An Asymmetric Multiple Traveling Salesman Problem with Backhauls to solve a Dial-a-Ride Problem

E. Osaba, E. Onieva, F. Diaz, R. Carballedo, P. Lopez, A. Perallos

Abstract—**Nowadays, public transportation has become an essential area for the actual society, which directly affects the quality of life. There are different sort of public transportation systems. One type that receives much attention these days because of its great social interest is the transportation on-demand. Some of the most well-known on-demand transports systems are the Demand Responsive Transit, and the Dial-a-Ride. In this paper, a real-world Dial-a-Ride problem is presented and modeled as a Multi-Attribute Traveling Salesman Problem. In addition, in this work a benchmark of this problem is presented, and the first resolution of this benchmark is offered. For the resolution of the problem an Adaptive Multi-Crossover Population Algorithm has been implemented.**

Keywords—On demand transportation, Traveling Salesman Problem, Combinatorial optimization, Genetic Algorithm, Meta-heuristic.

I. INTRODUCTION

Nowadays, transportation is a crucial sector for the society. Public transportation, for example, is a resource used by almost the whole population, and it directly affects the quality of life. There are different kinds of public transportation systems, each one with its own features. Nevertheless, they all share some disadvantages, as for example, the capacity of the vehicles, the frequency and schedules of the services, and the geographical area of coverage. These drawbacks lead to a difficulty to satisfy all user demands.

As a result of these inconvenients the concept of Transportation-On-Demand (TOD) arises [1]. TOD is concerned with the transportation of passengers or goods between specific origins and destinations at the request of users. Most TOD problems are characterized by the presence of three often conflicting objectives: maximizing the number of requests served, minimizing operating costs and minimizing user inconvenience.

There are different types of problems within the transportation on demand, as the Demand Responsive Transport, or Demand Responsive Transit (DRT) [2]. It is characterized by flexible routing and scheduling of small/medium vehicles operating in shared-ride mode between pick-up and drop-off locations according to passengers needs. In addition, the routes planed in a DRT may vary in real time. One possible application of a DRT system could be the transport service in rural areas or areas of low passenger demand, where a regular bus service may not be economically viable.

Another sort of on-demand transportation is the Dial-a-Ride Problem (DRP) [3]. The DRP is applicable in contexts where passengers are transported, either in groups or individually, between specified origins and destinations. The most common DRP application arises in door-to-door transportation services for elderly or handicapped people. In this context, users often formulate two requests per day: an outbound request from home to a destination, and an inbound request for the return trip. The social interest of this type of transport is undoubted because, above all, it helps to ensure welfare of people with special needs.

The DRP is the focus of many studies these days [4, 5], and it is currently in operation in several important cities of the world, as London[1], or Riverside (California)[2].

The aim of this paper is to address one DRP problem. For this purpose, the DRP problem has been modelized as a Multi-Attribute Traveling Salesman Problem (MATSP). Today, as can be read in [6], MATSPs, as well as the Multi-Attribute Vehicle Routing Problems (MAVRPs), are a hot topic in the scientific community. These kinds of problems are special cases of conventional routing problems, with the distinction of having multiple constraints and complex formulations. For this reason, this sort of problems has a great scientific interest. On the one hand, being NP-Hard, their resolution presents a scientific challenge. On the other hand, their applicability to real-world situation is greater than the conventional, or academic, versions of routing problems.

In this paper a MATSP is proposed, concretely, an Asymmetric Multiple Traveling Salesman Problem with Backhauls (AMTSPB), which is applicable to real-world on-demand transportation systems. In addition, in this work a benchmark of this problem is presented, and the first resolution of this benchmark is offered. For the resolution of the problem An Adaptive Multi-Crossover Population Algorithm (AMCPA) [7] has been implemented.

The rest of the paper is organized as follows. In Section II the AMTSPB is described and formulated. In Section III the proposed benchmark is described. The implemented AMCPA for its resolution is also depicted in this section. In Section IV the experimentation is shown. This paper finishes with the conclusions of the study and further work in Section V.

