NEW MOBILITY SCHEMA FOR HIGHLY-DEMANDED SERVICES AND INFRASTRUCTURES

Harbil Arregui
Research Assistant, Vicomtech-IK4, harregui@vicomtech.org
Estíbaliz Loyo
Researcher, Vicomtech-IK4, eloyo@vicomtech.org
Andoni Mujika
Research Assistant, Vicomtech-IK4, amujika@vicomtech.org
Oihana Otaegui
Head of Department, Vicomtech-IK4, ootaegui@vicomtech.org
Mikeletegi Pasealekua, 57 Parque Tecnológico 20009 Donostia / San Sebastián,
TEL: +[34] 943 30 92 30, FAX: +[34] 943 30 93 93

ABSTRACT

There are many scenarios where there is still a need to give a real alternative to the private vehicle, e.g., highly demanded services and infrastructures or big events that receive large influx of people that have to move at the same time to the same place. This paper describes the design and development of a new sustainable mobility solution, presenting an integrated real time management system that allows serving several users sharing a common origin or destination with similar schedules. Three different algorithms are tested in order to decide which one performs best to be included in the platform (linear programming, genetic and routing algorithms). The objective of the algorithms is to find the optimal set of routes, which is the one that uses the minimum number of vehicles to service all requests. Also, passengers are ensured that the travelling time increase due to other pick-ups will not exceed a certain percentage of the minimum time required to make the direct journey. Finally, it is concluded that the best approach stands on a combination of the presented algorithms.

INTRODUCTION

Many services and infrastructures receive large influx of people, usually concentrated in certain time slots. Business parks, hospitals, shopping areas or key transportation services such as airports and even cultural, sports or major events are good examples. They sometimes still need transportation solutions that will provide citizens with easy and comfortable access, either because of the lack of suitable scheduled public transport services for all the population, or because the strong orientation towards private car use generates inconvenient traffic congestion at entrances, pollution and parking problems.

This paper presents a new mobility schema that allows serving several users sharing a common origin or destination with similar schedules with a fleet of vehicles, maximizing the accessibility and minimizing the environmental and economical impact. This schema is assisted by an integrated real time management platform. Three algorithms have been designed in order to test their route calculation performance and decide which to be used in the system.

The problem presented is similar to the Travelling Salesman Problem (TSP) (1), which is one of the most intensively studied problems in computational mathematics. Several works have been developed, progressing from static routing to more dynamic solutions (2) reaching to on demand public transport systems (3). Most of them can be generically stated as Vehicle Routing Problems (VRP), where the aim is to minimize the total cost required to transport all the passengers or goods. Different models are classified according to the restrictions and requirements that the routes must fulfil: Vehicle Routing Problem with Time Windows (VRPTW) (4), Distance Capacitated Vehicle Routing Problem (DCVRP) (5), Split Delivery Vehicle Routing Problem (SDVRP), Pick Up and Delivery Problems (6), or Dial-A-Ride Problem (with or without maximum ride time). There are some other varieties of problem settings, where due to certain constraints not all the customers can be visited (7).

The system presented in this paper belongs to the category of a VRP, where due to certain constraints all the passengers have the same time and location requirements at destination (or origin). Scheduled timetables may not be accurate for the needs of passengers, resulting in long waiting times on arrival or excessive commuting times. The use of private car is the most widespread solution among users, but there are a lot of inconveniences:

- Relatively costly
- Time for parking
- Security of the vehicle (vehicle for long stays at the destination)
- Traffic congestion
- Environmental pollution, CO_2 emissions

NEW MOBILITY SCHEMA

Agents of those services and infrastructures mentioned before should consider the collective transportation to their facilities like a value added service they could offer. Transport operators are encouraged to integrate this vision in the market towards co-modality schemas. A user willing to go to the airport, e. g., chooses a pick-up location on a map through the web application and specifies the hour needed to be at the airport. The system receives this and the rest of requests from users that ask for transportation to the airport at the same time slot. The system runs the algorithm to calculate the routes and informs the users and drivers about the pick-up hours.

An innovative approach of the problem is proposed, integrating the benefits of both the use of public transport and the private vehicle in a simple way for the user. The solution includes a software application for designing dynamic optimal routes that will be integrated into a control centre allowing users to add their own stops.

