GD only estimates the convergence of the approximation set toward the reference front, and the results can be biased toward approximation sets of poor quality having large cardinality. The (iii) inverted generational distance, IGD, which compared with GD, reverses the order of the fronts considered as input, i.e., IGD is computed as the sum of the distances between the elements of the reference front to the closest element the approximation front, averaged over the size of the reference front. This makes IGD less sensitive to the size of the approximation set and intuitively measures more effectively convergence, spread, and distribution. It should be mentioned that, while the hypervolume is Pareto compliant, GD and IGD are not.

3 Algorithmic solutions

This section presents the adopted solution encoding, general genetic algorithms (GA) flow and briefly introduces the six algorithms used to solve the MO-DARP problem presented in Sec. 2.

3.1 Solution encoding and general GA

The solution of the MO-DARP problem corresponds to the assignment of drivers/vehicles to services. E.g., considering a set of services (S_1, S_2, S_3, S_4), ordered

DOI: 10.1007/978-3-030-50436-6_28

for instance by pick up time, a solution could be something like $(1, 2, 1, 3)$ meaning that services S_1 and S_3 are to be served by driver/vehicle 1, S_2 is to be served by driver/vehicle 2, and service S_4 for by driver 3.

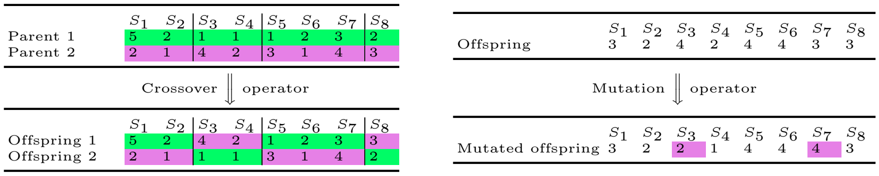
Given this representation using vectors of integers, the basic implementation of the based-GA is quite straightforward and can be summarized as follows [14]: (i) initialize the population by randomly attributing drivers/vehicles to services. (ii) Use the objective function values and number of violated restrictions to compute the fitness of each individual. (iii) Select parents supported on their rank and apply the crossover operator with probability, p_c. Figure 1 – left exemplifies a k-point crossover operation [32] where Parent 1 and Parent 2 are combined to generate Offspring 1 and Offspring 2. In the example, the operator randomly chooses $\kappa = 3$ (a parameter of the algorithm) cutting points which is followed by a swap between the blocks defined by those cutting points. (iv) In the next step, offspring suffer a mutation with probability p_m. In our case, the mutation operator randomly changes the vehicle/driver assigned to a service, e.g., see Fig. 1 – right. (v) As an optional step, not applied in our experiments, we can use a local search operator and/or solution correction. For instance, the local search can apply "neighbor" operator to improve offspring, generally improving the convergence in terms of fitness and time/number of iterations/number of objective function evaluations. Solution correction can also improve the algorithm performance by applying operations to ensure that solutions going to the next generation are feasible. (vi) Finally, the offspring are evaluated for their fitness (and possibly crowding distance, distance to reference points, etc.) and between parents and offspring, a new population, with the same size of the original one, is returned. This population moves to the next generation, returning to step (iii), or if the stopping criteria are met, the feasible non-dominated solutions in the last population are returned as the proposed solution.

As seen, GA evolve a population of solutions through a number of generations, using two main operators: crossover and mutation. In short, the first operator passes the parents lineament to the offspring and the second implements diversity on the population, preventing it from becoming trapped in some local minima and avoiding the stagnation of the population. Given a population and its offspring, in a greedy manner, the best individuals will be part of the next generation, maintaining a fixed population size (N). If everything goes well, the new population has moved towards the optimal solution, while maintaining its diversity.

One crucial difference between a generic (single objective) GA and a multiple objective GA, is the selection of the individuals that move to the next generation, as in a multiple objective problem, generally there is not a single best solution for the problem and therefore the selection of elements moving to the next generation is done using the Pareto dominance.

