Application of a genetic algorithm for solving the Dial-a-Ride problem

Stjepan Zelić[1], Marko Đurasević[2], Domagoj Jakobović[2], and Lucija Planinić[2] *

[1] Hypefy World, Zagreb, Croatia
[2] University of Zagreb Faculty of Electrical Engineering and Computing, Zagreb, Croatia
zellich93@gmail.com, marko.durasevic@fer.hr, domagoj.jakobovic@fer.hr, lucija.planinic@fer.hr

Abstract. The dial-a-ride problem (DARP) deals with the transportation of people from source to destination locations. One of the most common use cases is in the transportation of elderly or sick people, and as such it represents an important problem to consider. Previous studies demonstrated that various metaheuristic methods are suitable for solving this kind of problems. Therefore, in this study a GA is proposed and adapted for solving DARP. The obtained results show that the proposed algorithm can achieve better results than similar methods in previous studies. Additionally, the results demonstrate the results can be improved by considering some constraints as soft constraints and including them in the cost function to give the algorithm more flexibility.

Keywords: Genetic algorithm · Dial a ride problem · Optimisation.

1 Introduction

The Dial-a-ride problem (DARP) is a specific type of the Vehicle Routing Problem (VRP), which instead of transporting goods deals with the transportation of people. In DARP, users file requests to be picked up from a certain location at a specific time and to be delivered to another location until a specific time. The goal of the problem is to schedule a fleet of vehicles in a way that the user requirements are satisfied as much as possible, but also that the route duration is as small as possible. DARP has many practical applications in the real world, which include door-to-door transportation of elderly or disabled people [3], taxi services [11], rescue services [11], and demand-responsive transit [9].

DARP has already received a significant attention in the literature. One of the first studies dealing with DARP, in which a sequential insertion heuristic is proposed, was done by Jaw et al. [6]. The problem was also tackled in [8] with the use of simulated annealing. Tabu search (TS) was applied in [2] on a problem where the route durations need to be minimised by accommodating all

* This work has been supported in part by Croatian Science Foundation under the project IP-2019-04-4333.

user requests. A genetic algorithm (GA) for DARP was proposed in [7] which solves the problem from [2]. The major difference between these works is that several strict constraints are modelled as cost functions which are optimised, which gives the GA some flexibility. Another GA was used in [4], in which the authors test different algorithm configurations. An overview of different DARP models and solution methods is given in [3]. An extension of DARP which allows users to change vehicles during their trip is solved by an adaptive large neighbourhood search algorithm in [10]. A hyperheuristic approach to solving DARP is proposed in [12]. This method finds the best heuristic strategy of applying simple operators, which can be applied for new problems. An online version of DARP was considered in [9], in which the optimisation routine runs continuously during the system execution. A parallel extension of the TS method for DARP was proposed in [11]. A variant of the problem which examines different travel modes is examined in [5]. In [1] the authors consider a flexible DARP variant in which only a part of user requests are fixed up front.

The overview above shows that this problem is still widely researched, and many new DARP variants are being proposed and tackled. In this paper we consider the DARP variant which was described in [2] and [7]. The problem is solved using an adapted GA which includes some domain specific information in its evolutionary process. The goal of this research is to gain initial insights which will be used in subsequent studies to further improve the results and solve different DARP variants. The rest of the paper is organised as follows: Section 2 gives an introduction of DARP. The GA adapted for DARP is described in Section 3. The experimental setup and the results obtained by the proposed GA are described in Section 4. Finally, the conclusion of the paper and future research directions are outlined in Section 5.

2 Dial-a-ride problem

The considered DARP is modelled based on the problem defined in [2, 7]. In this problem there are n customer requests for transportation, given as a list with $2n$ locations. Each request has a pickup location, denoted with item i in the list, and delivery location (item $n + i$). The locations are modelled as a fully connected graph in which for all locations i and j a travel distance d_{ij} is defined. For each location a time window $[b_i, e_i]$ specifies the service of the request at that location, be it for pickup or delivery. Ideally, the service at locations should occur only withing those time windows. A service time s_i, required at each location, is also defined. Each customer request has a specified number of places which the user takes up in the vehicle, which are taken up at the pickup location and freed up at the delivery location. The customers also specify a maximum time they wish to spend in the vehicle. To satisfy the user requests, a fleet of m vehicles are available. Each vehicle k starts at the depot location D and returns to it after it completed all requests. Each vehicle has a constant capacity of C and a maximum route duration (both are same for all vehicles).