Deusto Instute of Technology, University of Deusto,
Av. Universidades, 24, Bilbao, Spain
Email: {e.osaba, roberto.carballedo, fernando.diaz}@deusto.es
{enrique.onieva, p.lopez, perallos}@deusto.es

[1] https://www.tfl.gov.uk/modes/dial-a-ride/
[2] http://www.riversidetransit.com/home/index.php/dial-a-ride

978-1-4799-8221-9/15/$31.00 ©2015 IEEE

II. Description of the AMTSPB

As explained in the introduction, in this paper a MATSP is proposed. The main characteristics of this kind of problems are their complex formulation with multiple constraints, and their applicability to real-world situations. This objective indirectly leads to an increased complexity of resolution, which entails, as the same time, to a major scientific challenge. The problem presented in this paper is an AMTSPB, which has three main characteristics.

1) *Asymmetry*: Unlike most routing problems treated in the literature, the costs in the AMTSPB are asymmetric, which means that the cost of traveling from one point i to another point j is different from the cost of the reverse trip. This feature adds complexity and realism to the problem, and it has been applied previously in the literature [8].
2) *Backhauling*: This feature is an adaption of the well-known backhauling system applied to some routing problems [9]. In this case, two sorts of nodes are available. The first types of nodes are the *pickup nodes*. In these points is where the people who have requested the transportation access to the vehicle. On the other hand, the *delivery nodes* is where people leave the vehicle. For this problem, all the *pickup nodes* have to be visited before visiting the *delivery nodes*. Furthermore, it is noteworthy that in one concrete node more than one person can take or leave the vehicle, which means that one route can visit more *pickup nodes* than *delivery nodes*, and vice versa.
3) *Multiple Vehicles*: This is a typical characteristic of the well-known Multiple Traveling Salesman Problem [10]. For this problem, there is a fleet K composed by a finite and fixed number of vehicles (k), which have to be used to meet the customers demands. Besides, there is a central depot in which all the vehicles must begin and end their route. This feature requires the problem to plan exactly k paths, one for each vehicle available. Furthermore, each route cannot visit more than a fixed q number of nodes.

With all this, the AMTSPB is a routing problem in which the costs of traveling between nodes are asymmetric, and the objective is to find exactly k number of routes of a maximum length of q nodes each, minimizing the total cost of the solution, and taking into account that all the *pickup nodes* have to be visited before the *delivery nodes*.

In this way, this problem can be defined on a complete graph $G = (V, A)$ where $V = \{v_0, v_1, v_2, \ldots, v_n,\}$ is the set of vertexes which represents the nodes of the system, and $A = \{(v_i, v_j) : v_i, v_j \in V, i \neq j\}$ is the set of arcs which represents the interconnection between nodes. Each arc has an associated distance cost c_{ij}, taking into account that $c_{ij} \neq c_{ji}$. The vertex v_0 represents the depot, and the rest are the visiting points. Besides, in order to facilitate the formulation, the set of customer V can be separated into two subsets, the first one for the *pickup nodes* $P = \{p_1, p_2, \ldots, p_p\}$, and the second one for the *delivery nodes* $D = \{d_{p+1}, d_{p+2}, \ldots, d_{p+m}\}$.

Furthermore, the presented AMTSPB be the mathematical formulated as follows:

Minimize:

$$\sum_{i=0}^{p+m} \sum_{j=0}^{p+m} \sum_{r=1}^{k} c_{ij} x_{ij}^r \tag{1}$$

Where:

$$x_{ij}^r \in \{0,1\}, \quad i,j = 0, \ldots, p+m, i \neq j; r = 1 \ldots k \tag{2}$$

