Available online at www.sciencedirect.com
ScienceDirect
Transportation Research Procedia 14 (2016) 1689 – 1698
Transportation Research Procedia
www.elsevier.com/locate/procedia

6th Transport Research Arena April 18-21, 2016

A flexible transportation service for the optimization of a fixed-route public transport network

Ioannis Bakas *, Richardos Drakoulis [a], Nikolaos Floudas [a], Panagiotis Lytrivis [a], Angelos Amditis [a]

[a]*Institute of Communication & Computer Systems (ICCS), 9 Iroon Politechniou Str. Zografou, Athens, Greece*

Abstract

In this paper, a flexible transportation service is proposed for a medium-sized city. By reviewing the literature of demand services, an algorithm solving the static multi-vehicle dial-a-ride problem with time windows and a fixed fleet of vehicles is developed and evaluated. Our contribution focuses on customers who specify their origin time from strategically chosen bus stations of the fixed infrastructure. A flexible objective function is used to reflect the customer's quality of service and also the operator's cost. Finally, simulated datasets of up to 500 user's requests are tested and compared against the public transport service of the city giving insight of how well a demand service performs.

Peer-review under responsibility of Road and Bridge Research Institute (IBDiM)

Keywords: Dial-a-ride problem; heuristic; routing; scheduling

1. Introduction

Well-designed public transport services are of crucial importance to ensure sustainable and environmentally friendly mobility to all citizens of modern cities. However, in sparsely populated areas with low demand for public

* Corresponding author. Tel.: +30-210-772-1076; fax: +30-210-772-229.
E-mail address: giannis.bakas@iccs.gr

Peer-review under responsibility of Road and Bridge Research Institute (IBDiM)
doi:10.1016/j.trpro.2016.05.134

transport more dynamic methods are required. Thus, it seems necessary to introduce some flexible and automatic reorganization of public transportation services in order to allow its optimization, both from the operator's and the citizens' point of view. The term flexible public transportation service refers to a range of strategies typically utilized in local public bus transportation. It is commonly applied to services, which incorporate - but not exclusively - elements of fixed-route or demand-responsive models.

In this paper a flexible public transportation service is proposed, which will be developed and simulated for the medium-sized city of Trikala, located in north-western Greece. The city public bus network is designed in a radial (star network) architecture, where all bus lines depart from the central square. The challenge of such a network is to optimally and efficiently serve the citizens to travel using public transport in neighboring not directly connected suburbs or cross the city without having to change multiple lines. The current solution for travelers involves using a bus line to the center and then transfer to another bus, which will get them to their destination increasing both the waiting and the travel time and therefore discouraging them from using public transport. Additionally, the possible introduction of a new dedicated circular line will be of relatively low and unpredictable demand to ensure viability.

In this work what is proposed is the use of the static multi-vehicle dial-a-ride problem (DARP) formulation to solve this situation. Several versions of DARP have been studied over the past 30 years. Dial-a-ride services may operate according to a static or a dynamic mode. In the first case, all requests are known in advance while in the second case requests are revealed through the day and vehicle routes are adjusted in real time. Our work is focused on the static multi-vehicle mode which assumes a fleet of M homogeneous vehicles starting from several depots located in a wide geographical area.

One of the first heuristics for the multi-vehicle static DARP was proposed by Jaw et al. (1986). Users specify a time window on the arrival time of their outbound trip and on the departure time of their inbound trip. The transporter then determines a planned departure time for the outbound trip and a planned arrival time for the inbound trip, while satisfying an upper bound on the ride time, which is expressed as a linear function of the direct ride time. A non-linear objective function combining several types of disutility is used to assess the quality of solutions. The heuristic selects users in order of earliest feasible pick-up time and gradually inserts them into vehicle routes so as to yield the least possible increase of the objective function. The algorithm was tested on artificial instances of 250 users and on real data set with 2627 users and 28 vehicles.