Usually several objectives are considered in DARP, out of which a single cost function is defined as a weighted linear combination of the individual cost functions. The considered cost functions are:

- f_1 - total route duration - the total duration of the routes for all vehicles
- f_2 - total ride time - the total time that the customers spent riding in the vehicles
- f_3 - total wait time - the time that the vehicles spent idle while waiting to service a request
- f_4 - total late time - the total time that the vehicle was late, meaning that it arrived at a location after its defined time window
- f_5 - total amount of time ride time violation - the excess amount of time that the customer spend riding in the vehicle above their requested ride time
- f_6 - total maximum route violation - the excess amount of time that the cars spent driving over their given maximum route duration

In [2] only functions $f_1 - f_3$ were minimised, whereas the remaining functions were not used since they were modelled as hard constraints (i.e. no lateness was allowed). However, in [7] the authors modelled some constraints as cost functions, which allowed the authors to obtain better results. The total cost function which is minimised is defined as $f = w_1 \cdot f_1 + w_2 \cdot f_2 + w_3 \cdot f_3 + w_4 \cdot f_4 w_5 \cdot f_5 + w_6 \cdot f_6$. The weights $w_1 \ldots w_6$ can be freely selected in order to determine the importance of the individual cost functions. The magnitudes of functions f_1 and f_2 are usually similar, whereas f_3 is usually smaller by one order of magnitude. However, the weights were set as $w_1 = w_2 = w_3 = 1$, since preliminary experiments demonstrated that in such a setting nevertheless focused quite well on f_3. Cost functions f_4 and f_5 usually had a magnitude around 5 times smaller than f_1 and f_2, whereas cost function f_6 was usually equal to 0. Therefore, their weights were set as $w_4 = w_5 = w_6 = 5$, to put equal focus focus on the cost functions which model stricter requirements.

3 Genetic algorithm for DARP

To find solutions for the considered DARP problem, a GA is adapted for the problem. The solutions are represented using two chromosomes, an integer and a permutation chromosome. The integer chromosome denotes to which vehicle each user request is associated. The permutation chromosome represents the order in which the customer requests will be handled. Figure 1 represents an example for a problem with 2 vehicles and 5 requests. Since there are 5 requests this means that the values 1-5 in the permutation vector represent pickup requests, whereas values 6-10 represent delivery requests. In this example, vehicle 0 will first handle the pickup of request 4 and then immediately its delivery (request 9). Then the vehicle will handle the second request. On the other hand, vehicle 1 will first handle three pickup request and then perform their delivery.

Instead of generating the initial population completely randomly, a simple heuristic initialisation was used for the construction of initial solutions. First,

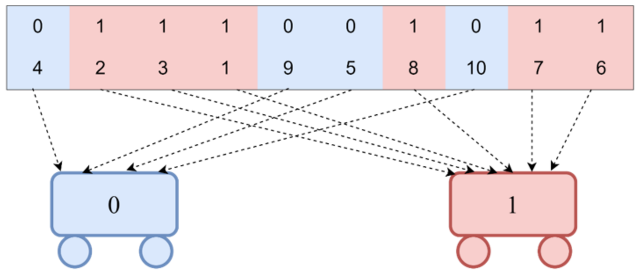

Fig. 1. Solution representation used by the GA

a permutation of the requests is randomly generated, by ensuring that each pickup request appears before its corresponding delivery request in the solution. After that, the vehicles are appointed to customer requests in a way that a vehicle is chosen which can arrive the closest to the middle of its time window. For example, for customer j this value would be calculated as $m_j = \frac{e_j - b_j}{2}$. The customer would then be associated with the vehicle which can arrive at the customer as close to m_j as possible. This solution initialisation method has demonstrated to achieve better results in preliminary experiments than when generating the initial population completely by random.

For the crossover operator an adapted PMX crossover is used. Unlike in the PMX crossover, in which two random crossover points are selected, in this variant a number of vehicles are selected and then all the genes associated to those vehicles are copied to the child individual. The remaining genes are then filled in a similar way as it is done in the original PMX crossover by copying over those orders form the second parent which are not yet present in the child individual. The mutation is performed by simply switching the vehicle to which the pickup and delivery requests of a user are allocated to another vehicle.

It is possible that during evolution a certain number of constraints are not satisfied. Therefore, a procedure which checks the validity of solutions and performs correction to them is used after each change in an individual. It first checks whether all delivery requests appear after their respective pickup requests. If this is not true, then the two requests are simply swapped. Secondly, it checks that the the capacity constraint of vehicles is satisfied at all points in time. This is fixed in a way that the point at which the capacity constraint is violated is located in the solution, and then a delivery request is moved before that point in order to free up space in the vehicle. By using these corrections, is is ensured that the algorithm works only with valid solutions throughout the evolution process.