This platform will provide fully flexible demand response solutions to people with similar needs. This solution promotes greater sustainability, encouraging responsible use of public transport but providing at the same time, the advantages provided by private transport. The platform covers, among others, the following:

- Allows residents of rural areas and small communities to have better access to the destinations.
- Allows disabled people to make use of public transport.
- Reduces congestion caused at the entrance to mass events and highly-demanded services and infrastructures.

The proposed system allows the user to make a request for collection at a specified point, and the vehicle is shared by all users who require transportation to the same destination, usually at the same time. Once all the requests for the same event are received through the user interface, the designed algorithms are run towards the best route planning. Afterwards, drivers are informed about the routes and users about their pickup times.

The final platform integrates the following elements:

- User interface to perform service requests
- Management interface for the system configurations.
- Algorithms for services planning
- Communication channels to inform both drivers and passengers about scheduled services.

The following diagram describes the general overview of the platform:

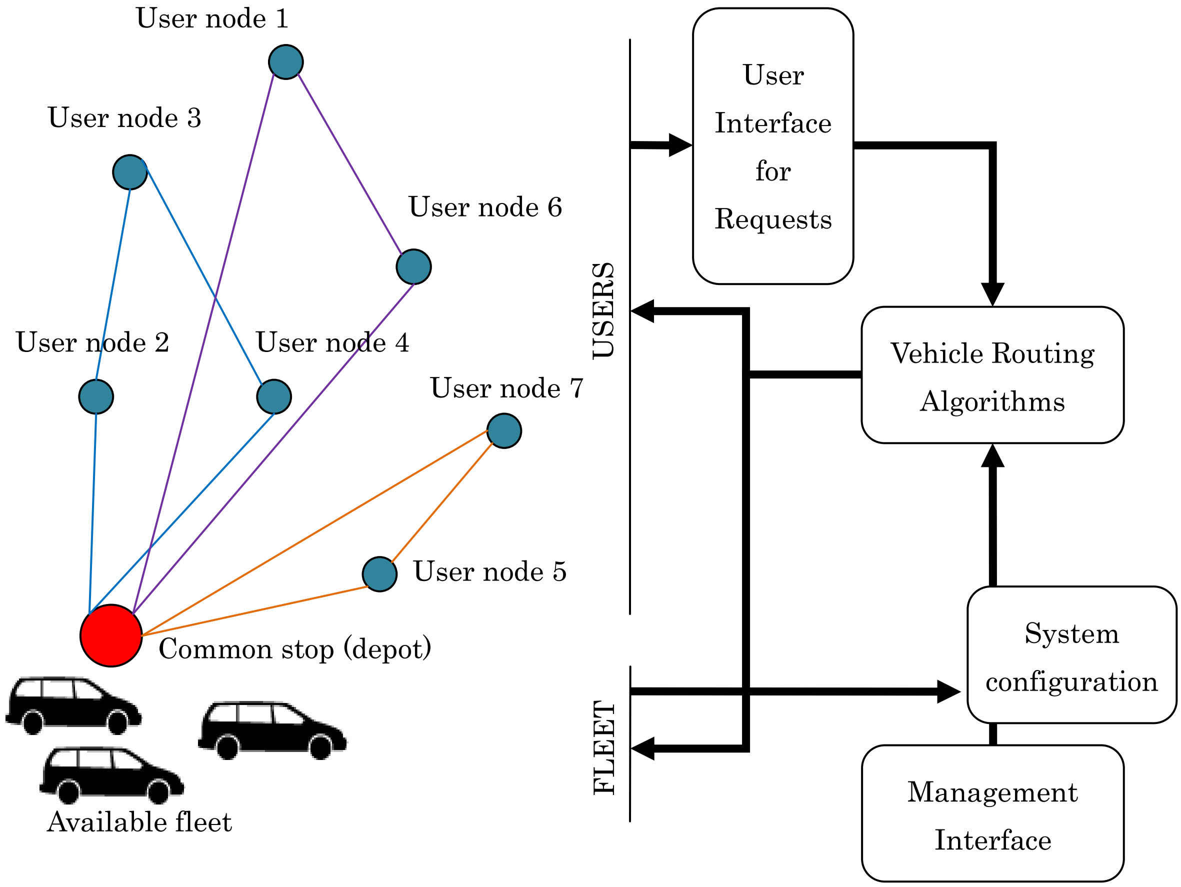

Fig. 1. General architecture

DESIGNED ALGORITHMS

The case in this paper is a special Capacitated VRP with Maximum Ride Time, as constraints must take into account the quality of service that human passengers require. A single depot is used (the common stop for all users). For a common destination, all the drivers leave from the depot and come back to the same point after having picked up all the passengers who have required the service. In the case of the common origin passengers are delivered at their stops. Each requirement is assisted by a vehicle, only once, which means that, if the capacity of the vehicle is not exceeded, all the passengers in the same stop travel in the same vehicle. The quality of service is measured by the maximum ride time or travel time rate (MRT > 1) of a passenger. The ride time for each user cannot exceed a percentage of the time required to complete the direct route between the user-defined stop and the depot. Relaxations like the use of symmetric cost matrixes cannot be applied, as one-way roads can be found in real life.

Under this scenario, three kinds of algorithms have been analyzed in order to determine the best way to solve this problem.

LINEAR PROGRAMMING ALGORITHMS