Bodin and Sexton (1986) have developed a cluster first, route second heuristic for the problem, employing a sequential insertion heuristic to form the clusters of users. The idea is to construct clusters by grouping users who are close together both in space and time dimension and then apply an insertion heuristic to construct the routes. This algorithm was tested on two instances extracted from Baltimore database and containing approximately 85 users each. Dumas et al. (1989) later improved this two-phase approach and solved instances with up to 200 users derived from real-life data taken from three Canadian cities.

Toth and Vigo (1996) have developed a new heuristic in which a parallel insertion procedure was used as route construction technique and then intra-route and inter-route exchanges were applied. It is assumed that users can specify requests with a time window on their origin or destination. The algorithm was tested on a real-life dataset involving between 276 and 312 requests from Bologna. Later improvements were applied by Toth and Vigo (1997) through the execution of a tabu thresholding post-optimization phase after the parallel insertion step.

In the last decade, a lot of focus has been given on stochastic algorithms. Recently, Parragh et al. (2010) and Parragh and Schmid (2012) used a variable neighborhood search and a hybrid large neighborhood search to solve the DARP. Cordeau and Laporte (2002) were the first to introduce a metaheuristic for solving the multi-vehicle static DARP. The authors applied a tabu search algorithm created by Fred W. Glover (1986), which iteratively removes a transportation request and reinserts it into another route. The interesting part of the algorithm is that intermediate infeasible solutions are allowed through the use of a penalized objective function. Later, Dominik Kirchler and Roberto Wolfler Calvo (2013) proposed a granular tabu search algorithm for the same problem, and proved that performs well in comparison with a classical tabu search algorithm, a genetic algorithm, and a variable neighborhood search.

This paper addresses the static DARP with time windows and a fixed fleet of vehicles. Our objective is to maximize the number of passengers served and the quality of service, as well as to minimize the overall operator's cost. Whereas all papers mentioned above use Euclidian distances and a mean velocity of a car to calculate the transition times between destinations, in our implementation a realistic travel time calculation was used, as will be described in section 4. Our contribution focuses on customers, who specify their origin time from strategically chosen bus stations of the fixed infrastructure, as opposed to choosing any-place in a pre-specified area, a formulation used extensively in the literature. The majority of the algorithms solving the DARP focus their objective on finding solutions with better service quality from the customer's point of view, our objective function was enhanced to express the disutility from the customer's perspective and also tries to capture the operator's cost by calculating the "consumption" of available vehicle resources and the distance routing costs, more details can be found in section 3.2. By taking into consideration the operator's cost, an insight of the viability of such services is given. Finally, algorithms developed in the last years are compared using known simulated datasets. The goal of this work is not to be compared against other algorithms, but to investigate that flexible demand services can solve the problems of public transport, especially in areas with low population density.

The remainder of this paper is organized as follows. In sections 2 and 3, the formulation of the problem and algorithmic details are provided. Finally, the experimental setup, results and conclusions are presented in sections 4 and 5 respectively.

2. Problem formulation

As this work is focused on the static mode of DARP, it is assumed that all requests are known beforehand and the system's customer is asked to specify, his desired pick-up time from a given origin, and his destination. Commonly in dial-a-ride services the origin and destination could be any place in a pre-specified area, in our case a collection of strategically chosen bus stations of the fixed infrastructure is used instead, a formulation which is closer to the definition of flexible transportation services. Table 1 shows the definitions of the quantities used.

Let there be N customers, each of which has specified a dpt_i and M a fleet of homogenous vehicles with capacity Q. The problem consists in finding a set of routes, such that an objective function is minimized and routing, capacity, and time window constraints are respected. Two time constraints (time windows) are defined for each customer, as follows:

