Computing Multicriteria Shortest Paths in Stochastic Multimodal Networks Using a Memetic Algorithm

Omar DIB
IRT SystemX, Paris-Saclay
Palaiseau, France
omar.dib@irt-systemx.fr

Alexandre CAMINADA
OPERA-UTBM-UBFC
Belfort, France
alexandre.caminada@utbm.fr

Marie-Ange MANIER
OPERA-UTBM-UBFC
Belfort, France
marie-ange.manier@utbm.fr

Laurent MOALIC
OPERA-UTBM-UBFC
Belfort, France
laurent.moalic@utbm.fr

***Abstract*— the human mobility is always organized nowadays in a multimodal context. However, the transport system has become more complex. For the sake of helping passengers, building Advanced Travelers Information Systems (ATIS) has therefore become a certain need. Since passengers tend to consider several other criteria than the travel time, an efficient routing system should incorporate a multi-objective analysis. Besides, the transport system may behave in an uncertain manner. Integrating uncertainty into routing algorithms may thus provide more robust itineraries. The main objective of this paper is to propose a Memetic Algorithm (MA) in which a Genetic Algorithm (GA) is combined with a Hill Climbing (HC) local search in order to solve the multicriteria shortest path problem in stochastic multimodal networks. As transport modes, railway, bus, tram and metro are considered. As optimization criteria, stochastic travel time, number of changes and walking time are taken into account. Experimental results have been assessed by solving real life itinerary problems defined on the transport network of the city of Paris and its suburbs. Results indicate that unlike classical deterministic algorithms and pure GA and HC, the proposed MA is efficient enough to be integrated within real world journey-planning systems.**

Keywords—multimodal route planning, stochastic networks, genetic algorithm, local search, memetic algorithm

I. Introduction

The demand for efficient routing methods that help passengers navigating through the intricate transport scheme has gained significant attentions in recent years. Several companies and transport operators are tending nowadays to build intelligent routing systems whereby passengers' routing queries are efficiently answered.

Building such systems requires considering the various properties of the transport system and meeting the needs and preferences of each passenger. To achieve that, several challenges have to be addressed. Firstly, an efficient representation for the transport system should be established in order to cope with basic and complex routing problems. Secondly, powerful routing algorithms should be designed in order to solve routing issues. The routing problem addressed in this paper refers for solving the multicriteria shortest path problem in a stochastic multimodal transportation network.

The first difficulty to solve this problem lies in handling a multicriteria optimization problem. Indeed, in a multiobjective context, there is not only one single optimal solution, but rather a set of nondominated solutions, from which the decision maker must select his/her most preferred one. Determining such Pareto set is a tedious task since one problem may have a huge number of nondominated solutions.

The second difficulty originates in the presence of stochastic elements in the problem, which represents travel times on edges. Stochastic travel times may be the result of many unanticipated events such as incidents and weather conditions. Indeed, handling uncertainty will undoubtedly increase the algorithm's search space. Consequently, the number of nondominated solutions will drastically increase and so does the algorithm's running time.

The aforementioned difficulties will make the use of standard shortest path algorithms infeasible due to their high computational time and inability to deal with time variant stochastic networks. To overcome that, we propose in this paper an approximate approach whereby high quality solutions are computed in a reasonable computational time.

The proposed method is based on a combination between two metaheuristics: a Genetic Algorithm (GA) that belongs to the population-based metaheuristics and Hill Climbing (HC) that belongs to single solution metaheuristics.

The main advantage of using GAs stems from the fact that they are robust adaptive optimization techniques. Thus, GAs have the ability to perform efficient search on poorly-defined spaces by maintaining an ordered pool of strings that represent regions in the search space. That is, GAs avoid randomness by intelligently visiting the search space.

Besides, GAs have shown high performances in solving real-world optimization problems whether in deterministic or stochastic environments. Furthermore, GAs have also resulted in high efficiency when dealing with various kinds of optimization problems such as mono/multicriteria problems.

