Diversion Issues in Real-Time Vehicle Dispatching

Soumia Ichoua[1,2], Michel Gendreau[1,2]
and Jean-Yves Potvin[1,2]

1 Centre de recherche sur les transports. Université de Montréal, C.P. 6128, succ. Centre-ville, Montréal, Québec, Canada H3C 3J7.

2 Département d'informatique et de recherche opérationnelle, Université de Montréal, C.P. 6128, succ. Centre-ville, Montréal, Québec, Canada H3C 3J7.

Abstract. Recent technological advances in communication systems now allow the exploitation of real-time information for dynamic vehicle routing and scheduling. It is possible, in particular, to consider diverting a vehicle away from its current destination in response to a new customer request. In this paper, a strategy for assigning customer requests that includes diversion is proposed and various issues related to it are presented. An empirical evaluation of the proposed approach is performed within a previously reported tabu search heuristic. Simulations compare the tabu search heuristic, with and without the new strategy, on a dynamic problem motivated from a courier service application. The results demonstrate the potential savings that can be obtained through the application of the proposed approach. In particular, a reduction in the objective value that ranges from 0.8 % to 4.3 % is observed.

1   Introduction

In the past few years, there has been a rapid growth in communication and information technologies (e.g., global positioning satellites, cellular phones, geographic information systems, geosynchronous satellite-based systems, etc.). These recent advances afford opportunities for using real-time information to enhance the performance of decision systems in the area of vehicle routing. According to the fraction of requests that are known in advance, vehicle routing problems may be classified as static or dynamic. In the static case, all data are known before the routes are constructed and do not change afterwards (e.g., location of transportation requests, demand, etc...). However, in the dynamic case, all or a fraction of all requests are revealed as the routes are executed. Hence, dispatchers are forced to react to events that occur in real time, such as new service requests, unexpected delays, accidents, etc... Numerous examples of dynamic vehicle routing and dispatching problems may be found in practice, like ambulance or police services, courier services, dial-a-ride problems (e.g., transportation-on-demand for the handicapped) and many others.

Due to the new information technologies mentioned above, real-time information is now available at lower costs. This explains the new interest that dynamic vehicle routing problems have gained recently. However, there is still a lack of methodologies that can efficiently solve dynamic vehicle routing problems through a judicious integration of real-time information.

In dynamic vehicle routing problems, one potential use of real-time information is to divert a vehicle away from its current destination to serve a request that just occurred in the vicinity of its current position. In this work, we propose an approach for the dynamic assignment of new requests that includes diversion and we examine different ways of implementing it. An empirical evaluation is performed within the tabu search heuristic reported in Gendreau et al. (96b).

The problem considered is motivated from a courier service application found in the local operations of long-distance shipping services, where the mail is collected at different customers' locations and brought back to a central office for further

processing and shipping. This problem belongs to the class of pick up (or delivery) only problems where a set of requests must be transported to (or from) a central depot. The goal is to design a set of minimum cost routes, originating and ending at the central depot, that satisfy the transportation requests. In a dynamic context, each new request is inserted in real time in the current set of planned routes, where a planned route is the sequence of requests that have been assigned to a vehicle but have not been serviced yet.

The paper is organized as follows. Section 2 presents a brief literature review dedicated to dynamic vehicle routing problems. Then, section 3 describes the problem considered in this study. Section 4 briefly introduces the original tabu search heuristic reported in (Gendreau et al. (96b)) that we subsequently modified to explore the benefits of a new dynamic assignment strategy. Section 5 presents and discusses several issues related to it. Section 6 explains how the original tabu search heuristic was modified to include it. Section 7 reports computational results based on different ways of implementing the proposed strategy within the tabu search heuristic. Finally, section 8 summarizes our findings.

2   Literature review

The earliest papers in the literature on dynamic vehicle routing and dispatching were presented in the seventies. Wilson was one of the first researchers interested in dynamic vehicle problems and his studies were mostly application-oriented. In 1977, Wilson et al. (77) developed a system for allocating passengers to buses in a dial-a-ride environment. A new incoming passenger was inserted at least cost among all feasible insertion places in the bus routes. Daganzo (78) and Stein (78) developed mathematic models to evaluate the performance of dynamic transportation systems and to provide analytic and parametric studies (without the need for a simulation, which was very expensive at this time).

By the end of the eighties, dynamic vehicle routing gained an increasing attention. Two major factors explain this tendency : new developments in information technologies and the need for decision systems that could exploit this information to better represent the real world. Interesting survey articles on dynamic vehicle routing can be found in Psaraftis (88, 95), Powell (95) and Lund (96).

Because real-time vehicle routing problems are NP-hard and quick response times are required, exact algorithms are not yet capable of handling problems of realistic sizes (Psaraftis (80, 83, 85), Dial (95)). This justifies the use of heuristics in real-time environments. According to how they deal with the dynamic aspect of the problem, the problem solving approaches reported in the literature can be classified in three major categories. They are reported in the following subsections.

2.1   Adaptation of static algorithms

This approach is based on the notion of a rolling horizon. As time unfolds, static problems are solved repeatedly over events found within an horizon of length L that extends from the current time t to $t + L$. Different strategies emerge when the length L of the horizon is modified. If L is very small, a myopic near-term strategy is observed. In some cases this strategy can provide near-optimal solutions (Bell et al.(83), Dial(95), Psaraftis(85), Powell(88)). On the other hand, if L is very large, the problem considered is richer but, as it contains long-term events, the solution obtained is typically weaker unless a fast and powerful solution procedure

4

is used (Trudeau et al.(89), Gendreau et al.(96a), Gendreau et al. (96b)). Note also that longer-term events may be postponed, since they are likely to lead to useless calculations and updates.

Adaptation of static procedures to dynamic vehicle routing problems can be divided in two classes :

(a) A sophisticated static problem-solving procedure, which typically involves a re-optimization of the routes, is applied each time an input update occurs. Some researchers have used this approach to solve pick up (or delivery) only problems (Bell et al.(83), Hill et al.(88), Brown et al.(87)), while others have used it for mixed pick up and delivery problems (Psaraftis(80, 83, 85), Powell et al.(88), Dial(95)).

The drawback of this approach is the amount of computation time resulting from repeatedly executing the static algorithm. This disadvantage is more dramatic when new events occur frequently and when the execution of the static algorithm requires more time.

(b) Fast local operations (e.g., insertion) are used for reacting to any input revision. Trudeau et al.(89) used this kind of methodology to solve an emergency service problem. Wilson(77), Rousseau and Roy(88), Solonki et al.(91), Madsen et al.(95) applied this strategy to dial-a-ride problems found in transportation on demand systems for the elderly or handicapped. This approach is easy to implement and is appropriate for a dynamic environment where time pressure is important (e.g. Lund et al.(96)). However, it is myopic since solutions are produced through consecutive insertions (while a complete reordering of the routes may lead to better solutions). To overcome this weakness, some authors combine local operations with re-optimization procedures. This is often achieved by executing a set of successive insertions followed by a local search (e.g., exchange procedures like 2-opt, see Lin(65)). For different applications reported in the literature, see Rivard(81), Roy et al.(85), Gendreau et al.(96a) and Gendreau et al.(96b).

It is worth noting that many of the studies mentioned above are motivated

by commercial projects. This can justify the frequent use of simple strategies like insertion procedures.

All studies mentioned above ignore the potential benefits of considering the stochastic aspects of the problems and trying to forecast the future. Stochastic methods are aimed at overcoming this weakness.

2.2 Stochastic methods

Real-time dispatching problems have a stochastic nature (e.g., accidents, congestion, unexpected changes in meteorological conditions, unpredictable cancelations, unforeseen delays, etc...). Stochastic methods can be viewed as a "natural" way to judiciously address these issues. The goal is to react properly to an event to insure a good quality of service to the customers disturbed by these events, while minimizing their undesirable impact on the whole system. Two major classes of stochastic approaches are reported in the literature : stochastic programming and Markov decision processes.

(a) Markov decision processes:

Bertsimas and Van Ryzin(91, 93) proposed and analyzed a Markov decision model for the dynamic traveling repairman problem, while Minkoff(93) used such a formulation to solve a dynamic vehicle dispatching problem where customers have stochastic demands. Powell(88) presented a study of the dynamic vehicle allocation problems in a truck-load carrier context. In this problem, special attention is given to dispatching and repositioning trucks in anticipation of forecasted future demands. The author proposed a Markov decision model. Unfortunately, Markov decision processes are confronted with the following limitations :

- The state space grows quickly with problem size.