$$ept_i = dpt_i \tag{1}$$

$$lpt_i = ept_i + ws_i \tag{2}$$

$$edt_i = ept_i + drt_i \tag{3}$$

$$ldt_i = lpt_i + mrt_i \tag{4}$$

Equations (1) – (2) denote that no customer will be picked up earlier than their desired pick-up time and no later than a pre-specified maximum ws_i. From equation (3) one can observe that if a customer is picked up on his desired time and delivered directly to his destination can achieve the earliest delivery time. Finally, there is an upper bound on the delivery time denoted by equation (4), where mrt_i is the maximum ride time allowed for customer i and is expressed by the linear function $\mathrm{mrt_i} = \alpha + \beta \cdot \mathrm{drt_i}\ \alpha, \beta \in \mathbb{Z}^{+}$. Variables α and β can be specified by any customer or by the system operator. In our implementation the operator is responsible for defining both these variables and the maximum acceptable deviation of customer i from his desired pick-up time, so from now on it shall be assumed for convenience that $mrt_i = MRT$ and $ws_i = WS$.

The main goal of the algorithm is to find an effective allocation of customers among vehicles and an associated time schedule of pick-ups such that the following equations are true:

$$adt_i - apt_i \leq MRT \tag{5}$$

$$dpt_i \leq apt_i \leq dpt_i + WS \tag{6}$$

There are also some other assumptions and service quality constraints that must be clarified before proceeding to the description of the algorithm and are listed below:

- The load of the vehicle cannot exceed the capacity Q.
- Dwell times – the amount of time needed to pick up and deliver customers – can be non-zero and different for every customer. In this implementation of the algorithm the dwell times are assumed to be zero.
- A vehicle is not allowed to wait idly when it is carrying passengers.

Table 1. Definition of quantities.

Notation	
N	The number of customers
M	The number of available vehicles available
Q	The vehicle's capacity
$+i$ $(-i)$	The pick-up and delivery node of customer i, respectively
$D(x, y)$	The time it takes a vehicle to go from point x to point y
drt_i	D(+i, -i), the direct ride time of customer i from origin to destination
dpt_i	The desired pick-up time of customer i
ept_i	The earliest possible time at which the pick-up of customer i can be made
lpt_i	The latest possible time at which the pick-up of customer i can be made
apt_i	The proposed by the algorithm time at which customer i will be picked up
adt_i	The actual delivery time of customer i to his destination
dv_i	$\mathrm{dv_i} = \mathrm{apt_i} - \mathrm{dpt_i}$, the deviation in time of customer i from his desired pick-up time
ws_i	The maximum acceptable deviation of customer i from his desired pick-up time, $\mathrm{dv_i} \leq \mathrm{ws_i}$

It is very important to clarify at this point the modes at which a vehicle can switch to. Active period is called, any time a vehicle is on the way of picking up, transporting, or delivering at least one customer. On the other hand slack or idle period is the time when the vehicle is empty and waiting.

3. Overview of the algorithm

The main goal of this paper is to make a comparison between a fixed-route and a flexible public transportation service and for this reason a modified version of ADARTW (Advance request dial-a-ride problem with time windows) algorithm developed by Jaw et al is proposed. ADARTW is a very interesting algorithm for two reasons. First, it addresses the most applicable and realistic version of the real-world problem in a way that avoids excessive abstraction and simplification. Second, it was tested on real data instances of about 2500 customers and gave high-quality solutions.

Consider that there are N customers and M available vehicles. The algorithm begins by indexing customers in the order of their desired pick-up time. For every customer i, the algorithm will search the best insertion into the work schedule of vehicle j. If it is infeasible to insert customer i into vehicle j the algorithm will continue with the rest of the vehicles. Having searched for all possible insertions in all available vehicles, the best one will be chosen, based on the insertion cost. If the insertion of customer i is infeasible in all vehicles then this customer has to be rejected by the system. One can recognize two basic parts of the algorithm, the search for feasible insertions and the optimization step, which is the cost calculation for each insertion.

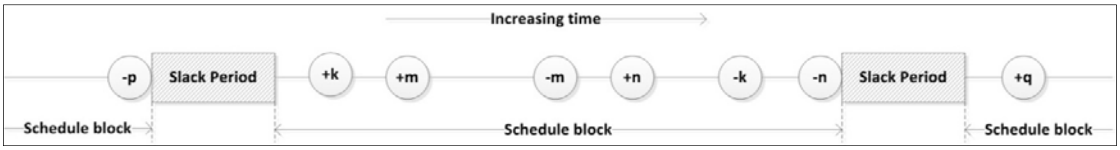

Fig. 1. Part of a vehicle's work schedule.