Despite that GAs have been successfully applied to many optimization problems, better performance can still be achieved by enhancing their search process. For this reason, Memetic Algorithms (MAs) [1] have been introduced. MAs can be seen as an extension of GAs since they exploit a population based global search technique in order to identify promising search regions. In contrast with GAs, MAs use local search procedures in order to perform local refinements; as a result, better search regions will be identified and probably better solutions will be obtained. A HC approach is used in this paper as a local search procedure.

The remaining of this paper is structured as follows: in next section, the formal description of the problem is introduced. In Section 3, some related works are presented. In Section 4, the way the multimodal network has been represented is explained. Section 5 is devoted to introduce the proposed MA. Experimental results are presented in Section 6. Finally, Section 7 gives some comments and outlines future works.

II. Problem Definition

The routing problem studied in this paper refers to a multicriteria shortest path problem in a time dependent and stochastic multimodal network. This problem can be formally described as follows: given a directed graph $G = (V, E)$, where V is the set of vertices and E is the set of edges with cardinality $|V| = n$ and $|E| = m$ and a d-dimensional function $w: E \rightarrow [\mathbb{R}+]^d$. Each edge e belonging to E is associated with a weight vector $w(e,t)$ that depends on the time at which e is crossed. A source vertex s that represents the departure station and a sink vertex z that represents the arrival station and a departure time dt are identified. A path p is a sequence of vertices and edges from s to z with respect to dt.

The cost vector $W(p)$ for linear functions of path p is the sum of the weight vectors of its edges, that is $W(p) = \Sigma_{e \in p} c(e,t)$. Given the two vertices s and z; let $P(s, z)$ denotes the set of all s-z paths in G. If all objectives are to be minimized, a path $p \in P(s, z)$ dominates a path $q \in P(s, t)$ iff $W_i(p) \leq W_i(q)$ for all indices $i, i \in \{1,\ldots,d\}$ and $W_j(p) < W_j(q)$ for at least one index $j, j \in \{1,\ldots,d\}$. A path p is Pareto-optimal if it is not dominated by any other path. The set of all nondominated solutions is called the Pareto-optimal set. The ultimate goal when solving the studied routing problem is then to compute the set of nondominated solutions Q $(Q \subset P(s,z))$ of $P(s,z)$ with respect to c and the departure time dt.

III. Related Works

Routing is a widely studied topic, mainly because of its relevance to real world applications in a wide range of fields such as transportation, energy and communication networks. Several works have been done in order to solve basic and complex routing issues. For instance, the most common way to solve the one-to-one shortest path problem in a deterministic graph is to apply the algorithm of Dijkstra [2]. This latter can also be generalized to solve the multimodal shortest path problem with respect to one criterion such as travel time, walking time and number of exchanges [3].

To cope with multicriteria shortest path problems in small and deterministic networks, several algorithms have been proposed such as the multilabel setting algorithm [4]. The main aim of such algorithms is to compute the whole set of non-dominated solutions to go from one place to another. Such exact algorithms were also accelerated by using speed up techniques that lie in computing some data in the offline mode to use them in the online mode [5].

Despite the fact that such accelerating techniques succeeded at enhancing the performance of traditional multicrieteria algorithms, applying them in real world contexts remains an issue due to their high computational time when the network's size is very large or the number of considered criteria becomes important [6]. Additionally, the use of speed up techniques may become infeasible due to the dynamic and stochastic properties of the transport system [7].

As a result, several researchers and practitioners have used heuristic approaches, and mainly metaheuristics, in order to solve multicriteria shortest path problems. For instance, [8] proposed a GA to solve the bicrieteria shortest path problem in a deterministic and time invariant graph with respect to two conflicting objectives; the transportation cost and the total travel time. [9] also proposed a hybrid metaheuristic in which a GA is combined with a VNS in order to solve the multicriteria routing issue in a deterministic network.