Linear Programming is a technique for the optimization of a linear objective function, subject to linear equality and linear inequality constraints (see (8)). In the considered case, the function to optimize (minimize) is the number of vehicles required for serving all the requests. Nevertheless, due to computational effort, the network to optimize is usually small and therefore only small instances of the problem can be considered. This is the reason why the results have not been presented in this paper and other types of algorithms to solve the problem have been considered.

HEURISTIC ALGORITHMS

The heuristic algorithms perform a limited exploration of the search space and typically produce good solutions quickly. They are relatively simple to implement and can be easily adapted to include the constraints encountered in real applications (9), (10). There are several categories:

- Constructive algorithms. They create new solutions by repeatedly adding to an empty solution until it is fulfilled.
- Two-phase algorithms. The problem is decomposed in its two natural parts, that is, grouping clients in feasible routes and planning routes.
- Improvement (or local search) algorithms. They start from an initial solution and try to improve it by making small changes such as relocating or exchange customers to invest sequences customers. Local search algorithms are of great importance and such is their utility that they are incorporated in many construction methods and are the basis of many metaheuristics.

In this paper local search algorithms have been used, specifically genetic algorithms. These types of algorithms have several steps. First, a fixed number of solutions are built randomly. They form a population. For a given number of vehicles, a solution specifies which stops each one visits in which order. In this case, solutions are forced to fulfil all the restrictions except for the travel time restriction. While the time restriction is not fulfilled, new generations of solutions are created. New generations creation is stopped when calculation time limit is reached. Finally, the best solution is chosen among the members of the last successfully created generation. That is the one that gives the best travel time proportion. This process is shown in Fig. 2.

To create a new generation, some members of the population have to be chosen, they must be combined somehow and mutation can occur. The result is then inserted into the population. Required steps that have to be developed to create a new generation are:

- Solutions choice: two solutions A and B are picked randomly.

- Combination: the first half of the stops in each car in solution A is combined with the second half of stops in the respective car in solution B. The same technique is applied in the other way round.
- Mutation: some stops are picked randomly and interchanged.
- Insertion: if the new solutions have a better proportion in the time restriction, they are added and the worst ones are removed. Otherwise, population is not changed.

Each time a new solution is created, local search is applied on it. That is, the best order of the stops in each car is searched.