Table 2. Description of the algorithm.

```
Algorithm: ADARTW
for each customer_i ∈ {1, ..., N} do
        C ← ∅ ;
        for each vehicle_j ∈ {1, ..., M} do
                if feasible_insertion(customer_i, vehicle_j) then
                        cost_j ← find_best_insertion(customer_i, vehicle_j) ;
                        C ← C ∪ {cost_j} ;
                end if
        end for
        if C = ∅ then
                continue: customer_i is rejected ;
        else
                find_best_vehicle(C)
        end if
end for
```

3.1. Feasible insertions

In order to understand how the algorithm recognizes feasible insertions into vehicle work schedules, the parts of it must be outlined. The parts of a vehicle's work schedule are depicted in Fig. 1. A schedule block is a continuous period of active vehicle time between two successive periods of vehicle slack time (idle time). A schedule block always begins with a vehicle starting on its way to pick up a customer and ends after the vehicle delivers its last customer. Associated with a schedule block is a "schedule sequence" indicating the sequence of stops in the block, and a "time schedule" indicating the time when each stop is scheduled to take place. For example, in Fig. 1 the schedule sequence associated with the middle schedule block is $\{\mathrm{apt_k},\ \mathrm{apt_m},\ \mathrm{adt_m},\ \mathrm{apt_n},\ \mathrm{adt_k},\ \mathrm{adt_n}\}$. An insertion of a customer into the work-schedule of a vehicle is feasible only if it does not violate any service- quality and time window constraints for the newly assigned customer and for all other customers already assigned to that vehicle. For example, with respect to the middle schedule block of Fig. 1, the number of possible schedule sequences involving the insertion of a customer *i* is 28 (e.g. $\{+\mathrm{i}, -\mathrm{i}, +\mathrm{k}, +\mathrm{m}, -\mathrm{m}, +\mathrm{n}, -\mathrm{k}, -\mathrm{n}\}, \ldots, \{+\mathrm{i}, +\mathrm{k}, -\mathrm{i}, +\mathrm{m}, -\mathrm{m}, +\mathrm{n}, -\mathrm{k}, -\mathrm{n}\}$). Generally the number of possible insertions of a customer inside a schedule block is $(\mathrm{s}+2)(\mathrm{s}+1)/2$, where *s* is the number of stops already in the schedule block. The four different ways of insertion inside a schedule block are listed below:

1. Both the pick-up and delivery of a customer are inserted at the end of the last schedule block.
2. Both the pick-up and delivery of a customer are inserted between two consecutive stops in the schedule-block.
3. The pick-up of a customer takes place somewhere within the schedule-block, while his delivery is inserted at the end of the schedule block.
4. The pick-up and delivery of customer are separated by at least one other stop and the delivery is not the last stop of the schedule block.

Finally, it is worth to note here that instead of inserting a customer inside an existing schedule block, the algorithm is taking into account the creation of an entirely new schedule block.

3.2. Optimization step

As mentioned before the algorithm is using an optimization step in order to decide which of the insertions of customer *i* is the best. For this reason a cost function in the form of a weighted sum of disutility is used. The cost function tries to express the disutility from the customer's point of view and also tries to capture the operator's cost by calculating the "consumption" of available vehicle resources and the distance routing costs. The customer's disutility is given by equation (7),

