18th Euro Working Group on Transportation, EWGT 2015, 14-16 July 2015, Delft, The Netherlands

Memetic algorithm for computing shortest paths in multimodal transportation networks

Abstract

Route planning in multimodal transportation networks is gaining more and more importance. Travelers ask for efficient routing methods allowing them to reach their destinations through the intricate multimodal transportation scheme. In this paper, we propose a new approach for computing multi-modal shortest paths. We only consider railway, bus and pedestrian networks. The travel time is the only metric in our cost function. Our proposed approach is a combination of two meta-heuristics: Genetic Algorithm (GA) and Variable Neighborhood Search (VNS). We compare our approach with the exact shortest path algorithm Dijikstra that has been modified to work in a multimodal environment, as well as, with a pure GA. Results have shown that the success rate of our approach in terms of converging to optimum/near optimum solutions is highly better than a pure GA. Moreover, in contrast to traditional algorithms like Dijkstra, our approach is fast enough for practical routing applications.

Keywords: Multimodal Shortest Paths; Hybrid Metaheuristics; Memetic Algorithm, Genetic Algorithm; Variable Neighborhood Search; Dijkstra;

1. Introduction

A Multimodal Transportation System (MTS) is the combination of all traveler modes and kinds of transportation systems operated through various systems (Bielli et al., 2006). That is, a set of choices of modes of transport that travelers can use simultaneously in order to reach their destinations. Nowadays, the human mobility within urban areas usually happens in a multimodal context (Liu, 2011). People are more prone to use more than one mode of transport during their travels. However, the transport system has become more and more complex. Users usually find themselves more confused with having several possibilities to go from one place to another. Consequently, for the sake of helping people efficiently to find optimal or near optimal routes through the complex transportation scheme, route planning in MTS has gained significant importance.

* Corresponding author. Tel.: +336 05 51 30 49.
  E-mail address: omar.dib@irt-systemx.fr

Generally, we can distinguish between private and public route planners. The formers usually consider private transport modes, which are qualified as continuous modes such as cars, bicycle and walking. On the other hand, public route planners are only dedicated to public transportation modes such as bus, subway, tram which work according to predefined timetables.

Nowadays, there exist several solutions to solve the conventional route planning in transportation systems. These solutions come in the form of free or paid software and applications such as Google and Bing Maps. Most of such routing tools support more than one transportation mode. However, few could provide a free combination of travel modes. In practice, travelers usually ask for an integrated solution that compute optimal or near optimal routes with combining all available transportation modes whether they are private or public.

In this paper, we introduce a new multi-modal routing approach. We focus our research on two public modes (Bus and railway) and one private mode (walking) that people can use to access transportation means, as well as, to make transfer between modes.

The remainder of this paper is organized as follows. We introduce some related works in Section 2. Section 3 describes the multi-modal Shortest Path Problem (SPP). We present in section 4 our modeling approach. Section 5 is devoted to present our novel approach. Experimental results and some discussions are shown in Section 6 and 7. Finally, we conclude this paper in section 8, as well as, we propose some future works.

2. Related work

Several approaches have been developed in order to represent the intricate multimodal transportation network. Some models are based on the hypergraph theory such as in (Nguyen et Pallottino, 1988), (Febbraro et Sacone, 1995), (Lozano et Storchi, 2002). Other models are based on the space-time-network (Pallottino et al., 1998) or on the multi-label networks (Ziliaskopoulos et Wardell, 2000). Using hyper-graph networks eliminates usually some arcs used for changing between transportation modes. As a result, the search space for the routing algorithms is usually less in comparison with other modeling approaches. However, the other models are more efficient when it comes to handle additional problem constraints.

Several algorithms have been proposed since 1956 for routing in static networks, such as (Dijkstra, 1959) Although such algorithms compute shortest paths in polynomial times, they become too slow to process real world data sets like continental networks, even on today's computers.

Researches have therefore focused on accelerating traditional algorithms using speed up techniques (Schultes, 2008). For more details about speed up techniques in static networks refer to (Delling, 2009 and Bast et al., 2014).

While routing in static networks has become quite easy, routing in dynamic networks turns out to be more difficult. Public transit networks such as bus and railway are inherently time-dependent. The travel time from one station to another is not static; it depends on the arrival time of the user at the departure station. Cooke and Halsey (1966) showed that standard algorithms like Dijkstra could be augmented to cope with the time dependency aspect of public transportation modes. However, that would be at the expense of additional computational efforts, especially when the size of the network becomes very important (Delling, 2009). Extensive works have therefore been investigated to augment speed up techniques used in static networks to accommodate for the new variant of SPP raised when dealing with time-dependent networks. For more details about accelerating techniques in dynamic networks, the reader can refer to (Bauer et al., 2011).

Although, speed up techniques are efficient and fast enough to compute shortest paths, they become less performant or even inapplicable when additional constraints are added to the SPP such as stochasticity and multi-criteria paths optimization. Therefore, there has been an application need to develop new routing approaches that provide optimal or near optimal routes in reasonable computational time in large-scale multimodal networks, as well as, that cope with additional problem constraints such as stochastic arcs' weights, multi-criteria optimization etc. We believe that meta-heuristics such as Genetic Algorithms, Local Search Procedures are efficient candidates to handle such requirements.