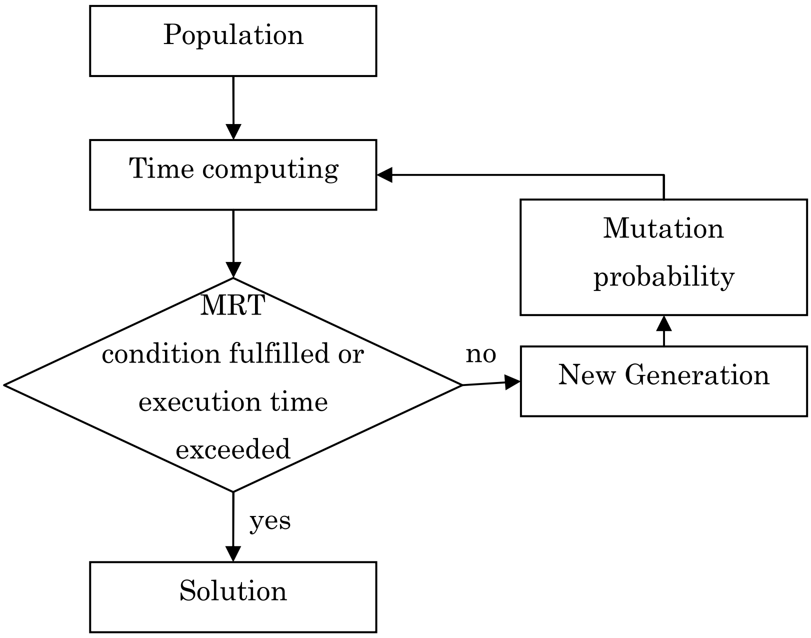

Fig. 2. Basic schema of the genetic algorithm.

ROUTING ALGORITHMS

This kind of algorithms usually study the routing problem as a graph theory problem in order to find the optimal solution. In the considered case, the optimal solution is that which minimizes the number of vehicles needed to meet demand, fulfilling the restrictions set. Furthermore, for the same number of vehicles, a second optimization is developed, in order to reduce as much as possible the time that passengers remain in the vehicles.

The first step is to calculate a priori the minimum number of vehicles needed to serve all passengers, considering only restriction the maximum capacity of each vehicle. This value is subsequently used as a reference for comparing the partial results obtained in each iteration of the algorithm. It is considered an initial stop ordering, it has been shown empirically that the optimal order a priori is the increasing order, taking into account the distances between stops.

Subsequently, the algorithm is designed for picking up the passengers waiting at each stop (for the established order) provided that the restrictions imposed initially are fulfilled. If one of the restrictions is not fulfilled, the following stop is considered attending the previously considered order. Once the solution is reached, it is necessary to determine whether the number of vehicles exceeds the minimum value originally estimated. If so, a new iteration of the algorithm has to be performed, considering a new order of initial stops and repeating all the steps previously explained. The algorithm ends when either the maximum time simulation or the minimum estimated number of vehicles is reached. In the following Fig. 3 the basic schema explaining the algorithm basis is shown.

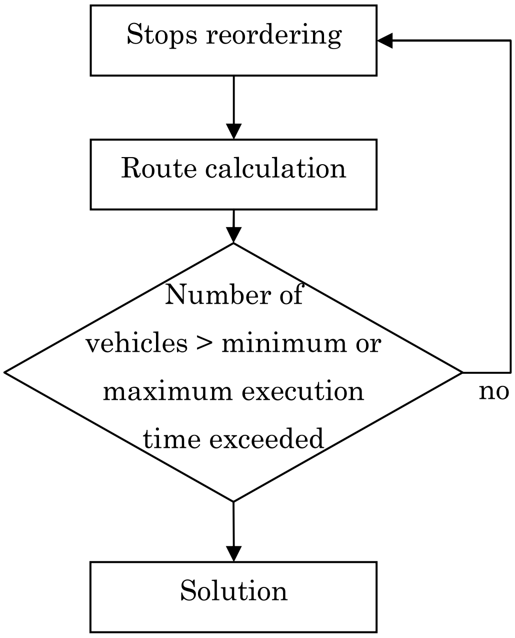

Fig. 3. Basic schema of the designed routing algorithm.

RESULTS

In order to analyze the performance of the studied algorithms, the following parameters have been used for testing:

- Distance matrix: Matrix A contains distances between 50 nodes uniformly spread over a surface.
- Requests: different random requests have been generated for 9, 19, 29, 39 and 49 user nodes among all the 50 existing nodes. One of the nodes for the algorithm will be the destination stop, always the same, implying no passenger is picked up there. To simplify things, in each requested node a unique passenger is assumed.
- Vehicle capacity: vehicles of 4, 7 or 50 passengers have been considered.

- Calculation time: different algorithms have been simulated during 1 minute, 10 minutes, 20 minutes. Results after that time (one hour) have also been reported in some cases if found significant.
- Maximum travel time rate (MRT limit): this parameter has been fixed at 1.4 (direct travel time exceeded in %40)

As it has been mentioned before, results for Linear Programming are not shown in this paper. Some simulations have been developed for a small number of nodes and passengers in each vehicle, but as these values increase performance gets much worse. Results referred to other two algorithms are presented. In the charts shown below, the integer numbers after the name of the algorithm (GA-12 or RA-11, e.g.) refer to the number of vehicles needed. The MRT values refer, in the same manner, to the highest travel time rate assigned to a passenger. Notice that the number of nodes in the algorithm is the number of service requests (plus the common stop) considering one passenger in each one.

