Innovative systems for the transportation disadvantaged: towards more efficient and operationally usable planning tools

INNOVATIVE SYSTEMS FOR THE TRANSPORTATION DISADVANTAGED: TOWARDS MORE EFFICIENT AND OPERATIONALLY USABLE PLANNING TOOLS

Marco Diana

12th November 2003

Diana, M. (2004) Innovative systems for the transportation disadvantaged: towards more efficient and operationally usable planning tools, *Transportation Planning and Technology*, vol. 27(4), pp. 315-331.

INNOVATIVE SYSTEMS FOR THE TRANSPORTATION DISADVANTAGED: TOWARDS MORE EFFICIENT AND OPERATIONALLY USABLE PLANNING TOOLS

Marco Diana

Institut National de Recherche sur les Transports et leur Sécurité
Département Economie et Sociologie des Transports
2, avenue du Général Malleret-Joinville
F-94114 Arcueil CEDEX – FRANCE
Tel. +33 1 47 40 72 54
Fax +33 1 45 47 56 06
E-mail marco.diana@inrets.fr

ABSTRACT

When considering innovative forms of public transport for specific groups, such as demand responsive services, the true challenge is to find a good balance between operational efficiency and "user friendliness" of the scheduling algorithm even when specialized skills are not available. Regret insertion-based processes have shown their effectiveness in addressing this specific concern. We introduce a new class of hybrid regret measures to better understand why the behaviour of this kind of heuristic is superior to that of other insertion rules. Our analyses show the importance of keeping a good balance between short- and long-term strategies during the solution process. We also use this methodology to investigate the relationship between number of needed vehicles and total distance covered, that is the key point of any cost analysis striving for a greater efficiency. Against expectations, in most cases decreasing the fleet size leads to savings in the vehicle mileage, since the heuristic solution is still far from optimality.

KEYWORDS

Demand responsive services; Routing and scheduling problems; Regret insertion heuristics

1. INTRODUCTION

It is well known that the task of fully meeting the specific needs of the transportation disadvantaged is far from being a simple matter. Most existing public transport systems and infrastructures have been conceived in ages in which the attention paid to these issues was lower, and subsequent actions aimed at putting things right are very expensive and not always fully successful, if at all possible. Given this situation, transport planners are increasingly turning their attention towards the possibility of creating new dedicated services from scratch, that could satisfy this specialized demand segment more effectively and at a lower cost. A well-known example is the passage of the American with Disabilities Act in the U.S. in 1990, that foreshadowed the diffusion of many dedicated demand responsive transport services (DRTS) across the country.

However, experience gained in the latest decade has shown that DRTS themselves can become very expensive for transit agencies. After their implementation, transport demand generally tends to steadily increase beyond any initial forecast, but in absence of skilfully designed technological architectures and management rules no economies of scale are generally possible, and costs increase consequently. The point is to succeed in serving more people with a single vehicle trip when the spatial and temporal demand density increases, thus progressively departing from a taxi-like service exploitation, but this cannot be done without an adequate approach.

In this paper we investigate issues related to the efficient exploitation of a many-to-many DRTS. That is, people can ask to be transported from any point to any other point of the transportation network, and the provider has to design a set of vehicle routes that satisfy all the requests, trying to maximize the economic effectiveness and respecting some quality requirements. This is the crucial passage that determines the overall efficiency of the service, upon which we will focus our attention. The underlying combinatorial optimisation problem is known in literature as Dial-a-Ride Problem[1,2], and presents some features that make it quite difficult to solve. In particular, it is usual practice to define a time window for every pickup and delivery point, i.e. a time interval in which the service must be given. Problems with time windows are particularly hard to solve, and methodologies that are efficient for pure routing problems may perform poorly when applied to this class[3].

Research efforts in past decades contributed in defining a wide range of possible solution methods for the problem of scheduling a DRTS. Among these, the so-called minimum incremental cost (or minimum cost) insertion is by far the most popular when we leave academic ambits and look at what has been implemented on the field. Following this approach, at each step of the scheduling algorithm, the following request according to a pre-determined ranking order (for example, by earliest pickup time) is inserted in the position that causes the minimum increase of the value of the objective function. Although this process cannot ensure the optimality of the solution found, it usually allows to build vehicle schedules of fair quality.

This simple and intuitive idea is currently being used both by researchers, as a kernel of more sophisticate solution approaches, and by practitioners when implementing commercial software. This happens basically because an insertion heuristic can give a sufficiently approximate solution for the optimisation problem in a reasonable and moreover foreseeable amount of time. Of course these solutions can be improved by resorting to more advanced methodologies (local searches or meta-heuristics such as tabu search); these latter however have computational times that usually cannot be easily predicted, and in some cases they have parameters that need to be tuned on the basis of the system we consider. For this, the possibility of their widespread utilization in a more operational context is often confined to particular ambits. The review of such methodologies is beyond the scopes of the present paper, since we are not presenting a competing approach. In the following we will limit ourselves to selectively mention only those works that add insights to our discussion, omitting other major contributions. We refer the interested reader to those papers that offer a more complete review of the state of the art in this field[4,5].

