The importance of information flows temporal attributes for the efficient scheduling of dynamic demand responsive transport services

Original
The importance of information flows temporal attributes for the efficient scheduling of dynamic demand responsive transport services / Diana, Marco. - In: JOURNAL OF ADVANCED TRANSPORTATION. - ISSN 0197-6729. - 40(1):(2006), pp. 23-46. [10.1002/atr.5670400103]

Publisher:
Wiley

Published
DOI:10.1002/atr.5670400103

(Article begins on next page)

The Importance of Information Flows Temporal Attributes for the Efficient Scheduling of Dynamic Demand Responsive Transport Services

Marco Diana

1st June 2005

Diana, M. (2006) The importance of information flows temporal attributes for the efficient scheduling of dynamic demand responsive transport services, *Journal of Advanced Transportation*, vol. 40(1), pp. 23-46.

From: Marco Diana [mailto:marco.diana@polito.it]
Sent: 14 September 2011 12:21
To: Permission Requests - UK
Subject: Article upload on institutional repository

Dear Permissions Department,

I am a researcher at Politecnico di Torino (Italy), where the institutional repository PORTO has been recently launched (http://porto.polito.it/). PORTO is a not-for-profit service for academic authors, providing access to the full-text of their publications. Full bibliographic details are given for each article, including the journal of original publication, etc.

I would like to deposit there the full text (not the editorial version) of the following article, published in 2006:

http://dx.doi.org/10.1002/atr.5670400103

and it is unclear for me if I am allowed to do so. […]

Thank you for your feedback on this issue.

Best regards

Marco Diana

****

From: Permission Requests - UK <permissionsuk@wiley.com>
To: Marco Diana <marco.diana@polito.it>
Date: Thu, 15 Sep 2011 14:59:08 +0100
Subject: RE: Article upload on institutional repository

Dear Marco Diana,

Thank you for your request. Your permitted uses as author as stated in the Copyright Transfer Agreement are:

Wiley-Blackwell licenses back the following rights to the Contributor in the version of the Contribution as originally submitted for publication, the right to self-archive on the Contributor's personal website or in the Contributor's institution's/employer's institutional repository or archive. This right extends to both intranets and the Internet. The Contributor may not update the submission version or replace it with the published Contribution. The version posted must contain a legend as follows: This is the pre-peer reviewed version of the following article: FULL CITE, which has been published in final form at [Link to final article].

We hope this clarifies the situation.

Best wishes,

Richard S Jones BSc MSc
Permissions Assistant

Wiley-Blackwell | 9600 Garsington Road | Oxford | OX4 2DQ | UK

The Importance of Information Flows Temporal Attributes for the Efficient Scheduling of Dynamic Demand Responsive Transport Services

Marco Diana

INRETS – French National Institute for Transport and Safety Research
DEST – Department of Transport Economics and Sociology
2, avenue du Général Malleret-Joinville
F-94114 Arcueil CEDEX, France
Tel. +33 (0)1 47 40 72 54 – Fax +33 (0)1 45 47 56 06
E-mail marco.diana@inrets.fr

Abstract

The operation of a demand responsive transport service usually involves the management of dynamic requests. The underlying algorithms are mainly adaptations of procedures carefully designed to solve static versions of the problem, in which all the requests are known in advance. However there is no guarantee that the effectiveness of an algorithm stays unchanged when it is manipulated to work in a dynamic environment. On the other hand, the way the input is revealed to the algorithm has a decisive role on the schedule quality. We analyze three characteristics of the information flow (percentage of real-time requests, interval between call-in and requested pickup time and length of the computational cycle time), assessing their influence on the effectiveness of the scheduling process.

Introduction

Demand Responsive Transport Services (DRTS) are a particular form of public transport characterized by the fact that the vehicles operate in response to calls from passengers to the transit operator, who then dispatches a vehicle to collect the clients and transport them to their destinations. Unlike traditional taxicabs, the scheduling algorithm tries to match the requests in order to share as many rides as possible. In a dynamic, or online system, the three main activities of a DRTS (collecting the requests, building the schedules and providing the service) are not carried out in a rigorous sequential order, but partially or totally overlap. Dynamic systems are more flexible and interesting on an operational point of view, since they are useful in handling many situations that arise in practice, such as incoming requests, customer no-shows at pickup points, vehicles breakdowns or traffic jams.

The operation of these systems implies the use of specialized algorithms. The schedule of the service could be obtained through the solution of a combinatorial optimization problem, known in the literature as the Pickup and Delivery Problem, or the Dial-a-Ride Problem. However its computational complexity makes it impossible to find the optimal solution for problem instances of practical interest. Research efforts are thus focused in defining heuristics that can find the best solution within a given computational time. It is important to note that most existing methodologies have been originally conceived to efficiently solve static problems, and have then been adapted to work in a dynamic environment. However the characteristics of an online problem are radically different, and it could hence be supposed that solution algorithms that work well in solving a static problem may perform in a less satisfying way when tackling its dynamic version. In order to shed some light on this concern, we preliminarily compare the performances of two different insertion heuristics that have already been evaluated for the solution of static versions of the Dial-a-Ride problem (Diana and Dessouky, 2004). It is shown that the findings of that paper, namely the clear superiority of one of the two heuristics, are no longer valid when the problem becomes dynamic.

These results suggest that the way the algorithm works is not the only determinant of the efficiency of the scheduling process (and maybe not even the most important). Another decisive factor is the “temporal dimension” of the dynamic problem, i.e. the way the information is revealed to the algorithm. This issue has comparatively received little attention in past research. It is quite intuitive to say that a dynamic problem can be solved in a less efficient way than its corresponding static version. Also, it is likely that "more dynamic" problems, in which the information is mostly revealed little in advance, are more difficult to solve than the "less dynamic" ones. Hence, having fixed the solution method and the problem instance (for example, the pool of requests to be served) there seems to be a relationship between the way the information becomes known and the efficiency of the computational procedure.

Our objective is to quantitatively investigate the above relationship. In order to do that, we focus our attention on the information flow that is feeding the algorithm (the list of incoming requests, network travel times etc.). It is then possible to define some temporal attributes related to the information flow, for example the advance with which we know a request, or the percentage of requests that is known before the start of the scheduling process. We will better define the attributes that are considered in the present research in a later section. The relevance of this aspect is given by the fact that some of these attributes can be more or less influenced by choices of the planner. The outcome of our research would then be the definition of policy guidelines that could contribute in incrementing the overall economic efficiency of the system.