4 Experimental study

4.1 Benchmark setup

The experimental study will be conducted on the dataset which is proposed in [2]. This dataset consists out of 20 problem instances which contain between 24 and 144 customers and between 3 and 13 vehicles. The instances are divided into two groups, instances from R1a to R10a were generated with narrow time windows, whereas instances from R1b to R10b have been generated with wide time windows. For each instance the GA was executed 10 times. The parameters of the GA were fine tuned in preliminary experiments. A population size of 200 individuals, mutation probability of 0.1, the 5-tournament selection for selecting individuals, and stopping criterion of 1500 generations were used.

4.2 Results

The results obtained by the proposed GA are presented in Table 1. The table outlines the three main objectives that were considered in previous studies: route duration, ride time, and waiting time. "Avg." denotes the average of 10 executions obtained for those objectives, whereas "Best" denotes the value for the objective obtained by the solution with the best fitness. In addition, the average values for the late times and ride time violations per customer are also included to outline how much the obtained solutions violate these constraints. The results are directly compared to the results obtained by Cordeau and Laporte [2] and Jorgensen et al. [7]. The results in the tables are denoted with '†' if they are better only than the results from Cordeau and Laporte, with '*' if they are better than those obtained by Jorgensen et al., with '+' if they are better than the results from both studies, and with '-' if they are worse than the results from both studies. It should be noted that the results for some instances are not marked, which is because they were not solved in the previous studies. The last row denotes the aggregated results across those instances that were also used in the previous studies, in order to make the cumulative scores comparable.

The results demonstrate that the proposed algorithm can achieve some improvements over the results obtained in the studies of Cordeau and Laporte and Jorgensen et al. For the waiting time cost, the proposed algorithm always achieved better results than both methods. For the route duration cost the algorithm always achieved better results than the method of Jorgensen et al. Finally, for the ride time cost for several instances better results were obtained than in the study by Cordeau and Laporte, however, the total obtained results for this criterion were worse than in both studies. It should be mentioned that in comparison with the results obtained by Cordeau and Laporte, the route duration and ride time costs obtained by the proposed method are similar (within a 1% margin). However, we obtained a much smaller value for the vehicle wait time, by a factor of 5. This improvement was possible because of the flexibility provided to us by treating some constraints as soft constraints (late time). Jorgensen et al. also treated late times as soft constraints; unfortunately, these values are not provided in the paper and it is not possible to determine to which extent these constraints were not satisfied. However, the proposed GA was able to achieve better results for the route duration and wait time objectives.

Although the problem was solved by treating late times as soft constraints, it can be seen that the late times and ride time violations are not extensive. The late average times are usually not larger than a minute, and the maximum late

Table 1. Overview of the obtained results

Instance	Route Duration Avg.	Route Duration Best	Waiting Time Avg.	Waiting Time Best	Ride Time Avg.	Ride Time Best	Late Time Avg.	Ride Time Violation Avg.
R1a	890*	972*	114+	201+	1138-	694†	0.49	0.07
R2a	1601+	1975+	164+	491+	2190-	1969†	0.72	1.60
R3a	2353+	2387+	117+	93+	3487†	2958†	0.34	0.92
R4a	3252	3598	270	548	4635	4495	0.53	1.20
R5a	3813*	3958*	193+	317+	5885†	4790+	0.98	1.32
R6a	4691	4773	279	323	7228	7133	0.92	2.58
R7a	1273	1354	133	162	1571	1295	1.50	0.19
R8a	2271	2254	46	14	3349	2803	0.89	2.86
R9a	3225*	3305*	64†	118†	5835*	5947*	4.72	2.52
R10a	4422*	4518*	97+	102+	8099*	7796*	5.19	4.26
R1b	788*	766*	31+	4+	984†	667†	0.48	0.06
R2b	1499*	1422*	55+	6+	2108†	1733†	0.42	1.32
R3b	2306	2282	69	34	3370	2555	0.34	0.42
R4b	3001	2941	75	38	4353	3636	0.20	0.82
R5b	3749*	3981*	135+	242+	5618†	5130†	0.58	1.34
R6b	4492*	4456*	149+	139+	6653†	6171+	1.13	1.01
R7b	1150*	1120*	19+	10+	1571†	1358†	0.54	1.12
R8b	2329	2355	100	88	3505	2658	0.96	2.17
R9b	3287*	3337*	45+	90+	5962-	5415†	1.88	2.98
R10b	4388*	4442*	66+	70+	7734*	7084*	3.62	2.00
Total	35659*	36637*	1249+	1882+	57264-	51711†	21.08	20.52

time was 5 minutes for instance R10a. On the other hand, the ride time violation was also usually in the range between one or two minutes. Such small constraint violations should not incur a large user dissatisfaction, but give the algorithm more flexibility in finding better solutions for other criteria.