In stochastic networks, several approaches have been proposed in order to solve shortest paths; however, in most of them, the travel time was the one and only one considered criterion. For instance, [10] proposed an efficient algorithm to compute the least expected shortest path in a time-varying network. In their work, stochastic edges are transformed to deterministic edges by using the expected value of their distribution functions. Experimentations show that their proposed algorithm succeeded in providing more robust solutions than traditional deterministic approaches; however, that was at the expense of nonpolynomial runtime complexity even when only one criterion is considered.

Other approaches were also proposed to various extensions of shortest path problems in stochastic networks such as [11] who proposed an efficient algorithm in order to solve the least expected shortest path with having a guarantee to arrive at the destination node before certain time.

To remedy the limitations of exact approaches, approximate approaches and mainly metaheuristics were used to solve optimization problems under uncertainties. For instance, [12] introduced a hybrid metaheuristic in which a Greedy Randomized Adaptive Search Procedure (GRASP) is combined with a Variable Neighborhood Search (VNS) in order to solve the vehicle routing problem with stochastic demands. For more information about how metaheuristic can be extended to deal with stochastic combinatorial optimization problems, readers can refer to [13].

IV. Modelling Approach

This section considers modelling a multimodal transportation network. It should be clarified that the term multimodal 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 appropriately modelling the timetable in order to enable the computation of journeys. Roughly speaking, a timetable consists of a set of stops (such as bus 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. The key point in this representation lies in modelling each transportation mode as a separate directed graph. An additional work is then done in order to integrate all sub-graphs into one larger graph.

A. Components Modelling

As a first step of modelling, two types of nodes are introduced, which correspond to stations and platforms. A station comprises a set of platforms where passengers wait for vehicles. It is worth mentioning that most of representations in the literature disregard platforms. Instead, they only focus on vehicles. However, platforms are essential since transfers inside stations are made between platforms. Moreover, in some routing issues such as evacuations, platforms play an essential role since they give ideas about the load of passengers or even the saturation of the transport system. As a result, we decided to integrate platforms into the modelling scheme.

A platform cannot belong to more than one station. Each platform has also a type (Bus, railway, tram...). This information can be used by routing algorithms that deal with users' preferences (e.g. a user may prefer to only take the bus mode along his/her journey).

After introducing the nodes that will construct the basis of this modelling approach, the interaction between them is now introduced. Interactions are always done via directed edges where each edge may have its own properties.

The first category of edges refers to modelling the fact that a station may comprise one or several platforms. A directed edge in inserted between a station *'S'* and each of its platforms.

Introducing this type of edges is essential for routing issues. For instance, if a routing request requires starting from one station, the search process has to start from every platform inside the station. Therefore, introducing separate nodes for stations and platforms, and linking both nodes is primal. It is important to mention that the degree of stochasticity associated with this type of edges is not very high since accessing a platform from the access point of a station is usually an easy task that does not depend on many external factors.

The second category of edges consists of modelling transfers between platforms. Transfers may happen between platforms belonging to the same station or platforms belonging to different stations. This category of edges plays an important role in the routing process since it gives an idea about the number of transfers that a passenger has to make before arriving at his/her arrival destination. Additionally, transfer edges help in computing the total walking time of the journey. Finally, transfer edges do usually depend on many external factors that may affect their associated travel times. Therefore, the degree of uncertainty along transfer edges is usually important and has to be considered.

The last category of edges consists of modelling a vehicle going from one platform belonging to one station to another platform belonging to another station. This category refers also to modelling timetable information. Since a timetable consists of time-dependent events (e.g. a vehicle departing at a stop) that happen at discrete points in time, time dependent edges are used to account for going from one platform to another by a vehicle. This latter may be a bus, railway or even a tram. The travel time of edges in this category may be affected by many factors such as delays, accidents, technical problems. Therefore, the degree of stochasticity along edges in this category is usually high.

Based on this modelling approach, we present in the next figure a multimodal network that consists of four stations, five platforms, two transfer edges, three edges accounting for vehicles and five edges to account for the linkage between stations and their associated platforms.