In the following, after a bibliographic review on dynamic demand responsive systems, we will describe the problem we want to study, and the simulation process we will use to solve it. After that, we will introduce some temporal parameters that can be seen as attributes of the information inflows of the scheduling process. We will then study the relationship between these attributes, the scheduling process and the quality of the solution, clarifying the importance of each factor and looking for possible policy implications. Finally, some concluding remarks on the overall research activity are reported.

Literature review

Generally speaking, the dynamic version of any routing problem has been much less studied than the corresponding static case. Psaraftis (1988, 1995) gives a survey on the state of the art of the research on the subject, and points out the potential benefits that the transportation sector could have in relying more consistently on these systems. Another comprehensive state of the art paper on the subject can be found in Powell et al. (1995). A comparative review of some of the former strategies to solve dynamic routing problems is reported in Powell (1988).

The aforementioned authors point out that no benchmark sets of dynamic problems, upon which to test different solution procedures, are available. Furthermore, the dynamic version of the above mentioned Pickup and Delivery problem is not univocally defined, since there are various ways to make a problem dynamic. Psaraftis (1995) proposes a taxonomy for dynamic routing problems that is based on some attributes of information. It is hence important to stress that the way the information is revealed is perhaps the most meaningful way to look at the problem.

Researchers proposed effective heuristics for the solution of dynamic Pickup and Delivery problems in more recent years. Ichoua et al. (2000) present a review of the approaches commonly being used. Most of these are more or less straightforward adaptations of methods formerly conceived for static problems. In particular, insertion methods can be easily adapted to work in a dynamic environment and hence have been widely used (Madsen et al., 1995; Dessouky and Adam, 1998b; Horn, 2002; Fu, 1999a; Fu, 1999b; Fu, 2002). In all the mentioned works, the insertion criterion is to minimize the additional cost of serving the incoming request, that is the heuristic rule that was primarily suggested by Solomon (1987). Other less applicable researches make use of reoptimization techniques. A new problem is solved every time there is a new input, the current solution generally being used as starting solution for the new problem. These approaches make use of a wide range of methodologies: local search procedures (Dial, 1995), metaheuristics (Jih and Hsu, 1999; Gendreau et al., 1999; Ichoua et al., 2000; Ichoua et al., 2003), incomplete optimization techniques (Savelsbergh and Sol, 1998; Jih and Hsu, 1999; Colorni and Righini, 2001), fuzzy logic (Teodorovic and Radivojevic, 2000). Heuristic rules that are more specifically designed to take into consideration the peculiarities of a dynamic routing and scheduling problem can be found in Mitrović-Minić and Laporte (2004) and in Mitrović-Minić et al. (2004).

In dynamic problems the need to quickly find a solution is much more impellent than in static cases. For this, these reoptimization techniques can only handle problems that are quite small; for example, Jih and Hsu (1999) study the single vehicle problem. On the other hand, some of the proposed algorithms need very powerful hardware to be used to solve realistic problems. The tabu search algorithm by Gendreau et al. (1999) and Ichoua et al. (2000, 2003) was implemented on a network of workstations, whereas the decentralized system envisioned by Dial (1995) foresees a computing unit on every vehicle of the fleet. Another approach may be to build a solution with a greedy rule, so that near-term events are immediately scheduled, and to simultaneously run a reoptimization procedure on another computer to improve the schedule of long-term events. One way to cope with the problem of the excessive computational burden is to limit the scheduling horizon, i.e. to consider only events that will happen up to a certain time point in the future (Savelsbergh and Sol, 1998; Colorni and Righini, 2001).

From these short notes, we can see that the definition and the qualitative characterization of a dynamic problem has been the focus of earliest researches, whereas heuristic solution approaches have more recently been proposed. The following step should be to focus on the time dimension of the information flows, according to the definition we have given in the introduction. Past research suggest that a simulation approach could be helpful in this case. Wilson et al. (1970) pioneered the use of simulations to compare different routing algorithms, as well the influence of the service area size, demand level and service quality on fleet size requirements. Regan et al. (1996) evaluated the performance of alternative load acceptance and assignment strategies for a dynamic goods distribution problem. Fu (2001) and Fu and Xu (2001) focus on the effect of trip cancellations and of the proportion of real-time demand trips on the operational performance of the system. The percentage of online requests is one of the attributes of the information flow that we also consider in the present work. The work of Larsen et al. (2002) is mostly interesting, since it examines the impact of dynamism on the quality of the solution and on the best-to-implement methodology for the so-called Partially Dynamic Travelling Repairman Problem. For this, a synthetic index, called degree of dynamism, is defined and a subsequent taxonomy is introduced. The degree of dynamism encompasses some of the temporal attributes that we will study in a more disaggregate way, as we later show.

Description of the studied system

The static version of DRTS we are going to investigate is formally described in Diana and Dessouky (2004). Thus, for the sake of briefness, in the following we will only give its qualitative synthetic description.

When making a reservation, the customer *k* has to specify the origin (O_k) and the destination (D_k) of the trip. He can also specify a time point ST_k, that can be either the pickup or the delivery time; on the other hand, the operator computes the maximum ride time as a linear function of the direct ride time. He also fixes the maximum wait time at the pickup point (for customers that specify the pickup time) or the maximum advance time at the delivery point (for customers that specify the delivery time). All these constraints, related to the quality of the service to be provided, imply the respect of time windows for all the pickup and delivery nodes, that we will denote by (EP_k, LP_k) and (ED_k, LD_k) respectively. In addition, we associate with each request a service time both at the pickup (sp_k) and at the delivery (sd_k) node, and the vehicles are allowed to stop and idle at any pickup location, waiting to serve the following request, if only no passengers are onboard. Passengers must have a seat, so that each vehicle has a maximum capacity given by the number of seats. We allow for requests being dynamically revealed to the algorithm when the scheduling process has already started. The objective function z consists in minimizing the weighted sum of the total distance traveled by all the vehicles, the excess ride time for all the customers and the total duration of the idle times.

Dynamic problems usually being considered involve some amount of stochastic information, i.e. information that is known a priori with some degree of uncertainty, which eventually lessens as time passes. Instead, we will study a deterministic problem. That is, a given piece of information is either not yet known, or completely known. For example, we do not know any spatial or temporal distribution of the service demand in order to forecast future requests, but we simply wait until the customer books the trip, and then we schedule it. We believe that this assumption can be considered realistic and useful when dealing with a system with a high level of ITS (Intelligent Transport Systems) technology adoption. There is a trade-off between the technological level of the system and the amount of incertitude associated with the input data we need in a scheduling process. In other words, a higher technological level may induce problems that are more dynamic and less stochastic. For example, a sufficiently sophisticated ITS architecture can detect the travel times on a road every a few seconds. Hence, we can have deterministic information that is denoted by a high degree of dynamism, instead of knowing a priori a distribution of the travel times that is valid for a longer period.