Subject to constraints:

$$\sum_{i=0}^{p+m} \sum_{r=1}^{k} x_{ij}^r = 1, \quad i = 0, \ldots, p+m; i \neq j \tag{3}$$

$$\sum_{j=0}^{p+m} \sum_{r=1}^{k} x_{ij}^r = 1, \quad j = 0, \ldots, p+m; j \neq i \tag{4}$$

$$\sum_{j=0}^{p+m} \sum_{r=1}^{k} x_{0j}^r = k \tag{5}$$

$$\sum_{i=0}^{p+m} \sum_{r=1}^{k} x_{i0}^r = k \tag{6}$$

$$\sum_{i=0}^{p+m} \sum_{j=0}^{p+m} x_{ij}^r \leq q, \quad r = 1 \ldots k \tag{7}$$

$$\sum_{i=0}^{p+m} x_{ij}^r - \sum_{l=0}^{p+m} x_{jl}^r, \ j = 0, \ldots, p+m; r = 1 \ldots k \tag{8}$$

$$\sum_{j=0}^{p+m} x_{ij}^r - \sum_{l=0}^{p+m} x_{li}^r, \ i = 0, \ldots, p+m; r = 1 \ldots k \tag{9}$$

$$\sum_{i=p+1}^{p+m} \sum_{j=1}^{p} \sum_{r=1}^{k} x_{ij}^r = 0 \tag{10}$$

The first formula is the objective function, which should be minimized, and it is the sum of the costs of all routes of the solution. The clause 2 expresses the nature of the binary variable x_{ij}^k , which is 1 if the vehicle k uses the arc (i, j), and 0 otherwise. Constraints 3 and 4 ensure that all nodes are visited exactly once. Functions 5 and 6 assure that the total number of vehicles leaving the depot is the same as the number of vehicles that return to it. That number is equivalent to k, i.e., the total amount of available vehicles. Sentence 7 indicates that there is no route which length exceeds the q maximum length. The flow of the route, i.e., all vehicles that arrive at a customer leave it with the aim of going to another customer, is ensured by the formulas 8 and 9. Finally, the restriction 10 ensures that all the *pickup points* are visited before delivery ones.

Finally, permutation representation is used in the proposed AMTSPB for the solution encoding [11]. In this way, each individual X is encoded by a permutation

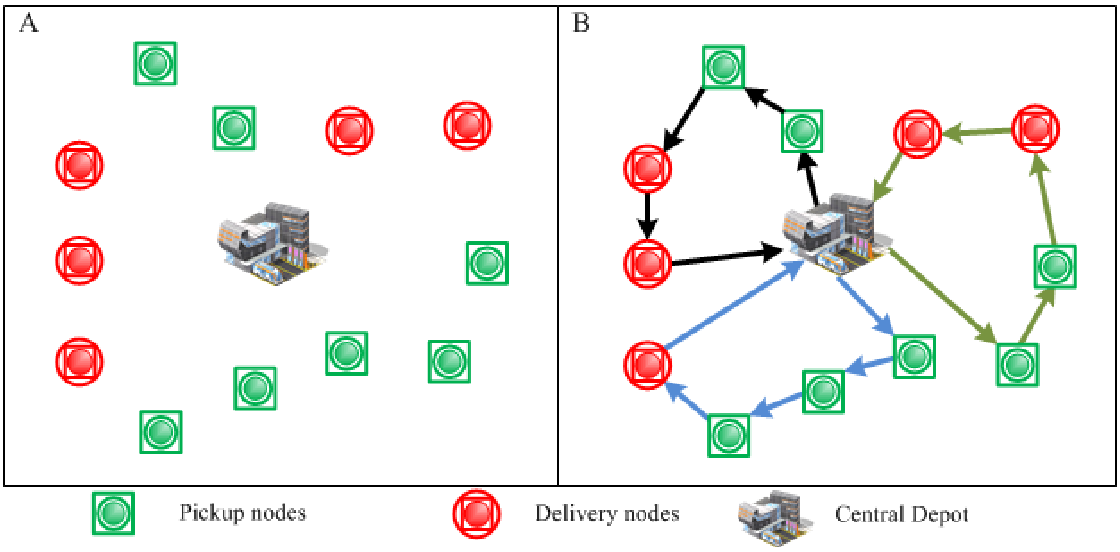

Fig. 1. A 12-noded and k=3 possible instance of the AMTSPB, and a possible solution

of numbers, which represents the path. In addition, to distinguish the different routes in a solution, they are separated by zeros. As an example, suppose a set of five *pickup nodes* $P = \{P_1, P_2, P_3, P_4, P_5\}$, and six *delivery nodes* $D = \{D_6, D_7, D_8, D_9, D_{10}, D_{11}, D_{12}\}$. One possible solution with m=3 would be $X = (P_5, P_3, D_6, D_7, 0, P_4, P_1, D_9, D_{12}, 0, P_2, D_{10}, D_8, D_{11})$.