Routing algorithm (RA)

As it can be seen in the following charts (Fig. 4 to Fig. 6), results obtained during the first minutes do not vary substantially. On the other hand, during the performed tests, there has not been found any variations in the solutions by increasing the capacity of the vehicles from 7 to 50 passengers. The reason is that no vehicle has had time to serve more than 7 passengers. Thus, the restrictive parameter in this case is the maximum travel time, not the number of passengers in the vehicle, and therefore, high-capacitated vehicles are not likely to offer important improvements.

Genetic algorithm (GA)

The described genetic algorithm with population of 10 individuals and the operating parameters stated above were used to complete the tables. Notice that not all the values shown in the charts are valid as good solutions, since travel time rates above 1.4 are not accepted (see restrictions of the considered problem). This algorithm needs a number of cars to be specified before running, if it cannot find any feasible solution with them, a new car is added and begins to run again. The minimum number directly depends on the capacity of the vehicles and the user requests, but that value might be, in general, utopian, wasting valuable calculation time. As the execution starts from a random initial population that does not comply with maximum travel time rules, significant time must be spent before receiving valid solutions. Since the result obtained with RA is assumed to fulfil all the restrictions, executions have been run at first for vehicle numbers obtained in RA and the whole maximum calculation time of 20 minutes have been applied to each run. In cases showing better results, executions have been repeated using less vehicles. On the contrary, for non-succeeding results, executions have been repeated adding more vehicles. In these cases, the best solution considered is the one obtained by RA. Additionally, making the algorithm run for 60 minutes in some executions improvements were found. For example, referring to results in Fig. 4, the travel time rates, in the case of considering 50 nodes, were reduced to 1.54106 for 12 vehicles, 1.44221 for 13 vehicles and 1.3333 for 14 vehicles.

For nodes below 40 the GA provides better solutions than the RA one in most cases after 20 minutes, needing 2 cars less after certain executions (30 nodes in Fig. 4 and Fig. 5). When requests are more numerous (more than 40), it is shown that 20 minutes cannot be enough to match the results of RA. In addition, some executions were not possible to perform because the time needed to generate an initial population was excessive (GA-13, GA-14 in Fig. 5 and GA-12 in Fig. 6, for 50 nodes).