We have already mentioned that in a dynamic environment we need to take into account the time dimension of the system. This in turn would imply the consideration of the duration of all the tasks within the system. In the remainder however we will consider only the computational time of the scheduling algorithm, assuming that the following quantities are negligible:

1) The duration of the phone call of the customer to the call center;
2) The time needed to transmit the schedule to the vehicles;
3) The time needed to the customer to be ready to be picked up at the convened point.

Dynamic scheduling process

<u>Call-in simulator</u>

The call-in simulator is a computer program that emulates the flow of service requests to the scheduling algorithm. The input is the following:

1) The list of the *n* requests, each request *k* being defined by the above mentioned quantities: O_k, D_k, (EP_k, LP_k), (ED_k, LD_k), sp_k and sd_k.
2) The distribution $f(\Delta t)$ of the time intervals Δt between call-in and requested pickup or delivery time, given in discrete form. Thus, for each class *i* of time intervals whose boundary values are [Δt_i, Δt_{i+1}) we specify the corresponding relative frequency f_i.
3) The percentage *p* of requests known in advance (offline requests).

The simulator starts by randomly selecting $p \cdot n$ requests from the list. These are considered as offline requests and are immediately sent to the scheduler. The remainder are real-time requests. For each real-time request *k* we need to know the call-in time IT_k. A sample whose cardinality is the number of real-time requests $(1-p) \cdot n$ is drawn from $f(\Delta t)$, thus giving the set (Δt_1, Δt_2, .., Δt_k, .., $\Delta t_{(1-p) \cdot n}$). Assuming that the probability that a request *k* is real-time and that the distribution of the time intervals are not dependent on its desired pickup or delivery time ST_k, we compute

$$IT_k = ST_k - \Delta t_k \qquad \text{for every } k .$$

Real-time requests are not sent to the scheduling algorithm at time IT_k. We set a maximum acceptable waiting time *C* for the user that is waiting for feedback, so that the scheduling horizon is divided into several cycle times of length *C*. Let (CT_i, CT_{i+1}] be the time limits of cycle *i*, $CT_{i+1} - CT_i = C$. Then, every request *k* such that $CT_i < IT_k \leq CT_{i+1}$ is sent to the algorithm at time CT_{i+1}. Buffering the requests is in fact likely to improve the performance of the heuristic.

Scheduling algorithm

In order to schedule the requests, two algorithms described in Diana and Dessouky (2004) have been used. The first one is the classical minimum incremental cost insertion procedure, in which a given request k is inserted in the position of the schedule that causes the minimum increment in the value of the objective function z. The latter algorithm makes use of a regret metric, that takes into account the drawback of not immediately inserting a request that could be more difficult to schedule at a later stage. Whenever the algorithm has to select the candidate request to be inserted, it computes a minimum cost matrix, i.e. the minimum cost c_{ij} of inserting each request i in every route j under construction. When an insertion is infeasible, the corresponding cell of the matrix is set to an arbitrarily large value. Let c_i* be the smallest value in each row of the matrix. The second step is to compute the regret cost r_i of each request i, given by

$$r_i = \sum_{j=1}^{n} \left(c_{ij} - c_i{}^* \right) .$$

Finally, the request to be inserted is the one with the maximum value of r_i.

Both the classical and the regret insertion algorithm have been adapted to work in the dynamic environment we introduced in the previous section. Their input consists of both static and dynamic data. Static data are network travel lengths and travel times, since in the present work we do not consider variation of speed flows within the simulation period. Hence, a shortest path problem solver is used to preprocess those data and obtain shortest paths from any possible pair of service points of future requests. Those shortest path are then handled to the scheduling algorithm. Also the above defined parameters that control the quality of the solution (maximum ride time and maximum wait or advance time) are supposed not to change during the simulation. The dynamic input are the requests coming from the call-in simulator. Thus, whenever a group of requests is revealed at time CT_i, these are inserted in the previously built routes and a new schedule is generated. Of course the schedule is updated only for those events (service times, vehicle idle times) that have still to take place. This feature can be used to increase the efficiency of the scheduling process, as it is discussed in the next section.

The flow chart of the simulation process is shown in figure 1.

Fig. 1.

<u>Dynamic insertion feasibility checks</u>

Since we are dealing with an extremely constrained dynamic problem, it is most important to quickly detect whether the insertion of a request in a certain position is feasible or not. For this, we developed a tailored procedure, generalizing the work carried out by Jaw et al. (1986) for static problems. Although specific algorithmic design issues are not the focus of this paper, we will give more details concerning this point, since we are not aware of a previous generalization of the often mentioned feasibility check methodology of Jaw et al. (1986) in order to make it work in a dynamic environment.

In the following we refer a pickup or delivery point of a request as a *node*, or *service point* of the schedule, whereas a *schedule block* is a succession of nodes delimited by two *idle times*, i.e. two pauses of the vehicle that is idling without passengers onboard. For each service point i in the schedule of a vehicle, let the control quantities $BTOP_i$, $BBOTTOM_i$, $ATOP_i$ and $ABOTTOM_i$ be respectively the maximum time interval by which all the nodes preceding and following i (i is included) can be pushed backward and forward, according to the formal definition given by Diana and Dessouky (2004). Unlike Jaw et al. (1986), these control quantities are not computed with reference to the schedule block of the insertion point, but refer to the whole schedule of the vehicle. This makes the management of insertions of requests across different blocks a lot easier. Then, a node i' can be inserted between nodes i and $i+1$ of the current schedule, without violating the time windows of all the nodes already in the schedule, only if $BTOP_i + ABOTTOM_{i+1}$ is greater than the additional travel time needed to serve i'.