Metaheuristics have been used to solve several routing issues. For instance, Abeysundara et al. (2005) used a Genetic Algorithm to solve the routing issue in road networks. Yu et al. (2012) proposed an improved Genetic Algorithm to solve the route-planning problem in a multimodal transportation network. Davies et al. (2003) introduced a new genetic approach for rerouting in dynamic and stochastic networks. Other metaheuristics have been used to solve routing issue such as in (Mohemmed et al., 2007). Although such approaches are very interesting, they have not been tested over large-scale real world multimodal networks. Therefore, there is no guarantee that such approaches still provide high performances in large-scale networks

In this paper, we propose a new approach for solving the routing issue in a multimodal transportation network. We introduce a new highly scalable hybrid GA-VNS algorithm (Genetic Algorithm with Variable Neighborhood Search) to compute multimodal shortest paths.

3. Problem description

The conventional routing issue addressed in this study is defined as follows :Given a directed graph $G = (V, E)$ with node set V of cardinality n, edge set E of cardinality m, costs $c(e) \in R$ for all edges $e \in E$, a source node $u \in V$, a target node $v \in V$ and a departure time t_0. We ask then for finding a path p $(e1, e2... ek)$ in the network with the following properties: 1) p must start at u; 2) the departure time at s must be t_0; 3) p must end at v; 4) the length of p is preferred to be the minimum among all the other routes that respect the properties 1-3.

4. Networks modeling

In this section, we present the approaches adopted to model our three networks. Each network is modeled as a separate graph. An additional work is then done to integrate all models into one larger model that is capable of adequately representing the networks infrastructures, as well as, yielding correct results when applying shortest path routing algorithms. Besides, since there are strong similarities between rail and bus timetables, we have decided to use the same approach to model both modes.

4.1. Private mode : pedestrian network

Modeling the pedestrian network is quite canonical. Junctions are represented by nodes in a directed graph G. An edge e (u, v) is inserted between u and v if u is linked with v by a footpath. The weight w(e) of an edge represents the average travel time along that edge. w(e) is computed by dividing the average walking speed s of a pedestrian into the geographical length of that segment.

4.2. Public modes

There are several approaches to model public transportation networks. The time-independent condensed model (Schulz et al., 2002) is the most basic approach. However, it is not used in our study since it represents the network structure without considering the scheduling. Therefore, it will yield incorrect results when computing shortest paths. To overcome such shortcomings, the time-expanded (Schulz et al., 2000) and time-dependent (Brodal et al., 2004) models have been developed. While the former allows more flexibility in considering additional constraints such as computing multi-objective shortest paths, the latter yields smaller inputs (Pyrga et al., 2007). Historically, there have been two versions of each model: A simple version and a realistic version. Unlike the former, the latter incorporates realistic transfer between vehicles at stations. The modeling approach used in this paper is based on the simple version of the time-expanded model. Typically, there exist two types of nodes to account for the timetable information (Delling et al., 2009). Physical nodes that represent stations and event nodes that account for departure and arrival events (Fig. 1.). Each event node has a link to its corresponding physical node. The event type decides the direction of the link. In case of arrival events, the direction is from the event node to the station node and the link weight represents in this case the alighting time. However, the direction is reversed in case of departure events and the link weight refers to the waiting and boarding time. Moreover, event nodes related to the same station are sorted in ascending order with respect to their timestamp and a link between two subsequent nodes is inserted to account for transferring or waiting. Finally, a link is inserted between two events belonging to different physical stations in order to represent the movement of a vehicle from one stop to another.
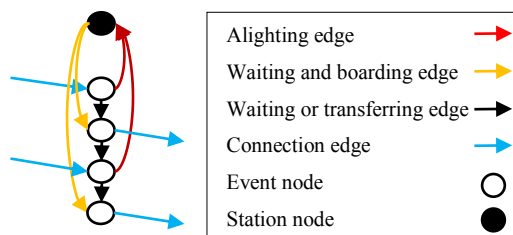