$$du_i = du_i^d + du_i^r \tag{7}$$

where

$$du_i^d = C_1 \cdot x_i + C_2 \cdot x_i^2, \ 0 \leq x_i \leq WS \tag{8}$$

and

$$du_i^r = C_3 \cdot y_i + C_4 \cdot y_i^2, \ 0 \leq y_i \leq MRT \tag{9}$$

Equation (8) denotes customer's disutility due to deviation from his desired time, where $\mathrm{x_i} = \mathrm{apt_i} - \mathrm{dpt_i}$ and equation (9) is the disutility due to excess ride time, where $\mathrm{y_i} = \mathrm{art_i} - \mathrm{drt_i}$. The quadratic terms allow the modeling of situations in which disutilities are believed to increase nonlinearly with x_i and/or y_i. Clearly, by varying C_1, C_2, C_3 and C_4, many different types of behavior can be produced. This algorithm contrary to other algorithms is not taking into account only the disutility of the inserted customer but it calculates also the disutility caused by the insertion to the already inserted customers. This is expressed by equation (10) below,

$$du_i^o = \sum_{k \ on \ j} \left(du_k^{new} - du_k^{old} \right) \tag{10}$$

where $\mathrm{du_k^{new}}$ and $\mathrm{du_k^{old}}$ are, respectively, the disutilities to customer *k* after and before insertion of customer *i* into the schedule of vehicle *j*. The summation is over all customers *k* who were already assigned to vehicle *j* prior to the assignment of customer *i*.

Apart from the customer's disutility the cost calculation procedure also considers the service operator's cost, which is expressed by the "consumption" of available vehicle resources and the distance routing costs, as denoted by equation (11).

$$vc_i = C_5 \cdot z_i + C_6 \cdot w_i + U_i(C_7 \cdot z_i + C_8 \cdot w_i) + C_9 \cdot rc_i \tag{11}$$

where C_5, C_6, C_7, C_8, and C_9 are externally set constants, z_i is the additional active vehicle time required to serve customer *i*, w_i is the change in vehicle slack time due to the insertion and U_i is an indicator of system workload defined as:

$$U_i = \frac{No.of \ system \ customers \ in \ [ept_i - T_a, \ ept_i + T_b]}{No.of \ vehicles \ availabled \ in \ [ept_i - T_a, \ ept_i + T_b]} \tag{12}$$

U_i is equal to the ratio of the number of customers demanding service to the available number of vehicles, during a period of time that has as midpoint the earliest pick-up time of customer *i*. It is clear that U_i will be larger during periods of high demand.

Finally, an important cost parameter is added, which was not included in the ADARTW. This is the distance routing cost which is calculated by equation (13) and it is the extra driving distance after the insertion of customer *i* in vehicle *j*.

$$rc_i = rc_j^{new} - rc_j^{old} \tag{13}$$

4. Experimental setup

Our experiment and evaluation of the algorithm is based on the transportation needs of the medium-sized city of Trikala, located in northwestern Greece. The city covers an area of 69.2 km2 with a population density of 900 inhabitants / km^2. The city public bus network is designed in a radial (star network) architecture where all the bus lines depart from the central square. The challenge of such a network is to optimally and efficiently serve the citizens to travel using public transport in neighboring not directly connected suburbs or cross the city without having to change multiple lines.

In order to calculate the travel times for both the public transport and the demand bus the traffic simulator SUMO (Simulation of Urban MObility) by Daniel Krajzewicz et al. (2012) is used. After importing the road network of the city from Open Street Maps, all bus stops of the fixed infrastructure were inserted. The road traffic was simulated using residential statistics like population density, shop and school working hours, in order to have a more realistic simulation of the city. Several simulations were run so as to derive realistic travel times between the bus stops, whereas most papers in the literature use Euclidian distances and a mean velocity of a car to calculate the transition times between destinations.

For the purpose of this paper simulated data were used. We made the assumption that during working days, demand is greater during the hours 07:00-09:30 a.m. and then it meets a smaller pick between 16:00 and 20:00, especially when the shops are open. From a mathematical point of view a mixture of two Gaussian distributions (14) to simulate this behavior was used.

$$f_{demand} = n_1(x, 5400, 3600) + n_2(x, 37800, 9450), \ where\ n_i(x, \mu, \sigma) \tag{14}$$

In Fig. 2 both the two Gaussian distributions (n_1 and n_2) are depicted with solid lines and the mixture of them with the green dashed line, which represents the demand during 14 hours of simulation time, starting from 07:00 a.m. The histogram in the same figure gives an insight of the number of customers that demand the service during simulation time, at specific time periods.