5 Conclusion

The obtained results demonstrate that with the initial adaptation of the GA it is possible to improve on the results for DARP. For one of the considered criterion the algorithm achieved a significant improvement over the existing results, whereas for the remaining two criteria the results are mostly similar to the other studies. Such performance was achieved by introducing more problem specific elements in the algorithm, but also by allowing some constraints to be unsatisfied. Since this study tackled the problem only briefly with only a few algorithm adaptations, there is still a lot of room to improve the results by further adaptation and fine tuning for the considered problem.

In future studies the goal is to test the proposed method on other data sets that were used in related surveys. Additionally, the intention is to adapt the proposed approach to cover other DARP variants which were not included in this paper. A more thorough study involving different metaheuristic algorithms will be conducted to propose alternative and more efficient methods for DARP. Finally, since several objectives are usually considered in DARP simultaneously, another obvious line of research would be to apply multi-objective algorithms.

References

1. Busing, C., Comis, M., Rauh, F.: The dial-a-ride problem in primary care with flexible scheduling (2021)
2. Cordeau, J.F., Laporte, G.: A tabu search heuristic for the static multi-vehicle dial-a-ride problem. Transportation Research Part B: Methodological **37**(6), 579–594 (2003). https://doi.org/https://doi.org/10.1016/S0191-2615(02)00045-0, https://www.sciencedirect.com/science/article/pii/S0191261502000450
3. Cordeau, J.F., Laporte, G.: The dial-a-ride problem (darp): Models and algorithms. Annals OR **153**, 29–46 (06 2007). https://doi.org/10.1007/s10479-007-0170-8
4. Cubillos, C., Rodriguez, N., Crawford, B.: A study on genetic algorithms for the darp problem. pp. 498–507 (06 2007). https://doi.org/10.1007/978-3-540-73053-8_50
5. Dong, X., Rey, D., Waller, S.T.: Dial-a-ride problem with users' accept/reject decisions based on service utilities. Transportation Research Record **2674**(10), 55–67 (2020). https://doi.org/10.1177/0361198120940307, https://doi.org/10.1177/0361198120940307
6. Jaw, J.J., Odoni, A.R., Psaraftis, H.N., Wilson, N.H.: A heuristic algorithm for the multi-vehicle advance request dial-a-ride problem with time windows. Transportation Research Part B: Methodological **20**(3), 243–257 (1986). https://doi.org/https://doi.org/10.1016/0191-2615(86)90020-2, https://www.sciencedirect.com/science/article/pii/0191261586900202
7. Jorgensen, R., Larsen, J., Bergvinsdottir, K.: Solving the dial-a-ride problem using genetic algorithms. Journal of the Operational Research Society **58** (10 2007). https://doi.org/10.1057/palgrave.jors.2602287
8. JR., J.W.B., KAKIVAYA, G.K.R., STONE, J.R.: Intractability of the dial-a-ride problem and a multiobjective solution using simulated annealing. Engineering Optimization **30**(2), 91–123 (1998). https://doi.org/10.1080/03052159808941240, https://doi.org/10.1080/03052159808941240
9. Lois, A., Ziliaskopoulos, A.: Online algorithm for dynamic dial a ride problem and its metrics. Transportation Research Procedia **24**, 377–384 (2017). https://doi.org/https://doi.org/10.1016/j.trpro.2017.05.097, https://www.sciencedirect.com/science/article/pii/S2352146517303782, 3rd Conference on Sustainable Urban Mobility, 3rd CSUM 2016, 26 – 27 May 2016, Volos, Greece
10. Masson, R., Lehuédé, F., Péton, O.: The dial-a-ride problem with transfers. Computers & Operations Research **41**, 12–23 (2014). https://doi.org/https://doi.org/10.1016/j.cor.2013.07.020, https://www.sciencedirect.com/science/article/pii/S0305054813001998
11. Pandi, R.R., Ho, S.G., Nagavarapu, S.C., Tripathy, T., Dauwels, J.: Gpu-accelerated tabu search algorithm for dial-a-ride problem. In: 2018 21st International Conference on Intelligent Transportation Systems (ITSC). pp. 2519–2524 (2018). https://doi.org/10.1109/ITSC.2018.8569472
12. Urra, E., Cubillos, C., Cabrera-Paniagua, D.: A hyperheuristic for the dial-a-ride problem with time windows **2015**, 1–12 (2015). https://doi.org/10.1155/2015/707056, https://doi.org/10.1155/2015/707056