Fig. 1. Representation of public modes

4.3. Combining networks

To compute multi-modal paths, the query algorithm has to use multiple networks simultaneously. For that reason, after modeling each network as a separate graph, we have to combine the different networks in order to form one larger multimodal graph. To do so, we use transfer links to connect each physical node in the public networks with the nearest node in the pedestrian network. That is, we use the pedestrian network to transfer from one mode to another. In the figure below, we represent a multimodal transportation network where the pedestrian network is the only middleware between the bus and railway network. Note that, the weight of a transfer link is the time required to going from a station node to the nearest node in the pedestrian network.
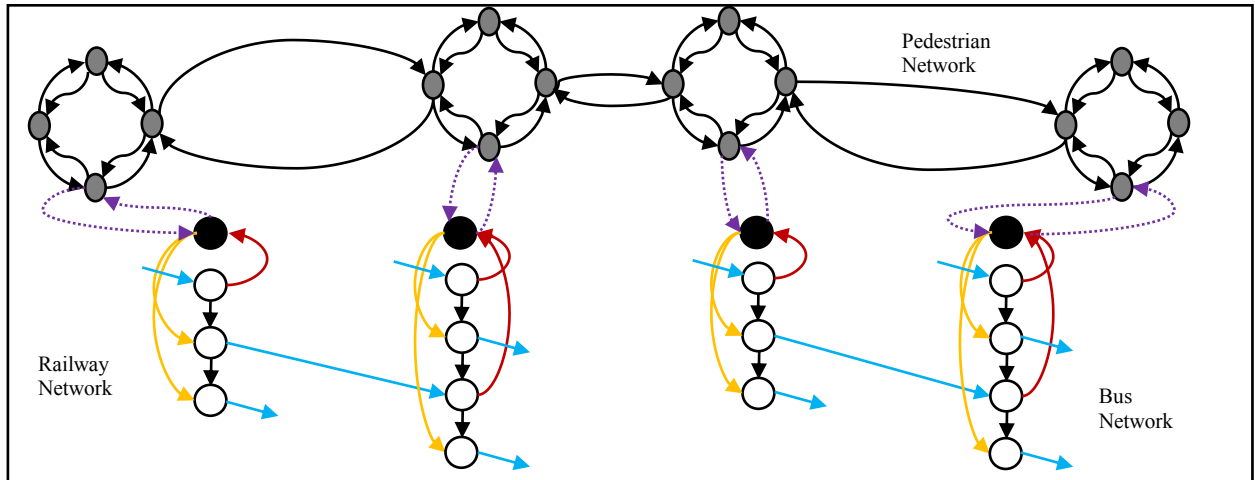

Fig. 2. Representation of a multimodal transportation network

After combining the multiple networks into one single connected graph, we present in the next section our novel approach to computing optimal or near optimal routes to go from a physical node (station or pedestrian node) at the user's departure time t_0 to another physical node.

5. Proposed Approach

The main contribution of this paper is to apply an approximate method based on a collaboration between two meta-heuristics to compute multimodal shortest paths. That is, we apply a Memetic Algorithm in which we use VNS inside a GA to solve the Earliest Arrival Problem (EAP) in a multimodal environment. We suppose in this paper that the reader is familiar with GAs, VNS and Memetic Algorithms (MAs). For more details about the metaheuristics used in this work, the reader can refer to (Talbi, 2002 and Hansen et al., 2005 and Moscato et al., 2004)

As traditional Genetic Algorithms work, our approach maintain a population of solutions at each iteration. Initial solutions are generated using a double search algorithm that is able to provide feasible paths between any two nodes. The details of this algorithm is described later in this article.

Once a population of solutions is successfully generated, we do an enhancement operation over each individual in the first population. That is, we apply a VNS over the first population. We decided with improving initial solutions since their quality would possibly help the algorithm find better or even optimal solutions. More details about VNS adaptation to the problem will be discussed later.

Once we accomplish the VNS, we send individuals to an evaluation process. Our fitness function is the sum of the weights of each edge included in a path. After the improving phase, we repeatedly perform genetic operations in the goal of increasing the algorithm's chance to find the optimal or high-quality routes.

We begin with the crossover operation that is responsible for forming two new paths (offspring) from two initial solutions. Crossovers are usually used to exploit regions within the search space. We have used single-point as a crossover technique. We will discuss later and in more details the crossover operation.

After crossover operation, we perform a special mutation operation based on VNS. That is, we apply VNS over each individual (path) in a population. Thanks to this technique, our algorithm will have more chances to exploit and explore new and distant regions within the search space.

The whole process is repeated until the algorithm reaches our stopping criteria. The following scheme represents the algorithms' steps that we will discuss in more details in next paragraphs.
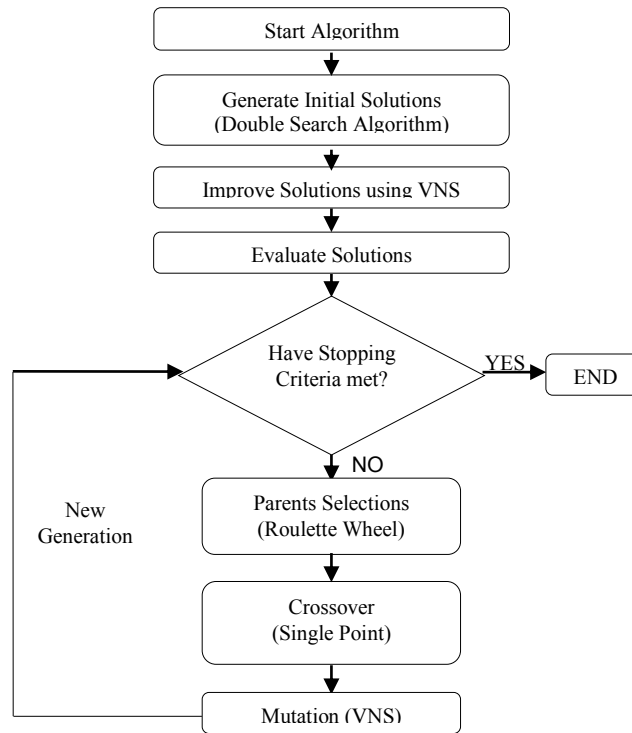
```
┌─────────────────────────┐
│     Start Algorithm     │
└─────────────────────────┘
             │
             ▼
┌─────────────────────────┐
│ Generate Initial Solutions │
│  (Double Search Algorithm) │
└─────────────────────────┘
             │
             ▼
┌─────────────────────────┐
│ Improve Solutions using VNS │
└─────────────────────────┘
             │
             ▼
┌─────────────────────────┐
│    Evaluate Solutions   │
└─────────────────────────┘
             │
             ▼
        ◇ Have Stopping ◇  ──YES──▶  END
        ◇ Criteria met? ◇
             │
             NO
             │
             ▼
┌─────────────────────────┐
│    Parents Selections   │
│    (Roulette Wheel)     │
└─────────────────────────┘
             │
             ▼
┌─────────────────────────┐
│       Crossover         │
│    (Single Point)       │
└─────────────────────────┘
             │
             ▼
┌─────────────────────────┐
│     Mutation (VNS)      │
└─────────────────────────┘

New Generation
```