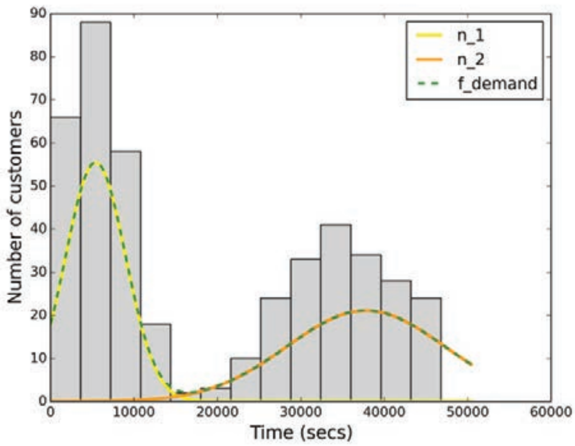

Fig. 2. Gaussian distribution mixture representing the demand.

Finally, it is worth to note that the origin and destination of every customer is selected randomly but is guided so as the destination could not be reached directly using only one public transport bus. As mentioned in the introduction this is the target group of users to which demand services could be applied to.

5. Results and conclusions

Before running the final experiment, several runs to gain insight of the weights $C_{1\text{-}9}$, were made. For this reason a small dataset of 100 customers was generated and tested with varying values of C_i. Table 3 shows the best parameters of the algorithm chosen for the final experiment. Four statistics are used to evaluate the effectiveness and quality of the demand service 1) the average deviation from the customer's desired pick-up time, 2) the average ride time, which is the time the customer is in the vehicle, 3) the average ride ratio, which is the ratio between the actual ride time and the direct ride time of the customer and finally, 4) the driving distance made by the vehicles.

Without loss of generality the frequency of the three public transport lines serving the area is assumed to be 15 minutes for the time period of the simulation, which is actually bigger than in the real world and is giving a small precedence to the public transport service. For the final experiment, a dataset of 500 customers was generated following the procedure described in section 4. The fleet consists by four mini-buses with capacity of 8 persons.

Table 3. Parameters of the algorithm used for the final experiment.

C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	*WS*
75.0	2.0	0.3	0.0	0.0	0.0	0.8	1.0	1000.0	10 min

In this work, we intend to evaluate the performance and quality of a flexible demand service against a public transport service, for this reason the statistics mentioned above were also calculated for the public transport service.

In Fig. 3 two bar plots comparing the performance of the two services are depicted. In both diagrams it is clear that from the customer's disutility perspective the demand service performs better during the whole simulation. Taking a closer look at both the average waiting time and average ride ratio bars of demand service, one can observe that their values are analogous. This can be explained by the fact that in low demand periods customers are likely to be delivered directly from their origin to their destination so that their service times are close to their desired times and the ride ratio is close to 1. On the other hand, in high demand periods more customers share the same ride, which results both in higher waiting time and ride ratio values.