From our previous discussion we can see there is a sort of dichotomy between common scheduling tools that still rely on relatively simple heuristics, and more efficient approaches that are hard to implement in concrete applications. Reducing this gap should become a research priority. For this, we believe it could be of interest to make an effort to improve insertion methods, thus keeping their strengths that make them easily utilizable (intuitive mechanism, constancy of the computational time given the dimension of the problem instance), but trying to improve their myopic behaviour, essentially due to the fact that requests which could be more difficult to insert are not specially considered. Some researchers already considered this aspect, proposing for example procedures that heuristically try to anticipate the insertion of “difficult” requests[6,7].

Another proposed approach is to change the metric, i.e. the way the successive request to be inserted is chosen by the algorithm. Previous work on this issue has often involved the utilization of a metric based on the regret cost. For example, Potvin and Rousseau[8], Kontoravdis and Bard[9] and Liu and Shen[10] applied this methodology for the solution of the standard vehicle routing problem with time windows. Another paper[11] proposed a regret insertion heuristic for the solution of a realistic formulation of the dial-a-ride problem with time windows. In all these works, 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

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

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

Diana and Dessouky[11] carried out a comparison between the minimum cost and the regret insertion, showing the better performance of the latter at the price of an increase of the computational complexity. However their analysis simply assesses the existence of such a performance gap, without further investigating the mechanisms that make this happen. Clarifying this point would of help to define a new class of scheduling algorithms that are as easy to implement as those currently being used, but with a substantial improvement of their performances. In order to pursuit this goal, the following we will further analyse the behaviour of a regret heuristic.

The remainder of the paper is thus organized as follows. In section 2 we better characterize the problem we are going to investigate by presenting its linear programming formulation. Then we describe a numerical experience that can help us to better explain the rationale that lies behind our new hybrid regret measure definition. The latter is then introduced in section 3.2. In section 4 we present computational experiences with the new measure. In particular, we assess whether the efficiency of the regret insertion scheme can be improved by implementing algorithmic variants based on the hybrid measure. Finally, we use these variants to investigate the relationship between fleet dimension and distance covered when we keep the quality of the service constant. For the sake of efficiency, it is in fact of interest for the provider to find the best balance between these two cost components. Finally we draw some conclusions from our research activity and we indicate its possible developments.

2. PROBLEM FORMULATION

In the envisaged DRTS, time windows and service times are associated to both the pickup and the delivery point of any request in order to provide a service of acceptable quality. A request may involve the transportation of more than one passenger; in case some of these needs to be accompanied. The service time is the time interval between the arrival of the vehicle at the service point and its departure, and it must be entirely comprised within the time window. Depending on the kind of service being provided and on the typology of passenger, the service time can range from a few seconds to 1-2 minutes.

The provider has to schedule the service, trying to match the requests in order to maximize the rideshare. The vehicles have a capacity constraint, so than all the passengers will have a seat, and eventually all the wheelchair will be accommodated. Furthermore, the vehicles can eventually stop and idle at any pickup node until they meet the time window of the incoming request, if only no passengers are already onboard. The fleet size is fixed and all the vehicles start and end their journey at the same location (depot). The objective function of the optimisation process is the minimization of the weighted sum of the miles travelled, the passengers' overtime (defined as the difference between the duration of the journey onboard the vehicle and the ride time of a direct trip from origin to destination) and the idle times.

The mathematical programming formulation of this problem has never been published, to the best of our knowledge. However the problem we study is a generalization of the Pickup and Delivery Problem with Time Windows, that was introduced and defined by Dumas et al.[12]. As we said, in our case we have an additional constraint that helps in realistically modelling a people transportation service. We force the vehicles not to idle if there are customers onboard. We name this a *selective no-idling constraint*, since only on a subset of the nodes of the network the vehicles cannot stop and wait. This subset depends on the schedule and cannot be determined a priori. This constraint was originally proposed by Jaw et al.[2], that applied it in modelling a real paratransit service.

Our objective function is also somewhat different from those usually being used in solving this class of problems, since many researches focus on goods transportation services in which simply the minimization of the vehicle fleet and/or of the distance covered is needed. On the contrary, the exploitation of transit services must keep into account the satisfaction of the passengers. Our effort has been aimed at modelling this aspect through a linear expression, so that linear programming solution techniques could eventually be applied. It is however worth mentioning that non-linear objective functions are generally used in order to better represent the customer satisfaction of public transportation services[2,13].

Beyond the aforementioned work of Dumas et al.[12], a mathematical programming formulation of the Pickup and Delivery Problem can be found in Desrosiers et al.[14], in which the multiple depot and heterogeneous fleet case is presented. The formulation of our problem is based on these, with the additional selective no-idling constraint. Another possible approach is that proposed by Lu and Dessouky[15], in which the index related to the vehicle number is not required for binary arc flow variables.

We have to serve n requests, indexed by i. Every request has a pickup node i and a delivery node $n+i$, and the depot is the first and the last point of each vehicle route, so that it will be labelled as 0 and $2n+1$. Of course with this notation different nodes of the schedule may correspond with the same physical location. Therefore, the node set of the network is $N = \{0, 1, ..., n, n+1, ..., 2n, 2n+1\}$. We also define the pickup node set $P^+ = \{1, 2, ..., n\}$, the delivery node set $P^- = \{n+1, n+2, ..., 2n\}$ and the service node set $P = P^+ \cup P^-$.