- Simplifying assumptions are made to make the model more tractable.

These limitations often prevent Markov decision models from being applied to complex real-world problems.

(b) Stochastic programming:

The only work that we are aware of in this category is the one done by Powell(88) in his comparative review of dynamic vehicle allocation problems. The author proposed a hybrid model that combines insights from Markov decision processes and classical network formulations.

At this point, stochastic programming does not seem to be "mature" enough to provide a strong background for dynamic vehicle routing studies.

2.3 Other methods

A new generation of approaches arises that try to replicate the dispatcher's decision making process. This is achieved by automating the decision procedure based on previous decisions taken by a skilled dispatcher.

(a) Neural networks:

Shen and Potvin(95) proposed an expert consulting system for a dispatcher working in a courier service company. The assignment of any new incoming request is provided by a learning module based on a neural network. The shortcomings of this approach are three-fold : **(i)** many parameters must be adjusted during the learning phase (e.g., connection weights); **(ii)** there is a need for a data base of decisions previously taken by an expert; **(iii)** the decisions can only be as good as those made by the expert.

(b) Genetic programming:

This work was done by Benyahia and Potvin(95), again in a courier service context. They defined a program as an utility function which evaluates a vehicle based on different characteristics, such as the additional tardiness resulting from the insertion of a new request in a planned route. The major weakness of genetic programming is the difficulty to interpret the generated program.

2.4 Diversion strategies

Apart from the study of Regan et al.(94, 95), all approaches that we are aware of fix the current destination of each vehicle. However, diverting a vehicle away

from its current destination may be quite beneficial. Technically, diversion is now possible due to recent advances in information technologies. Regan et al. (94, 95) have proposed different diversion strategies in the context of a truck-load carrier. These studies will be discussed in detail in section 5.

3   Problem definition

The problem considered is a dynamic pick up (or delivery) problem motivated from courier service applications. The static version of the problem can be stated as follows. Given a fixed size fleet of m identical vehicles and a set of requests to be served, the goal is to find a set of minimum cost routes that service these requests. More precisely:

- The vehicle routes must originate from and terminate at a fixed depot

- Each vehicle services one route and the service point of each request is visited exactly once by exactly one vehicle;

- No capacity constraint is considered (since small items are transported);

- Each service point, (including the depot) has its own time window $[e, l]$, where e is the earliest service time and l is the latest service time. Each route must start and end within the time window associated with the depot. Moreover, the service points have "soft" time windows. Thereby, a vehicle can arrive before the lower bound or after the upper bound: if the vehicle arrives too early, it must wait to start its service; if the vehicle is too late, a penalty for lateness is incurred in the objective function.

The objective is to minimize a weighted sum of total distance and total lateness over all customers. In the dynamic version of the problem, a number of service requests are not known completely ahead of time, but are rather dynamically revealed as time goes by. As illustrated in figure 1, in a dynamic setting, a vehicle route can be divided into three parts at any instant t :

- completed movements which form the part of the route already executed. Thereby, this part can not be modified anymore;

- current movement to reach the current destination;

- planned movements which constitute the portion of the routes not yet executed by the vehicle (planned route).
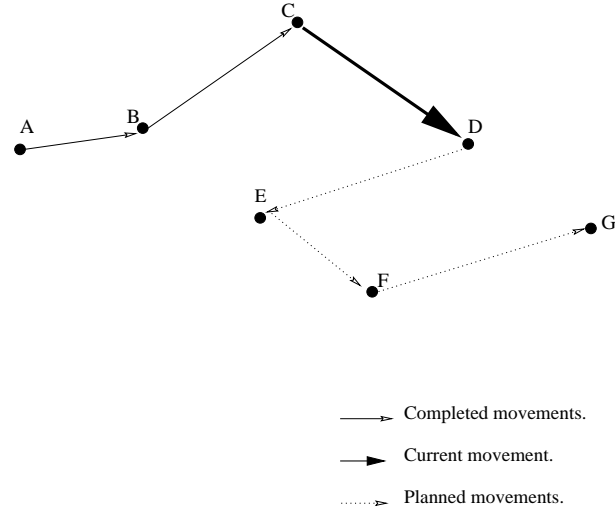
Figure 1: A vehicle route in a dynamic setting.

Given a new request at instant t, the problem is to assign this request to a particular vehicle and include it in its planned route at minimum cost.

In the following section, we briefly describe the original tabu search heuristic developed to tackle this problem.

4 A parallel tabu search algorithm

At any instant t, a solution to our dynamic problem is a set of planned routes, each beginning with the current destination of the associated vehicle. This current destination is fixed in the procedure to be described and can not be modified (i.e., the vehicle must reach its current destination).

The algorithm developed in Gendreau et al. (96b) is a parallel tabu search heuristic with an adaptive memory. Tabu search is an iterative local search technique that starts from some initial solution. At each iteration, a neighborhood is generated around the current solution and the best solution in this neighborhood becomes the new current solution (even if it does not provide an improvement). By allowing a degradation of the objective, it is possible to escape from bad local optima, as opposed to pure descent methods. The interested reader will find more details about this approach in Glover and Laguna (97).

The algorithm developed in Gendreau et al.(96b) can be described as follows :

- Construct I different initial solutions with a stochastic insertion heuristic (where the choice of the next customer to be inserted is randomized).

- Apply tabu search to each solution and store the resulting routes in the adaptive memory.

- While a stopping criterion is not met do :

  - Use the routes stored in the adaptive memory to construct the initial solution.

  - Decompose the problem into subproblems obtained through a decomposition procedure.

  - Apply tabu search to each subproblem.

  - Add the routes of the resulting solution in the adaptive memory.

In the following, the main points of this algorithm are briefly introduced.

Adaptive memory

An adaptive memory which stores previously found elite solutions, is used to generate new starting points for the tabu search. This is achieved by combining routes taken from different solutions in memory (Rochat and Taillard (95)). Any new solution produced by the tabu search is included in the memory if it is not filled yet. Otherwise, the new solution replaces the worst solution in memory, if it is better.

Neighborhood structure of tabu search

The procedure for generating the neighborhood is called the CROSS exchange. Basically, two segments of variable lengths are taken from two different routes and are swapped.

Parallel implementation

To cope with real-time environments, the algorithm was implemented on a network of workstations. The parallelization of the procedure was achieved at two different levels :

(1) Different tabu search threads run in parallel, each of them starting from a different initial solution.

(2) Within each search thread, many tabu searches run independently on sub-problems obtained through a decomposition procedure.

Given the available platform, a master-slave scheme was chosen to implement the procedure. The master process manages the adaptive memory and creates initial solutions for the slave processes that run the tabu search.

A dynamic environment

Given the dynamic context, it is important to maintain the consistency of the adaptive memory with the current environment. Also, the memory itself must be consistent, that is, the current destination of each vehicle (which is fixed) must be the same for all solutions in memory. Thus, whenever a new event occurs, the search threads are first interrupted. Then, after an appropriate update of the adaptive memory, it is possible to restart the search threads with new solutions constructed from this updated memory. A new event may be of two types:

- A vehicle has finished serving its current customer. In this case, its next destination (i.e., the next customer to be serviced) has to be determined. The best solution in memory is used for this purpose. The remaining solutions in the adaptive memory are updated by removing this customer from its current location and by inserting it in first position in the planned route of the vehicle.

- A new service request has just occurred. In this case, the new request is inserted in all solutions in adaptive memory. If there is no feasible insertion position in any solution, the request is rejected. Otherwise, the request is accepted and all solutions with no feasible insertion places are discarded from memory. The best solution is then re-optimized using a local search heuristic based on CROSS exchange in order to have at least one solution of high quality that includes the new request.

Section 5 will now consider the case where the next destination of each vehicle is not fixed. To address this issue, a dynamic assignment strategy that includes diversion as a special case is proposed. Its implementation within the tabu search heuristic presented here is reported in section 6.

5 A new strategy for handling customer requests

5.1 Motivation

Apart from the study of Regan et al.(94, 95), all approaches that we are aware of fix the current destination of each vehicle. However, diverting a vehicle away from its current destination may be quite beneficial. This could be considered, for example, when a new request occurs in the vicinity of the current position of some vehicle, while the driver is on his way to his current destination. Technically, diversion is now possible due to recent advances in communication technologies. In the following, we present an approach that includes diversion as a special case. We also investigate the trade-off between computation time and solution quality, given that this strategy takes place in a context where time pressure is important (vehicles are moving fast and diversion opportunities may be quickly lost).