When dynamically computing the control quantities, it must be considered that the part of the schedule that has already been deployed must not be changed when trying to insert a request. Let us consider a schedule with n requests, thus containing $2n+2$ nodes (n pickup and n delivery points, plus the origin and destination depot) indexed with i, $i = 0, 1, ..., 2n+1$. Let us also assume that the vehicle will be heading towards node j at the end of the present computational cycle, and that k is the last node of the schedule block containing j. Then the following holds:

1) Nodes 0, 1, .., j cannot be moved. Ichoua et al. (2000) studied the possibility of diverting the vehicle, i.e. changing node j in the schedule. We do not consider this case, since we assume to study a DRTS in an urban area in which the mean length of a trip between two nodes is quite low and several vehicles operate. Hence we put equal to zero $ATOP_i$, $BTOP_i$, $ABOTTOM_i$ and $BBOTTOM_i$ for every $i = 0, 1, ..., j$. Since the detour for inserting any node in any position is strictly greater than zero (i.e. the triangular inequality holds), this also implies that no node can be inserted in these positions.
2) Nodes $j+1$, $j+2$, ..., k cannot be moved if they come before the node being inserted. In fact, this would imply the need of a movement in nodes that have already been served, since by definition there is no idle time between $j+1$, $j+2$, ..., k and j, $j-1$, ... This condition is satisfied by putting equal to zero the quantities $BTOP_i$ and $BBOTTOM_i$ for every $i = j+1, j+2, ..., k$. On the other hand, the quantities $ATOP_i$ and $ABOTTOM_i$ for every $i = j+1, j+2, ..., k$ are computed like the static case, as shown in details by Diana and Dessouky (2004).
3) Nodes $k+1$, $k+2$, .., $2n+1$ are not affected by the fact that the problem is dynamic, and the related control quantities are computed like the static case.

We report in figure 2 a schematic representation of these new dynamic insertion feasibility checks.

Fig. 2.

Finally, we remark that in a dynamic environment there may be requests whose pickup or even delivery time window are entirely before the actual arrival time at node j, for every vehicle on duty. This is particularly likely to happen when the customers specify a delivery time for a long journey and are requesting the service too late. These requests of course cannot be served and are detected by the call-in simulator, in order to avoid handling them to the scheduling algorithm. This is done to spare useless computations, but moreover to keep a distinction between those requests, that simply cannot be accepted, from those that cannot be scheduled because the algorithm does not find a feasible insertion. In order to correctly evaluate the performance of the heuristic, only these latter are to be considered. A more detailed description of this aspect can be found in Diana (2003).

Experimental design

In the following we will consider five problem instances containing 1000 requests that are representative of the transportation service for elderly and disabled people in Los Angeles County. These instances have been generated and statically solved by Diana and Dessouky (2004). Thus, we already know the minimum number of vehicles needed to serve each of the requests when they are all known in advance. In this case the regret insertion heuristic was proven to outperform the minimum cost-insertion one. This is not a surprise, since the regret heuristic has been carefully designed on the basis of the characteristics of a static problem.

As we stated in the introduction, at a preliminary stage we would like to verify if the performance gap among different solution procedures of static problems is unchanged when the problem becomes dynamic. In our formulation the quality of a schedule is given by the number of rejected requests and by the value of the objective function z. The formulation of z is the same as for the static simulations. The relative importance of the components of z related to the total distance, the excess ride time and the idle time is respectively expressed by these three numbers: 0.45, 0.50 and 0.05. However the weights actually being used in the simulation must take into account that the physical units are different for distance and time, so that a scaling factor has been incorporated. The related computational procedure is shown in Diana (2003).

We will express the solution quality in relative terms, comparing the outcome of the static and of the dynamic simulations. This will also allow us to evaluate the cost of not knowing all the information at the beginning of the process. The most important point is that there are various ways to run a dynamic simulation, as described in the previous section, on the basis of Los Angeles datasets. In fact, a given piece of information can be transmitted to the scheduling process in several different ways. Our goal is to assess whether the temporal characterization of the information flow has an influence on the quality of the solution, beyond that of the scheduling algorithm. For this, we define an experimental plan that considers the effect of the following five factors on the number of rejected requests and on the value of the objective function z:

1) the kind of insertion heuristic being used;
2) the percentage p of offline requests;

3) the expectation E(Δt) of the distribution $f(\Delta t)$ of the time intervals between call-in and requested pickup time;
4) the length C of the cycle time;
5) the number v of vehicles in operation.

The influence of the second factor (the percentage of offline requests) has been previously studied by Fu and Xu (2001). There also is a clear relationship between the second and the third factor and the "effective degree of dynamism" as defined by Larsen et al. (2002), whereas to the best of our knowledge the influence of the length of the computational cycle time has never been investigated before. Preliminary statistical analyses have shown that the effect of the four quantitative factors is strongly nonlinear, so that we chose to set them at three different levels that have been fixed as follows.

In the actual transportation service in Los Angeles only a fraction of the requests is known before the beginning of the scheduling process. Hence, we run three different scenarios in which 100, 200 and 400 randomly chosen requests are handed to the heuristic before the scheduling process starts, thus fixing p = 10%, 20% and 40%.

Dessouky and Adam (1998a) derived the distribution $f(\Delta t)$ from real data concerning three days of service operations in Los Angeles (figure 3). In this distribution E(Δt) is 138.9 minutes. It can be seen that the mean time interval is quite long; a DRTS for the general population would probably have requests with a shorter advance notice. Then we run additional scenarios supposing that E(Δt) is halved and reduced to one fourth. This was obtained simply by reducing the width of the classes of Δt from 100 minutes to 50 and 25 minutes respectively, without changing the shape of the distribution.

Fig. 3.

Concerning the cycle time, it is worth pointing out that the planner can freely set this parameter. Hence, we tested the following values of C: 5 minutes, 1 minute and 10 seconds. The first value is somewhat a limit, beyond which it is surely necessary to call the customer back later. The other two are more realistic and could represent the responsiveness of a medium- and high-quality service respectively.

Including the fleet dimension among the factors under control seems not so insightful. It is in fact obvious that it strongly influences the quality of the solution in a positive way. However we are interested in looking whether it is possible to serve all the requests in a dynamic system, given a sufficiently large value of v. The number of vehicles used in the static solutions of the problem v_s is a sort of lower bound, since it is anticipated that the fleet that has been used there is insufficient in a dynamic environment, i.e. some requests would be rejected. The variable v was set at the following levels: $v = v_s$, $v = v_s$ + 10% and $v = v_s$ + 30%.

A full factorial experimental plan would imply running simulations for $2 \cdot 3^4 = 162$ different scenarios, one for each possible combination of levels. However at a preliminary stage we are interested in studying only the main effects of the factors on the quality of the solution. Setting up a true model would require a more focused study on only some of the above five factors and it will be the goal of further research efforts. For this, we adopt a fractional design that will allow us to reduce the number of the considered scenarios to 18. We report in figure 4 a block diagram showing our design, illustrating the parameter settings for each scenario, numbered from C1 to C9 (when the classical metric is used) and from R1 to R9 (when the regret algorithm is used). It can be seen that the effect of the fleet size is monitored through the sequences of scenarios C2-C1-C3 and R2-R1-R3, that of the percentage of offline requests through C1-C4-C5 and R1-R4-R5, that of the mean time interval through C1-C6-C7 and R1-R6-R7 and that of the cycle time length through C1-C8-C9 and R1-R8-R9.