Request i involves the transportation of u_i people. Each node i in P has a time window $[ET_i, LT_i]$ and a service time s_i. We have a set of vehicles $V = \{1, 2, ..., |V|\}$ indexed by k. Each vehicle can carry a maximum number of persons CAP, excluding the driver. For each distinct node i and j in N, let tt_{ij} and td_{ij} represent respectively the travel time and the distance to go from i to j. Finally, we have the tern a, b and c of weights of the components of the objective function.

We use four sets of decision variables in our formulation. Flow variables X_{ijk}, $i, j \in N$, $k \in V$, are equal to one if vehicle k travels from node i to node j, and equal to zero otherwise. Time variables AT_i, $i \in P$, represent the time at which the vehicle leaves node i after having finished its service. Hence, the vehicle arrives at node i at $AT_i - s_i$. Slack variables D_i, $i \in P$, represent the idle time of the vehicle at node i. Finally, load variables Y_i, $i \in P$, represent the number of passengers that are onboard the vehicle while it is approaching node i.

The mathematical programming formulation of our problem is thus the following one:

$$\min \left[a \cdot \left(\sum_{i \in N} \sum_{j \in N} \sum_{k \in V} td_{ij} X_{ijk} \right) + b \cdot \left(\sum_{i \in P^+} \left(u_i \cdot \left(AT_{n+i} - AT_i \right) \right) \right) + c \cdot \left(\sum_{i \in P} D_i \right) \right] \qquad (2)$$

subject to

$$\sum_{j \in P \cup \{2n+1\}} \sum_{k \in V} X_{ijk} = 1 \qquad \forall i \in P^+ \qquad (3)$$

$$\sum_{i \in P \cup \{0\}} X_{ijk} - \sum_{i \in P \cup \{2n+1\}} X_{jik} = 0 \qquad \forall j \in P \quad \forall k \in V \qquad (4)$$

$$\sum_{j \in P^+ \cup \{2n+1\}} X_{0jk} = 1 \qquad \forall k \in V \qquad (5)$$

$$\sum_{i \in P^- \cup \{0\}} X_{i,2n+1,k} = 1 \qquad \forall k \in V \qquad (6)$$

$$\sum_{j \in P} X_{ijk} - \sum_{j \in P} X_{j,n+i,k} = 0 \qquad \forall i \in P^+ \quad \forall k \in V \qquad (7)$$

$$AT_i + tt_{i,n+i} \le AT_{n+i} - s_{n+i} - D_{n+i} \quad \forall i \in P^+ \qquad (8)$$

$$X_{ijk} = 1 \;\; \Rightarrow \;\; AT_i + tt_{ij} = AT_j - s_j - D_j \quad \forall i, j \in P \quad \forall k \in V \qquad (9)$$

$$ET_i + s_i \le AT_i \le LT_i \quad \forall i \in P \qquad (12)$$

$$Y_i > 0 \;\; \Rightarrow \;\; D_i = 0 \quad \forall i \in P \qquad (11)$$

$$X_{ijk} = 1 \;\; \Rightarrow \;\; Y_i + u_i = Y_j \quad \forall i \in P^+ \quad \forall j \in P \quad \forall k \in V \qquad (12)$$

$$X_{ijk} = 1 \;\; \Rightarrow \;\; Y_i - u_{i-n} = Y_j \quad \forall i \in P^- \quad \forall j \in P \quad \forall k \in V \qquad (13)$$

$$X_{0jk} = 1 \;\; \Rightarrow \;\; Y_j = 0 \quad \forall j \in P^+ \quad \forall k \in V \qquad (14)$$

$$X_{ijk} = 1 \;\; \Rightarrow \;\; Y_j \ge u_i \quad \forall i \in P^+ \quad \forall j \in P \quad \forall k \in V \qquad (15)$$

$$Y_i + u_i \le CAP \quad \forall i \in P^+ \qquad (16)$$

$$X_{ijk} \in \{0,1\} \qquad \forall i, j \in N \quad \forall k \in V \tag{17}$$

$$D_i \geq 0 \qquad \forall i \in P \tag{18}$$

It can be seen that the objective function of our problem is a linear expression, and also all the constraints can be put in a linear form. The proof that the second addend of equation (2) minimizes the overtime for all the passengers is straightforward[16]. Constraints (3) through (7) ensure the correctness of vehicle routes, whereas constraints (8) and (9) force variables AT_i to be coherent with the travel times, the service times and the slack periods. Constraints (10) and (11) are the time windows and the selective no-idling constraints. Constraints (12) through (15) concern the number of passengers onboard at each node and constraints (16) ensure the respect of the capacity limit.