In 1994, Regan et al.(94) have proposed different diversion strategies in the context of a truck-load carrier. This problem is a combined pick up and delivery problem with no consolidation (i.e. at any time, a vehicle is either empty or carrying a single load). In our application, on the other hand, a vehicle may be used to carry many loads concurrently. Also, in the context of a truck-load carrier, requests are more distant from each other and the activities take place over a longer time horizon (e.g., a few days). Finally, diversion was evaluated in a context where simple local dispatching rules were used, thus resulting in myopic approaches.

In Regan et al.(94), the authors identify, under simple assumptions (i.e., one vehicle, two customers only), the probability for diversion to be beneficial in terms of reducing the total distance traveled. A diversion decision is based on the distance between the current position of the vehicle and the new pick up point. Since there is no consolidation, diversion is allowed only if a vehicle is moving empty. Numerical results show that diversion provides a gain of 1 to 7 % when compared with the basic strategy that serves the requests in their order of occurrence. Their procedure is also extended to 100 requests evaluated sequentially on a pairwise basis (i.e., after serving a customer, a new demand appears and the procedure is restarted). As compared to the basic strategy previously mentioned, a gain of 1.5
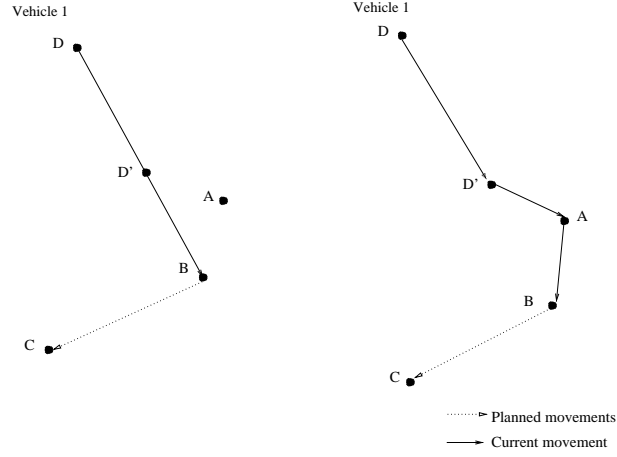
Figure 2: Diversion.

to 12.5 % is observed.

Later, in Regan et al.(95), the authors have presented another study on diversion. Once again, only one vehicle is considered and a queue of at most 5 requests is associated with the vehicle. Diversion is compared with a strategy that re-sequences the requests in the queue whenever the vehicle reaches its next destination. The goal is to minimize the total travel time. An acceptance rule is also introduced to decide whether to accept or to reject a new request. Namely, if the ratio of empty to loaded distance in the solution with the new load is greater than or equal to a given threshold value, the load is rejected. It is reported that diversion performs better than the basic strategy without diversion.

5.2 A broader view

Diversion consists of allowing a vehicle to be diverted away from its current destination to serve a request that just occurred in the vicinity of its current position. Figure 2 shows an example for one vehicle route. We suppose that a new request unfolds at point A at instant t, while vehicle 1 is at position D' on his way to service point B. The only modification that can result from en-route diversion is the modification of the current destination of one vehicle. This occurs if the vehicle services the new request before its planned current destination B. In this
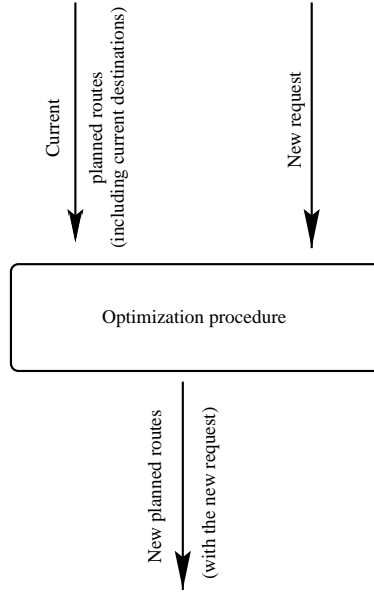
Figure 3: A dynamic assignment strategy that includes diversion.

paper, we propose a broader approach where a change that occurs in the system may lead to redirecting one or more vehicles.

Figure 3 illustrates this new strategy in a schematic way. We first assume that the current destination of each vehicle is part of the planned route (and may thus change). Whenever a new request occurs, this request and the current routes are provided as input to some optimization procedure. The latter then produces a new set of routes with the new request. The optimization procedure, may be anything that we can think of, such as:

- A simple insertion procedure. Diversion then happen if the new request is inserted just before the current destination of some vehicle.

- An insertion followed by some re-optimization procedure (e.g., tabu search).

- An exact algorithm (assuming that it is fast enough).

Depending on the optimization procedure, it may well happen that the current destination of each vehicle remains the same after its application (although
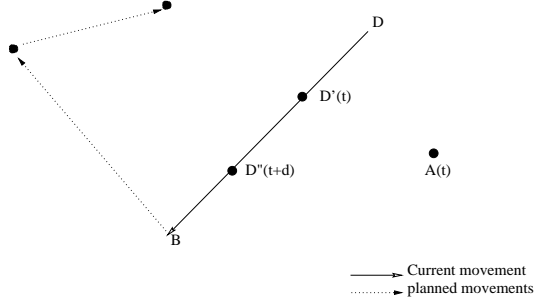
Figure 4: Time projection

the planned routes have changed). Conversely, the current destination of one or more vehicles may change, but not necessary for servicing the new request (which appears further away on a planned route). Thus, diversion is simply a particular outcome of the dynamic assignment strategy presented here.

5.3 Solution quality versus computation time trade-off

Let us assume that a new request appears at point A at instant t and that it is possible to estimate the position D' of the vehicle (between the point D just served and the current destination B), see Fig. 4.

Since new opportunities for including the new request are now offered through the inclusion of the current destination in the planned route, the solution obtained should be better. However, some amount of time δt is required to apply the optimization procedure. Hence, a decision based on the situation at instant t when a new request appears, does not correspond to the state of the system at $(t + \delta t)$ when the decision becomes available (i.e. the vehicle will be at position D'' rather than D'). Consequently, the situation should be assessed at instant $(t + \delta t)$ rather than instant t. The goal is to provide a decision that will fit in the context found at the end of the time allocated to the optimization procedure.

Clearly, this amount of time is a very important factor. In dynamic settings, finding a value that achieves a good trade-off between execution time and solution quality is very challenging. If δt is too small, not enough time is available for the optimization procedure. Therefore, the situation may be incorrectly assessed,

resulting in a bad decision. On the other hand, if δt is too large, opportunities for diversion may be lost.

In order to assess the benefits that can be gained from the application of our dynamic assignment strategy, the parallel tabu search heuristic developed by Gendreau et al. (96b) was used as the optimization procedure. The next section shows how the original algorithm was modified to implement it.

6   Implementation

6.1   Inclusion of the current destination

In the original implementation of the tabu search heuristic (Gendreau et al. (96b)) reported in Section 4, the current destination of each vehicle is a fixed node which is not part of the planned route to be optimized. We thus included the current destination of each vehicle in the planned route and use instead the current position of the vehicle as the fixed node. With this modification, the re-optimization procedure based on CROSS exchanges can now move the current destination around (as any other service point on the planned route).

6.2   Time projection

When a new request is received at time t, all solutions in the adaptive memory are updated according to the state of the system at time $(t + \delta t)$, where δt is the amount of time allocated to the optimization procedure. The planned routes of the best solution in adaptive memory are used for this purpose. A dummy point $F_{t+\delta t}^i$ is created for every vehicle route i of every solution in the adaptive memory. This dummy point $F_{t+\delta t}^i$ represents the position of vehicle i at time $(t + \delta t)$ and is the fixed node associated with route i.