Fig.3. Flowchart of the proposed GA-VNS algorithm

5.1. Encoding scheme

Encoding is the process of representing an individual. It is a crucial step in GA and it is often closely dependent on the problem characteristics. Encoding affects the work of genetic operators as well as the whole performance of the algorithm. Several techniques have been proposed for encoding solutions such as binary, octal and hexadecimal encodings. The efficiency of those latters is highly related to the problem itself. 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 solutions, 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 paths with different nodes and edges may exist to go from an origin point to a destination one.

5.2. Generating initial solutions

Generating initial solutions for meta-heuristics is not always straightforward. 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. In fact, we simultaneously run a forward search from the starting node with respecting the departure time t_0 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 can get feasible routes having different lengths to go from one node to another. Note that other techniques can be used for getting initial solutions such as in (Davies et al,2003 and Yu et al.,2012), however, they are not used in this work since they require significant computational effort.

5.3. Enhancing initial solutions using VNS

Enhancing initial solution(s) in the context of combinatorial optimization problems may rapidly guide the search process towards important regions in the solution space. Moreover, having a good repartition of initial solutions within the search space may increase the algorithm's chance in finding the optimal or near optimal solutions. For these reasons, we have enhanced initial solutions by applying the metaheuristic VNS over each individual in the first population. Unlike traditional local search approaches such as tabu search (Glover et al., 2013), VNS does not follow a trajectory. It explores rather increasingly distant neighborhoods of the current incumbent solution, and escapes from this solution to another if and only if an improvement has been made. Therefore, VNS is more likely to prevent the optimization process from rapidly falling into local optima. Although, VNS is irrelevant to the problem and is suitable for all kinds of optimization problems, however, its implementation is not always straightforward.

One of the vital requirements for the VNS is the definition of neighborhood structures. To deal with that issue, our algorithm performs a preprocessing operation during the generation phase of networks in order to compute additional data required for constructing the neighborhood structures. As a result, we obtain a VNS method having two neighboring structures. The first neighboring structure consists of replacing two successive nodes in the current solution by a suboptimal path computed and stored during the generation phase of networks. The second neighboring structure is to replace three nodes in the incumbent solution by a suboptimal path calculated in the preprocessing step. It is worth mentioning that using more than two neighboring structures might enhance the performance of the VNS in terms of the quality of final solution, however, that would be at the expense of additional running time. We therefore limit our method to only two neighboring structures.

We present in Fig.4 the general steps of our VNS approach. Indeed, our VNS takes as input two neighboring structures and a single initial solution and it performs then local refinements in order to enhance its initial solution. More precisely, our VNS proceeds with the first neighboring structure. It examines then all the neighbors' solutions belonging to the current solution. VNS then selects the best neighbor among all neighbors according to the fitness. VNS then compares the best-selected neighbor with the current solution; it moves to the best neighbor if its fitness is better than the current solution. If not, VNS changes the neighboring structure. The process is repeated while the current solution can be enhanced by using any of the neighboring structures.

Input:
- Neighboring structures N_K, (K=1,2);
- An individual (solution) x;

Iteration
1.   **While** stopping rule is not satisfied **do**
2.   k=1;
3.   **While** k<2 **do**
4.   Exploration of neighborhood: find the best neighbor x' of x (x'$\in N_k(x)$)
5.   Move or Not:
6.   **If** f(x') < f(x) **then**
7.   x ← x', k ← 1;
8.   **else**
9.   k ← k+1;
10.   **end if**
11.   **end while**
12.   **end while**
13.   **return** the best solution

Fig.4. Steps of VNS

5.4. Evaluation function

Evaluating individuals is of great importance for Genetic Algorithms. A fitness function is usually defined to evaluate how good a potential solution is relative to other potential solutions. Since we optimize in our work one single criterion, our fitness is a scalar value that represents the length of the current individual. The result is therefore a non-negative number representing a path length. More the fitness decreases, more the chance for an individual to pass its genes to the next generation increases.