3. METHODOLOGICAL APPROACH

3.1. Comparisons Between Minimum Cost And Regret Insertion Procedures

As we mentioned above, a more thorough comparison between minimum cost and regret insertion is needed in order to gain further insights concerning the reasons of the better performance of the regret scheme. As a first step, we can notice that when we use the regret heuristic to solve the same problem instances with the number of vehicles that are needed by the minimum cost scheme, it is possible to directly compare the two solutions in terms of objective function values, since the underlying combinatorial optimisation problem, as defined in section 2, becomes the same. In fact, it can be seen that the number of vehicles is not a decision variable in the above introduced problem formulation, but it has an obvious influence on the number of constraints and on the first term of the objective function (2).

In table 1 we sum up the results of this preliminary analysis, using data of the system actually in operation in Los Angeles County to serve the transportation disadvantaged. It can be seen that the improvement of the objective function value is of about 4%. This is still a significant gain, although not so impressive as the one found in Diana and Dessouky[11] when minimizing the fleet dimension, that was of about 8%. Hence, we can say that the regret scheme is relatively more efficient in decreasing the number of needed vehicles and of travelled miles, than in optimising the problem, given our objective function as we defined it. If we compare the values of the three components of the objective function, we can see that the first term, related to the mileage, shows the percentage reduction that we anticipated. The second one instead has a decisive increment, thus signifying that customers would have to spend more time onboard. It is thus clear that the regret algorithm does not improve the schedule on the customers' point of view, even if the quality of the service is a component of the objective function. On the other hand, the decrease of the mileage and the increase of the user's overtime causes an increase of the rideshare, that in turn would benefit the economic effectiveness of the service.

Table 1

3.2. Hybrid Regret Measure Definition

Given the peculiar behaviour of the regret heuristic that has been shown in the preceding subsection, the idea is to gain further insights by modifying the definition of the metric, i.e. equation (1), thus keeping the steps of the algorithm unchanged. From the previously discussed results, it can be seen that the regret metric has a less myopic behaviour than the minimum cost algorithm, but in turn it is not so good in avoiding insertions that worsen the objective function value. This seems also to be confirmed by the fact that when solving less constrained versions of the problem, in which the myopic behaviour of a greedy heuristic is less harmful, the efficiency gain in using the regret scheme is diminished.

Stemming from this consideration, our proposal is to use a hybrid methodology between the minimum cost insertion and the regret measure, trying to catch the best of each. Thus we would like to set a good long-term scheduling strategy, contemporarily avoiding requests insertions that could worsen too much the objective function value. In order to achieve both these results, for a generic request *i* we define the following hybrid regret cost h_i:

$$h_i = \sum_{j^*=1}^{n} \left[\left(\frac{n - j^* + 1}{n-1} \right)^p \left(c_{ij} - c_i^* \right) \right] - q \cdot c_i^* \tag{19}$$

and we use it in the above described regret algorithm, instead of the regret cost r_i as defined in equation (1). In the following we will name r_i the classical regret measure, as opposed to the hybrid regret h_i. In equation (19) p and q are user-specified parameters. It can be seen that if $p = q = 0$ then $h_i = r_i$.

The intuitive meaning of the above formula is the following. The second term keeps into account the minimum insertion cost c_i*, and we change its sign since the request to be inserted is the one with the largest value of h_i. The first term is a modification of the classical regret cost definition, in which the different addends of the sum are weighted in a different way. Every time we consider a request, we sort the respective insertion costs in ascending order. We label j* the index of this order, that is used in equation (19). It can thus be seen that every addend of the sum is multiplied by a term that is smaller as j* increases.

Let us explain the rationale of this approach through an example. If a request i can be inserted in three different vehicle routes, we will have three insertion costs c_{i1}, c_{i2} and c_{i3}. Let us suppose that the first one is the minimum, so that $c_{i1} = c_i$*, and that the last one is the maximum. We also take $p = q = 1$ for simplicity. When we apply equation (19), we have two terms of the sum that are greater than zero; the first one is multiplied by 1 and the second one by 0.5. That is, in the computation of the regret cost, the term $c_{i3} – c_i$* counts for half in comparison with the term $c_{i2} – c_i$*. In fact, if request i is not inserted at this stage, it is likely that it can still be inserted in the following one at a cost that is nearer c_{i2} than c_i* or c_{i3}. This is of course not a deterministic rule, since the insertion costs at successive stages will depend on the insertion that we are going to perform. Our guess simply tries to model the fact that when we compute the regret of not inserting now a request, we should give less importance to the worsening of the objective function due to insertions that would probably take place in a more remote future.

Parameters p and q in the above equation allow us to tune the sensitivity of the hybrid regret metric in relation with the aforementioned features. In particular, when we increase the value of p we give more importance to candidate near-term insertion over more remote moves, whereas when q is increasing more attention is paid in avoiding moves that could worsen too much the objective function value.