Updating the solutions in the adaptive memory is illustrated in figure 5. In this figure, a fleet of two vehicles is considered: a new request occurs at instant t and δt is the amount of time allocated to the optimization procedure. In the best solution S^* at instant $t + \delta t$, vehicle 1 will have finished service at customer 1 and customer 3, while vehicle 2 will not have yet reached customer 2. Hence, before applying the optimization procedure, customer 1 and customer 3 are removed from S^* and the dummy points are inserted at the beginning of the two routes. To update solution S, customer 1 and customer 3 are first removed from it (since they are serviced). Then the dummy fixed points are added at the beginning of the two routes. It is worth noting that the customers that have not been serviced yet are found in the same order on the planned routes before and after the update. Only the service times are likely to change.
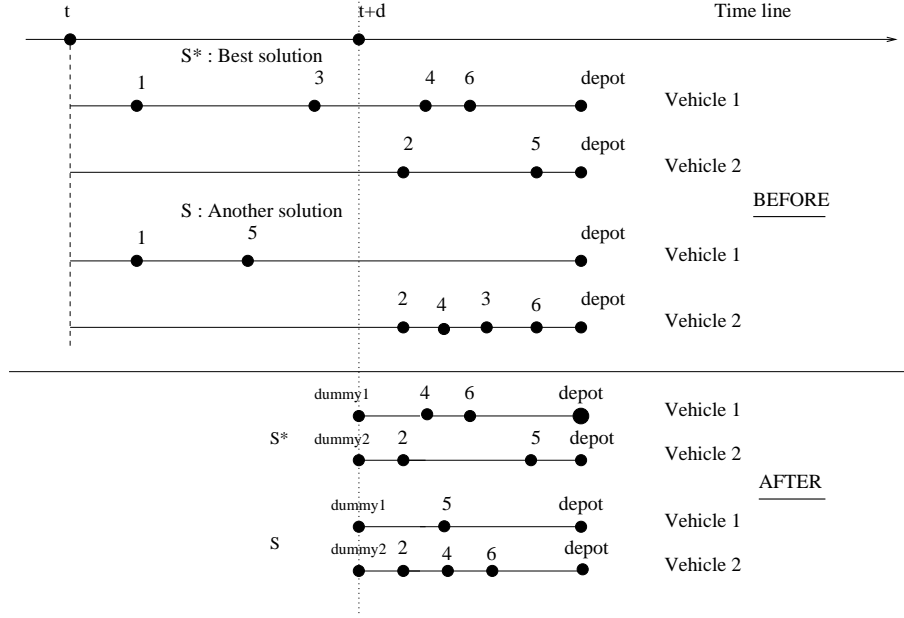
Figure 5: Update of solutions in the adaptive memory

6.3 Handling new events

Suppose that a new event occurs at instant t. The new events that occur during the time interval $[t, t + \delta t]$ will not interrupt the search threads. Instead, these events are handled as follows :

(a) When a vehicle has finished serving a customer, its next destination is determined by the best solution found in the adaptive memory. That is, in the time interval $[t, t + \delta t]$, everything is "frozen" and we simply follow the best solution in memory.

(b) When a new request occurs, we must decide whether to accept or reject it, since the customer may be waiting for a prompt answer about the ability to handle his request (a request is rejected if the end time of the planned route exceeds the upper bound of the time window at the depot after its insertion). Basically, the new request is inserted in all solutions in a copy of the adaptive memory. Solutions with no feasible insertion places are discarded. If at least

one feasible solution remains, the new request is accepted and is kept in a list of pending requests. Otherwise, it is rejected.

At instant $(t + \delta t)$, all search threads are then interrupted. The adaptive memory is updated using the best solution found by the optimization procedure and the pending requests are handled one by one, as in the original algorithm (Gendreau et al.(96b)). Finally, the search threads are restarted with new solutions constructed from the adaptive memory.

6.4   Setting δt

The amount of time allocated to the optimization procedure must be related to the rate at which new events occur. When few events occur, a large evaluation time can be allowed. Conversely, in case of a highly dynamic environment, the evaluation must be done quickly. Based on these observations, three rules are considered.

Rule 1: δt is chosen in such a way that the optimization procedure ends before any vehicle begins its service at its current destination. Let us assume that a service request is received at time t, and that service at the current destination on route i begins at time t_i. Then, choose δt such that : $t + \delta t \leq t_i$, $\forall i$, that is, $\delta t \leq min_i(t_i) - t$.

By choosing δt less than or equal to $min_i(t_i) - t$, we only need to add a dummy fixed node at the beginning of each planned route, which corresponds to the position of the vehicle at time $t + \delta t$. However, δt is often very small as there is typically one or more vehicles that are close to their current destination. Consequently, this rule was not retained in our computational experiments.

Rule 2: The amount of time δt is fixed a priori and is related to the average inter-request time interval. This interval is proportional to a moving average of the l last inter-request intervals, that is $\delta t = \alpha_1 T^-$, where α_1 is a constant and T^- is the moving average. By relating δt to the average request inter-arrival times, our choice is based on the intensity of occurrence of new requests.

Rule 3: In rule 2, the computation of δt may take into account a service request that is not so urgent (i.e., its time window is much later than the current time). This is true, in particular, in contexts where the calls are received much before the requested service times. Therefore, another rule is proposed. Let X be the length of some time horizon and l_X the number of requests on the planned routes found within that horizon. Then the average time per request is defined as X/l_X and the interval δt is such that $\delta t = \alpha_2 X/l_X$, where α_2 is a constant.

Rule 2 and rule 3 represent two different strategies. In rule 2, intensity is measured using past events, since the moving average is calculated over the l last request inter-arrival times. In rule 3, intensity is assessed using events to come, since the average time per request is evaluated over requests found on the planned routes. Results obtained with these two strategies for setting δt are reported in the next section.

7 Computational results

7.1 Simulation framework

Simulations were performed to assess the benefits of the proposed approach. Data were taken from Solomon's 100-customer Euclidean problems (Solomon(87)), as in Gendreau et al. (96b). In the following, we describe the essential components of the simulator. The interested reader will find more details about this simulator in Gendreau et al.(96b).

- The customer locations are generated within a $([0, 100])^2$ square.

- The travel times are proportional to the corresponding Euclidean distances.

- Six different sets of problems are defined, namely C_1, C_2, R_1, R_2, RC_1, and RC_2. The customers are uniformly distributed in the problems of type R, clustered in groups in the problems of type C and mixed in the problems of type RC. In the problems of type 1, only a few customers can be serviced on each route due to a narrow time window at the depot, as opposed to problems of type 2 where each route may have many customers.

- The set of requests is divided into two subsets. The first subset contains requests that are assumed to be known at the start of the day. They are randomly selected among the entire set of requests with a bias in favor of requests with early time windows. The second subset contains requests that are received in real-time. In this case, the arrival time of each request is adjusted to create two different types of scenarios : scenario 1 with half of the requests known in advance and scenario 2 with only 25 % of the requests known in advance. In the latter case, a higher degree of dynamism is obtained, as there are more requests that occur in real-time within the same time horizon.

- Using minutes as time units for time-related data in Solomon's files, the time horizon is set to 15 minutes. The intent is to have a realistic rate of new

unfolding requests (about 3 requests per minute for scenario 1 and 5 requests per minute for scenario 2).

7.2 Numerical results

The experiments reported in this section were performed on a network of 9 SUN UltraSparc-IIi workstations (300 MHz). Each process was programmed in C++ and communication between the processes was handled by the Parallel Virtual Machine Software. In these experiments, the objective function to minimize is the sum of the total distance traveled and the total tardiness over all customer locations. The fleet size was set to the number of routes in the best solution reported in the literature for each problem (Gendreau et al. (96b)). The value of the parameters found in the original algorithm were kept as in Gendreau et al. (96b). Parameters related to our new dynamic assignment strategy are tested in the following.

7.2.1 Preliminary tests

Scenario 1 (with half of the requests known in advance) was first considered for the preliminary experiments. The new algorithm was tested under rule 2 (see section 5) where the interval δt is computed as a multiple of a moving average T^- of inter-arrival times. That is, $\delta t = \alpha_1 T^-$, where α_1 is a multiplier. The moving average is taken over l loads, with $l = \beta_1 K$, K is the number of requests that are known in advance and β_1 is a multiplier. Parameters α_1 and β_1 are determined empirically.

Preliminary experiments were performed over 2 problems selected in each of the six categories C_1, C_2, R_1, R_2, RC_1, and RC_2. In these experiments, the values 0.10, 0.25, 0.50, 0.75 and 0.90 were tested for α_1 while the values 0.10 and 0.25 were tested for β_1. The results obtained did not clearly show an improvement over the original algorithm. Thus, more investigations were required, leading to an important refinement in our algorithm.

As mentioned in section 6, a copy of the adaptive memory is used to decide whether a new request should be accepted or rejected. This can lead to incon-

sistencies with the true memory. In particular, two important problems were identified:

- A new best solution that is received from a tabu search thread is not inserted in the copy (only in the adaptive memory itself). Consequently, requests that could have been inserted in the solutions found in the adaptive memory at time $t + \delta t$ may have been wrongfully rejected.