Fig. 4.

For each scenario the five problem instances simulating the paratransit service in Los Angeles have been solved. Since the call-in simulator randomly assigns to each request a call-in time according the distribution of figure 3, if we run the simulation more than once over a given set of requests we will get different results. We made two replications of each sample to keep this aspect into account. To sum up, 10 simulations (2 replications over 5 problem instances) have been run for each scenario, allowing us to statistically analyze the results of 180 different simulations in the next section.

Computational results

Analysis of variance

An analysis of variance has been performed in order to assess the explanatory power of the considered factors concerning both the number of rejected requests and the objective function value. The first step is to identify the factors that have a real influence on these two experimental responses, so at this stage we will not consider the variation of v (scenarios C2, R2, C3 and R3), since as we said its influence is obvious. Then we perform hypothesis tests on the sample constituted by the 14 remaining scenarios (i.e., 140 simulations in total), in order to assess the significance levels of the kind of algorithm, p, $E(\Delta t)$ and C. The results are reported in terms of P-values in table 1. Considering the "number of rejected requests" response (second column of table 1), it turns out that the kind of algorithm, p and $E(\Delta t)$ are three factors that influence the response, whereas C is not significant at a 5% level. This shows the importance of considering the temporal characterization of the information flow in order to optimize the scheduling process of a dynamic system, beyond the attention traditionally paid to algorithmic aspects. On the other hand, none of the considered factors seems to influence the value of the objective function (last column of table 1), so that the quality of the schedule of a dynamic system under the point of view of the combinatorial optimization problem seems to be much less influenced by the algorithm than by other sources of variability.

Table 1.

As we previously mentioned, the effects of the factors over the responses are highly nonlinear and the resulting regression model has an overall fairly low explicative power; hence it is not presented here. In the following paragraphs we will instead make comments on the relationship among solution quality and single effects, that can give a first set of useful policy indications for decision makers. We report in table 2 the averages of the value of the objective function and of the number of rejected requests for the considered scenarios.

Table 2.

Classical versus regret metric

We see from table 2 that the number of rejected requests is lower when we use the minimum incremental cost insertion, except in one case. The value of the objective function is often greater, but we showed that this is not statistically significant. Thus we can overall say that the classical insertion outperforms the regret scheme when the considered problem becomes dynamic. We have a confirmation of the inadequacy of the regret criterion considering what happens when the number of requests that is simultaneously handled to the scheduler is lowered. Considering the sequence of scenarios C1, C8 and C9 and R1, R8 and R9 it can be seen that the performance gap between the two algorithms lessens. This is fully understandable, since the two methodologies become more similar when the algorithm can handle only few requests. These findings contrast with the conclusions of Diana and Dessouky (2004), where the same the static version of the same problem was solved. As a reference, we report in the first line of table 2 the results from the computations of that paper. These results clearly show that the regret algorithm can schedule more requests at a lesser cost when the considered problem is static (*R-static* versus *C-static* scenarios). This confirms the importance of not exclusively focusing on the scheduling algorithm when designing a dynamic DRTS.

We finally point out that our results cannot be generalized as such, i.e. we cannot say that the classical insertion consistently outperforms the regret one when solving dynamic routing and scheduling problems. Our goal is instead to show that the relative performance of different heuristics cannot be taken for granted with reference to a dynamic problem simply on the basis of results that have been validated when studying their corresponding static versions.

Sensitivity to the percentage of offline requests

Comparing scenarios 1, 4 and 5 we conclude that we do not improve the quality of the schedule when we increase the percentage p of offline requests from 10% to 20%. On the contrary, when $p = 40\%$ there is a significant improvement of the solution. It can thus be said that, concerning the efficiency of the scheduling process, if the percentage of offline requests is less than 20%-30%, the problem still basically behaves in a “dynamic” way. The non-monotonicity of the response implies that there is a benefit in knowing in advance a greater number of requests only if p is above a certain threshold. In the considered case, this threshold seems to be quite high, somewhere located between $p = 20\%$ and $p = 40\%$. From a decision maker perspective, the determination of this threshold has a practical relevance, for example in the definition of an effective pricing policy. Economic incentives for customers that book an offline trip should be foreseen only if this information is valuable in terms of an increase of the efficiency of the service schedule. Fu and Xu (2001) study the relationship between *p* and the vehicle productivity (trips/hour) when *p* ranges between 80% and 95% (figure 3 of that paper). Their results seem to confirm that when the problem is "mostly static" decreasing the number of real-time requests can lead to important efficiency gains.

<u>Sensitivity to the time difference between call-in and requested pickup time</u>

Results from table 2 show that there is a strong interaction between E(Δt) and the kind of heuristic, and the analysis of the variance told us that both factors are significant. We believe that this can give us an explanation of the counter-intuitive finding that, when using the classical insertion scheme, the lower E(Δt) the better the solution. More research should be addressed to clarify this interaction. Furthermore, we see here a clear indication of the importance of designing an heuristic that can make a difference between short term and long term events in dynamic environments. Various strategies have been implemented in the past. Not scheduling requests that fall beyond a certain planning horizon is the simplest one, whereas more elaborate approaches make use of periodic reoptimization techniques. From a policy perspective, fare incentives for booking a real-time trip as soon as possible should be considered only if an increase of the value of E(Δt) improves the computational efficiency. We have found that this depends on the capability of the algorithm, whereas on the contrary from the previous paragraph we conclude that the influence of the percentage of offline requests is more dependent on the structure of the problem itself (i.e., demand patterns).

<u>Sensitivity to the length of the cycle time</u>

The analysis of variance told us that *C* is not significant at a 5% level (P = 0.06). Furthermore, different results are obtained when considering the two concurring algorithms (scenarios C1, C8 and C9 versus scenarios R1, R8 and R9). As a consequence of this, it is not a good idea to set up long cycle times, thus lowering the responsiveness of the system and the quality level that is perceived by the users. Hence, a cycle time of only a few seconds is likely to be the best solution, since it does not represent a problem for a customer waiting for a response.

Sensitivity to the fleet size

Using more vehicles to operate the service of course lessens the number of rejected requests, but even incrementing this number by 30% (scenarios C3, R3) is not sufficient to satisfy all the demand, whereas the objective function value z steadily increases, due to the corresponding increase of vehicle miles traveled. Looking at the values of the three components of z, not reported here for briefness, we see that the ridesharing and the increase of ride time are not sensitive to the increase of the fleet size. In other words, from the point of view of the customer, the quality of the service is not changing when more vehicles are used, but the probability of being rejected decreases.