4. COMPUTATIONAL EXPERIENCE OUTCOMES

4.1. Comparisons Between Classical Regret And Hybrid Regret Insertion

In order to assess our approach, we coded a parallel insertion algorithm that uses the hybrid regret cost as defined in equation (19). Then we solved five problems involving 1000 requests, that are representative of the paratransit service for elderly and disabled people that is actually in operation in Los Angeles County. These instances are those previously used by Diana and Dessouky[11]. We solved several times these five problems with the hybrid regret algorithm, using different values of parameters *p* and *q* in order to see if there are settings that outperform all the others. More specifically, we tested three values for *p* (0, 1 and 10) and three values for *q* (0, 1 and *n*, where *n* is the number of vehicles), thus exploring nine different regret metrics, including the classical one that as aforementioned is obtained when we have $p = q = 0$. We report the results of these computations in table 2. The first line of the table shows the results we have when we use the classical regret, whereas the second reports the best solution that has been found with the hybrid regret, together with the corresponding values of *p* and *q*.

Table 2

It can be seen that the hybrid regret algorithm improved the solution in four out of five cases. Moreover, in three of these it was even possible to serve all the requests with a smaller number of vehicles, contemporarily lowering the total travelled miles. We will come back to this very important result in the next subsection. Concerning the fleet dimension, there is not a unique value of *p* and *q* that consistently gives the best results; however, there are some combinations of values of the two parameters that behave slightly better than others on average. We graphically show this in figure 1, where we report the difference between the minimum number of vehicles needed by the classical regret insertion and the minimum number of vehicles needed by each of the other insertion methods, for each of the five considered problem instances. Thus, it can be seen that the combinations of (*p*, *q*) that give the best results are (1, 1) and (1, *n*), if we focus our attention on the minimization of the fleet dimension (which is not considered in our objective function). On average, the improvement over the classical regret has been respectively of 0.4 and 0.6 vehicles.

Fig. 1

Since the variability of this datum across the five samples is marked, and the gains over the classical insertion not so high in relative terms (although not irrelevant), this should only be considered as an indicative result. Assuming that the populations of the minimum number of vehicles needed when using different regret metrics have the same variance, we inferred the difference in the mean values. On the basis of the observed data, statistical analysis allows us to say that the (1, *n*) metric improves over the classical one with a degree of confidence of about 75%, whereas the same conclusion for the (1, 1) metric can be drawn only with a very low degree of confidence (about 60%). These confidence intervals could be improved by running more simulations, but we recall that our point is to explore the capabilities of the regret metric, more than proposing an adoption of our methodology “as it is”. In any case, on the basis of these preliminary tests, on an operational point of view we can already say that if the fleet minimization is our primary concern (e.g. because of an unexpected shortage of vehicles), then the utilization of a hybrid regret metric with $p = 1$ is likely to decrease the probability of rejecting some requests.

Focusing on the objective function minimization, we run additional simulations using the same number of vehicles for all the considered algorithmic variants. For each instance, the dimension of the fleet is the minimum one that allowed to find a feasible solution by using all the nine metrics. The results are reported in figure 2, where we show the difference between the objective function value when using the eight hybrid regret metrics and the value obtained with the classical one. In these experiments, the classical regret behaved slightly better than the other algorithms. It can however be seen that the gaps among the different metrics are practically irrelevant, the mean values over five samples being really near each other. Moreover, in none of the five replications the classical regret reported the best minimum value; in other words, the variance of its solutions is smaller, so that the performance of the classical regret algorithm is more constant concerning the value of the objective function.

Fig. 2

4.2. Relationships Between Number Of Vehicles And Distance Covered

Beyond the obvious interest in improving the performance of the classical regret metric, our computational experiments can also give a contribution to shed some light on another issue, i.e. the relationship between fleet dimension and efficiency of the scheduling process in terms of objective function value. We will focus our attention on the first component of our function, that is the total length of the vehicles routes. This is because the operator could theoretically provide a service of a certain quality using different combinations of fleet dimension vs. total distance covered. It is thus interesting to explore the nature of the relationship between these two quantities in our problem, since it would be possible to choose the most efficient solution on the basis of the cost structure of the enterprise, not changing the quality of the service that is provided.

Given our problem formulation in section 2, we would expect that when we lower the number of vehicles in operation, the distance covered increases in order to serve all the requests. This should happen until the fleet dimension becomes insufficient and we cannot find a feasible solution (i.e. some service requests are rejected by the system). In order to verify how things are going in reality, we can compare the distance covered in the previously described simulations, whose results are represented in figures 1 and 2. We recall that in the former we used the minimum number of vehicles that originates a feasible solution within each metric, whereas in the latter we used the minimum number of vehicles that can generate a feasible solution with all the nine metrics. Thus, the fleet dimension in the former case is always smaller, or at least equal, to that in the latter.

In this way, it has been possible to build five plots, one for each problem instance, that show the relationship between fleet dimension and miles travelled when using different algorithms. We report in figure 3 two of these plots, that represent the two distinct trends that have been observed. Plot 3a shows that, when using more vehicles in the simulation, the total mileage increased in all the considered regret metrics. On the contrary, in plot 3b we can see that there are five metrics that give a better solution when we increase the fleet dimension. Here it is clear that there are two distinct groups which behave quite homogeneously (the segments in figure are nearly parallel). Moreover, it can be seen that the five abovementioned metrics are those with $q = 0$ and $q = 1$, whereas the other three have q equal the number of used vehicles. So, we can overall say that there is not a unique trend, even if all the samples are taken from the same population and all the heuristics are similar to each other. It is however predominant the behaviour that goes against our intuition, i.e. in most cases the distance covered increases when we increase the dimension of the fleet.