- A new solution may be received from a tabu search thread, while the adaptive memory is full. In this case, the worst solution in adaptive memory is replaced by the new one. At this point, there is an inconsistency with the copy that still includes this worst solution.

To improve the implementation, both the adaptive memory and its copy are updated whenever a new solution is received from a tabu search thread. Table 1 shows the kind of improvement that can be obtained with the refined implementation, using $\alpha_1 = 0.9$ and $\beta_1 = 0.25$. The four numbers in each entry are the average number of unserved customers, distance traveled, total lateness and objective value (sum of total distance traveled and total lateness), respectively, for each problem set. The row "Overall" contains averages taken over the entire set of 56 test problems. Note that when the number of unserved customers is maintained, the objective value is improved. On the other hand, when the number of unserved customers is reduced, this is achieved with only a slight increase in the objective value.

7.2.2 Parameter values

To determine the best values for parameters α_1 and β_1 under rule 2, tests were performed over a small sample of problems selected in each of the six categories C_1, C_2, R_1, R_2, RC_1, and RC_2. Several values had to be tested for every problem in the sample. Thereby, a significant amount of computation time is required if the size of the sample is too large. On the other hand, this size has to be large enough to get a correct calibration. For these experiments, the size of the sample

was set to 4. As scenario 1 and scenario 2 represent different degrees of dynamism (requests occur more frequently in scenario 2), the experiments were conducted separately for the two scenarios. Table 2 and table 3 report results obtained with different combinations of values using scenario 1 and scenario 2, respectively. As in Table 1, the four numbers in each entry are the average number of unserved customers, distance traveled, total lateness and objective value, respectively, for each problem set. Based on the results obtained, the values $\alpha_1 = 0.50$, $\beta_1 = 0.15$ for scenario 1 and $\alpha_1 = 0.50$, $\beta_1 = 0.10$ for scenario 2 were finally selected.

Under rule 3, the interval δt is computed as a function of the length of some rolling time horizon, that is, $\delta t = \alpha_2 X / l_X$, where α_2 is a constant, l_X is the number of requests on the planned routes found within the horizon X; $X = \beta_2 L$, where L is the length of the time horizon for the simulation and β_2 is a multiplier. The tests were conducted as for rule 2. Table 4 and table 5 report results obtained with different combinations of values using scenario 1 and scenario 2, respectively. The values $\alpha_2 = 0.50$, $\beta_2 = 0.25$ for scenario 1 and $\alpha_2 = 0.25$, $\beta_2 = 0.25$ for scenario 2 were finally selected.

Note that under rule 2, δt is about 5 seconds for scenario 1, and 2.8 seconds for scenario 2. Under rule 3, δt is about 4 seconds for scenario 1 and 2 seconds for scenario 2.

7.2.3 Assessing the new implementation

Tables 6 and 7 compare the new algorithm (with the best parameter values found under rule 2 and rule 3) with the original algorithm reported in Gendreau et al.(96b), for scenario 1 and scenario 2, respectively. Here, the tests were conducted over all problems in each category. The results highlight the fact that the new implementation performs well against the original algorithm for the two scenarios. More precisely:

Rule 2: For scenario 1, both the number of unserved customers and objective value were improved with regard to the original algorithm (apart from subset R_2). The reduction ranged from 33.3% to 100% for the number of unserved customers and from 0.6% to 3.0% for the objective value. In the case of subset R_2, the in-

crease in the objective value was marginal when compared to the reduction in the number of unserved customers : 0.3% against 60%. Note that the lowest reduction observed in the objective value with the new strategy is still larger than the only increase observed (0.7% against 0.3%).

For scenario 2, the results demonstrate that applying our approach under rule 2 reduces both the number of unserved customers and the objective value in all categories. The reduction ranges from 11.2% to 67% for the number of unserved customers, and from 1.0% to 2.7% for the objective value.

Rule 3: For scenario 1 and for categories C_1 and RC_2, where the original algorithm succeeds in serving all customers, the new algorithm under rule 3 improves the objective value. This reduction ranges from 1.3% to 4.3%. In the case of R_1 and C_2, both the number of unserved customers and objective value were improved. The reduction ranged from 66.8% to 100% for the number of unserved customers, and from 2.6% to 2.7% for the objective value. For RC_1 and R_2, the number of unserved customers was improved under rule 3 at the cost of a small increase in the objective value (i.e., 67% against 0.04% for RC_1 and 80% against 0.6% for R_2). Using rule 3 instead of rule 2 results in a small reduction of the objective value ranging from 0.5% to 2.0% when the number of unserved customers is the same. Rule 3 succeeds in reducing the number of unserved customers, as obtained by rule 2 for RC_1 and R_2, at the cost of a small increase in the objective value (i.e. 50% against 1.8%, and 51% against 0.2%).

For scenario 2, the comparison with the original algorithm indicates a reduction of both the number of unserved customers and the objective value over all categories. This reduction ranges from 16.8% to 100% for the number of unserved customers, and from 2.0% to 4.3% for the objective value. The results also indicate that, apart from RC_1 and R_2, rule 3 leads to the same number of unserved customers than rule 2, but it reduces the objective value by 0.9% to 2.2%. For R_2 and RC_1, when compared with rule 2, rule 3 succeeds in reducing both the number of unserved customers (i.e. 12% and 100% respectively), and the objective value (i.e. 0.6% and 1.1% respectively).

Problem set	$\alpha_1 = 0.90, \beta_1 = 0.25$ new	$\alpha_1 = 0.90, \beta_1 = 0.25$ old
C_1	0 [a]	0
9	835.781 [b]	838.784
problems	2.025 [c]	1.403
	837.807 [d]	840.187
R_1	0.583	0.75
12	1183.00	1182.49
problems	51.125	52.55
	1234.12	1235.04
RC_1	0.375	0.25
8	1317.37	1316.12
problems	47.058	47.852
	1364.43	1363.97
C_2	0	0.25
8	608.321	602.624
problems	0	0
	608.321	602.624
R_2	0.181	0.181
11	1058.76	1073.06
problems	57.20	66.64
	1115.97	1139.70
RC_2	0	0
8	1188.09	1244.00
problems	45.166	53.276
	1233.26	1297.27
Overall	0.21	0.27
56	1040.62	1050.80
problems	35.69	39.02
	1076.31	1089.82

[a]Nb. of unserved customers.

[b]Total distance traveled.

[c]Total lateness.

[d]Objective value.

Table 1: Comparison of the refined implementation (new) with the old one using scenario 1

Problem set	$\alpha_1 = 0.90$ $\beta_1 = 0.15$	$\alpha_1 = 0.90$ $\beta_1 = 0.25$	$\alpha_1 = 1.20$ $\beta_1 = 0.25$	$\alpha_1 = 0.50$ $\beta_1 = 0.10$	$\alpha_1 = 0.50$ $\beta_1 = 0.15$
C_1 4 problems	0 [a] 829.89 [b] 0 [c] 829.89 [d]	0 832.35 0 832.35	0 828.535 0 828.535	0 852.31 0.65 852.96	0 828.94 0 828.94
R_1 4 problems	0.25 1253.02 63.435 1316.46	0 1284.25 38.307 1322.56	0.25 1262.32 43.90 1306.22	0.25 1266.03 47.12 1313.15	0 1281.79 45.385 1327.18
RC_1 4 problems	0.5 1334.86 74.472 1409.33	0.75 1324.50 62.985 1387.49	0.5 1310.76 55.962 1366.72	0.75 1318.16 67.762 1385.92	0.25 1330.19 83.022 1413.22
C_2 4 problems	0 642.978 0 642.978	0 625.628 0 625.628	0 612.602 0 612.602	0 616.238 0 616.238	0 627.51 0 627.51
R_2 4 problems	0.5 1096.48 24.182 1120.66	0.25 1106.52 30.155 1136.67	0 1106.13 19.337 1125.47	0.75 1126.82 53.782 1180.60	0 1107.19 14.645 1121.84
RC_2 4 problems	0 1226.10 24.045 1250.14	0 1184.14 38.74 1222.88	0 1188.22 10.745 1198.96	0 1198.09 45.41 1243.50	0 1180.53 17.482 1198.01

[a]Nb. of unserved customers.

[b]Total distance traveled.

[c]Total lateness.

[d]Objective value.

