A METAHEURISTICS BASED SIMULATION TOOL TO OPTIMIZE DEMAND RESPONSIVE TRANSPORTATION SYSTEMS

Eneko Osaba, Pablo Fernandez
Mobility Unit, Deusto Technology Center, Av. Universidades, 24, Bilbao, Spain
eneko.osaba@deusto.es, pablo.fernandez@deusto.es

Roberto Carballedo, Asier Perallos
Department of Software Engineering, University of Deusto, Av. Universidades, 24, Bilbao, Spain
roberto.carballedo@deusto.es, perallos@deusto.es

Keywords: Demand simulation tool, intelligent transport, demand responsive transport, evolutionary computing.

Abstract: Public passenger transport is an important area that affects our quality of life. The design of routes and frequencies of such systems is very important to ensure their economic viability. The work presented in this paper focuses on the design of a software tool that assists in the creation of routes and schedules of passenger transport systems. For this, we have developed an application that is based on evolutionary computing techniques to simulate passenger demand and adjust the routes and frequency of the services to meet those demands. The result of work done is a software tool, and a metaheuristic algorithm that can be used for solving optimization problems.

1 INTRODUCTION

There are different kinds of public transportation systems, each one with its own features but they all share some disadvantages. One major disadvantage of the conventional transportation systems is the lack of resources to satisfy all the users' demands. In rural areas and places with low demographic density, the existence of regular public transportation systems is not profitable. There are some areas where public transportation vehicles do tours and stops without picking or delivering any passenger. If this situation is constantly repeated the route will eventually be eliminated. Whenever the supply of public transport services in an area is reduced, we could say that the quality of life of people living in that region decreases. To avoid this loss of quality of life, we should analyze the demand of passengers, and to adapt transportation systems to passenger demand. This is essentially the concept of transport systems on demand..

Transportation-On-Demand (TOD) (Jorgensen, Larsen and Bergvinsdottir, 2007) is concerned with the transportation of passengers or goods between specific origins and destinations at the request of users. Most TOD problems are characterized by the presence of three often conflicting objectives: maximizing the number of requests served, minimizing operating costs and minimizing user inconvenience. As is common in many combinatorial optimization problems, these objectives are conflicting and it is needed to sort them by importance.

Many of the techniques used to solve these problems do not yield an exact solution. This is because the types of problems to be solved are classified as NP-hard (Garey and Johnson, 1990). For this reason, heuristics techniques are used for obtaining good approximations.

In this work an algorithmic solution to this problem has been developed. This solution consists in a hybrid algorithm combining two evolutionary computing techniques: a Genetic Algorithms and a Simulated Annealing algorithm.

Besides the algorithm, a simulation tool has been developed. The tools is oriented to the bus public transport, it allows the user to create virtual environments with dynamic bus stations, stops and clients requests. The tool is able to obtain the best route through several bus stations that satisfies passengers' requests. The design of the route is made by means of the hybrid algorithm mentioned.

The aim of the new tool is to support the decision to create or delete a particular route. The developed simulation tool has also been used to validate the algorithmic solution proposed.

This paper is structures as follow: the first section makes a brief state-of-the-art about the most common transportation problems and the existing traffic simulation systems. Then the simulation tool is presented as well as the proposed algorithms. The final solution is presented in the next section and the development of the hybrid algorithm is explained. Finally the conclusions and the future work is presented.

2 CONTEXT

2.1 Transportation problems

Most of the problems arisen in transportation on demand topic have similar characteristics, which means that they can be framed as instances of other generic and well know problems. In this section, we present the most common traditional problems in the field of transportation on demand.

Travelling Salesman Problem (TSP) (Applegate, Bixby, Chvátal and Cook, 2006): The Travelling Salesman Problem (TSP) is an NP-hard problem in combinatorial optimization studied in operations research and theoretical computer science. Given a list of cities and their pair-wise distances, the task is to find a shortest possible tour that visits each city exactly once. This type of problem is used as a benchmark for many optimization algorithms.