In Figure 1(a) an example of a AMTSP instance with 12 nodes, and k=3 is depicted. Furthermore, in Figure 1(b) a possible solution for this instance is shown.

III. Proposed Benchmark and Used Technique

This section is divided in two different parts. First, in Section III-A, the benchmark developed in this study for the proposed AMTSPB is described. On the other hand, in Section III-B the technique used to address the presented AMTSPB is detailed.

A. The AMTSPB Benchmark

As can be read in the good practices presented by the authors of this study in [12], the existence of a benchmark to solve an optimization problem is crucial. The benchmark presented in this work for the proposed AMTSPB is a modification of the ATSP Benchmark that can be found in the TSPLib Benchmark [13]. These ATSP instances are all the available ones in the well-known TSPLib webpage[3].

In total, 19 different instances have been designed for the AMTSPB, which have from 17 to 443 nodes. The first node of each instance is considered as the depot. Furthermore, an extra parameter $type_i$ has been added to the rest of the nodes, which indicates whether the node i is a *pickup node* or a *delivery node*. This parameter has been set using the following procedure:

$$type_i = pickup\ node, \quad \forall i \in \{1, 3, 5, \ldots, n\}$$

$$type_i = delivery\ node, \quad \forall i \in \{2, 4, 6, \ldots, n\}$$

In addition, the number of vehicles available for each instance has been established set in k=4. On the other hand, the maximum length of each path has been set in $q = n/3$, where n is the total number of nodes of the instance. Finally, the geographical location and costs have remained the same as in the original instances.

With the aim of allowing the replication of this experimentation, the benchmark developed is available under request to the corresponding author of this paper.

B. The proposed AMCPA

Since the problem presented in this work is applicable to real-life situations, it has been opted for an algorithm which is simple to implement and quick to execute. Thus, the technique can offer results in a short time, something appreciated in real environment applications. As has been said in Section I, the selected meta-heuristic has been an AMCPA. The AMCPA is an adaptive variant of the classic Genetic Algorithm (GA), which main characteristics are the following:

- AMCPA reverses the philosophy of conventional GAs. Unlike GAs, it starts with high value for the mutation probability p_m, and a very low (or null) value for the crossover probability p_c.
- The proposed algorithm combines the p_c adaptation and the multi-crossover mechanism.
- The presented technique adapts its p_c depending on the search performance in recent iterations, and current generation number. In contrast, most of the previous studies rely the p_c adaptation in the population fitness [14, 15].

Regarding the adaptive mechanism, p_c of the algorithm is modified every generation depending on the results get in the previous iteration. This way, if the best solution found by

[3] *https://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/*

Algorithm 1: Pseudocode of the proposed AMCPA

$P \leftarrow$ Initialize the population;
Selection of the first Crossover Function ;
p_c = 0.0;
p_m = 1.0;
repeat
    $P_m \leftarrow$ Mutation phase;
    Parents selection process (P);
    $P_c \leftarrow$ Crossover phase;
    $P \leftarrow$ Survivor selection process ($P \cup P_m \cup P_c$) ;
    if *best solution has been improved* **then**
        p_c is restarted;
    else
        if $Maxp_c$ *has been reached* **then**
            Change the crossover function;
            p_c is restarted;
        else
            p_c is increased;
        end
    end
until *termination criterion reached*;
Return the best individual found;

the algorithm has been improved in the last generation, p_c is restarted. In other case, the value of p_c is increased based on the following formula:

$$p_c = p_c + \frac{2 \cdot G_{wi} + G}{N_I^3}$$

where:

N_I: number of individuals in the population,

G: total number of generations executed,

G_{wi}: number of generations executed without improvements,

As can be seen, p_c increases proportionally to the number of generations without any improvement in the best solution (G_{wi}) and the total number of generations (G).