Table 2: Searching for the best parameter values under rule 2 for scenario 1

Problem set	$\alpha_1 = 0.50$ $\beta_1 = 0.10$	$\alpha_1 = 0.50$ $\beta_1 = 0.15$	$\alpha_1 = 1.20$ $\beta_1 = 0.15$	$\alpha_1 = 0.90$ $\beta_1 = 0.25$	$\alpha_1 = 0.25$ $\beta_1 = 0.25$
C_1	0 [a]	0	0	0	0
4	882.42 [b]	871.055	908.478	925.06	881.658
problems	0.105 [c]	0	19.79	15.027	2.127
	882.525 [d]	871.055	928.268	940.088	883.785
R_1	0.25	0	0.25	0	0.50
4	1296.70	1284.24	1303.97	1303.56	1296.28
problems	76.00	63.982	73.315	67.71	53.067
	1372.70	1348.22	1377.28	1371.28	1349.35
RC_1	0.25	0.75	0.5	1.00	1.00
4	1366.52	1349.24	1405.42	1402.88	1354.10
problems	54.99	80.737	77.917	61.32	64.82
	1421.50	1429.98	1483.34	1464.20	1418.92
C_2	0	0.25	0	0	0
4	607.572	619.515	613.62	621.495	612.33
problems	0	0	0.002	0	0
	607.572	619.515	613.622	621.495	612.33
R_2	0	0.75	0	0.25	0
4	1113.66	1093.96	1119.40	1140.72	1106.98
problems	17.63	93.605	23.73	14.032	22.36
	1131.29	1187.56	1143.13	1154.76	1129.34
RC_2	0	0	0	0	0
4	1207.95	1235.45	1213.50	1144.22	1218.88
problems	37.50	19.827	35.867	60.822	37.622
	1245.45	1255.28	1249.37	1205.04	1256.50

[a] Nb. of unserved customers.

[b] Total distance traveled.

[c] Total lateness.

[d] Objective value.

Table 3: Searching for the best parameter values under rule 2 for scenario 2

Problem set	$\alpha_2 = 0.50$ $\beta_2 = 0.15$	$\alpha_2 = 0.90$ $\beta_2 = 0.15$	$\alpha_2 = 1.20$ $\beta_2 = 0.15$	$\alpha_2 = 0.50$ $\beta_2 = 0.25$	$\alpha_2 = 0.25$ $\beta_2 = 0.25$
C_1	0 [a]	0	0	0	0
4	829.26 [b]	834.125	857.005	829.112	839.488
problems	0 [c]	0	0.375	0	0
	829.26 [d]	834.125	857.38	829.112	839.488
R_1	0.25	0.25	0.25	0.25	0.25
4	1272.71	1274.41	1267.60	1275.08	1265.19
problems	51.997	52.532	43.627	39.537	45.217
	1324.71	1326.94	1311.23	1314.62	1310.41
RC_1	0.75	0.75	1	0.75	0.75
4	1339.43	1344.64	1321.06	1325.66	1352.71
problems	62.587	74.707	71.185	60.04	64.92
	1402.02	1419.34	1392.25	1385.70	1417.63
C_2	0	0	0.5	0	0
4	615.518	614.925	616.928	616.878	624.21
problems	0	0	0	0	0
	615.518	614.925	616.928	616.878	624.21
R_2	0	0	0.5	0	0
4	1110.10	1095.61	1113.52	1091.81	1105.60
problems	60.165	61.09	46.065	56.442	64.455
	1170.27	1156.70	1159.58	1148.25	1170.06
RC_2	0	0	0	0	0
4	1186.05	1198.42	1172.54	1177.27	1198.94
problems	23.472	29.502	23.862	25.457	32.652
	1209.52	1227.92	1196.40	1202.73	1231.60

[a]Nb. of unserved customers.

[b]Total distance traveled.

[c]Total lateness.

[d]Objective value.

Table 4: Searching for the best parameter values under rule 3 for scenario 1

Problem set	$\alpha_2 = 0.25$ $\beta_2 = 0.25$	$\alpha_2 = 0.50$ $\beta_2 = 0.25$	$\alpha_2 = 0.90$ $\beta_2 = 0.15$	$\alpha_2 = 1.20$ $\beta_2 = 0.15$	$\alpha_2 = 0.50$ $\beta_2 = 0.15$
C_1 4 problems	0 [a] 880.675 [b] 1.977 [c] 882.652 [d]	0 889.275 0 889.275	0 880.555 0.90 881.455	0 955.958 1.695 957.652	0 882.895 0 882.895
R_1 4 problems	0 1282.77 46.415 1329.18	0.25 1302.15 51.71 1353.86	0.25 1293.42 60.865 1354.29	0.25 1282.50 58.432 1340.93	0.25 1285.20 56.23 1341.42
RC_1 4 problems	0 1366.24 71.422 1437.66	1.00 1358.48 89.21 1447.69	1.00 1380.77 60.082 1440.86	1.00 1363.36 71.322 1434.68	0.25 1334.95 68.03 1402.98
C_2 4 problems	0 601.972 0 601.972	0.25 650.912 0 650.912	0 612.018 0 612.018	0 630.335 0 630.335	0 603.122 0 603.122
R_2 4 problems	0 1089.10 14.995 1104.09	0.50 1090.41 36.237 1126.65	0 1083.09 22.917 1106.00	0.50 1104.65 14.192 1118.84	0.50 1101.28 40.322 1141.60
RC_2 4 problems	0 1221.02 35.692 1256.72	0 1207.80 28.942 1236.75	0 1191.57 40.562 1232.13	0 1207.86 46.522 1254.38	0 1218.99 27.662 1246.66

[a] Nb. of unserved customers.

[b] Total distance traveled.

[c] Total lateness.

[d] Objective value.

Table 5: Searching for the best parameter values under rule 3 for scenario 2

Problem set	Original algorithm	New rule2 $\alpha_1 = 0.50$ $\beta_1 = 0.15$	algorithm rule3 $\alpha_2 = 0.50$ $\beta_2 = 0.25$
C_1 9 problems	0 ^{a} 835.098 ^{b} 5.535 ^{c} 840.633 ^{d}	0 830.024 3.341 833.366	0 829.628 0 829.628
R_1 12 problems	0.5 1219.42 43.267 1262.69	0.166 1194.75 48.625 1243.38	0.166 1178.54 51.460 1230.00
RC_1 8 problems	0.375 1349.65 44.22 1393.87	0.25 1304.58 64.955 1369.53	0.125 1333.41 60.965 1394.38
C_2 8 problems	0.125 612.798 0.326 613.124	0 609.232 0 609.232	0 597.011 0 597.011
R_2 11 problems	0.454 1028.34 31.96 1060.30	0.181 1035.09 29.053 1064.15	0.09 1020.55 46.334 1066.88
RC_2 8 problems	0 1203.75 30.811 1234.56	0 1175.87 22.108 1197.98	0 1160.83 20.672 1181.50
Overall 56 problems	0.27 1049.83 27.20 1077.03	0.10 1034.11 29.10 1063.22	0.07 1027.95 31.79 1059.74

^{a}Nb. of unserved customers.

^{b}Total distance traveled.

^{c}Total lateness.

^{d}Objective value.

Table 6: Comparison with the original algorithm for scenario 1

Problem set	Original algorithm	New	algorithm
	Without diversion	$rule2$ $\alpha_1 = 0.50$ $\beta_1 = 0.10$	$rule3$ $\alpha_2 = 0.25$ $\beta_2 = 0.25$
C_1 9 problems	0 [a] 877.332 [b] 10.455 [c] 878.378 [d]	0 843.924 11.227 855.152	0 833.061 8.027 841.089
R_1 12 problems	0.4 1226.21 55.431 1281.64	0.333 1203.99 67.167 1271.16	0.333 1197.30 49.121 1246.42
RC_1 8 problems	0.375 1400.50 56.391 1456.89	0.125 1358.78 73.28 1432.06	0 1346.98 69.002 1415.98
C_2 8 problems	0 610.775 0 610.775	0 600.796 0 600.796	0 595.635 0 595.635
R_2 11 problems	0.818 1043.96 216.696 1260.65	0.727 1043.43 193.885 1237.32	0.636 1027.82 207.639 1235.46
RC_2 8 problems	0 1222.22 27.441 1249.66	0 1189.48 41.023 1230.51	0 1184.77 19.568 1204.34
Overall 56 problems	0.30 1070.75 68.10 1138.85	0.23 1048.45 70.61 1119.06	0.19 1039.11 65.25 1104.36

[a]Nb. of unserved customers.

[b]Total distance traveled.

[c]Total lateness.

[d]Objective value.