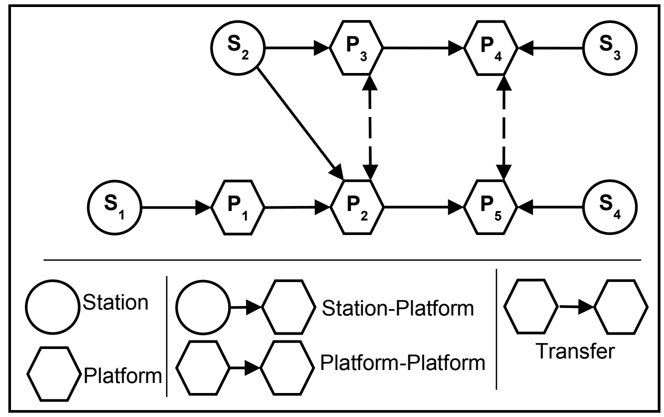

Figure1: Modelling a multimodal network

B. Cost Modelling

We explain in this section the cost value of an edge in the final multimodal graph. Indeed, the cost only represents a travel time since other criteria (i.e. number of transfers and walking time) will be computed on the fly while the algorithm performs its search process.

In the first order, the travel time can take a static value that does not depend on time. Edges belonging to this category will have one and only one static travel time value whatever the situation is. We consider in this work that edges linking stations and platforms are static and time-invariant. The cost of an edge in this case represents the minimal travel time required to access a platform from its parent station.

In the second order, edges can have stochastic and time-invariant costs. In this category, the travel time may take a random value within an interval; however, the travel time distribution is the same whenever the edge is crossed. In this work, we consider transfer edges as stochastic and time-invariant. In another term, the transfer from one platform to another is a stochastic value, but an edge has one and only one time distribution function.

In the third order, unlike the other categories, edges here are stochastic and time-dependent. Example of edges belonging to this category are those linking two platforms via a vehicle. To model costs in this category, for each vehicle departing, we add an interpolation point to the corresponding edge. The timestamp of the interpolation point represents the departure time of the vehicle whereas the weight represents the travel time along this edge. Evaluating this edge at a specific time t is done by identifying the interpolation point, which represents the first vehicle departing respecting t, and then extracting the weight corresponding to this interpolation point.

To account for uncertainty among all stochastic edges, we consider that the travel time is a random variable that varies according to a custom probability law. Based on the cost modelling, we show in the next figure the three cases that may arise while assigning costs to edges in the multimodal graph.

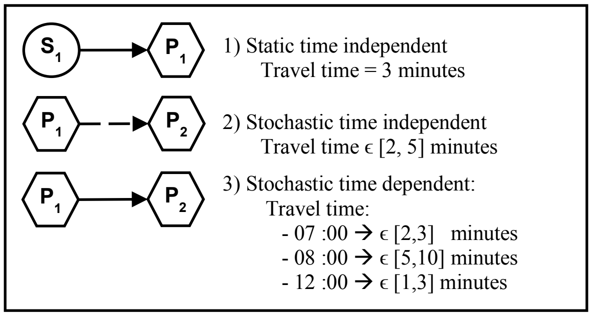

Figure2: Cost modelling

Up to this modelling level, the different components of the transport system, as well as, its various properties required to solve the emerging routing issue are present in the final multimodal graph. In the next Section, we will introduce the proposed MA that has been applied over this model in order to deal with the studied routing problem.

V. Memetic Algorithm

As in standard GAs, the proposed approach proceeds with a set of initial solutions. These latter are generated using a constructive heuristic based on a double search algorithm [14]. A forward search is performed starting from the departure station with respect to the departure time; a backward search is also performed from the arrival station. A solution (path) is then found whenever the two searches intersect. The result of this operation is a set of initial feasible solutions to go from the departure to the arrival station respecting the departure time.

A. Encoding Scheme

To encode solutions, a list of nodes is used. Each node corresponds to a platform. This list forms a path from the departure to the arrival station. The information stored in each node is related to the identifier of the platform, its transportation mode and the time at which the node is traversed in addition to the value of the different objectives considered. The length of a chromosome is variable and may not be greater than the number of platforms in the network.