Related to the multi-crossover feature, the proposed technique has more than one crossover operator which are alternated during the execution of the algorithm. At the initialization phase of the technique, one operator is assigned randomly. Then, when necessary, this function is replaced at random by another available, allowing repetition. For this purpose, a maximum value $Maxp_c$ for p_c is defined. If over the generations the p_c value exceeds $Maxp_c$, the crossover function is randomly replaced by another one, and p_c is restarted to its initial value.

It is noteworthy that $Maxp_c$ is an adjustable parameter, which has to be high enough to prevent a premature function change. Additionally, its value cannot be too high, in order to avoid an excessive runtime waste. A pseudocode of the metaheuristic is depicted in Algorithm 1.

IV. Experimentation

In this section the performed experimentation is described. First, in Section IV-A the parameterization of the developed AMCPA is detailed. Thereafter, in Section IV-B the results obtained are shown.

A. Parameters of the AMCPA

An initial population composed by 50 randomly generated individuals has been used for all the instances of the problem. In addition, as can be seen in Algorithm 1 the p_c starts at 0.0, and the p_m has been set in 1.0. When the best solution found has not been been improved, the p_c increases following the formula seen in Section III-B, otherwise, it returns to 0.0. Moreover, $Maxp_c$ has been stablished in 0.4.

In relation to the parents selection criteria, the well-known binary tournament has been used. On the other hand, regarding the survivor function, a 50% elitist - 50% random function has been used (which means that the half of the population is composed by the best individuals, and the remaining ones are selected randomly). About the ending criteria, the execution of the AMCPA finishes when the population converges. This same criteria has been used many time in the literature [16]. In the present study, the convergence is assumed when there are $n + \sum_{g=1}^{n} g$ generations without improvements in the best solution, where n is the size of the problem.

Furthermore, three crossover functions has been used for the proposed AMCPA. These functions are the Short Route Crossover (SRX), the Random Route Crossover (RRX), and the Large Routes Crossover (LRX). These operators are a particular case of the traditional crossover, in which the cut point is made always in the middle of the chromosome. The working way of the first of them is the following: first of all, half of the routes (the shortest ones) of one of the parents are inserted in the child. After that, the nodes already selected are removed from the other parent, and the remaining nodes are inserted in the child in the same order (taking into account the vehicle capacity). Assuming a 16-node instance (including the depot), an example could be the following:

$$P_1 = (1, 2, 3, 4, \mathbf{0}, 9, 10, 11, 12, \mathbf{0}, 13, 14, 15, \mathbf{0}, 5, 6, 7, 8)$$

$$P_2 = (1, 12, 6, 3, \mathbf{0}, 2, 4, 7, 11, \mathbf{0}, 5, 14, 9, \mathbf{0}, 8, 13, 10, 15)$$

The resulting offprings could be as follows:

$$O_1 = (1, 2, 3, 4, \mathbf{0}, 9, 10, 11, 12, \mathbf{0}, 6, 7, 5, 14, \mathbf{0}, 8, 13, 15)$$

$$O_2 = (1, 12, 6, 3, \mathbf{0}, 2, 4, 7, 11, \mathbf{0}, 9, 10, 13, 14, \mathbf{0}, 15, 5, 8)$$

RRX works similar to the SRX. In this case, the routes selected in the first step of the process are selected randomly, instead of choosing the best ones. Finally, in the case of LRX, the selected routes are the longest ones. Regarding the mutation function, the Vertex Insertion Function has been used. This function selects one random node from one randomly chosen route of the solution. This node is extracted, and inserted in another randomly selected route, respecting the route length constraint.