Vehicle Routing Problem (VRP) (Dantzig and Ramser, 1959): The vehicle routing problem (VRP) is a generalization of the TSP. The aim of the problem is to service a number of customers with a fleet of vehicles. Often the context of this type of problem is related to deliver goods located at a central depot to customers which have placed orders for such goods. Implicit is the goal of minimizing the cost of distributing the goods. Many variants of the VRP are described in the literature (Pisinger and Ropke, 2007). These problems include the addition of variables and constraints. One of the most popular variants includes time windows for deliveries. These time windows represent the time within which the deliveries (or visits) must be made. (Repoussis, Tarantilis and Ioannou, 2009)

Demand Responsive Transport (DRT): Demand Responsive Transport or Demand-Responsive Transit (DRT) or Demand Responsive Service is an advanced, user-oriented form of public transport. It is characterized by flexible routing and scheduling of small/medium vehicles operating in shared-ride mode between pick-up and drop-off locations according to passengers needs. DRT systems provide a public transport service in rural areas or areas of low passenger demand, where a regular bus service may not be economically viable. DRT systems are characterized by the flexibility of the planning of vehicle routes. These routes may vary according to the passenger' needs in real time. This is the type of problem that we used to benchmark the algorithm proposed in this paper.

2.2 Traffic Simulation Tools

The simulation tools help us to measure the performance of a system in a virtual environment. This is a usual practice before making a large scale change in an existing platform because there are involved a lot of costs associated with this process. Another major feature of the simulation tool is the ability of predicting the behaviour of one environment in a specific context.

Talking about simulation systems in the transportation context is talking about prediction and traffic optimization. One of the main goals in these simulations systems is predicting the effects and the impact of a vehicle or road accident in the regular traffic as stated in (Burghout, Koutsopoulos, and Andreasson, 2010), this is a critical fact in places where the traffic density is high because a vehicle accident can cause several inconveniences to the users of the road.

Another important goal of the transportation related simulation systems is the traffic light calibration. Thanks to the traffic simulators the agents are able to setup the traffic lights according to the simulated traffic and other natural factors like the weather or the time of the day as written in (Lopez-Mellado, 2010), in order to optimize the light cycles.

3 SIMULATION TOOL

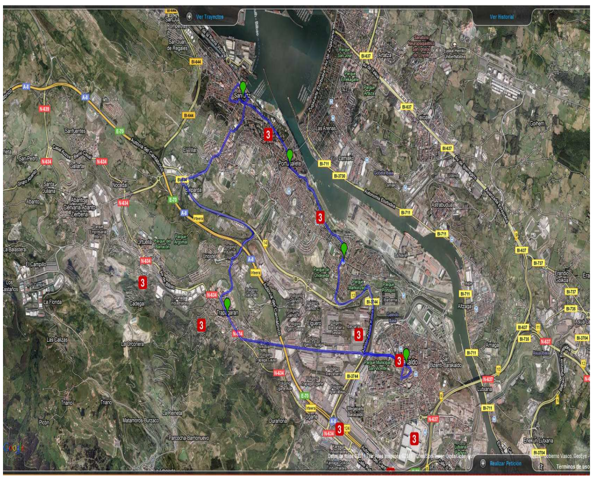
Nowadays there are many kinds of public transportation systems: on demand or regular ones. This requires the public transport to be more demanding and sophisticated. But before performing variations to the regular service, by adding more transport units, new stops or creating new routes or lines, it is necessary to make preliminary studies to with the objective of improving the quality of service and the client satisfaction without making mistakes. To achieve this excellence level, different techniques and tools are used and applied before deploying the changes, in order to guarantee that the people in charge will be able to have an idea of the impact of the new variations.

In this work it has been developed a simulation tool, oriented to bus public transport system (on demand or regular), in rural or urban environments. With this tool, different tests can be made to check the performance improvement when adding or removing a new stop to/from a route, or the creation of a new line.

With this tool the user will be able to create different environments composed of stations, that user may place wherever he desires. The application will calculate the optimized route to navigate through all the placed stations using the artificial intelligence algorithm developed that will be further explained later. Once the route is created, the user may make requests, which the system will manage on an efficient way. The system will also be able to make a simulation of how the bus would be completing the route and managing the dynamic requests made in real time.

It's important point out the difference between primary and secondary stations. The primary stations are the ones the bus must pass through, the secondary are the ones that will only be part of the route if it exists enough demanding from the users. This feature allows to create on demand transport lines, and helps to select which stations will be primary or secondary.

Figure 1, shows an image of main interface of the tool developed. That image represents an environment composed by some primary and secondary stops.