Another thing worth mention is that, as the representation used allows for invalid solutions (e.g., a vehicle has two services that overlap in time), these are treated in the algorithm by implementing a restriction-violation policy, meaning that each solution will have assigned to it the number of restrictions it violates,

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50436-6_28



	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8
Parent 1	5	2	1	1	1	2	3	2
Parent 2	2	1	4	2	3	1	4	3

Crossover ⇓ operator

	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8
Offspring 1	5	2	4	2	1	2	3	3
Offspring 2	2	1	1	1	3	1	4	2

	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8
Offspring	3	2	4	2	4	4	3	3

Mutation ⇓ operator

	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8
Mutated offspring	3	2	2	1	4	4	4	3

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

and solutions that do not violate any restriction will always be considered "better" than those that violate any restriction, this means that the former will always be moved to the next generation (regardless of their fitness values) prior to any of the latter, and if any of these last will survive, the ones with less violations are preferred.

3.2 Algorithms overview

NSGA-II As the name suggests, Non-dominated Sorting Genetic Algorithm II, NSGA-II, proposed by Deb *et. al.* [11], is a genetic algorithm specially adapted for the optimization of multiple objective problems. NSGA-II uses Pareto dominance to rank solutions to their respective "layer" in the approximation set, i.e., non-dominated solutions are in rank 1, solutions dominated by first rank solutions are in rank 2, solutions dominated by solutions in rank 1 and 2 are placed in rank 3, and so on. The individuals that pass to the next generation are consecutively selected from the lowest rank layers. When not all individuals in a given rank can pass to the next generation, NSGA-II uses the notion of crowding distance to maintain diversity, so that the solutions discarded are those in a more crowded space.

NSGA-III Another algorithm is NSGA-III [10], characterized by the authors as "*a reference-point-based many-objective evolutionary algorithm following NSGA-II framework that emphasizes population members that are non-dominated, yet close to a set of supplied reference points*". In this algorithm, the maintenance of diversity among population is aided by supplying and adaptively updating a number of well-spread reference points (r), replacing the crowding distance operator. Equal to NSGA-II, "lowest" layers' elements are added while not surpassing the population's size. When adding a layer that surpasses the population size then some elements are selected taking into consideration a set of reference points which can either be predefined in a structured manner or supplied (preferentially) by the user. Worth mentioning that the population size is not a parameter, but is determined automatically using the number of reference points.

PESA2 Proposed by Corne *et al.* [8] the Pareto Envelope-based Selection Algorithm 2, PESA2, uses a selection technique in which the unit of selection are

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50436-6_28

hyper-boxes located in the objective space. The selective fitness is made over the hyper-boxes populated by at least one individual of the current approximation set. The selected individual is then randomly chosen from the elected hyper-boxes. The method was proven to return solutions which are better spread along the Pareto frontier than individual based selection tested methods.

SPEA2 Proposed by Zitzler, Laumanns, and Thiele [34], the improved Strength Pareto Evolutionary Algorithm, SPEA2, starts by initializing the population with a set of candidate solutions. Then, the best solutions are stored in an archive, insulated from the population. Non-dominated individuals are featured by combining dominance count and dominance rank methods. In other words, for each individual the number of dominated individuals and the number of dominating individuals in the population allows to compute a fitness value. In each iteration, non-dominated individuals from the union of the archive and the current population are updated, maintaining archive's size, either by including dominated individual from the current pool or removing individual from the archive, basing the decision on the nearest neighbor Euclidean distance. The mating pool used to generate the next population is filled by the individuals of the archive and the offspring are then generated by a set of variation operators.

ϵ-MOEA Deb, Mohan and Mishra [13] proposed a steady-state Multi-Objective Evolutionary Algorithm, ϵ-MOEA, based on the ϵ-dominance, concept introduced in [22]. This algorithm uses a two co-evolving populations: an EA population, and an archive population. The former starts with an random population, and the later gets the non-dominated solutions from the former. In each generation, two solutions, one from each population, are mated to create offspring, which are then checked for inclusion in both population and archive, using the ϵ-dominance concept. Worth mention that, while the population has a fixed number of individuals, meaning that an individual must be removed in order to incorporated an offspring in the population, the archive can grow as large as needed, to incorporate all non-dominated solution. A detailed explanation can be found in [13].