5.5. Crossover

To accomplish the crossover process, a single tournament selection mechanism 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 (offsprings). 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. However, there is a big chance that the same individual is duplicated in the population as the generations go on. We therefore, replace the duplicated individuals with newly generated chromosomes. This process will undoubtedly increase the diversity within the population. 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 (Fig.5).
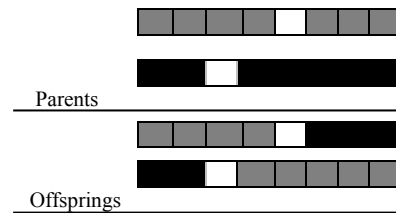
Fig.5. Single point crossover

One point worth mentioning is that after applying our crossover, we do not care about the feasibility of offsprings. We will always end up with feasible paths from the source to the destination. Therefore, the algorithm does not lose time to check the validity of offsprings nor to perform additional operations to repair wrong individuals. Finally, experimental results show that using more than one crossover point might increase the diversity of our approach. Therefore, the algorithm's chance to find better solutions will also increase. However, that may be at the expense of additional computational efforts. We consequently decided with only applying the simple crossover technique.

5.6. Mutation (VNS)

Mutation is an important part of the genetic search since it helps to prevent the population from premature convergence. Mutation usually consists of altering one or more gene values in a chromosome in the hope of guiding the search process towards new regions within the solution space. Unlike most GAs that use conventional mutation techniques such as boundary, non-uniform and uniform, we have used the advanced local search procedure VNS to perform the mutation operation. That is, we apply VNS over each individual in the population resulting from the crossover operation. By doing so, we ensure that the genetic algorithm will maintain a sufficient diversity level that prevents premature convergence. We therefore increase the algorithm's chance to find better solutions.

Unlike traditional GAs, the mutation operation in our approach is applied with the same probability as the crossover. The purpose of doing that is to allow the algorithm to avoid local minima by preventing the population of chromosomes from becoming too similar to each other, thus slowing or even stopping evolution.

We did apply other mutation techniques such as order changing, but we have realized that the mutation becomes less performant and it may provide infeasible solutions (invalid paths). An additional process should therefore be introduced to reform infeasible paths. As a result, the mutation computational time will increase. We decided thereby against using such traditional techniques.

5.7. Stopping conditions

In contrast to conventional exact algorithms, meta-heuristics do not guarantee finding optimal solutions. Additional terminating conditions should therefore be introduced in order to allow the convergence of the method. Our proposed approach terminates in two cases: i) when the maximum number of evolutions is reached (Generation Number) ii) when no better solutions are found during several evolutions (Fitness Convergence).

6. Test and illustration

To evaluate our approach, we have produced a multimodal graph based on the data of the French region Ile-de-France that includes the city of Paris and its suburbs. The data comprise geographical information, as well as, timetable information for railway and bus modes. The pedestrian network includes 275 606 nodes and 751 144 edges. The bus network consists of 72 511 bus stations and 5 078 443 arrival/departure events. The railway network encompasses 1 746 railway stations and 122 284 arrival/departure events for one day.

We assumed in this work that the walking speed is 5 km/h. Additionally, our algorithm has the following parameters i) the initial population size is 5 ii) the probability of crossover and mutation is 0.5 iii) the maximum number of generation is 500 iiii) The number of generations used to ensure a fixed state in the population is 100.

The performance of our approach has been evaluated in comparison with two other approaches: One exact approach (Dijkstra) that has been implemented using a priority queue and one approximate approach (a pure GA). The only differences between our hybrid GA-VNS approach and the pure GA is that in the latter we do not enhance initial solutions. Moreover, the mutation operation is performed traditionally i.e., without using VNS. In addition, the probability of crossover is very high (0.9) compared to the probability of mutation (0.1).

The comparison between the different approaches is done regarding two axes i) the CPU computational time (time performance) ii) the solutions quality. We run algorithms on an Intel core I5 machine of 8 GB RAM and we used java as a programming language. Besides, we made extensive use of generic programming techniques in order to avoid runtime overheads. We also put particular efforts into carefully implementing efficient data structures when storing and manipulating the graphs' components.

We present in Table 1 results obtained from applying the various algorithms in different scenarios. We consider in each scenario one or a combination of several transportation modes. We run in each case algorithms over 1000 random generated origin/destination queries.

Table 1. Experimental Results.

Scenario	Average Running Time (*s*)			Average Speed		Average GAP (%)	
	Dijkstra	Pure GA	GA-VNS	Pure GA	GA-VNS	Pure GA	GA-VNS
Pedestrian	0.17	0.002	0.005	57	34	6.6	1.2
Bus	3.234	0.006	0.007	485	462	12.1	1.8
Railway	0.078	0.0007	0.001	110	78	5.12	1.1
Pedestrian + Bus	3.315	0.0067	0.008	490	414	12.10	2.65
Pedestrian + Railway	0.281	0.003	0.006	80	46	9.12	1.9
Railway + Bus	3.221	0.005	0.007	550	460	10.16	2.5
Pedestrian + Bus + Railway	4.125	0.0058	0.008	700	515	15.56	2.99

Results showed that the running time of Dijkstra increases gradually with the size of network. This can be explained by that fact that the time complexity of Dijkstra is totally linked to the number of nodes and edges in the graph. More the graph components' increases more the search space of Dijkstra increases. Results also showed that in the last scenario where we consider all networks, the average time required by Dijkstra to answer travelers' queries increases to 4,125 seconds, which is too slow to be employed in real world routing applications.