Table 7: Comparison with the original algorithm for scenario 2

8 Conclusion

In this paper, a new strategy for the dynamic assignment of new requests that includes diversion as a special case was proposed. Tests were conducted to assess its benefits within a previously reported tabu search heuristic. Some issues related to the proposed model were addressed and explored. Results show that a reduction in the sum of total distance traveled and total lateness, as well as in the number of unserved customers is observed when compared with the original heuristic, where the current destination of each vehicle is fixed.

Future work will be aimed at considering additional sources of uncertainty like congestion, accidents, vehicle breakdowns, and others. A taxonomy of such events will be developed to help us in the task of determining appropriate responses.

References

[1] **P. Badeau, M. Gendreau, F. Guertin, J.Y. Potvin, E. Taillard,** *A Parallel Tabu Search Heuristic for the Vehicle Routing Problem with Time Windows,* Technical Report CRT-95-84, Centre de recherche sur les transports, Université de Montréal, Canada (1995).

[2] **M.O.Ball, T.L. Magnanti, C.L. Monna and G.L. Nemhauser (eds),** *Network Routing,* Handbooks in Operations Research and Management Science, Volume 8, North-Holland: Amsterdam (1995).

[3] **D.O. Baush, G. G. Brown, D. Ronen,** *Consolidating and Dispatching Truck Shipment of Mobil Heavy Petroeum Products,* Interfaces **25** (1995), 01-17.

[4] **W.J. Bell, L.M. Dalberto, M.L. Fisher, A.J. Greenfield, R. Jaikumar, P. Kedia, R.G. Macj, P.J. Prutzman,** *Improving the Distribution of Industrial Gases with an On-Line Computrized Routing and Scheduling Optimizer ,* Interfaces **13** (1983), 04-23.

[5] **R.E. Bellman,** *On a Routing Problem ,* Quarterly of Applied Mathematics **16**(1958), 87-90.

[6] **I. Benyahia and J.Y. Potvin,** *Decision Support for Vehicle Dispatching Using Genetic Programming ,* publication CRT-95-23, Centre de recherche sur les transports, Université de Montréal, Canada (1995).

[7] **D. Bertsimas and G. Van Ryzin,** *A Stochastic and Dynamic Vehicle Routing Problem in the Euclidian Plane,* Operations Research **39** (1991), 601-615.

[8] **D.J. Bertsimas and G. Van Ryzin,** *Stochastic and Dynamic Vehicle Routing in the Euclidian Plane with Multiple Capacitated vehicles,* Operations Research **41** (1993), 60-76.

[9] **J.R. Birge and J.K. Ho,** *Optimal Flows in Stochastic Dynamic Networks with congestion,* Operations Research, **41** (1993), 203-216.

[10] **J.Y. Blais, J.M. Rousseau,** *HASTUS: A Model for the Economic Evaluation of Drivers' Collective Agreements in Transit Companies,* INFOR **20** (1982), 03-15.

[11] **L. Bodin, B.L. Golden, A. Assad and M.O. Ball,** *Routing and Scheduling of Vehicles and Crews: the State of the Art,* Computers and Operations Research **10** (1983), 62-212.

[12] **J.W. Braklow, W.W. Graham, S.M. Hassler, K.E. Peck and W.B. Powell,** *Interactive Optimization Improves Service and Performance for Yellow Freight System,* Interfaces **22** (1992), 147-172.

[13] **G.G. Brown, C. J. Ellis, G. Lenn, W. Graves, David Ronen,** *Real Time, Wide Area Dispatch of Mobil Tank Trucks ,* Interfaces **17** (1987), 107-120.

[14] **C.F. Daganzo,** *An Approximate Analytic Model of Many-To-Many Demand Responsive Transportation Systems ,* Transportation Reaserch **12** (1978), 325-333.

[15] **M. Desrochers, J. Desrosiers and M. Solomon,** *A New Optimization Algorithm for the Vehicle Routing Problem with Time Windows,* Operations Reaserch **40** (1992), 342-354.

[16] **J. Desrosiers, Y. Dumas, M.M. Solomon and F. Soumis,** *Time Constrained Routing and Scheduling,* in Handbooks in Operations Research and Management Science, Volume 8: Network Routing, M.O. Ball, T.L. Magnanti, C.L. Monma and G.L. Nemhauser (eds), pp. 35-139, North-Holland: Amsterdam (1995).

[17] **Robert B. Dial,** *Autonomous Dial-A-Ride Transit Intoductory Overview,* Transportation Reaserch **03C** (1995), 261-275.

[18] **M. Dror and W. Powell,** *Special Issue in Stochastic and Dynamic Models in Transportation* ,Operations Research **41** (1993), 11-17.

[19] **Y. Dumas,** *Confection d'Itinéraires pour le Transport Adapté*, Publication G-88-44, GERAD, H.E.C., Canada (1988).

[20] **Y. Dumas, J. Desrosiers, F. Soumis,** *Large Scale Multi-Vehicle Dial-A-Ride Problems*, Publication G-89-30, GERAD, H.E.C., Canada (1989).

[21] **M.L. Fisher, A. J. Greenfield, R. Jaikumar and P. Kedia,** *Real Time Scheduling of a Bulk-Delivery Fleet: Practical Application of Lagrangian Relaxation*, Report 82-10-11, Decision Sciences Dept., University of Pennsylvania (1982).

[22] **M.L. Fisher, H.J. Gang and T.B. Xing,** *Scheduling Bulk-Pickup-Delivery Vehicles in Shangai* , Interfaces **16** (1986), 18-23.

[23] **M. Gendreau, P. Badeau, F. Guertin, J.Y. Potvin, É. Taillard,** *A Solution Procedure for Real-time Routing and Dispatching of Commercial Vehicles*, Technical Report CRT-96-24, Centre de recherche sur les transports, Université de Montréal, Canada (1996a).

[24] **M. Gendreau, F. Guertin, J.Y. Potvin and É. Taillard,** *Tabu Search for Real-Time Vehicle Routing and Dispatching*, Technical Report CRT-96-47, Centre de recherche sur les transports, Université de Montréal, Canada (1996b).

[25] **M. Gendreau, A. Hertz, G. Laporte and M. Stan,** *A Generalized Insertion Heuristic for the Traveling Salesman Problem with Time Windows*, Technical Report CRT-95-07, Centre de recherche sur les transports, Université de Montréal, Canada (1995).

[26] **B.E. Gillet,** *A Heuristic Algorithm for the Vehicle-Dispatch Problem*, Operations Research **22** (1974), 340-349.

[27] **F. Glover and M. Laguna,** *Tabu Search,* Kluwer (1997).

[28] **W.G. Graves, and B. A. Whinston,** *An Algorithm for the Quadratic Assignment Problem*, Management Science, **17** (1970), 453-471.

[29] **A. Hill, V. Mabert and D. Montgomory,** *A Decision Support System for the Courrier Vehicle Scheduling Problem*, Omega Int. J. of Mgmt Sci. **16** (1988), 333-345.

[30] **J.N. Hooker and N.R. Natraj,** *Solving a General Routing and Acheduling Problem by Chain Decomposition and Tabu Search*, Transp. Science **29** (1995), 30-44.

[31] **P. Jaillet,** *The Probabilistic Traveling Salesman Problem*, Ph.D. Thesis, Department of Civil Engineering, MIT (1985).

[32] **P. Jaillet,** *Probabilistic Routing Problems in the Plane*, Operational Research **29** (1991), 675-688.

[33] **I.Z. Jarrah and G. Yu,** *A Decision Support Framework for Airline Flight Cancelations and delays*, Transp. Science **27** (1993), 266-280.

[34] **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*, Working Paper MIT-UMTA-82-3, M.I.T., Cambridge Massachusetts (1982).

[35] **J.J. Jaw, A.R. Odoni, H.N. Psaraftis and N.H.M. Wilson,** *A Heuristic Algorithm for the Multi-Vehicle Advance Request Dial-A-Ride Problem with Time Windows*, Transp. Res. **20B** (1986), 243-257.

[36] **B. Kalantani et al.,** *An Algorithm for the Traveling Salesman Problem with Pickup and Delevery Customers*, Journal of Operational Research **22** (1985), 377-386.

[37] **J.G. Klincewicz and B.M. Rosenwein,** *The Airline Exception Scheduling Problem*, Transportation Science **29** (1995), 04-16.

[38] **S. Lin,** *Computer Solutions of the Traveling Salesman Problem,* Bell System Technical Journal **44** (1965), 2245-2269.