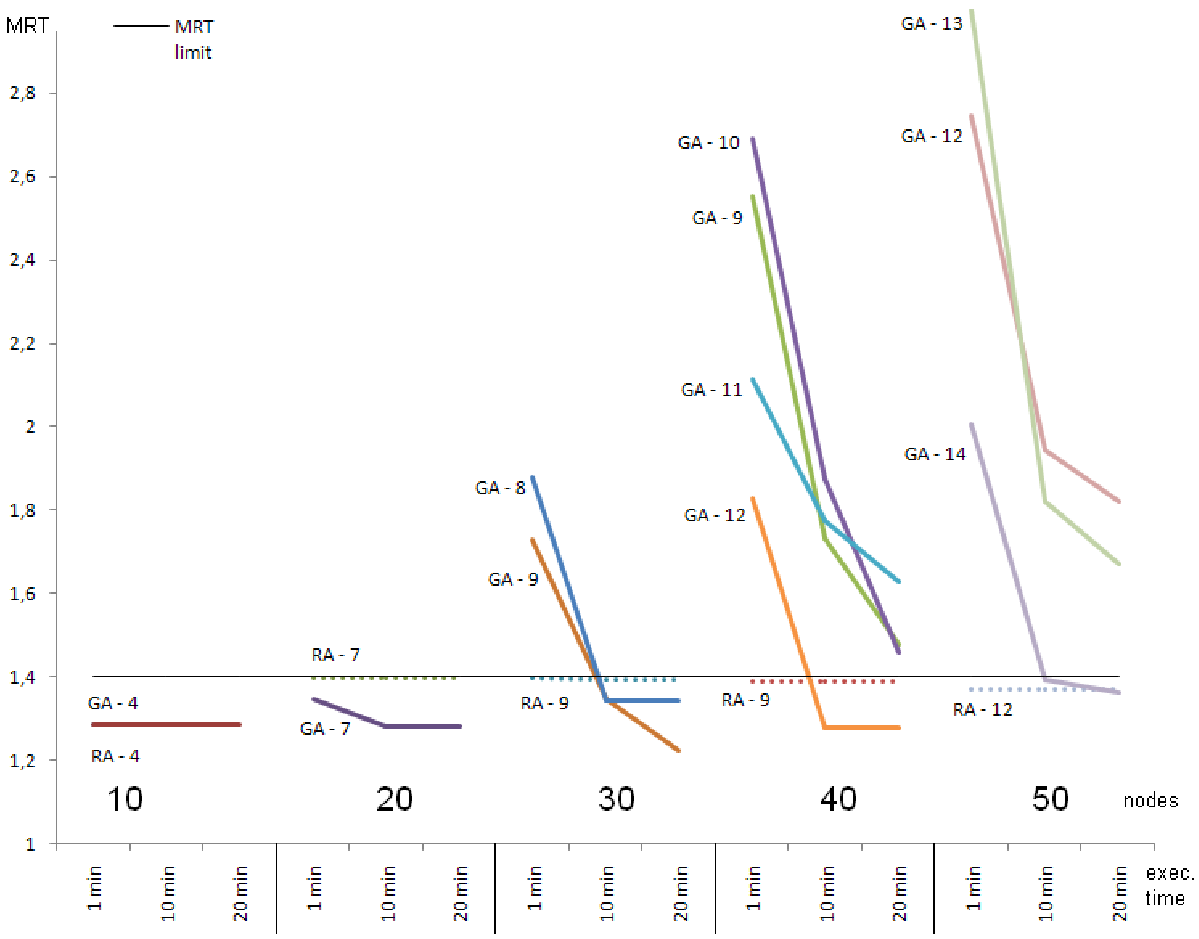

Fig. 4. Highest MRT rates according to algorithm and number of 7-passenger vehicles.