TABLE I. RESULTS OBTAINED BY THE AMCPA FOR THE PROPOSED AMTSPB BENCHMARK. FOR EACH INSTANCE RESULTS AVERAGE, STANDARD DEVIATION, MEDIAN, INTERQUARTILE RANGE AND TIME AVERAGE ARE SHOWN

Instance	Avg.	S. dev.	Median	I. R.	Time
AMTSPB_br17	69.2	2.8	68.5	3.7	0.78
AMTSPB_ftv33	2106.8	70.7	2113.5	95.5	1.05
AMTSPB_ftv35	2243.4	72.07	2222.5	122.7	1.01
AMTSPB_ftv38	2464.4	69.6	2471.0	108.2	1.09
AMTSPB_p43	5943.3	27.3	5943	48.2	1.10
AMTSPB_ftv44	2583.5	100.0	2552.0	147.2	1.35
AMTSPB_ftv47	2892.4	101.1	2887.5	120.2	1.21
AMTSPB_ry48p	21138.4	1017.7	20686.5	1328.0	1.32
AMTSPB_ft53	11332.2	502.4	11297.5	668.7	1.52
AMTSPB_ftv55	2904.3	193.6	2892.0	185.7	1.74
AMTSPB_ftv64	3147.0	189.2	3164.0	195.5	2.00
AMTSPB_ftv70	3528.0	152.8	3546.0	226.5	2.19
AMTSPB_ft70	47614.7	480.2	47637.5	652.7	3.01
AMTSPB_kro124p	56630.2	2438.7	57075.0	3356.7	6.02
AMTSPB_ftv170	6586.5	343.4	6523.5	443.2	18.66
AMTSPB_rbg323	2013.3	51.5	2006.0	46.7	54.22
AMTSPB_rbg358	2031.5	82.8	2022.5	86.2	69.07
AMTSPB_rbg403	3003.1	29.5	2998.0	45.5	79.76
AMTSPB_rbg443	3472.7	38.0	3467.5	48.7	92.73

B. Results

All the tests have been performed on an Intel Core i7 3930 computer, with 3.20 GHz and a RAM of 16 GB. Java has been used as programming language. All the 19 instances proposed for the AMTSPB have been used in the experimentation. The name of each AMTSPB instance has a number that displays the number of nodes it has. For each instance 30 runs have been executed, and the average fitness value (Avg.) and standard deviation (S. dev.) are shown. As said in Section II, the fitness value is the sum of the costs of all routes of the solution. Additionally, average runtime is also displayed (in seconds) as well as the median and the interquartile range (I.R.). These results can be seen in Table I.

Furthermore, in Table II the fitnees of the best solution found for each instance is depicted. In addition, the generations that have been needed to reach this solution, and the runtime of this execution are also depicted. Since this is the first appearance of the AMTSPB in the literature, these solutions are considered the best solutions found until the publication of this paper.

V. CONCLUSIONS AND FURTHER WORK

In this paper a new MATSP has been presented. This problem has been proposed with the aim of addressing different kind of DRP problems. Specifically, the developed problem is an AMTSPB, which objective is to find an exactly number number of routes which visit all the nodes once, minimizing the total cost of performing them. In addition, it has to be taken into account that the costs of traveling between nodes are asymmetric, and that two sort of nodes coexist: *pickup nodes* and *delivery nodes*. All the *delivery nodes* have to be visited after *pickup nodes*.

Furthermore, a benchmark composed by 19 instances has been proposed for the AMTSPB. These benchmark is an adaption of the well-known ATSP benchmark that can be found in the TSPLib. Finally, a first resolution for this benchmark has been presented. These solutions are considered the best ones, since it is the first time that the AMTSPB has been addressed.

TABLE II. BEST SOLUTIONS FOUND BY THE AMTSPB FOR EACH INSTANCE OF THE PROPOSED BENCHMARK

Instance	Fitness	Generation	Time
AMTSPB_br17	65	71	0.78
AMTSPB_ftv33	1988	1249	1.73
AMTSPB_ftv35	2130	836	1.04
AMTSPB_ftv38	2357	2906	1.70
AMTSPB_p43	5901	859	1.07
AMTSPB_ftv44	2429	2806	2.76
AMTSPB_ftv47	2671	3296	2.21
AMTSPB_ry48p	19954	5303	3.71
AMTSPB_ft53	10442	2340	1.68
AMTSPB_ftv55	2662	4380	2.47
AMTSPB_ftv64	2646	5114	4.08
AMTSPB_ftv70	3229	6608	4.09
AMTSPB_ft70	46746	9572	6.76
AMTSPB_kro124p	51995	11366	7.00
AMTSPB_ftv170	5673	28811	24.05
AMTSPB_rbg323	1942	69185	71.40
AMTSPB_rbg358	1942	71422	89.99
AMTSPB_rbg403	2967	47421	75.28
AMTSPB_rbg443	3426	60395	129.16