It is worth underlining that even in scenario C3 and R3 only one out of twenty simulations originated a schedule in which all the requests were served. The variability of the response is high, so that dimensioning the fleet in order to serve all the requests with a reasonable degree of confidence is quite a difficult task. This leads to a clear policy indication: to persist in serving all the requests might lead to an abnormally high number of vehicles to be used in certain circumstances. Hence, the utilization of a taxicab service for serving a small portion of requests can be economically justifiable (and perhaps practically unavoidable), even if the marginal cost of serving these would be high. This is generally not the case of static services, the attentive planning of which can generally avoid the use of external resources. For this particular case study, a good solution could be to dimension the fleet as for scenarios C3 and R3 ($v = v_s + 30\%$), and to serve the remaining requests through taxicabs. An economic analysis that keeps into account the cost structure of a DRTS would be needed to find the optimal fleet dimension and the percentage of requests to be served outside the system.

Conclusions

In this paper we defined an evaluation framework for a dynamic DRTS that allowed us to assess the effect of different factors on the efficiency of the scheduling process. On a methodological point of view, a generalization of the insertion feasibility checks proposed by Jaw et al. (1986) has been discussed. Our finding can be summed up as follows.

- The kind of algorithm being used is only one of the factors that can influence the quality of the schedule of a dynamic DRTS and perhaps not the most important. Furthermore, performance gaps among heuristics, that have been detected in solving a static problem, are not guaranteed to stay unchanged when the problem becomes dynamic.

- The relationship between the percentage of offline reservations and the number of rejected requests is not monotonic and is dependent on demand patterns. It follows that fare discounts for early reservations have a positive effect on the scheduling efficiency only if they are determinant to assume a critical percentage of offline requests. This threshold needs to be determined through a market study.
- On the contrary, the benefit of knowing a real-time request sufficiently far in advance basically depends on the capabilities of the algorithm. Pricing policies concerning this point should then be considered together with the performances of the scheduling software. We point out that most of the commercial software implements a classical insertion heuristic. For this kind of algorithm, our results are counterintuitive: smaller advance times lower the number of rejected requests.
- The length of the cycle time does not significantly affect the solution. It seems thus advisable to shorten it as much as possible in order to increase the system responsiveness to customers calls.
- In a dynamic environment trying to serve all the requests by a DRTS can lead to oversized fleets. If policy regulations do not allow for rejections, then it is more convenient to foresee the possibility of outsourcing a fraction of these to taxicabs.

We believe that the above guidelines are only a preliminary finding of a new direction of research that is worth considering more carefully. In particular, future efforts will be aimed at developing predictive models for subsets of the factors here considered. Nonlinearities that have been detected require in fact a more extended experimental study, possibly using datasets from different sites, as well as a wider range of scheduling procedures.

Beyond the attention traditionally paid to the heuristic design, we have shown that, when we face a dynamic problem, issues linked to the temporal characterization of the information flow play a major role. What is even more important, an overall assessment of these "temporal" effects is a powerful tool for defining useful guidelines for planners and decision makers, beyond continuing efforts in improving scheduling heuristics, on crucial issues such as the definition of the most convenient fare structure or the opportunity of outsourcing to taxicabs part of the service operations.

Acknowledgements

This research was made possible by a grant from the Turin Transit Agency ATM (now GTT – Gruppo Torinese Trasporti), Italy. Thanks are also due to Dr. Maged Dessouky of the Department of Industrial and Systems Engineering, University of Southern California, Los Angeles, and to Access Services, Inc. for providing the data upon which the problem instances were built, and to anonymous referees for their helpful comments.

References

Colorni, A., Righini, G., 2001. Modeling and optimizing dynamic dial-a-ride problems. International Transactions in Operational Research 8, 155-166.

Dessouky, M.M., Adam, S., 1998a. Real-time scheduling of demand responsive transit service - Final report. University of Southern California, Department of Industrial and Systems Engineering, Los Angeles.

Dessouky, M.M., Adam, S., 1998b. Real-time scheduling rules for demand responsive transit systems. Proceedings of the 1998 IEEE International Conference on Systems, Man and Cybernetics, New York, 2956-2961 vol. 3.

Dial, R.B., 1995. Autonomous Dial-a-Ride Transit: Introductory overview. Transportation Research 3C, 261-275.

Diana, M., 2003. Methodologies for the tactical and strategic design of large-scale advance-request and real-time demand responsive transit services. Ph.D. Dissertation, Politecnico di Torino, Dipartimento di Idraulica, Trasporti e Infrastrutture Civili, Torino, Italy.

Diana, M., Dessouky, M.M., 2004. A new regret insertion heuristic for solving large-scale dial-a-ride problems with time windows. Transportation Research 38B, 539-557.

Fu, L., 1999a. Improving paratransit scheduling by accounting for dynamic and stochastic variations in travel time. Transportation Research Record 1666, Transportation Research Board, National Academic Press, Washington, D.C., 74-81.

Fu, L., 1999b. On-line and off-line routing and scheduling of dial-a-ride paratransit vehicles. Computer-aided Civil and Infrastructure Engineering 14, 309-319.

Fu, L., 2001. Simulation model for evaluating intelligent paratransit systems. Transportation Research Record 1760, Transportation Research Board, National Academic Press, Washington, D.C., 93-99.

Fu, L., 2002. Scheduling dial-a-ride paratransit under time varying, stochastic congestion. Transportation Research 36B, 485-506.

Fu, L. and Xu, Y. 2001. Potential effects of automatic vehicle location and computer-aided dispatch technology on paratransit performance. Transportation Research Record 1760, Transportation Research Board, National Academic Press, Washington, D.C., 107-113.

Gendreau, M., Guertin, F., Potvin, J.-Y., Taillard, E., 1999. Parallel tabu search for real-time vehicle routing and dispatching. Transportation Science 33, 381-390.

Horn, M.E.T., 2002. Fleet scheduling and dispatching for demand-responsive passenger services. Transportation Research 10C, 35-63.

Ichoua, S., Gendreau. M., Potvin, J.Y., 2000. Diversion issues in real-time vehicle dispatching. Transportation Science 34, 426-438.

Ichoua, S., Gendreau. M., Potvin, J.Y., 2003. Vehicle dispatching with time-dependent travel times. European Journal of Operational Research 144, 379-396.

Jaw, J.J., Odoni, A.R., Psaraftis, H.N., Wilson, N.H.M., 1986. A heuristic algorithm for the multi-vehicle many-to-many advance request dial-a-ride problem. Transportation Research 20B, 243-257.