[39] **K. Lund, O.B.G. Madsen, J.M. Rygaard,** *Vehicle Routing Problems with Varying Degrees of Dynamism* , Technical Report IMM-REP-1996-1, Institute of Mathematical Modeling, Technical University of Denmark, Lyngby, Denmark (1996).

[40] **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 Op. Res. **60** (1995), 193-208.

[41] **H. Min,** *The Multiple Vehicle Routing Problem with Simultaneous Delivery and Pickup points,* Transp. Res. **23A** (1989), 377-386.

[42] **A. Minkoff,** *A Real Time Dispatching of Delevery Vehicles,* Technical Report, Massachusetts Institute of Technology, Cambridge, Mass. (1985).

[43] **A.S. Minkoff,** *A Markov Decision Model and Decomposition Heuristic for Dynamic Vehicle Dispatching* , Operations Research, **41** (1993), 77-90.

[44] **J.Y. Potvin and J.M. Rousseau,** *Constraint-Direct Search for the Advanced Request Dial-A-Ride Problem with Service Quality Constraints,* in Computer Science and Operations research: New Developpement in the Interfaces, Balci, Sharda and Zenios (eds), pp. 457-474, Pergamon Press (1992).

[45] **W. Powell,** *A Comparative Review of Alternative Algorithms for the Dynamic Vehicle Allocation Problem,* in Vehicle Routing: Methods and Studies, B.L. Golden and A.A. Assad (eds), pp. 249-291, North Holland: Amsterdam (1988).

[46] **W. Powell, Y. Sheffi, K.S. Nickerson, K. Butterbaugh and S. Atherton,** *Maximizing Profits for North American Van Lines Truckload Division: A New Framework for Pricing and Operations,* Interfaces, **18** (1988), 21-41.

[47] **W.B. Powell, P. Jaillet and A. Odoni,** *Stochastic and Dynamic Networks and Routing*, in Network Routing, Handbooks in Operations Research and Management Science, Volume 8, M.O. Ball, T.L. Magnanti, C.L. Monma and G.L. Nemhauser (eds), pp. 141-295, North-Holland: Amsterdam (1995).

[48] **H.N. Psaraftis,** *A Dynamic Programming Solution to the Single-Vehicle Many-to-Many Immediate Request Dial-A-Riade Problem* , Transportation Science **14** (1980), 130-154.

[49] **H.N. Psaraftis,** *K-Interchange Procedures for Local Search in a Precedence-Constrained Routing Problem*, European Journal of Operational Research **13** (1983), 391-402.

[50] **H.N. Psaraftis,** *An exact Algorithm for the Single Vehicle Many-To-Many Dial-A-Ride Problem with Time Windows*, Transportation Reaserch **17** (1983), 351-357.

[51] **H.N. Psaraftis,** *Analysis of $O(N^2)$ Heuristic for the Single Vehicle Many-To-Many Euclidian Dial-A-Ride Problem*, Trasportation Reaserch **17B** (1983), 133-145.

[52] **H.N. Psaraftis, J.B. Orlin, D. Bienstock and P.M. Thompson,** *Analysis and solution algorithms of sealift routing and scheduling problems : Final Report*, Working Paper No. 1700-85, Sloan School of Management, M.I.T. (1985).

[53] **H.N. Psaraftis,** *Scheduling Large Scale Advance-Request Dial-A-Ride Systems*, Amer. J. Math. Management Sc. **06** (1986), 327-367.

[54] **H.N. Psaraftis,** *Dynamic Vehicle Routing Problems*, in Vehicle Routing: Methods and Studies, B.L. Golden and A.A. Assad (eds), pp. 223-249, North Holland: Amsterdam (1988).

[55] **H.N. Psaraftis M.M. Solomon, T.L. Magnanti and T.U. Kim,** *Routing and Scheduling on a Shoreline with release times*, Management Science **36** (1990), 212-223.

[56] **H.N. Psaraftis,** *Dynamic Vehicle Routing: Status and Prospects* , Annals of Operations Research **61** (1995), 143-164.

[57] **B. Ran, D.E. Boyce and L.J. Leblanc,** *A New Class of Indtantaneous Dynamic User-Optimal Traffic Assignmrnt Models*, Operations Research **41** (1993), 192-202.

[58] **H. Rappoport,** *A Planing Heuristic for Military Airlift*, Interfaces **22** (1992), 73-87.

[59] **A.C. Regan, H.S. Mahmassani and P. Jaillet,** *Real-Time Information for Improved Efficiency of Commercial Vehicle Operations*, Proc., First World Congress on Applications of Transport Telematics & Intelligent Vehicle-Highway Systems, **Nov.** (1994)

[60] **A.C. Regan, H.S. Mahmassani and P. Jaillet,** *Improving Efficiency of Commercial Vehicle Operations Using Real-Time Information: Potential Uses and Assignment Strategies*, Transportation Research Record **1493** (1994), 188-198.

[61] **A.C. Regan, H.S. Mahmassani and P. Jaillet,** *Dynamic Vehicle Allocation for Fleet Management: Operational Changes for Improved Efficiency*, Proc., 2nd World Congress on Applications of Transport Telematics & Intelligent Vehicle-Highway Systems, **Nov.** (1995)

[62] **A.C. Regan, H.S. Mahmassani and P. Jaillet,** *Dynamic Decision Making for Commercial Fleet Operations Using Real-Time Information*, Transportation Research Record **1537** (1995), 91-97.

[63] **R. Rivard,** *Construction des parcours des véhicules et des Horaires des Chauffeurs pour le Transport des Personnes Handicapées*, publication CRT-240, Centre de recherche sur les transports, Université de Montréal, Canada (1981).

[64] **Y. Rochat and E. Taillard,** *Probabilistic Diversification and intensification in Local Search for Vehicle Routing*, Journal of Heuristics, **1** (1995), 147-167.

[65] **J.M. Rousseau, and S. Roy,** *RAO Répartition Assistée par Ordinateur: la Déscription du Prototype* , publication CRT-564, Centre de recherche sur les transports, Université de Montréal, Canada (1988).

[66] **S. Roy J.M. Rousseau, G. Lapalme and J.A. Ferland,** *Routing and Scheduling for the Transportation of Disables Persons: the Algorithm*, Publication CRT-412, Centre de recherche sur les transports, Université de Montréal, Canada (1985).

[67] **M.W.P. Savelsberg and M. Sol,** *The general Pickup and delivery Problem*, Tranp. Science **29** (1992), 17-29.

[68] **R. Séguin, J.Y. Potvin, M. Gendreau, T.G. Crainic and P. Marcotte** *Real-time Decision Problems: an Operational Research Perspective*, publication CRT-95-58, Centre de recherche sur les transports, Université de Montréal, Canada (1995).

[69] **Y. Shen and J.Y. Potvin,** *A Computer Assistant for Vehicle Dispatching with Learning Capabilities*, Operations Reaserch **61** (1995), 189-211.

[70] **M.M. Solomon,** *Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints*, Operations Research **35** (1987), 254-265.

[71] **R.S. Solanki,** *An execution Planning Algorithm for Military Airlift*, Interfaces **21** (1991), 121-131.

[72] **D.M. Stein,** *Scheduling Dial-A-Ride Transportation Systems*, Transportation Reaserch **12** (1978), 232-249.

[73] **C. Sutcliffe and J. Board,** *Optimal Solution of a Vehicle Routing Problem: Transporting Mentally Handicapped Adults to an Adult Training Center*, J. Opl Res. Soc. **41** (1990), 61-67.

[74] É. Taillard, P. Badeau, M. Gendreau, F. Guertin and J. Y. Potvin, *A Tabu Search Heuristic for the Vehicle Routing Problem with Soft Time Windows*, Transportation Science **31** (1997), 170-186.

[75] P.M. Thompson and H.N. Psaraftis, *Cyclic Transfer Algorithms for Multivehicle Routing and Scheduling Problems* , Operations Research **41** (1993), 935-946.

[76] P. Trudeau, J.M. Rousseau, J.A. Ferland and J. Choquette, *An Operations Research Approch for the Planning and Operating of an Ambulance Service*, INFOR **27** (1989), 95-113.

[77] I. Webb, *Period and Phase of Customer Replenishment: A New Approch to Inventory/Routing Problems*, Technical Report, Massachusetts Institute of Technology, Cambridge, Mass. (1989).

[78] N.H.M. Wilson and N.H. Colvin, *Computer Control of the Rochester Dial-a-Ride System*, Report R77-31, Dept. of Civil Engineering. M.I.T. (1977).