Fig. 3a,b

We believe that this result is due to the fact that our solution methodologies are heuristic, that is more or less far from optimality. It is thus possible to think that the degree of efficiency with which we solve our problem is not constant, but that as we lower the fleet dimension, thus narrowing the solutions space, the algorithm is “forced” to come closer to optimality in order to find a feasible solution. The outcome of this analysis seems to be that the idea of lowering the number of vehicles in activity in order to decrease the mileage is not good, at least when we use a scheduling algorithm that is really not much sophisticate.

5. CONCLUSIONS

In this paper we studied a generalization of the Pickup and Delivery Problem with time windows, in which service times are greater than zero and vehicles are allowed to idle if only no passengers are onboard. These features allow to model a passenger transport service in a more effective and realistic way. This problem class is well known for being quite hard to solve. On the other hand, in operational environments it is often necessary to quickly find solutions for big instances. There is thus a need for solution approaches that represent a good compromise between the minimum cost insertion, currently being the core of most routing and scheduling software, and more effective but slower and hardly adaptable methodologies that have been proposed in scientific literature.

We explored the possibility of addressing this issue by using an insertion algorithm in which the criterion for choosing the request to be inserted is different from the minimization of insertion cost. Previous work has shown the effectiveness of the regret metric in order to achieve this result. In this paper, we further investigated its potentialities, trying to see why the regret metric works better. In order to do that, we generalized its definition through the introduction of a hybrid regret measure. Computational experiments have shown that the hybrid regret is a promising way of improving the effectiveness of an insertion algorithm without worsening the computational burden, since it is possible to decrease the dimension of the fleet. The key point to fully exploit the potentialities of the regret metric has been shown to be the definition of the best balance between the need of avoiding bad insertions and that of building a long-term strategy in the solution process.

As a further application of our analysis tool, the implementation of several variants of the regret metric allowed us to shed some light on another issue of relevant practical interest, that is the trade-off between number of vehicles being used and distance covered. The results show that when using greedy heuristics that work more or less far from optimality, such as those usually embedded in commercial software packages, increasing the number of vehicles in order to limit the travels is not recommendable, since there is a loss of efficiency in the scheduling process. Hence, the best policy seems always to consist in keeping the minimum number of vehicles in activity, even if the minimization of the variable costs is our primary concern (for example, because resources such as vans and drivers have already been deployed in a previous planning phase).

On the basis of our analyses, we have come to the conclusion that there is room for improving the scheduling algorithm of software applications designed for the operation of innovative transport services, without necessarily having to implement burdensome methodologies that are often more designed for research purposes, or calibrated in specific contexts. A widespread efficiency increase of the scheduling processes currently in use would help overcome financial problems that those dedicated transport systems are suffering in many different countries, thus averting the risk of retrenchment of services that have proven their efficacy in meeting the needs of specific social groups.

REFERENCES

[1] Desrosiers, J., Dumas, Y. and Soumis, F. *The multiple dial-a-ride problem*. Lecture Notes in Economics and Mathematical Systems 308: Computer-Aided Transit Scheduling. Springer-Verlag, Heidelberg, 1988.

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

[3] M.M. Solomon, *Algorithms for the vehicle routing and scheduling problems with time windows constraints*. Operations Research 35, 254-265 (1987).

[4] Desaulniers, G., Desrosiers, J., Erdmann, A., Solomon, M.M. and Soumis, F., *The VRP with pickup and delivery*. Cahiers du GERAD G-2000-25, Ecole des Hautes Etudes Commerciales, Montréal, 2000.

[5] Savelsbergh, M.W.P. and Sol, M., *The general pickup and delivery problem*. Transportation Science 29, 17-29 (1995).

[6] W.-C. Chiang and R.A. Russell, *Simulated annealing metaheuristics for the vehicle routing problem with time windows*. Annals of Operations Research 63, 3-27 (1996).

[7] P. Toth and D. Vigo, *Heuristic algorithms for the Handicapped Persons Transportation Problem*. Transportation Science 31, 60-71 (1997).

[8] J.Y. Potvin and J.M. Rousseau, *A parallel route building algorithm for the vehicle routing and scheduling problem with time windows*. European Journal of Operational Research 66, 331-340 (1993).

[9] G. Kontoravdis and J.F. Bard, *A GRASP for the vehicle routing and scheduling problem with time windows*. ORSA Journal of Computing 7, 10-23 (1995).

[10] F.H. Liu and S.Y. Shen, *A route-neighborhood-based metaheuristic for vehicle routing problem with time windows*. European Journal of Operational Research 118, 485-504 (1999).

[11] M. Diana and M. Dessouky, *A new regret insertion heuristic for solving large-scale dial-a-ride problems with time windows*. Forthcoming on Transportation Research B, (2004).

[12] Y. Dumas, J. Desrosiers and F. Soumis, *The pickup and delivery problem with time windows*. European Journal of Operational Research 54, 7-22 (1991).