As future work, the resolution of the benchmark by some different techniques has been planned, in order to try to find better solutions for the developed instances. Additionally, the authors of this study intend to find some other real-world problems, with a great social interest, with the aim of modelizing them as MAVRP, and solving them.

REFERENCES

[1] J.-F. Cordeau, G. Laporte, J.-Y. Potvin, and M. W. Savelsbergh, "Transportation on demand," *Transportation, handbooks in operations research and management science*, vol. 14, pp. 429–466, 2004.

[2] J. Brake, J. D. Nelson, and S. Wright, "Demand responsive transport: towards the emergence of a new market segment," *Journal of Transport Geography*, vol. 12, no. 4, pp. 323–337, 2004.

[3] J.-F. Cordeau and G. Laporte, "The dial-a-ride problem: models and algorithms," *Annals of Operations Research*, vol. 153, no. 1, pp. 29–46, 2007.

[4] U. Ritzinger, J. Puchinger, and R. F. Hartl, "Dynamic programming based metaheuristics for the dial-a-ride problem," *Annals of Operations Research*, pp. 1–18, 2014.

[5] J. Paquette, J.-F. Cordeau, G. Laporte, and M. Pascoal, "Combining multicriteria analysis and tabu search for dial-a-ride problems," *Transportation Research Part B: Methodological*, vol. 52, pp. 1–16, 2013.

[6] T. Vidal, T. G. Crainic, M. Gendreau, and C. Prins, "Heuristics for multi-attribute vehicle routing problems: A survey and synthesis," *European Journal of Operational Research*, vol. 231, no. 1, pp. 1–21, 2013.

[7] E. Osaba, F. Diaz, E. Onieva, R. Carballedo, and A. Perallos, "Amcpa: A population metaheuristic with adaptive crossover probability and multi-crossover mechanism for solving combinatorial optimization problems," *International Journal of Artificial Intelligence*, vol. 12, no. 2, pp. 1–23, 2014.
[8] G. Laporte, H. Mercure, and Y. Nobert, "An exact algorithm for the asymmetrical capacitated vehicle routing problem," *Networks*, vol. 16, no. 1, pp. 33–46, 1986.
[9] M. Goetschalckx and C. Jacobs-Blecha, "The vehicle routing problem with backhauls," *European Journal of Operational Research*, vol. 42, no. 1, pp. 39–51, 1989.
[10] T. Bektas, "The multiple traveling salesman problem: an overview of formulations and solution procedures," *Omega*, vol. 34, no. 3, pp. 209–219, 2006.
[11] P. Toth and D. Vigo, *The vehicle routing problem*. Siam, 2001.
[12] E. Osaba, R. Carballedo, F. Diaz, E. Onieva, and A. Perallos, "A proposal of good practice in the formulation and comparison of meta-heuristics for solving routing problems," in *International Joint Conference SOCO-CISIS-ICEUTE*. Springer, 2014, pp. 31–40.
[13] G. Reinelt, "Tspliba traveling salesman problem library," *ORSA journal on computing*, vol. 3, no. 4, pp. 376–384, 1991.
[14] M. Srinivas and L. M. Patnaik, "Adaptive probabilities of crossover and mutation in genetic algorithms," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 24, no. 4, pp. 656–667, 1994.
[15] L. Wang and D.-B. Tang, "An improved adaptive genetic algorithm based on hormone modulation mechanism for job-shop scheduling problem," *Expert Systems with Applications*, vol. 38, no. 6, pp. 7243–7250, 2011.
[16] A. Moraglio and R. Poli, "Geometric crossover for the permutation representation," *Intelligenza Artificiale*, vol. 5, no. 1, pp. 49–63, 2011.