ϵ-NSGA-II Kollat and Reed [20] have extended the NSGA-II algorithm, by adding ϵ-dominance [22], archiving and adaptive population sizing. The ϵ-dominance allows the user to specify the precision with which they want to quantify each objective. The ϵ-NSGA-II uses a series of "connected runs", and, as the search progresses, the population size is adapted based on the number of ϵ-non-dominated solutions found and stored in the archive, which are then used to direct the search using an injection scheme, where 25% of the subsequent population are composed by ϵ-non-dominated solutions from the archive, and the other 75% will be generated randomly.

As seen, the last two algorithms use the ϵ-dominance concept to maintain the diversity of the population. Basically, this means that the search space is divided in a number of grids (or hyper-boxes) of ϵ-size, and in each grid (or hyper-box)

DOI: 10.1007/978-3-030-50436-6_28

Algorithms	κ	p_c	p_m	ϵ	r	N
NSGA-II	4	0.9	0.01	–	–	100
NSGA-III	4	0.9	0.01	–	12	–
SPEA2	4	0.9	0.01	–	–	100
PESA2	4	0.9	0.01	–	–	100
ϵ-NSGA-II	4	0.9	0.01	0.001	–	100
ϵ-MOEA	4	0.9	0.01	0.001	–	100

Table 1. Tuned algorithms' parameters.

only one solution in allowed (see [22] for a detailed explanation), guaranteeing this way the inexistence of any area for the solutions to converge.

4 Experimental results

Starting by a brief characterization of the private chauffeur company's testing data, this section compares the algorithms presented in Sec. 3. So, data used in this section is part of the dataset of the services requested between January 2012 and December 2019 to a private chauffeur company of Algarve. Data is characterized by a seasonal variation where the summer months, in particular July, August, and September, have the largest number of services. The number of services per hour also varies during a tipical day. As a curiosity, the peak number of services in a single hour was reached in a Saturday of September 2019 with 70 services requested between 8 and 9 a.m. On the other other hand, the peak number of services for a single day was also reached in September 2019 with 552 requests.

The code was developed is Python, using the Platypus framework for evolutionary computing, which includes optimization algorithms and analysis tools for MO optimization [17]. To compute the distance between the different service locations, the Open Source Routing Machine (OSRM) [23] was used, which is a C++ routing engine for shortest paths in road networks, supported on Open Street Maps cartography [25]. Implementations were run on an Intel i7-5820K at 3.3Ghz, 32GB RAM, running Microsoft Windows 10. To simplify the presentation, algorithms' parameters were previously tuned varying their values over large predetermined sets, being the final values summarized in Tab. 1. The stopping criteria was set as 50,000 evaluations of the objective function and the algorithms were run 10 times for each instance. Furthermore, given the large number of days/problem instances available and in order to represent the seasonality of this type of service, 4 scenarios with different degrees of complexity were selected, namely, by order of scenario, with 40, 99, 300 and 526 requests, and 14, 25, 52 and 70 vehicles, respectively. As a note, the vehicle set used in each scenario was limited to the vehicles used by the company on the selected day.

Table 2 shows the achieved mean values (standard deviation values inside parenthesis) for HV, GD, and IGD indicators. In bold are the best achieved in-

DOI: 10.1007/978-3-030-50436-6_28

Indicator	Scenario	NSGA-II	NSGA-III	SPEA2	PESA2	ϵ-NSGA-II	ϵ-MOEA
HV	1	0.896 (.02)	0.898 (.02)	0.907 (.01)	0.905 (.01)	0.918 (.02)	**0.934** (.02)
	2	0.819 (.03)	0.859 (.02)	0.831 (.02)	0.826 (.04)	0.798 (.02)	**0.881** (.03)
	3	0.654 (.07)	0.738 (.03)	0.626 (.04)	0.699 (.06)	0.540 (.07)	**0.784** (.06)
	4	0.593 (.06)	**0.757** (.05)	0.605 (.06)	0.695 (.07)	0.478 (.05)	0.755 (.08)
GD	1	0.006 (.00)	0.004 (.00)	0.004 (.00)	0.005 (.00)	0.002 (.00)	**0.001** (.00)
	2	0.012 (.00)	0.006 (.00)	0.004 (.00)	0.008 (.00)	0.003 (.00)	**0.002** (.00)
	3	0.022 (.01)	0.014 (.00)	0.010 (.00)	0.015 (.00)	0.014 (.00)	**0.004** (.00)
	4	0.031 (.01)	0.017 (.00)	0.018 (.01)	0.020 (.01)	0.028 (.00)	**0.010** (.00)
IGD	1	0.078 (.01)	0.070 (.01)	0.066 (.01)	0.067 (.00)	0.051 (.01)	**0.040** (.01)
	2	0.122 (.02)	0.078 (.01)	0.087 (.01)	0.098 (.02)	0.114 (.02)	**0.057** (.01)
	3	0.236 (.05)	0.146 (.02)	0.199 (.03)	0.167 (.04)	0.272 (.06)	**0.106** (.02)
	4	0.350 (.08)	0.178 (.04)	0.272 (.05)	0.213 (.05)	0.407 (.05)	**0.160** (.05)

