A Genetic Algorithm for solving the multicriteria routing problem in public transit networks

O. Dib[1], M. Manier[2], L. Moalic[3] and A. Caminada[4]

1. IRT SystemX, 8 Avenue de la Vauve, 91120 Palaiseau France
omar.dib@irt-systemx.fr

2. OPERA – UTBM, Belfort France
{omar.dib ; marie-ange.manier ; Laurent.moalic ; alexandre.caminada}@utbm.fr

Keywords: Multimodal networks, Multicriteria shortest paths, Genetic algorithm.

1 Introduction

Route planning in urban public transport systems constitutes a common decision problem faced by travelers. As in real life, commuters do not only seek short time travels. However, they tend to consider other elements into their journeys such as monetary cost, comfort (quality of mode) and effort (walking distance, number of transfer, waiting time…). Therefore, there is a real need to develop a seamless application that provides passengers with efficient itineraries according to their needs and preferences.

Routing applications whether they arise in transportation area or other domains such as communication networks refer for solving Shortest Path Problems (SPPs). While solving some routing problems can be done in a straightforward manner, computing shortest paths under certain circumstances is not always an easy task. For instance, solving the one-to-one SPP in static networks can be easily accomplished by applying the well-known algorithm of Dijkstra. On the other side, computing multicriteria shortest paths appears to be more difficult especially in large-scale dynamic networks. Computing itineraries w.r.t several criteria refers to the Multiobjective Shortest Path Problem (MOSP), a fundamental problem in the field of multiobjective optimization. Solving the emerging problem consists of finding the set of non-dominated journeys from which the user chooses his/her most preferred one. Given two journeys j_1 and j_2, we say that j_1 dominates j_2 if there is at least one criterion for which j_1 has a better value than j_2 and there is no criterion for which j_2 has a better value than j_1. A journey j is then called Pareto-optimal if it is not dominated by any other journey. The main difficulty in multiobjective contexts stems from the fact that, in many optimization problems, determining the entire set of nondominated solutions is a tedious task since one problem may have a huge number of nondominated solutions (even in case of two objectives). Additionally, and in contrast to single criteria search, one cannot abort the search after finding a first optimal solution. Even after finding all Pareto-optima, search algorithms require a substantial amount of time to guarantee that no further solution exists. Therefore, in many optimization problems, especially those requiring real time answers, we do not focus on finding the optimal Pareto solution set. Rather, we try to use approximate methods whereby we compute near optimal solutions in reasonable computational time.

The Genetic Algorithm (GA) that belongs to the population-based metaheuristics and was introduced by Holland in 1987 [1] is one of the approximate approaches that has been efficiently used for solving a wide range of multicriteria problems. For instance, [2] used a GA to solve the one-to-one SPP in large-scale road networks. [1] and [4] worked with evolutionary algorithms to compute single source shortest paths using single-objective fitness. [5] proposed a GA to find shortest paths in computer networks. [6] and [7] also worked with GAs to find shortest paths in data networks. [8] proposed effective crossover operators for the all-pairs SPP. In this work, we develop a GA for solving the multi-criteria routing problem in time-dependent public transit network. As optimization criteria, we use travel time, travel cost, number of transfers and total walking time. As transportation modes, we use Railway, Bus, Tram, Metro and pedestrian. We assess the performance of the proposed GA by solving a wide range of real-world routing queries defined on the public transport network of the City of Paris and its suburbs. We compare the performance of the introduced GA with an exact multi-criteria shortest path algorithm. Experimental results indicate that the proposed GA is efficient enough to give near optimal solutions in reasonable computational time.

2 Modeling Approach

This section considers modeling a time-dependent public transit network. It should be clarified that the term "transport network" is used in the sense of multiple fixed scheduled transport services. A key difference to static networks is that public transit networks are inherently time-dependent, since certain segments of the network can only be traversed at specific, discrete points in time. As such, the first challenge concerns modeling the timetable appropriately in order to enable the computation of journeys. Roughly speaking, a timetable consists of a set of stops (such as bus stops or train platforms), a set of routes (such as bus or train lines), and a set of trips. Trips correspond to individual vehicles that visit the stops along a certain route at a specific time of the day. Trips can be further subdivided into sequences of elementary connections, each given as a pair of (origin/destination) stops and (departure/arrival) times between which the vehicle travels without stopping. Each network in our approach is modeled as a separate directed graph. An additional work is then done to integrate all sub-graphs into one larger graph. As a first step of modeling, we introduce three types of nodes that correspond to stations, platforms and departure events. A station comprises a set of platforms where passengers wait for vehicles. An edge is inserted between a platform and its parent station; its weight represent the minimal time required for accessing that platform from the entrance point of the station. A platform cannot belong to more than one station, however, a station can contain one or several platforms. Each platform has also a type (Bus, railway, tram...) to differentiate between modes. Since a timetable consists of time-dependent events (e.g., a vehicle departing at a stop) that happen at discrete points in time, we use the idea of the time-expanded model that builds a space-time graph to unroll time. Roughly speaking, the model creates a vertex for every event in the timetable that consists of vehicle departing from a platform x at dt and arrives to another platform y at at. Timestamps are inserted into event nodes to account for the departure and arrival times. Event nodes are ordered in the way that a higher-level node refers to an earlier event. In addition, waiting, boarding and alighting arcs are inserted between event nodes and platforms. To account for transfers between and inside stations, we inserted transfer edges with transfer time between platforms.