On the other hand, the running time of approximate approaches is highly better than Dijkstra. It does not exceed the threshold of 8 msec which is an acceptable time for travelers seeking real time answers. While the average speed of our hybrid GA-VNS algorithm, with respect to the basic algorithm of Dijkstra, reaches a factor of 515, the pure GA outperforms both Dijkstra and our hybrid GA-VNS in terms of speed (with a factor of 700 compared to Dijkstra). This can be explained by the fact that the pure GA converges rapidly to suboptimal solutions since the mutation operation responsible for avoiding premature convergence is not very efficient and is applied with low probability. However, our approach spends more time to intelligently visit distant regions within the solution space thanks to the VNS approach used in the mutation operation.

Although approximate approaches outperform Dijsktra in terms of speed, however, they do not always provide optimal routes. While the average GAP to the optimality of the pure GA may increase to 15 %, our hybrid GA-VNS always results in solutions that are very close to the optimal solution. The average gap to the optimality of our

approach may reach a maximum of 3 % in the worst case, i.e., when all networks are considered (scenario 7). We can easily notice that the quality of solutions in our approach is better than in a pure GA. This can be explained by the fact that the special mutation operation as well as enhancing initial solutions have played an important role in helping the algorithm find better solutions.

7. Discussions

Results presented in the previous section has proven the efficiency of our approach in terms of time performance and solutions quality. We discuss in this section some other strengths of our approach as well as some situations where applying the proposed hybrid GA-VNS method may also result in significant performance.

First, unlike classical shortest path algorithms like Dijkstra, our algorithm results in a set of feasible and high quality solutions to reach a destination. Consequently, users will have access to several routes to reach their targets. Such multiplicity of solutions is very beneficial in some real-life applications such as routing in data networks. For instance, having several ways to route data packets along computer networks may help network administrators or even routing protocols in alleviating congestions and conflicts.

Second, the travel-time is the only metric used in our work. However, travelers do not only seek short-time journeys, but they also endeavor to optimize other criteria such as cost, effort (walking distance, number of transfer, waiting time…), safety etc. As a result, an efficient routing algorithm should also handle multi-objective scenarios. Delling (2009) showed that traditional exact shortest path algorithms like Dijsktra can be augmented to cope with multi-criteria issue. However, that would be at the expense of unacceptable computational effort. To overcome such problem, approximate methods have to be used to identify highly attractive connections in reasonable time. Although we haven't tested our approach in a multi-criteria environment yet, we believe, however, that combining GA and VNS will provide a high performance tool for computing multi-criteria paths. From one side, we benefit from the power of GA in exploiting a large solution space, and from the other side, we capitalize on the capacity of VNS in escaping from local optima and visiting distant solution regions.

Finally, although the proposed approach is efficient and more flexible than conventional methods, it has some drawbacks. Generally, approximate methods and especially meta-heuristics suffer from parameterization problems. There are no standard parameters for meta-heuristics to achieve the best performance. For instance, in the proposed algorithm, parameters like the number of generations, probability of genetic operators, number of neighborhood structures, etc. have been chosen because they provided high performance over some optimization problems. However, there is no guarantee that such parameters are the best for our problem. Future work will be undoubtedly done to deal with the parameterization issue of our approach.

8. Conclusion

Solving the multimodal route planning is a highly important issue to develop seamless advanced traveler information systems. This paper proposed a hybrid GA-VNS approach for route planning in multimodal environment. We focused our study on three modes bus, railway and the pedestrian network. Networks have been modeled separately and an additional work has been done to connect networks via transfer links.

Experimental results showed that traditional shortest path algorithms such as Dijkstra could optimally compute multimodal routes. However, the computational effort will increase rapidly with the size of the network. Therefore, using such algorithms to support travelers' query may not be convenient for users seeking real-time answers.

On the other hand, the proposed approach succeeded in finding optimal or near optimal multimodal routes in a reasonable computational effort even when the network's size increases. Therefore, our approximate approach has resulted in a good tradeoff between the solution quality and the average speed .We have also proven in this paper that combining metaheuristic such as GA and VNS did increase the quality of final solutions in comparison with applying pure metaheuristic such as GA. Finally, several works have been planned to be accomplished in the near future. We will first perform a sensitivity analysis to study the impact of parameters over the performance of our approach. We will then use our novel approach to compute multi-objective shortest paths. Finally, we will compare our approach with other metaheuristics and approaches such as tabu search and hyperpath theory.

Acknowledgements

References

Abeysundara, Sachith, Baladasan Giritharan, and Saluka Kodithuwakku. "A genetic algorithm approach to solve the shortest path problem for road maps."Proceedings of the International Conference on Information and Automation. 2005.

Bast, Hannah, et al. "Route planning in transportation networks." Technical Report MSR-TR-2014-4. Microsoft Research, Microsoft Corporation, 2014.

Bauer, Reinhard, Daniel Delling, and Dorothea Wagner. "Experimental study of speed up techniques for timetable information systems." Networks 57.1 (2011): 38-52.

Bielli, M., Boulmakoul, A., & Mouncif, H. (2006). Object modeling and path computation for multimodal travel systems. European Journal of Operational Research, 175(3), 1705-1730.

BRODAL, G. S. AND JACOB, R. 2004. Time-dependent networks as models to achieve fast exact time-table queries. In Proc. 3rd Workshop on Algorithmic Methods and Models for Optimization of Railways (ATMOS 2003). Electronic Notes in Theoretical Computer Science, vol. 92. Elsevier.