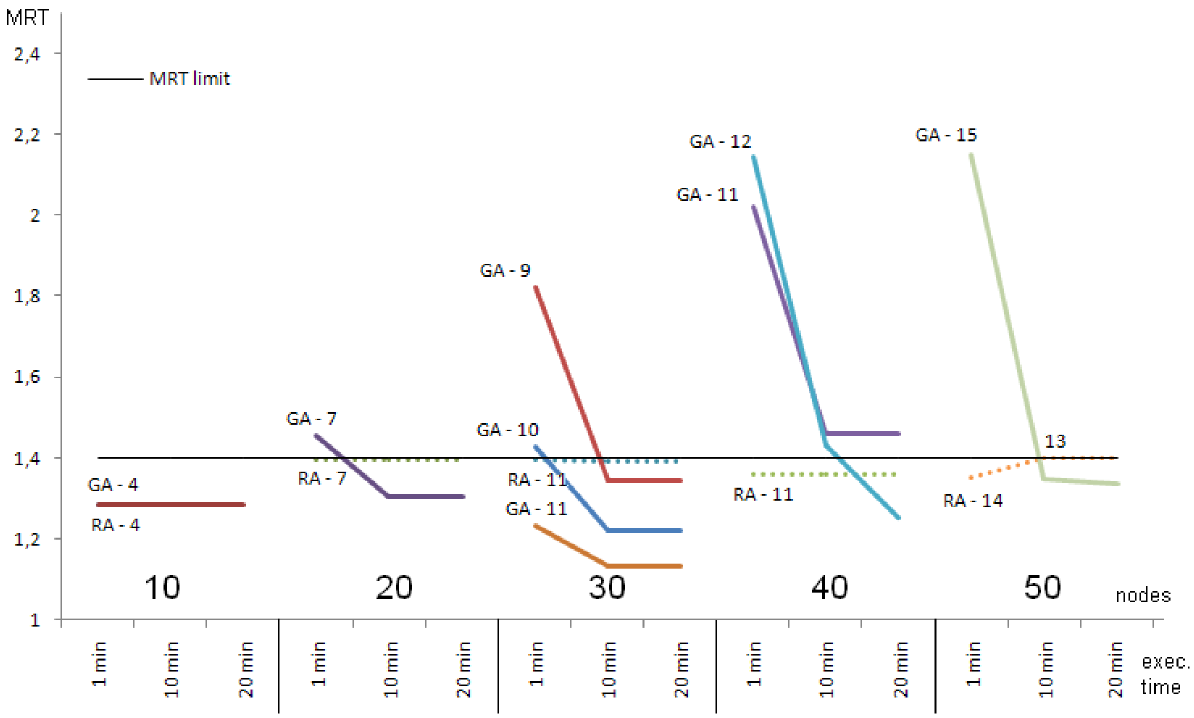

Fig. 5. Highest MRT rates according to algorithm and number of 4-passenger vehicles.

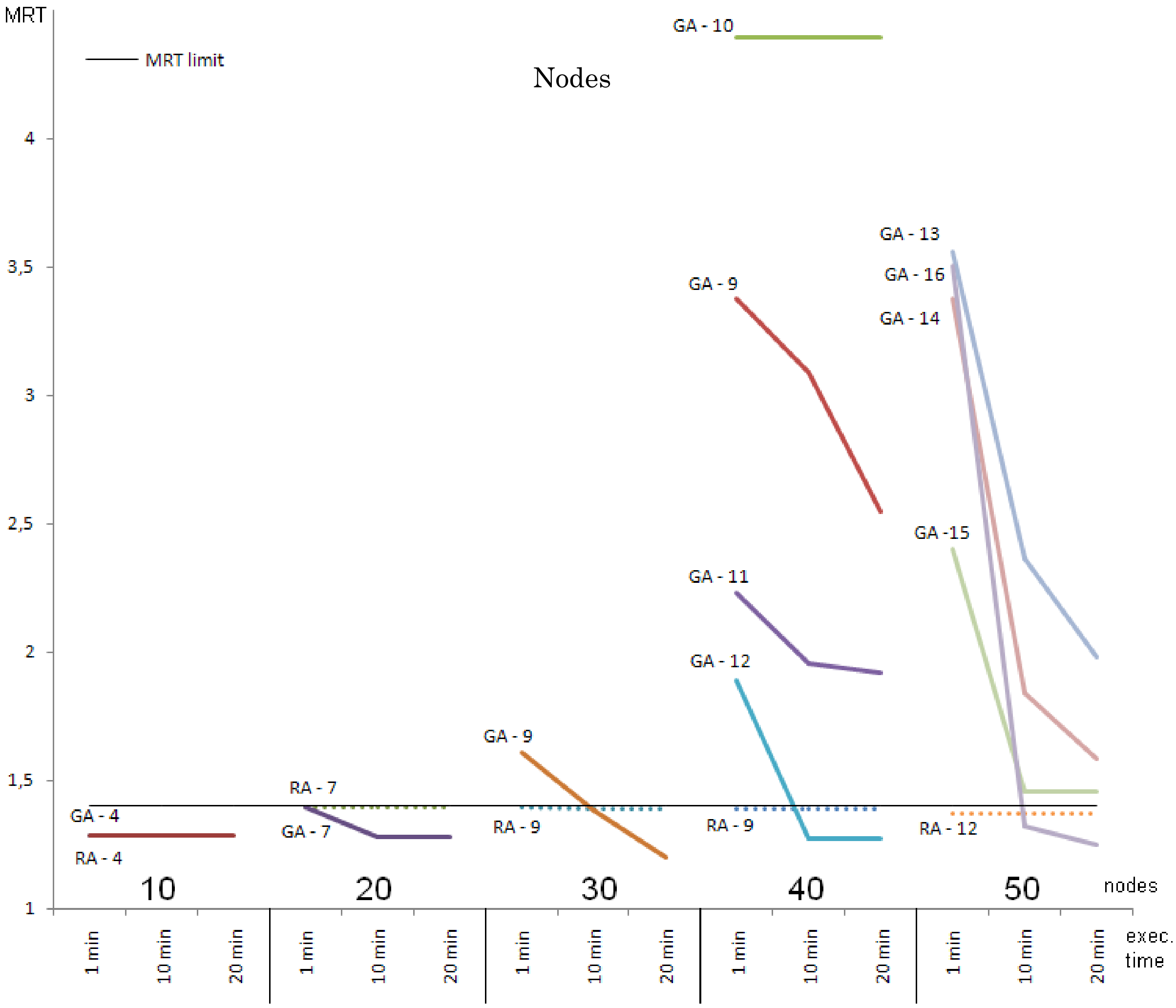

Fig. 6. Highest MRT rates according to algorithm and number of 50-passenger vehicles.