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

[14] J. Desrosiers, Y. Dumas, M.M. Solomon and F. Soumis, *Time constrained routing and scheduling*, in Handbooks in Operations Research and Management Science, vol. 8: Network Routing. M.O. Ball, T.L. Magnanti, C.L. Monma and G.L. Nemhauser (Editors), North-Holland, Amsterdam, 35-139, 1995.

[15] Q. Lu and M. Dessouky, *An exact algorithm for the multiple vehicle pickup and delivery problem*. Forthcoming on Transportation Science (2003).

[16] M. Diana, *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, 2003.

LIST OF FIGURES

LIST OF TABLES

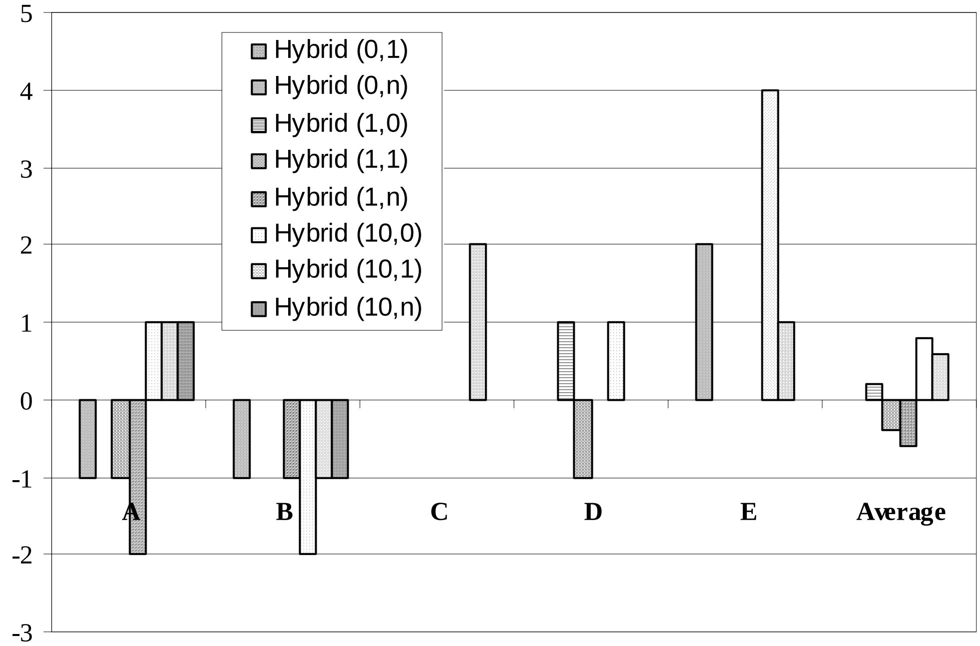


Fig. 1. Differences in the number of vehicles needed in comparison with the classical regret insertion