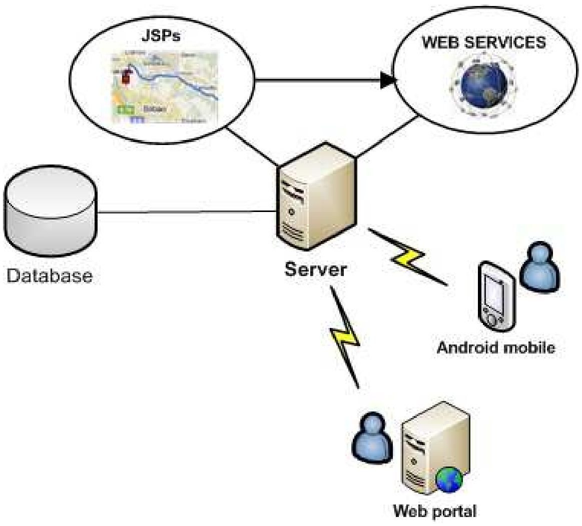
low

Figure 2, shows the application's main screen. Several deployable tabs can be seen. These tabs contain diverse information. The top ones for example, have information about the route of each of the buses and the history of the active bus. The bottom one contains a history of all the requests made, a panel to make transportation requests and some controls to manage the simulation of the bus tour.

'ed Tabs

Apart from this, a mobile application has been developed. Thanks to this, the users can kwon the active bus route, in map or text format. Furthermore, it also offers the possibility of making requests in the same way as the web application.

To finish, in Figure 3 a conceptual schema of the final system architecture is shown.



chitecture

As mentioned before, there exist two ways to use the simulation tool. One is based on a web application and the other one is an android based mobile application. These two alternatives access to a central server via internet, this server can be divided into three different components. The first one is a database where all the information about the made requests, the active routes and the statistics are stored. The second component is composed by some web services that offer several access methods to the database and business logic. The web services are consumed by some Java Server Pages (JSP), the third component of the server. These JSP pages represent the presentation layer and user access to both web application and mobile application.

4 PROPOSED ALGORITHMS

As mentioned in the preceding section, the implemented simulation tool uses an artificial intelligence algorithm as base. This algorithm is used to resolve the route planning problem, in a static way in regular transport lines, as well as in a dynamic way, in on demand transport lines. In application level, the algorithm will be in charge of finding in every moment the optimized route the bus has to go through to visit all the stations of the environment. To implement the solution, the problem has been treated as a DRT one, as mentioned in the introduction of the article, and resolved with a heuristic method that obtains good approximations. To perform this task, we designed a hybrid algorithm that combines simulated annealing methods and genetic algorithms. Then we explain the details of each technique separately.

Simulated Annealing (Rutenbar, 1989): This is one of the most popular local search techniques. It is based on the physical principle of cooling metal. Using that analogy, it generates an initial solution and the process proceeds by selecting new solutions randomly. The new solutions are not always better than the initial solution, but as time passes and the temperature decreases (the metal becomes stronger), each new solution must be better than previous solutions.

Genetic algorithm (Zhang, Yao and Zheng, 2009): This algorithm is inspired by the laws of natural selection and the evolution of the animal species. An initial population of solutions is defined. This initial population consists of a number of individuals (solutions of the problem.) Then, with the combination and evolution of these individuals, the algorithm tries to get a better solution.

Hybrid Algorithm (Kaur and Murugappan, 2008): The hybrid algorithm is the resultant from joining the genetic and the simulated annealing algorithms. It was decided to use this algorithm after a test period detailed in the following section.

5 TEST AND FINAL SOLUTION

In the field of artificial intelligence, when you create a new algorithm, or modify an existing one, results that show that the new solution is better than other known solutions have to be submitted. To do this, there are sets of problems used to compare results in computational resources and parameters related to the quality of the solution. In our case, there is not a set of sample problems for comparing the performance of design algorithms for demand responsive transport routing problems. For this reason, we have defined our own set of testing, and we are going to make comparisons between our algorithm, a brute-force optimal algorithms and each of the techniques used in our hybrid algorithm.

As indicated above, for the design of our hybrid algorithm, separate versions of simulated annealing and a genetic algorithm have been implemented. In addition, we have implemented a "brute force" algorithm, to find out the optimal solution for small instances of the problem (with few intermediate stops).

With these 3 algorithms, there have been a series of tests to measure the performance of each algorithm and the ability of each one to solve the problem. As a result of these tests, we have obtained several conclusions:

1. The “brute force” algorithm is optimal because it always finds the best solution. Even so, it has the disadvantage that the execution time is unacceptable when the number of stations increases to more than 9 (for a large number of stations cannot even get a solution). This algorithm cannot be used in a real scenario.
2. The simulated annealing algorithm only finds the optimal solution when the first and last station does not vary during the resolution process. Running time is always the same regardless of the number of stops.
3. In the case of genetic algorithm, the execution time is constant if the number of generations is also constant. An advantage of this algorithm is that the probability of finding a good solution is independent of the number of stops.