Davies, Cedric, and Pawan Lingras. "Genetic algorithms for rerouting shortest paths in dynamic and stochastic networks." European Journal of Operational Research 144.1 (2003): 27-38.

Delling, Daniel, and Dorothea Wagner. "Time-dependent route planning." Robust and Online Large-Scale Optimization. Springer Berlin Heidelberg, 2009. 207-230.

Delling, Daniel. Engineering and augmenting route planning algorithms. Diss. Karlsruhe Institute of Technology, 2009.

Delling, Daniel, Thomas Pajor, and Dorothea Wagner. "Engineering time-expanded graphs for faster timetable information." Robust and Online Large-Scale Optimization. Springer Berlin Heidelberg, 2009. 182-206.

Dijkstra, Edsger W. "A note on two problems in connexion with graphs."Numerische mathematik 1.1 (1959): 269-271.

Dowsland, Kathryn A., and Jonathan M. Thompson. "Simulated annealing."Handbook of Natural Computing. Springer Berlin Heidelberg, 2012. 1623-1655.

E. Pyrga, F. Schulz, D. Wagner, and C. Zaroliagis. E_cient models for timetable information in public transportation systems. Journal of Experimental Algorithmics (JEA), 12:Article 2.4, June 2007.

Febbraro et Sacone, 1995 A. Di Febbraro, S. Sacone. An on-line information system to balance traffic flows in urban areas. Porceeding of the 36th IEEE conference on decision and control, 5, p. 4772-4773, 1995.

Frank Schulz, Dorothea Wagner, and Christos Zaroliagis. Using Multi-Level Graphs for Timetable Information in Railway Systems. In Proceedings of the 4th Workshop on Algorithm Engineering and Experiments (ALENEX'02), volume 2409 of Lecture Notes in Computer Science, pages 43{59. Springer, 2002.

F. Schulz. Timetable Information and Shortest Paths. Phd thesis, University of Karlsruhe (TH), 2005.

Glover, Fred, and Manuel Laguna. Tabu Search∗. Springer New York, 2013.

Hansen, Pierre, and Nenad Mladenović. Variable neighborhood search. Springer US, 2005.

K. Cooke and E. Halsey. The Shortest Route Through a Network with Time- Dependent Intermodal Transit Times. Journal of Mathematical Analysis and Applications, (14):493{498, 1966.

Liu, L. (2011). Data model and algorithms for multimodal route planning with transportation networks. Technische Universität München, München.

Lozano et Storchi, 2002 A. Lozano et G. Storchi. Shortest viable hyperpath in multimodal networks. Transportation Research Part B 36, p. 853–874, 2002.

Mohemmed, Ammar W., Nirod Chandra Sahoo, and Tan Kim Geok. "A new particle swarm optimization based algorithm for solving shortest-paths tree problem." Evolutionary Computation, 2007. CEC 2007. IEEE Congress on. IEEE, 2007.

Moscato, Pablo, Carlos Cotta, and Alexandre Mendes. "Memetic algorithms."New optimization techniques in engineering. Springer Berlin Heidelberg, 2004. 53-85.

Nguyen et Pallottino, 1988 S. Nguyen et S. Pallottino. Equilibrium traffic assignment for large scale transit network. European Journal of Operational Research, 37(2), p. 176-186, 1988.

Pallottino et Scutellà, 1998 S. Pallottino, and M. G. Scutellà. Shortest path algorithms in transportation models: classical and innovative aspects. In Equilibrium and Advanced Transportation Modelling, Kluwer, p. 1245-281, 1998.

Raidl, Günther R. "A unified view on hybrid metaheuristics." Hybrid Metaheuristics. Springer Berlin Heidelberg, 2006. 1-12.

Schultes, Dominik. "Route Planning in Road Networks." Ausgezeichnete Informatikdissertationen. 2008.

SCHULZ, F.,WAGNER, D., ANDWEIHE, K. 2000. Dijkstra's algorithm on-line: An empirical case study from public railroad transport. ACM Journal of Experimental Algorithmics 5, 12.

Talbi, E-G. "A taxonomy of hybrid metaheuristics." Journal of heuristics 8.5 (2002): 541-564.

Yu, Haicong, and Feng Lu. "A multi-modal route planning approach with an improved genetic algorithm." Advances in Geo-Spatial Information Science(2012): 193.

Ziliaskopoulos et Wardell, 2000 A. Ziliaskopoulos et W. Wardell. An intermodal optimum path algorithm for multimodal networks with dynamic arc travel times and switching delays. Elsevier Science, European Journal Of Operational Research, 125, p. 486–502, 2000.

Corrections

The authors would like to thank the reviewers for their valuable comments and suggestions, which helped to considerably improve the quality of the manuscript. Below, we address all comments point-by-point, discussing the subsequent modifications. Reviewers' comments in italic fonts, answers in normal fonts.

Referee 1:

1. The reviewer believe that the comparisons should be extended to the field of metaheuristics in order to have a genuine benchmarking.