Table 2. Hypervolume, generational distance, and inverted generational distance mean (standard deviation) values for the 4 scenarios.

dicator's values. Required to calculate the indicators' values, the reference front for each scenario was computed as the best know approximation front, i.e., it was computed has the non-dominated elements of all known solutions in the objective space. From the table's analysis and over the introduced arrangement, it is deductible that ϵ-MOEA achieves the best results. Only on Scenario 4, NSGA-II had best HV but with a difference of 0.002 units. This results allows us to believe that ϵ-MOEA has better convergence, spread, and distribution when compared with the others, again over this set of instances. We should also point out that in general, algorithms achieved a population of non-dominated solutions in every case, i.e., all solutions/population elements belong to their approximation set. This fact allowed us to avoid taking measures relative to feasibility of the solution (e.g., implementing post optimization/feasibility operators), maintaining the purity of the algorithms.

To better illustrate the difference in the solutions, only extreme solutions (those with the minimum values in each objective) will be examined next. This does not mean that a decision maker will select on of those. Figure 2 shows three typical scheduling solutions for Scenario 1 obtained with ϵ-MOEA, where each line represents a vehicle's scheduling (red bars for service time and blue bars the connection time, i.e., moving without passengers). The plots shows the extreme solutions for each objective, namely: (top) minimal total distance, (middle) minimal difference in drivers' wages, and (bottom) minimal number of empty seats. Argumentably, although expectable, it is possible to assert that the top solution (with a total of 2745 Km) has less transit without passengers (blue bars) than the others (with 3088 Km and 2900 Km, respectively). Easy to observe is the fact that, the bottom solution has 3 vehicles without any service, as these were vehicle with more seats and were not scheduled. This last solution shows that is conceivable for the company to diminish the number of large vehicles, at least in the low season.

As an example of a larger instance, Fig. 3 presents the scheduling for Scenario 4 considering the minimal distance (extreme) solution, obtained with ϵ-MOEA. The presented solution accounts for 38987 Km and 924 empty seats. If the num-

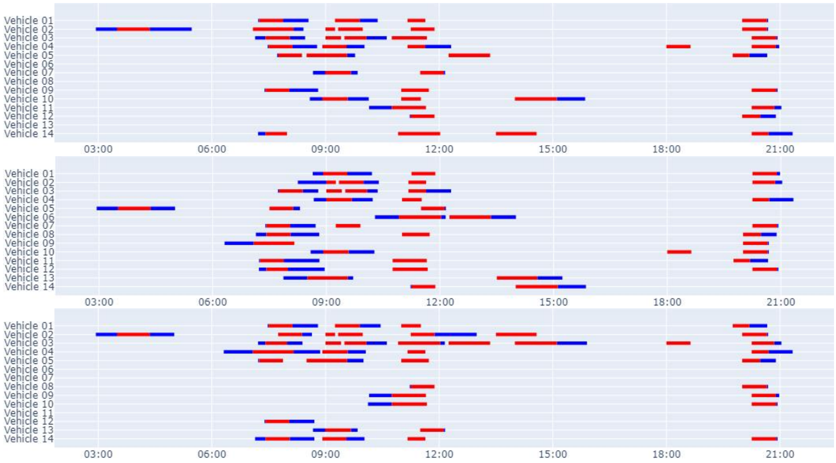


Fig. 2. Examples of scheduling obtained with ϵ-MOEA for Scenario 1, considering the objective-extreme: (top) total distance, (middle) drivers' wages difference, and (bottom) number of empty seats.

ber of empty seats was to be minimized (solution is not presented), the number of empty seats extreme solution accounted for 41660 Km and 844 empty seats. As conclusion, these solutions require a decision maker which, in accordance with the company policy, must decide, for instance, if saving 2673 Km is better or worse than using larger vehicles (with, probably, higher fuel consumption). Or simply choose a more "balanced" solution between the ones returned by the elected algorithm.