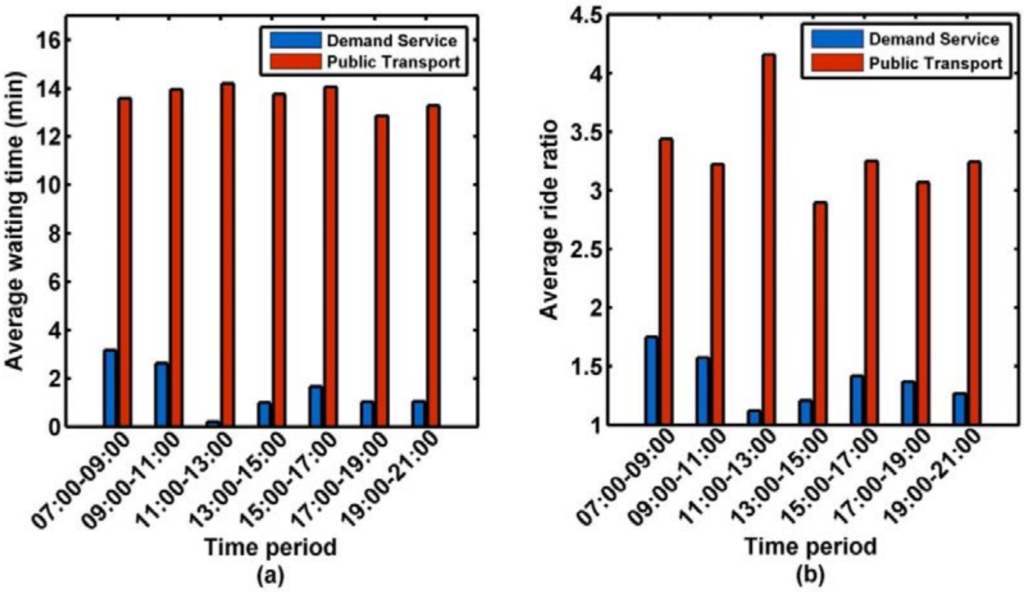

Fig. 3 (a) Comparison of average waiting time; (b) Comparison of average ride ratio.

It is interesting to get a closer look at the waiting time and the ride ratio of every customer individually, as illustrated by Fig. 4 (a) – (d). Diagrams (a) and (b) show how the waiting time and the ride ratio of the demand service are affected by the demand during the simulation. Comparing Fig.4 (a) – (b) with Fig. 2 we can see that the bar values are following the trend of f_{demand}, verifying the assumption that in high demand periods both the waiting time and the ride ratio are increasing. On the other hand, as it is shown by diagrams in Fig. 4 (c) – (d) there is no relationship between the demand and the quality statistics of the public transport service. The fact that there is a relationship between the demand and the quality of service for the algorithm, gives motivation for future work.

A relaxation mechanism can be implemented, which will adjust the $C_{1\text{-}9}$ parameters at runtime especially on high demand periods in order to give better results and also avoid rejecting customers.

Finally, from the operator's point of view, the four mini-buses used for the demand service covered the distance of 1063.67 km where the public transport buses used for the simulation drove about 901.6 km. Although, the difference is really small we have to mention that the demand service's routing cost is linearly increasing to the number of customers, as depicted by Fig. 5. This is the main drawback of demand service, but it is proved that is working well in regions with low demand.

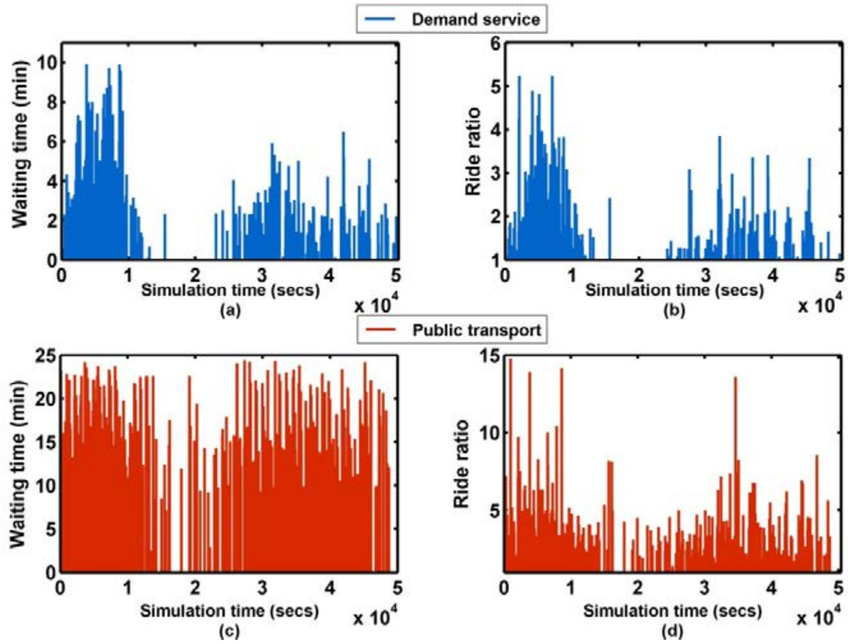

Fig. 4 (a) Demand service customer's waiting time; (b) Demand service customer's ride ratio; (c) Public transport customer's waiting time; (d) Public transport customer's ride ratio