We totally agree with the reviewer. We compared in this paper our approach with a pure Genetic Algorithm (refer to "test and illustration" section). We tried also when conducting this study to compare our approach with other metaheuristics from the literature such as ACO, PSO. However, we encountered some difficulties. From one side, none of the metaheuristics was applied over the real data of the French region Île-de-France. From another side, most metaheuristics were tested over small and unreal networks. As a result, comparing our approach with such methods was not possible. We added therefore in the related works section few sentences to emphasis on the lack of applying metaheuristics over large-scale real-world multimodal networks. We also added in the conclusion section that one of our future works is to apply metaheuristics such as tabu search etc over the real-world data of the French region Île-de-France, and to compare after that such methods with our approach.

2. As the authors themselves argued, a sensitivity analysis should be performed in order to highlight the role of the parameters.

A sensitivity analysis should be accomplished in order to study the impact of the parameters over the performance of our approach. Currently, we are working on that. So far, we have found that the performance of our approach increases when the genetic operators (crossover, mutation) are performed with the same probability of 0.5. Additionally, a small number of individuals in the first population (<10) is usually sufficient for achieving high performances. A more detailed study will appear in the near future in order to present our findings.

3. Another issue is that, actually, in their choice transportation users consider at list the monetary cost in addition to travel time. From a transportation planning/managing perspective, this issue could not be neglected (differently from data network managers).

We agree with the referee. Travelers do not only seek short-time journeys, but they also ask for minimizing other criteria such as cost, walking distance, number of transfer, waiting time etc. Although the travel time was the only metric used in our work, we mentioned in the discussion section that the proposed approach could handle multi-criteria issue. From one side, we benefit from the power of GA in exploiting a large solution space, and from the other side, we capitalize on the capacity of VNS in escaping from local optima and visiting distant solution regions. Additionally, we mentioned in the conclusion section that one of our future works is to apply our approach in a multi-objective context.

Referee 2:

1. The main problem of the paper is that the hyperpath theory is not at all considered in the literature and as possible solution method. Therefore, it is not clear if the proposed method can be better of the hyperpath assignment. Moreover, the multimodal transportation network is actually a transit network where coexists rail and bus lines; the hyperpath assignment model allows to solve the problem in this case.So, in order to be published, the paper should at least compare the proposed method with the hyperpath method.

We thank the Reviewer for raising this issue. We agree that the hyper-path theory could be an efficient candidate for solving routing issues in multimodal networks. In accordance with the referee's wishes, we expanded our literature review to mention models based on the hypergraph theory.We have added the following paragraph.

"Indeed, several approaches have been developed in order to represent the intricate multimodal transportation network. Some models are based on the hypergraph theory such as in (Nguyen et Pallottino, 1988), (Febbraro et Sacone, 1995), (Lozano et Storchi, 2002). Other models are based on the space-time-network (Pallottino et al., 1998) or on the multi-label networks (Ziliaskopoulos et Wardell, 2000). Using hyper-graph networks eliminates usually some arcs used for changing between transportation modes. As a result, the search space for the routing algorithms is usually less in comparison with other modeling approaches. However, the other models are more efficient when it comes to handle additional problem constraints"

Besides, the referee suggests making a comparison between our approach and the hyperpath approach. With all due respect to the reviewer, the main contribution of this paper is not to prove that our approach is the fastest among all other approaches nor to compare the hypergraph approach with the space time approach. We instead want to present a model based on the space-time network. We also want to prove that applying Dijkstra over this model lead to correct shortest paths. From another side, we want to show that metaheuristic could be used for computing basic and complex variants of shortest path problems and the proposed approach provides high quality solutions within reasonable computational time.

Finally, regarding the highly interesting reviewer suggestion and for the lack in the literature, we have decided to provide in a future work a full comparison between the various modeling approaches used for representing a multimodal transportation network. We will also make a comparison between the hyper path approach and our routing approach where this latter is applied over a model based on the hypergraph theory.

2. At page 3 the definition of the pedestrian network considers one-way and two-way links where, usually, the pedestrian links are always two-way.

The referee is right to point out that the pedestrian links are usually two-ways. In fact, the reason why we made a distinction between one-way and two-way links is just for being consistent with the other networks. To remove the ambiguity, we eliminated the sentence "If in reality the road between u and v is a two-way road, then two edges (u, v) and (v, u) are inserted into the graph".

3. At page 3 some references are with all capital letters

All the references have been corrected with respect to the template

4. In Table 1 a comparison about the quality of the compared path is not reported but only on the computing times; moreover some number are with . decimal separator and other ones with the ,

We evaluate in table1 the performance of two approximate approaches (our algorithm and a pure GA) with respect to an exact approach (Dijsktra's algorithm). The comparison is done regarding two criteria: the time performance and the solutions quality. While the former is introduced by the average speed columns, the latter is expressed by the average GAP to the optimality. Therefore, the quality of the compared path is expressed by the GAP columns. Data in the table were updated to fulfill the template requirements.

5. In the discussion, the sentence "Such multiplicity of solutions is very beneficial in some real-life applications. For instance, having several ways to route data packets along computer networks may help network administrators or even routing protocols in alleviating congestions and conflicts." in my opinion is not correct for the transports.

The sentence has been changed to point out that such multiplicity of solutions might be useful in some applications such as routing in data networks.

6. The references are not alphabetically ordered

We have alphabetically ordered the references