5 Conclusion

This paper studies algorithmic solutions for the scheduling operation of a private chauffeur company. The problem was previously formalized as a multiple objective dial a ride problem, with three objectives: (a) minimization of total distance traveled by the vehicles to serve all requests, (b) minimization of the drivers' wages difference, and (c) minimization of the total number of empty seats while satisfying all requests. Six algorithms (ϵ-MOEA, NSGA-II, NSGA-III, ϵ-NSGA-II, SPEA2, and PESA2) were compared using hypervolume, generational distance, and inverted generational distance indicators over 4 different size instances of the problem. Over this arrangement, ϵ-MOEA could be considered the best solution as it presented the best indicator values, except for one indicator in one scenario.

As future work an analysis on the usage of this and other algorithms online (or quasi-online) is due. Another not yet studied issue is the problem's dynamic, where new requests or vehicles are inserted/removed during or after optimization process finished.

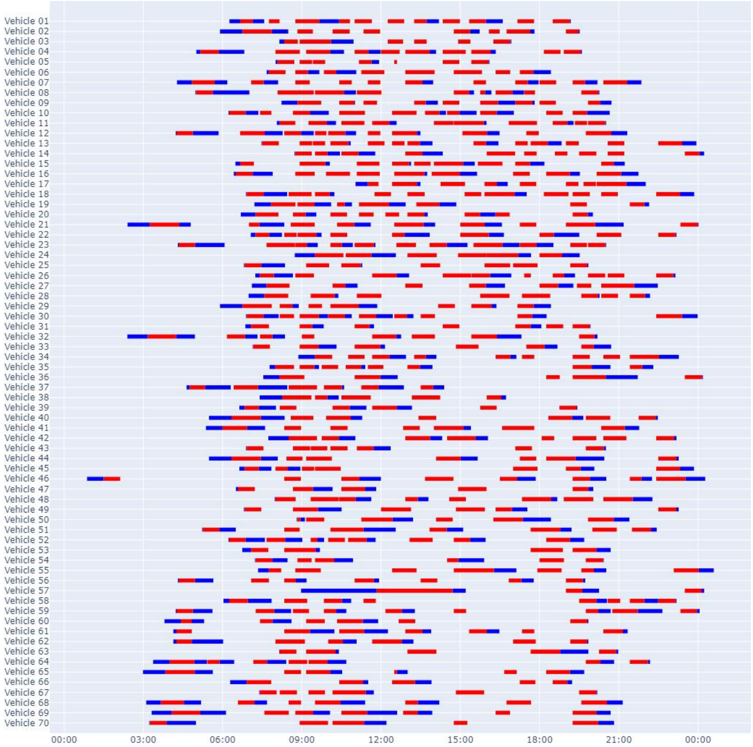


Fig. 3. Examples of scheduling obtained with ϵ-MOEA for Scenario 4, considering the objective-extreme total distance

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). https://doi.org/10.1016/j.parco.2003.12.001
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. Bezerra, L.C.T., López-Ibáñez, M., Stützle, T.: An empirical assessment of the properties of inverted generational distance on multi- and many-objective optimization. In: Lecture Notes in Computer Science, pp. 31–45. Springer International Publishing (2017). https://doi.org/10.1007/978-3-319-54157-0_3
4. 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
5. Cordeau, J.F.: A branch-and-cut algorithm for the dial-a-ride problem. Operations Research **54**(3), 573–586 (2006). https://doi.org/10.1287/opre.1060.0283