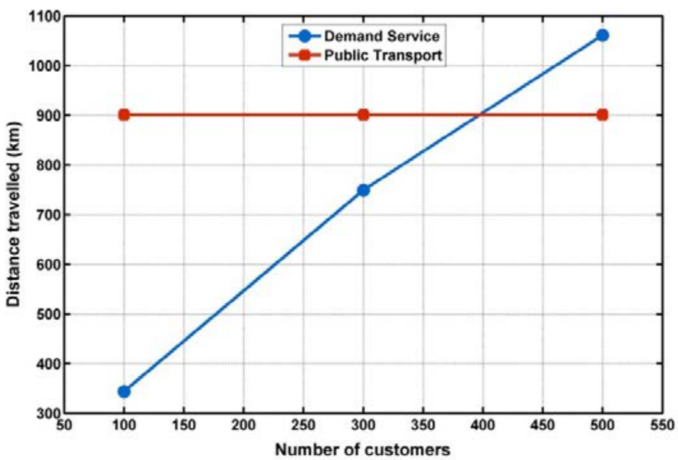

Fig. 5. Routing cost comparison between the demand service and the public transport.

Acknowledgements

This work is being developed and supported in the framework of TEAM, which is an EU research project that envisions advanced integrated mobility system, including the Collaborative Public Transport Optimization (CPTO) application. The main goal of the CPTO application is to optimally improve the flexibility of the transport infrastructure in order to serve dynamically the demand of cities and citizens.

References

Bodin L., Sexton T., 1986. The multi-vehicle subscriber dial-a-ride problem, TIMS Studies in Man- agement Science 2, pp. 73–86.

Cordeau J-F., Laporte G., 2003, A tabu search heuristic for the static multi-vehicle dial-a-ride problem, Transportation Research Part B: Methodological 37, pp. 579-594.

Dumas Y., Desrosiers J., Soumis F., 1989a. Large scale multi-vehicle dial-a-ride problems, Les Cahiers du GERAD, G–89–30, École des Hautes Études Commerciales, Montréal.

Glover F., 1986. Future paths for integer programming and links to artificial intelligence, Computers and Operations Research 13, pp. 533–549.

Jaw J., Odoni A., Psaraftis H., Wilson N., 1986. A heuristic algorithm for the multi-vehicle advance- request dial-a-ride problem with time windows, Transportation Research B 20, pp. 243–257.

Kirchler D., Wolfler Calvo R., 2013. A Granular Tabu Search algorithm for the Dial-a-Ride Problem, Transportation Research Part B: Methodological 56, pp. 120-135.

Krajzewicz D., Erdmann J., Behrisch M., Bieker L., 2012. Recent Development and Applications of SUMO - Simulation of Urban Mobility, International Journal On Advances in Systems and Measurements, pp. 128-138.

Parragh S.N., Doerner, K.F., Hartl, R.F., 2010. Variable neighborhood search for the dial-a-ride problem, Computers & Operations Research 37 (6), pp. 1129– 1138.

Parragh S.N., Schmid, V., 2012. Hybrid column generation and large neighborhood search for the dial-a-ride problem, Computers & Operations Research 40 (1), pp. 490–497.

Tomorrow's Elastic Adaptive Mobility (TEAM), FP7 integrated project, Nov. 2012 - Oct. 2016, website: https://www.collaborative-team.eu/

Toth P., Vigo D., 1996, Fast local search algorithms for the handicapped persons transportation problem, Meta-heuristics: Theory and applications, Kluwer, Boston, pp. 677– 690.

Toth P., Vigo D., 1997. Heuristic algorithms for the handicapped persons transportation problem, Transportation Science 31, pp. 60–71.