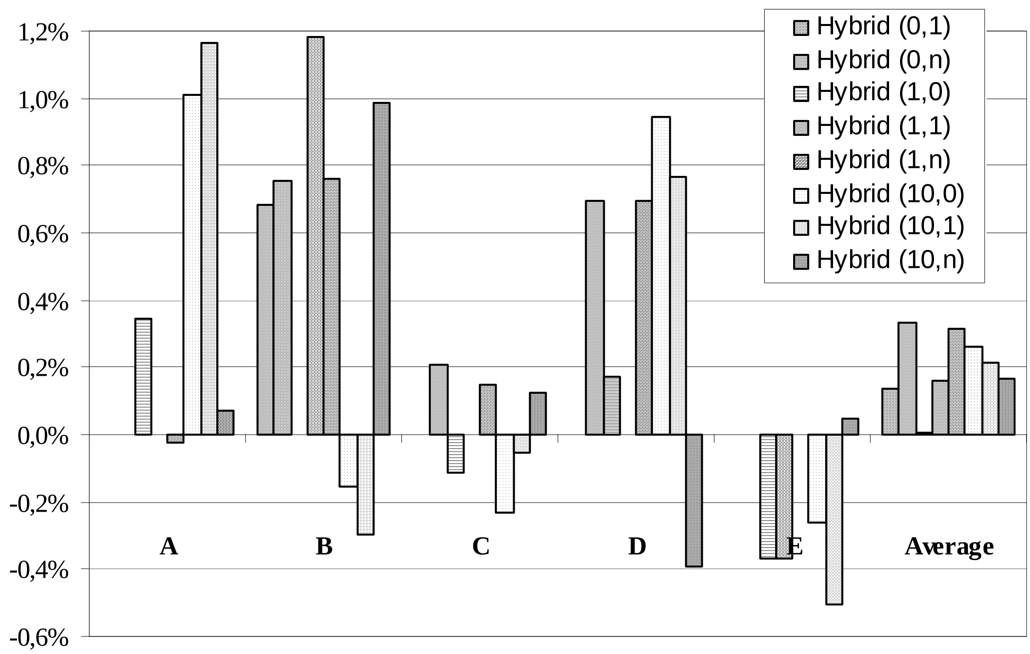
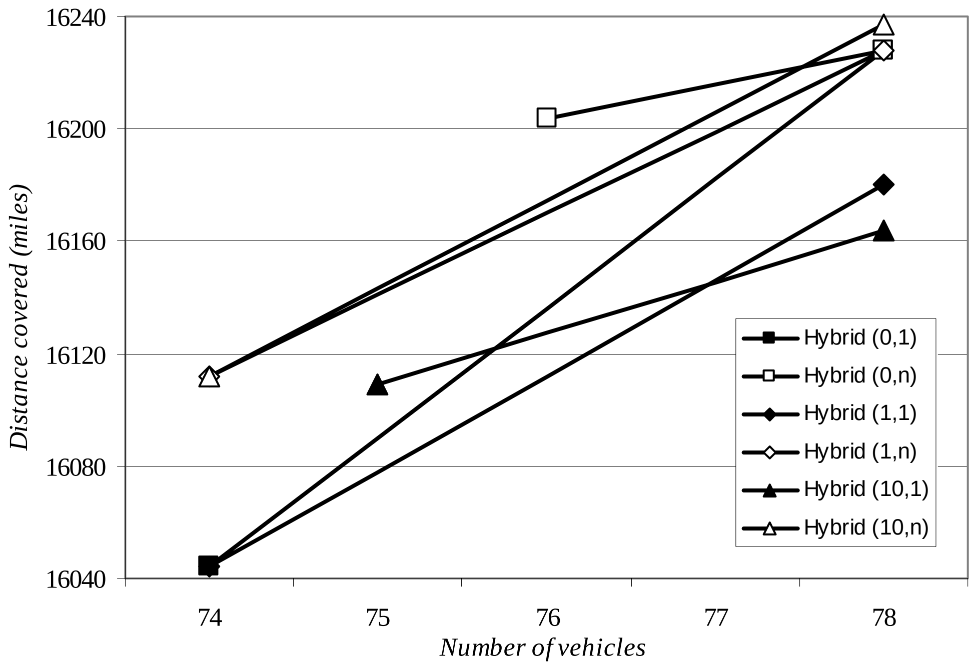
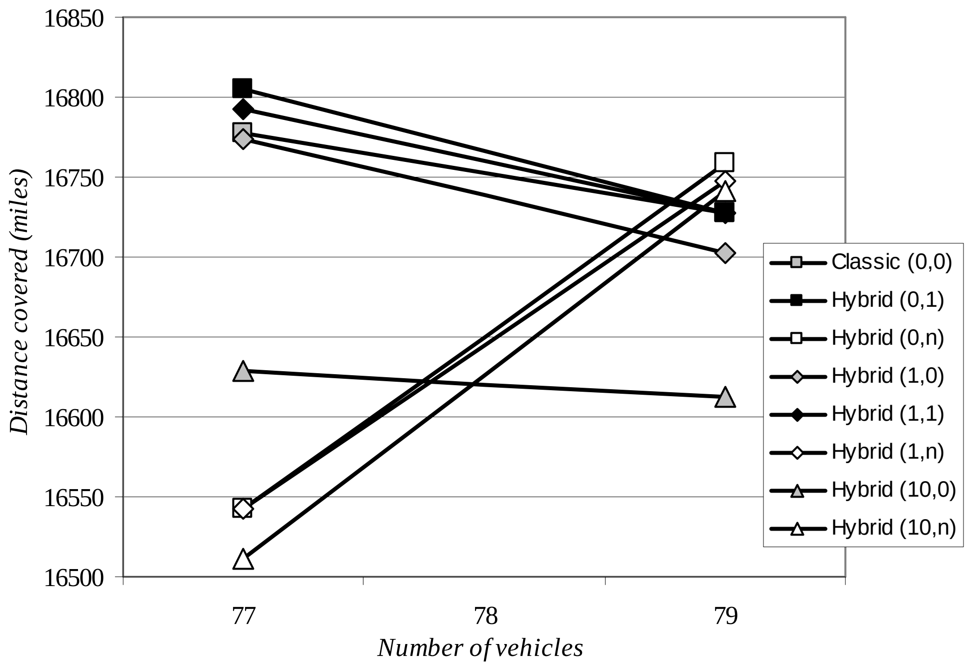


Fig. 2. Differences in the objective function value in comparison with the classical regret insertion



(a)



(b)

Fig. 3. Relationship between number of vehicles and distance covered when using different regret metrics. (a) First and (b) second problem instance

Table 1. Mean values of the objective function and of its three components for the minimum cost and the regret insertion algorithm

Algorithm	Objective function	Components		
		1st	2nd	3rd
Min Cost	19588	17728	1194	666
Regret	18796 (-4.04%)	16559 (-6.59%)	1549 (+29.73%)	688 (+3.29%)

Table 2. Computational results of the regret insertion algorithms

Regret algorithm	Samples				
	A	B	C	D	E
Classical	78[a] 17108[b]	80 17039	77 16778	73 16247	74 16044
Hybrid	76[a] 17099[b] (1,76)[c]	78 16871 (10,0)	77 16511 (10,77)	72 16228 (1,1)	74 16044 (0,0)

[a] Number of vehicles
[b] Total miles travelled
[c] Value of parameters (p, q)