Jih, W.-r., Hsu, J.Y.-j., 1999. Dynamic vehicle routing using hybrid genetic algorithms. Proceedings of the 1999 IEEE International Conference on Robotics & Automation, Detroit, Michigan, 453-458.

Larsen, A., Madsen, O., Solomon, M., 2002. Partially dynamic vehicle routing – models and algorithms. Journal of the Operational Research Society 53, 637-646.

Madsen, O.B.G., Ravn, H.F., Rygaard, J.M., 1995. A heuristic algorithm for a dial-a-ride problem with time windows, multiple capacities, and multiple objectives. Annals of Operations Research 60, 193-208.

Mitrović-Minić, S. and Laporte, G., 2004. Waiting strategies for the dynamic pickup and delivery problem with time windows. Transportation Research 38B, 635-655.

Mitrović-Minić, S., Krishnamurti, R. and Laporte, 2004. Double-horizon based heuristics for the dynamic pickup and delivery problem with time windows. Transportation Research 38B, 669-685.

Powell, W., 1988. A comparative review of alternative algorithms for the dynamic vehicle allocation problem. Vehicle routing: methods and studies, B.L. Golden, A.A. Assad (eds.). North-Holland, Amsterdam, 249-291.

Powell, W., Jaillet, P., Odoni, A., 1995. Stochastic and dynamic networks and routing. Handbooks in Operations Research and Management Science, vol. 8: Network Routing, M.O. Ball, T.L. Magnanti, C.L. Monma, G.L. Nemhauser (eds.). North-Holland, Amsterdam, 141-295.

Psaraftis, H.N., 1988. Dynamic vehicle routing problems. Vehicle Routing: methods and studies, B.L. Golden, A.A. Assad (eds.). North-Holland, Amsterdam, 223-248.

Psaraftis, H.N., 1995. Dynamic vehicle routing: status and prospects. Annals of Operations Research 61, 143-164.

Regan, A.C., Mahmassani, H.S., Jaillet, P., 1996. Dynamic decision making for commercial fleet operations using real-time information. Transportation Research Record 1537, Transportation Research Board, National Academic Press, Washington, D.C., 91-97.

Savelsbergh, M.W.P., Sol, M., 1998. DRIVE: dynamic routing of independent vehicles. Operations Research 46, 474-490.

Solomon, M.M., 1987. Algorithms for the vehicle routing and scheduling problems with time windows constraints. Operations Research 35, 254-265.

Teodorovich, D., Radivojevic, G., 2000. A fuzzy logic approach to dynamic dial-a-ride problem. Fuzzy Sets and Systems 116, 23-33.

Wilson, N.H.M., Sussman, J.M., Higonnet, B.T. and Goodman, L.A., 1970. Simulation of a computer-aided routing system (CARS). Highway Research Record 318, Highway Research Board, Washington, D.C., 66-76.