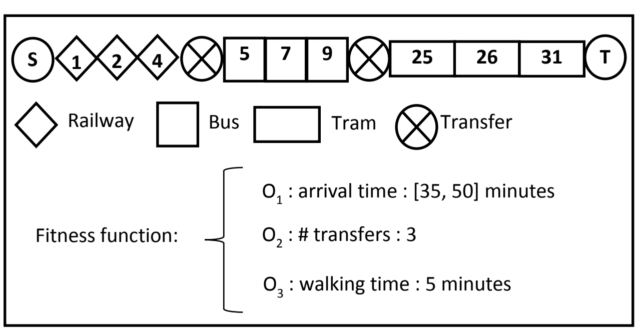

Figure 3: Example of a solution with its fitness function

B. Evaluation

The fitness function of an individual is a three dimensional vector where each dimension represents one criterion. We show in Figure 3 an example of how a route from S to T is encoded. As can be seen, each node that represents a platform and has its own type and properties is encoded within the vector. Additionally, a special type of nodes is inserted to account for transfers. The path from S to T consists of taking three transport modes. The fitness function is represented by a 3-dimensional vector containing the value of each criterion.

Since custom distribution functions are used to account for travel times, the expected value extracted from the travel time distribution at the destination station is then used to compare the travel time criterion between two solutions.

To go from one station to another with respect to a departure time, travel time distributions on links have to be summed up. We assume in this paper that links are completely correlated (i.e. random variables are stochastically dependent). To merging two distributions, the pointwise sum is used. In case that random variables are stochastically independent, the convolution product can be used instead in order to propagate probabilities. In real world situations, and especially in public transport modes, disruptions delaying a vehicle will also cause delays for successors vehicles. Thus links are more likely to be dependent variables.

C. Local Search Operator

After generating initial solutions, the quality of the initial population is enhanced by applying a Hill Climbing local search over each individual. By doing so, good initial individuals will be used in the evolution process.

In the following, we explain how the local search is applied. As in most of single solution metaheuristics, the main challenge lies in finding an efficient strategy to move from one solution to another. To do so, a neighboring structure should be identified. In this work, the neighborhood structure consists of applying the double search algorithm used for generating initial solutions in order to replace a subpath with a newly generated one. That is, a random edge $e(x, y)$ is selected from a path (individual); a forward search is then launched from the tail x and a backward search is launched from the head y; an alternative path is then found from x to y when the two searches intersect.

Replacing the edge e in the initial individual by one of its alternatives leads then to a neighbor solution. By searching the alternatives of each edge included in the initial path, the whole list of neighboring solutions can be constructed. The size of this list highly depends on the density of the network.

We show in the next figure an example of a solution and its neighborhood solutions. Assuming that the path P *(1→2→5)* is a solution extracted from the initial population. As can be seen, the edge (2, 5) can be replaced by the path (2→3→5) and (2→4→5). Therefore, the paths (1→2→3→5) and (1→2→4→5) are the neighborhood solutions of the initial path (1→2→5).

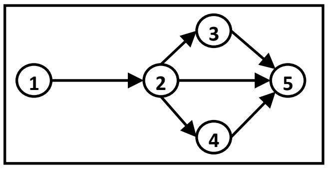

Figure 4: Neighborhood structure

After defining the neighborhood structure, we now explain the processes whereby a neighbor solution is chosen to replace the current incumbent solution. We use in this work the best neighborhood strategy. Deciding which solution is the best among all solutions in the neighborhood list is not an easy task since several criteria are taken into consideration.

To overcome that, we compute for each solution I in the neighborhood list, the number of solutions that I dominates in the list itself. More a solution dominates other solutions, better it is. Therefore, the best solution among all solutions in the neighborhood list is the one that has the highest number of dominated solutions. In case that two solutions have the same number of dominated solutions, one is randomly chosen.

After identifying the best neighbor solution I^*, the number of solutions dominated by the initial solution I_0 is also computed. If I^* dominates more solutions than I_0, then I^* replaces I_0. Otherwise, it can be said that the local search is get trapped into a local minimum.