3 Genetic Algorithm

After introducing the modeling approach in the previous section, we explain in this section the proposed GA. As in most GA's schemes, the first challenge lies in the genetic representation. A solution (individual) in our work is any route that allows going from the starting node at the user's departure time to the destination node. To represent such solution, we have used the permutation encoding. Typically, each chromosome consists of a string of positive integers that represent the IDs of edges included in the route. The size of chromosomes is not fixed since several routes with different nodes and edges may exist to go from an origin point to a destination one. After encoding, we explain the generation of initial solutions. The composition of the initial population in our approach is remarkably different compared to traditional GAs. Our initial solutions are a set of feasible paths generated using a construction heuristic based on a double search algorithm. More precisely, we simultaneously run a forward search from the starting node with respecting the departure time dt and a backward search from the destination node. A feasible path is then found when the two searches intersect. As a result of this operation, we get feasible routes having different lengths to go from one node to another. When it comes to the evaluation phase, we do not use a simple scalar value like in single criterion optimization. However, the fitness function in our work is represented by a p-dimensional vector where in each dimension we evaluate the value of one criterion. As in traditional GA scheme, the proposed algorithm evolves through three operators: *selection*, *crossover* and *mutation*. To accomplish the crossover, a single tournament selection is employed. After ordering the individuals in the population, each chromosome in the odd position is mated with the next chromosome in order to produce new individuals. By doing so, we produce a new population having twice the size of the current population. The best half individuals are then selected for the next generation and the rest are ignored. Duplicated individuals are replaced with newly generated chromosomes to avoid reprocessing the same individual. Single point crossover technique has been used in our approach in order to produce offspring. An intersection node between two individuals is selected to be the crossover point. Current individuals exchange then part of genes with each other before or after the crossover point to generate offsprings. To accomplish the mutation, we randomly select one individual and we try to replace one subsequent route by a new feasible one. By doing so, we increase the diversity in the population and thereby, we prevent the algorithm from premature convergence. Finally, to allow the convergence of the proposed method, two terminating criteria are use i) when the maximum number of evolutions is reached (Generation Number) ii) when no better solutions are found during several evolutions (Fitness Convergence).

4 Experimental Results

To assess the performance, we have applied the proposed algorithm to solve 10000 routing queries based on the real data of the French region Île-de-France that includes the city of Paris and its suburbs. The start time, departure and arrival stations are uniformly picked at random. Transportation data are provided by the transport organization authority that controls the Paris public transport network and coordinates the different transport companies operating in Île-de-France. Data comprise geographical information, as well as, timetable information for four transport modes (Bus, Metro, Railway, and Tram). More precisely, data encompass 17950 stations; 41047 platforms; 195000 transfers; 303000 trips and 6800000 events for one day. We use in this GA the following parameters: the initial population size is 5; the probability of crossover is 0.9; the probability of mutation is 0.1; the maximum number of generation is 500; The number of generations used to ensure a fixed state in the population is 100. We run algorithms on an Intel core I5 machine of 8 GB RAM and we used java as a programming language.We compare the performance of the proposed GA with an exact shortest path algorithm regarding two axes: the CPU computation time and the quality of solutions. Results indicate that the average running time to solve routing queries w.r.t four criteria (travel time, cost, number of transfers and walking time) is 390 seconds when using the exact algorithm. However, the GA spends less than 117 milliseconds to accomplish its search process. The average gap to the optimality of the GA w.r.t the exact approach may reach a maximum of 4%. Therefore, the proposed GA is efficient enough to be used in a real world routing system. As future works, we have planned to integrate other transportation modes such as Car and Bike Sharing, as well as, consider some stochastic parameters such as accidents and delays.

Acknowledgments

This research work has been carried out in the framework of the Technological Research Institute SystemX, and therefore granted with public funds within the scope of the French Program "Investissements d'Avenir".

References

[1] Goldberg, D. E., & Holland, J. H. (1988). Genetic algorithms and machine learning. Machine learning, 3(2), 95-99.

[2] Dib, O., Manier, M. A., & Caminada, A. (2015, September). A Hybrid Metaheuristic for Routing in Road Networks. In Intelligent Transportation Systems (ITSC), 2015 IEEE 18th International Conference on (pp. 765-770). IEEE.

[3] Surender Baswana, Somenath Biswas, Benjamin Doerr, Tobias Friedrich, Piyush P. Kurur, Frank Neumann: Computing single source shortest paths using single-objective fitness. FOGA 2009: 59-66

[4] Benjamin Doerr, Edda Happ, Christian Klein: Tight Analysis of the (1+1)-EA for the Single Source Shortest Path Problem. Evolutionary Computation 19(4): 673-691 (2011)

[5] Kumar, Rakesh, and Mahesh Kumar. "Reliable and Efficient Routing Using Adaptive Genetic Algorithm in Packet Switched Networks." IJCSI International Journal of Computer Science Issues 9.1 (2012): 1694-0814.

[6] Behzadi, Saeed, and ALIA ALESHEIKH. "Developing a Genetic Algorithm for Solving Shortest Path Problem." NEW ASPECTS OF URBAN PLANNING AND TRANSPORTATION (2008).

[7] Kumar, Dr Rakesh, and Mahesh Kumar. "Exploring Genetic algorithm for shortest path optimization in data networks." Global Journal of Computer Science and Technology 10.11 (2010).

[8] Benjamin Doerr, Daniel Johannsen, Timo Kötzing, Frank Neumann, Madeleine Theile: More Effective Crossover Operators for the All-Pairs Shortest Path Problem. PPSN (1) 2010: 184-193