6. Cordeau, J.F., Laporte, G.: The dial-a-ride problem: models and algorithms. Annals of Operations Research **153**(1), 29–46 (2007). https://doi.org/10.1007/s10479-007-0170-8
7. Cordeau, J.F., Laporte, G.: The dial-a-ride problem (DARP): Variants, modeling issues and algorithms. Quarterly Journal of the Belgian, French and Italian Operations Research Societies **1**(2) (jun 2003). https://doi.org/10.1007/s10288-002-0009-8
8. Corne, D.W., Jerram, N.R., Knowles, J.D., Oates, M.J., J, M.: PESA-II: Region-based selection in evolutionary multiobjective optimization. In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2001. pp. 283–290. Morgan Kaufmann Publishers (2001)
9. 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
10. Deb, K., Jain, H.: An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, Part I: Solving problems with box constraints. IEEE Transactions on Evolutionary Computation **18**(4), 577–601 (Aug 2014). https://doi.org/10.1109/TEVC.2013.2281535
11. 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 (April 2002). https://doi.org/10.1109/4235.996017
12. Deb, K.: Multi-objective Optimization using Evolutionary Algorithms. John Wiley & Sons (2001)
13. Deb, K., Mohan, M., Mishra, S.: A fast multi-objective evolutionary algorithm for finding well-spread pareto-optimal solutions. Tech. rep. (2003), http://www.iitk.ac.in/kangal/papers/k2003003.pdf
14. Gendreau, M., Potvin, J.Y.: Handbook of metaheuristics, vol. 2. Springer (2010)
15. Guerreiro, P.M.M., Cardoso, P.J.S., Fernandes, H.C.L.: Applying NSGA-II to a multiple objective dial a ride problem. In: Rodrigues, J.M.F., Cardoso, P.J.S., Monteiro, J., Lam, R., Krzhizhanovskaya, V.V., Lees, M.H., Dongarra, J.J., Sloot, P.M. (eds.) Computational Science – ICCS 2019. pp. 55–69. Springer International Publishing, Cham (2019). https://doi.org/10.1007/978-3-030-22750-0_5
16. 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 Papers On Line **49**(12), 1198–1203 (2016). https://doi.org/10.1016/j.ifacol.2016.07.671
17. Hadka, D.: Platypus - Multiobjective Optimization in Python (2015), https://platypus.readthedocs.io, accessed in 08-Fev-2019
18. 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
19. 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)
20. Kollat, J.B., Reed, P.M.: The value of online adaptive search: A performance comparison of NSGAII, ϵ–NSGAII and ϵ–MOEA. In: Lecture Notes in Computer Science. vol. 3410, pp. 386–398. Springer, Berlin, Heidelberg (2005). https://doi.org/10.1007/978-3-540-31880-4_27

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50436-6_28

21. 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
22. Laumanns, M., Thiele, L., Deb, K., Zitzler, E.: Combining Convergence and Diversity in Evolutionary Multiobjective Optimization. Evolutionary Computation **10**(3), 263–282 (sep 2002). https://doi.org/10.1162/106365602760234108, http://www.mitpressjournals.org/doi/10.1162/106365602760234108
23. 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
24. 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)
25. OpenStreetMap contributors: Planet dump retrieved from https://planet.osm.org. https://www.openstreetmap.org (2017)
26. Parragh, S.N., Doerner, K.F., Hartl, R.F.: A survey on pickup and delivery problems. Journal für Betriebswirtschaft **58**(1), 21–51 (2008). https://doi.org/10.1007/s11301-008-0033-7
27. 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). https://doi.org/10.1287/trsc.2014.0524
28. 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
29. 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). https://doi.org/10.1002/net.20177
30. 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
31. 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). https://doi.org/10.1007/978-3-319-66939-7_28
32. Umbarkar, A., Sheth, P.: Crossover operators in genetic algorithms: A review. ICTACT Journal on Soft Computing **06**(01), 1083–1092 (oct 2015). https://doi.org/10.21917/ijsc.2015.0150
33. Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C.M., da Fonseca, V.G.: Performance assessment of multiobjective optimizers: an analysis and review. IEEE Transactions on Evolutionary Computation **7**(2), 117–132 (April 2003). https://doi.org/10.1109/TEVC.2003.810758
34. Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: Improving the strength pareto evolutionary algorithm (2001). https://doi.org/10.3929/ETHZ-A-004284029, http://hdl.handle.net/20.500.11850/145755
35. 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). https://doi.org/10.1109/4235.797969