D. Selection

After enhancing initial solutions, the algorithm continues to perform genetic operations until a stopping criterion is met. We start with the selection operation that consists of selecting individuals for the reproduction phase.

To perform the selection, a stochastic roulette wheel technique is used. More precisely, a roulette wheel is used where all chromosomes are placed according to their selection probability. Therefore, the better the chromosomes are, the more chances to be selected they have.

Since several criteria are considered, a comparison mechanism is implemented to decide which are the best chromosomes in the current population. For this purpose, a rank is assigned to each individual according to its objectives' values. An individual is then better than another if its associated rank is higher. The rank of an individual represents the average value of all ranks determined by sorting all individuals in the population according to each objective.

To illustrate the selection process, an example is given in Table 1. It is assumed here that the current population consists of seven individuals that represent solutions for a two criteria minimization problem. After sorting individuals according to the first objective f_1, a rank $R(f_1)$ is computed; a second rank $R(f_2)$ is computed according to the second objective f_2. A global rank GR is finally computed by dividing the sum of two ranks by the number of objective.

Based on the global rank, a selection probability is computed for each individual by dividing its rank over the sum of ranks of all individuals in the population.

Table 1. Computing ranks and assigning selection probability

I	$S(f_1)$	$S(f_2)$	$R(f_1)$	$R(f_2)$	GR	p_i
I_1	5	1	1	6	7/2	0.13
I_2	4	2	2	4	6/2	0.11
I_3	3.5	1.5	3	5	8/2	0.15
I_4	2	1.5	5	5	10/2	0.19
I_5	3	4	4	2	6/2	0.11
I_6	2	3	5	3	8/2	0.15
I_7	1	5	6	1	7/2	0.13

I: individual *S*: Score *R*: Rank *GR*: Global Rank
Pi: probability of selection

As can be seen from the table above, the global average rank of individuals I_2 and I_5 is the smallest average *(6/2)*. This property reflects very well that I_2 and I_5 are dominated by I_4; therefore, their survival probability will decrease. Since I_4 represents the individual that dominates the maximum number of individuals in the current population, the selection strategy results in a high GR for I_4. It is worth mentioning here that by using this selection technique, poor solutions will not be totally prevented from passing their genes to the next generation; however, their survival probability is lower than good solutions. Therefore, this selection operator can be also seen as a diversification mechanism for the genetic evolution.

E. Crossover

To perform the crossover, two individuals are selected using the aforementioned selection operator and some information are then exchanged in order to provide offspring.

The multiple point crossover has been used as a crossover operator. A crossover point is chosen to be a node where passengers exchange from one mode to another. In case that no crossover points can be found between parents, another parent is selected. After selecting two individuals, new individuals are produced. By doing so, a new population having twice the size of the current population is produced. The best half individuals are then selected for the next generation according to their global rank and the rest are ignored.

It is important to mention that after producing offsprings via this crossover technique, the algorithm ignores the feasibility of solutions since offsprings will always represent a correct itinerary from the origin to the destination station. However, time at each node has to be adjusted according to the departure and arrival times at platforms. This will also require modifying the values of the different criteria. We show in Figure 5 an example of the crossover operation applied over two individuals P_1 and P_2. We assume in this example that P_1 and P_2 have two transfers in common. After exchanging genes between parents, new individuals I_1 and I_2 are produced.

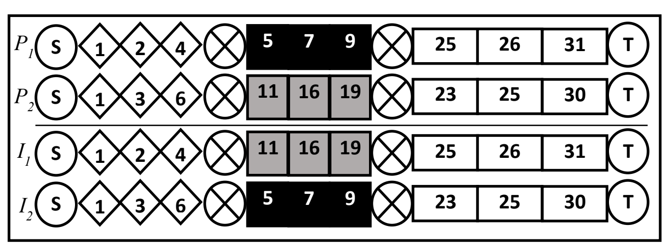

Figure 5: Crossover operation

To avoid losing the elite solution, the best solution (i.e. the solution with the highest selection probability) at each generation is copied as it is to the next generation. An archive is also used to store all nondominated solutions while going from one generation to another. Moreover, since there is a chance that the same individual is duplicated in the population as the generations go on, duplicated individuals are therefore, replaced with newly generated chromosomes.