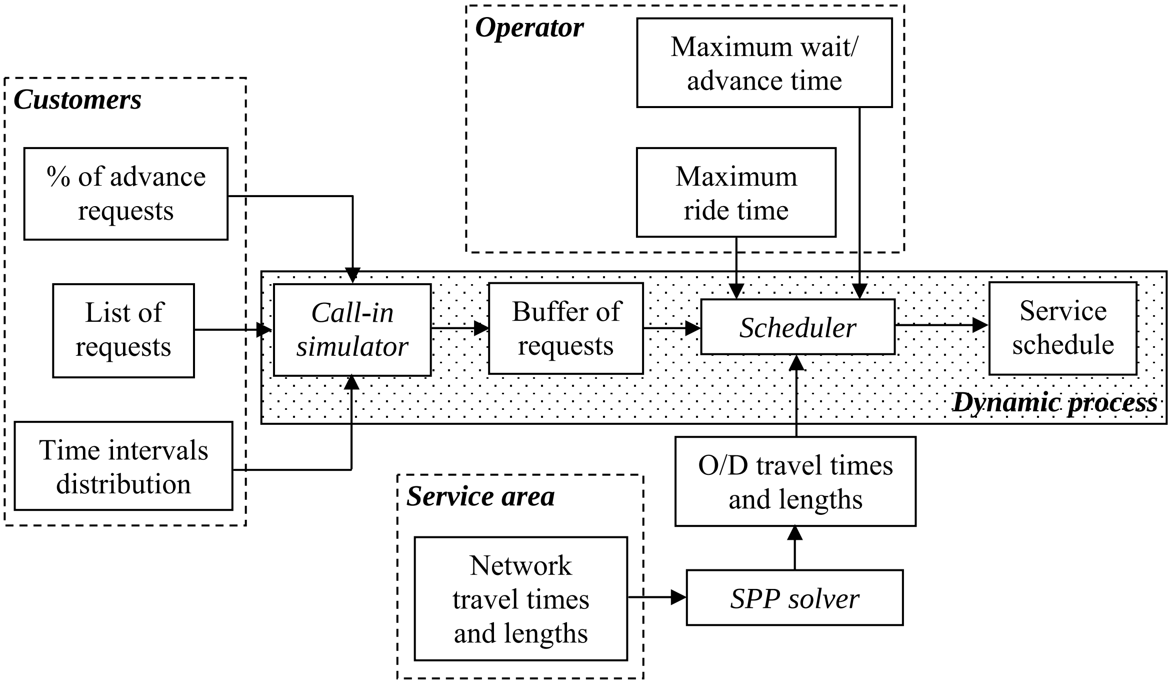

Fig. 1. Flow chart of the simulation process

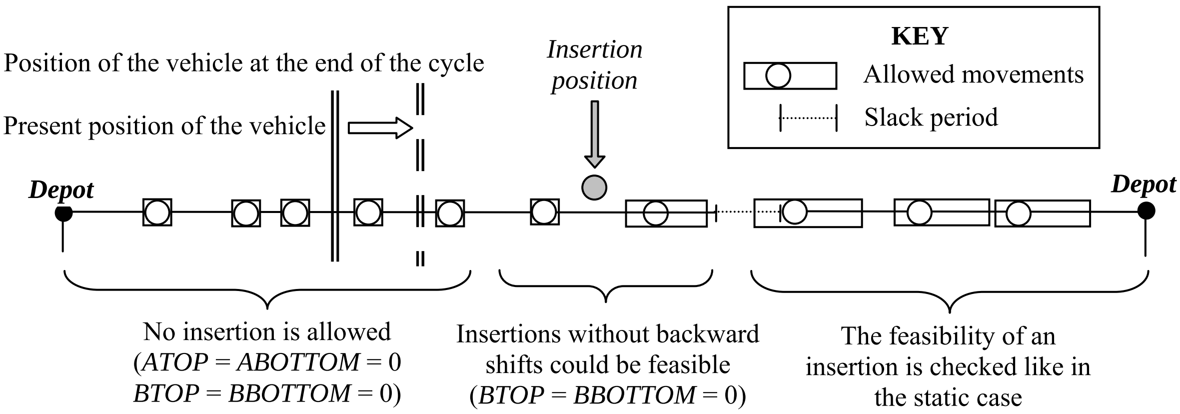

Fig. 2. Dynamic insertion feasibility checks

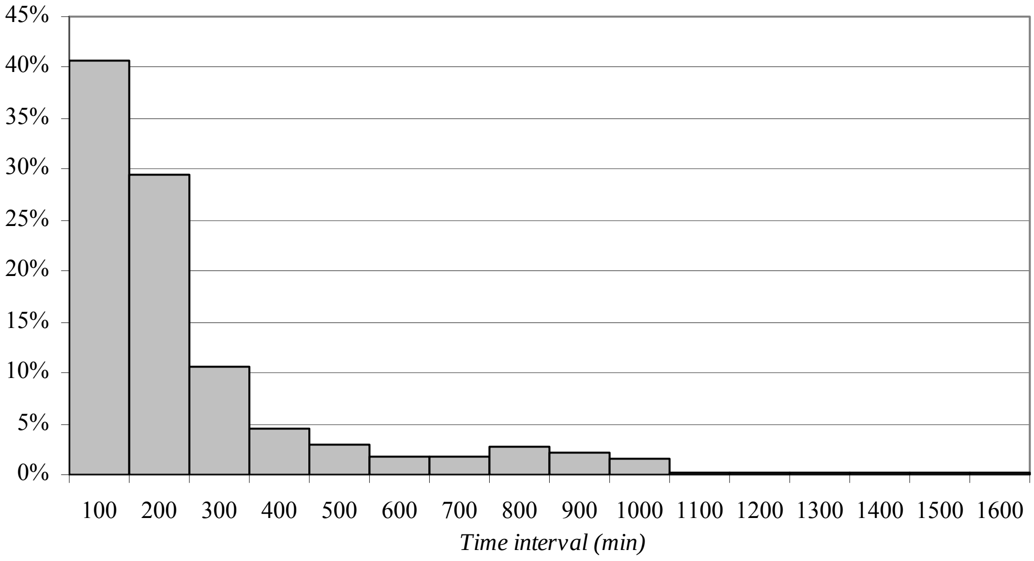

Fig. 3. Distribution of the time intervals between call-in and pickup time

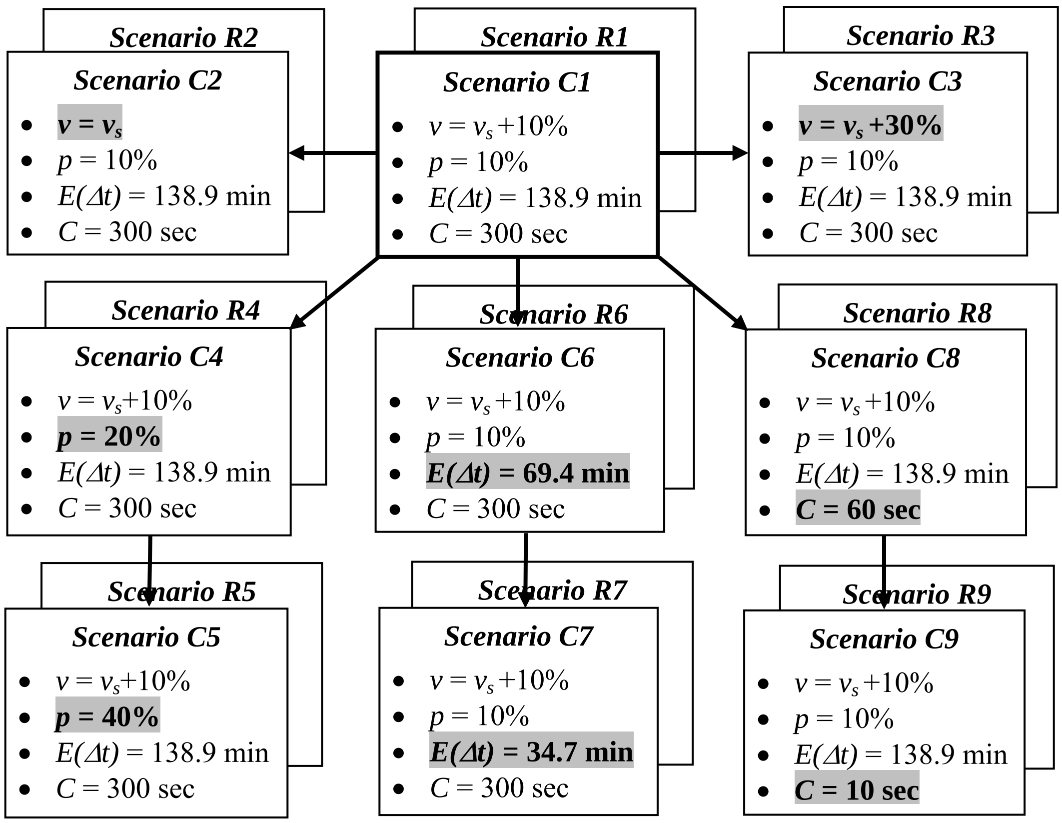

Fig. 4. Design of experiments for the factor analysis

Table 1. Significance levels of the four considered factors (*P*-values)

Factor/Response	Rejected requests	Objective function
Algorithm	<0.0001	0.60
Offline requests	<0.0001	0.14
Advance notice	0.0006	0.96
Cycle time	0.06	0.22

Table 2. Number of rejected requests and objective function values, mean values over two replications of five samples

Scenario	Rejected requests	Objective function	Scenario	Rejected requests	Objective function
C-Static	*7.6*	*19182*	*R-Static*	*0*	*18731*
C1	44.4	19766	R1	71.5	19710
C2	78.0	18564	R2	99.0	18768
C3	10.9	21438	R3	30.6	21361
C4	52.3	19205	R4	75.6	19176
C5	27.3	20247	R5	44.4	19856
C6	38.3	19725	R6	53.0	19946
C7	35.0	19542	R7	76.9	19087
C8	49.8	19808	R8	59.6	19685
C9	51.5	19632	R9	47.0	19979