After preliminary analysis of algorithms separately, we came to the conclusion that the results of runtime and solution quality were not good. For this reason we decided to combine the two heuristics.

5.1 Our hybrid algorithm

Our hybrid algorithm came up with the aim to combine the advantages of genetic algorithms and simulated annealing:

- Rapid and constant execution time (simulated annealing).
- Probability of finding a good solution for the problem instances with many stops (genetic algorithm).

The solution would avoid the main drawback of the two algorithms:

- The solution should be optimal or very close to it.

With all these goals, it thought about making the hybrid. By nature of the two algorithms, it is appropriate to insert the execution of simulated annealing algorithm in the execution of genetic algorithm. That is because the first algorithm is focused on only one solution and the second works simultaneously with different solutions.

Having decided the model of integration, there were two options to do the integration:

- Integrate the simulated annealing in the process of creating the initial population.
- Integrate the simulated annealing in the process of reproduction, right after generating the new population.

After several tests, we concluded that the most effective solution was to apply the simulated annealing algorithm just after the reproduction process. Below is a table showing the results of the tests. The table shows the number of stops, the number of generations used in the genetic algorithm, runtime, and the percentage of times the algorithm finds the optimal solution.

Table 1: Results of the tests.

Stations	Generations	T. of execution	% of optimal solution
9	5	3 seconds	80%
9	10	5 seconds	100%
10	5	3 seconds	80%
10	10	5 seconds	100%
11	5	3 seconds	80%
11	10	5 seconds	100%

Comparing the proposed alternative with each of the separate algorithms, we can ensure that the execution time is right, regardless of the number of stops. Moreover, in situations where the optimal solution is not found, the average deviation for the optimal solution does not exceed 3% of the value of the optimal solution.

6 CONCLUSIONS AND FURTHER WORK

The work presented is the result of a research project funded by the Basque government. The project focuses on the design of a software tool that assists in the creation of routes and schedules of passenger transport systems. For this, we have developed an application that is based on evolutionary computing techniques to simulate passenger demand and adjust the routes and frequency of the services to meet those demands. The result of work done is a software tool, and a metaheuristic algorithm that can be used for solving optimization problems.

A part of the results of this project, and as there is no benchmark problems for this type of transportation problem we are currently developing a framework for algorithm validation. This new framework will serve to validate the performance of demand responsive routing algorithms. During the design of this benchmark problems we will define the basic criteria to measure algorithms performance. Examples of such criteria are running time, solution quality, ease of implementation, robustness and flexibility.

7 REFERENCES

R M Jorgensen, J Larsen and K B Bergvinsdottir: Solving the Dial-a-Ride problem using genetic algorithms. Journal of the Operational Research Society, 58, pp. 1321–1331. 2007

Garey, M. R. and Johnson, D. S: Computers and Intractability; a Guide to the Theory of Np-Completeness. W. H. Freeman & Co. 1990.

Applegate, D. L.; Bixby, R. M.; Chvátal, V.; Cook, W. J. The Traveling Salesman Problem. ISBN 0691129932. 2006

Dantzig, G.B.; Ramser, J.H. The Truck Dispatching Problem. Management Science Vol. 6, No. 1, October 1959, pp. 80-91.

D Pisinger, S Ropke: A general heuristic for vehicle routing problems. Computers and Operations Research, Volume 34 Issue 8, August, 2007.

Repoussis, P.P.; Tarantilis, C.D.; Ioannou, G.: Arc-guided evolutionary algorithm for the vehicle routing problem with time windows. IEEE Transactions on Evolutionary Computation, Volume 13 Issue 3, June 2009.

Burghout, Wilco; Koutsopoulos, Haris N; Andreasson, Ingmar: Incident Management and Traffic Information Tools and Methods for Simulation-Based Traffic Prediction. Transportation Research Record, Issue 2161, pp. 20-28, 2010

Lopez-Mellado Ernesto: A modeling framework for urban traffic systems microscopic simulation. Simulation Modelling Practice and Theory, Volume 18, Issue 8 pp. 1145-1161, Sep 2010

Rutenbar, R.A.: Simulated Annealing algorithms: an overview. IEEE Circuits and Devices Magazine 5 (l), pp. 19-26. 1989

L Zhang, M Yao, N Zheng: Optimization and improvement of Genetic Algorithms solving Traveling Salesman Problem. Image Analysis and Signal Processing, pp. 327-332 April 2009.

Kaur, D. Murugappan, M.M: Performance enhancement in solving Traveling Salesman Problem using hybrid genetic algorithm. Fuzzy Information Processing Society, pp. 1-6, May 2008.