CONCLUSIONS

A new mobility solution is presented to promote greater sustainability and encourage a responsible use of road transport, providing many advantages of the private car. Three different kinds of algorithms used to solve routing problems have been introduced, based on: Linear Programming, Genetic Algorithms and Routing Algorithms. Since the amount of data is high, Linear Programming has been discarded for this study and experimental results obtained by the other two algorithms have been presented.

It is concluded that this two algorithms can be combined to take advantage of their characteristics. Algorithm RA offers fast solutions that can be acceptable in some cases where real time response is required even for large amounts of requests. It provides in a short execution time (1 minute) a solution which fulfils all the restrictions. However, the amount of cars needed to serve all the requests or the travel time that users spend on route can sometimes be improved. Algorithm GA takes longer to give good solutions; but it is suitable for those scenarios explained in this work, when travelling requirements of users are known in advance, obtaining solutions in real time may not be a priority. On the other hand, when the number of nodes is too high to achieve an improvement with the GA in the considered execution time, the solution given by the RA is perfectly assumable. Consequently, as a global solution, an initial result achieved with the RA feeds the GA, effectively spending the available calculation time on feasible or near-feasible options.

The optimization of the services, therefore, is closely related to the immediacy required in the response. User requests that need to be served in short future force the system to change the already generated solutions if the vehicles are already on route. If the location of the vehicles is known by the system, the best solution possible must take into consideration the stops left to visit, and try to reeschedule them fast with RA in order to include the new request if possible.

ACKNOWLEDGEMENT

The authors would like to thank the Gipuzkoa Provincial Council, which has funded on demand transport project in Aia, constituting a starting point for this work. On the other hand, this work has been partially supported by Basque Government, Etortek call (IE08-221).

REFERENCES

(1) Lawler, E., Lenstra, J., Rinnooy Kan, A. and Shmoys, D, *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization.* Wiley, New York, 1985.

(2) Huisman, D. and Wagelmans, A., “A solution approach for dynamic vehicle and crew scheduling”, *European Journal of Operational Research*, vol. 172, no. 2, 2004, pp. 453–471
(3) Horn, M., “Fleet scheduling and dispatching for demand-responsive passenger services”, *Transportation Research*, vol. 10, 2002.
(4) Desrochers, M., Desrosiers, J. and Solomon, M., “A new optimization algorithm for the vehicle routing problem with time windows”, *Operations Research.*, vol. 40, no. 2, 1992, pp. 342–354.
(5) Faulin, J., Gilibert, M., Juan, A.A., Vilajosana, X. and Ruiz, R., “A Simulation-Based algorithm for the capacitated vehicle routing problem” in *Proceedings of the 2008 Winter Simulation Conference,* S. J. Mason, R. R. Hill, L. Moench, O. Rose, eds. IEEE, 2008.
(6) Savelsbergh, M. and Sol, M., “The General Pickup and Delivery Problem” *Transportation Science*, vol. 29, pp. 17–29, 1995.
(7) Garcia, A., Arbelaitz, O., Otaegui, O., Vansteenwegen P. and Linaza, M., “Public Transportation Algorithm for an Intelligent Routing System”, in *Odysseus Fourth International Workshop on Freight Transportation and Logistics*, 2009.
(8) Kara, I. and Bektas, T., “Integer Linear Programming Formulation of the Generalized Vehicle Routing Problem”, *5th EURO/INFORMS Joint International Meeting*, vol. 41, July 2003.
(9) Thompson P. and Psaraftis, H., “Cyclic transfer algorithms for multivehicle routing and scheduling problems”, *Operations Research*, vol. 41, no. 5, pp. 935–946, 1993.
(10) Brysy, O. and Gendreau, M., “Metaheuristics for the Vehicle Routing Problem with Time Windows”, *Transportation Science*, vol. 39, no. 1, pp. 119–139, 2005.