F. *Mutation & Termination Conditions*

To accomplish the mutation operation and in contrast to standard GAs, the hill climbing local search is used. That is, the local search is applied over survival individuals in order to enhance their quality.

The algorithm stops when it fails to find interesting solutions during α continuous steps. An interesting solution is a new generated individual that is not dominated by any of the individuals in the current population P or is a solution that at least dominates one individual in P. Another stopping criterion is when the algorithm reaches a β maximum number of generations. These two parameters can vary according to the preference and need of the user.

VI. EXPERIMENTAL RESULTS

To evaluate this work, a routing application based on the real data of the French region Île-de-France that includes the city of Paris and its suburbs has been developed.

A. *Experimental Setup*

Data that comprise geographical information and theoretical timetable information for four transport modes (Bus, Metro, Railway, and Tram) are provided by the transport organization authority that controls the Paris public transport network. More precisely, data encompass 17950 stations; 41047 platforms; 195000 transfers; 303000 trips and 6800000 events for one day. Travel times on stochastic edges are assigned custom distribution functions resulting from a prediction module based on historical data and observations.We show in Figure 6 the region where this study is taking place.

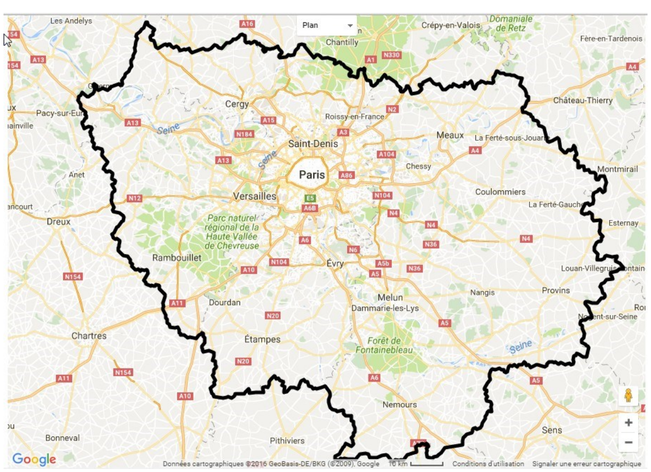

Figure 6: Case Study : Ile-de-France Region

The proposed MA is compared with the following approaches a) a multicrietria label-setting algorithm that can compute the set of nondominated solutions [5]; this algorithm is applied a priori; the travel time in this case is static and represents the exact theoretical time extracted from timetable information. b) A pure GA that has the same properties of the proposed MA except that initial solutions are not enhanced via local search. Moreover, the mutation used is not based on local search but on the algorithm used to generate initial solutions. That is, a subpath is replaced by one of its alternatives to accomplish a random walk in the search space. c) A pure hill climbing approach that starts with one single solution and tries to apply local refinements until a local optimum is reached d) A standard multicriteria algorithm that is applied a posteriori (after knowing the state of the network). This algorithm is the reference since it gives the optimal set of non dominated solutions a posteriori.

The following parameters are used to initialize the MA: the initial population size is 5; the probability of crossover is 0.9; the mutation rate (HC) is 0.9. The pure GA has the following parameters: the initial population size is 100; the crossover probability is 0.9; the mutation probability is 0.1. The number of generations used to ensure a fixed state in the population is 100 in both GA and MA. These parameters result in the best performance for both the proposed MA and the pure GA. We present later in this paper some experimentations made in order to tune some of these parameters.

Algorithms have been tested on an Intel core I5 machine of 8 GB RAM and developed in Java. The comparison is done by solving 1000 routing queries, each having a departure, arrival stations and a departure time uniformly generated at random.

For each query, 1000 scenarios are considered. A scenario is an instance of the multimodal graph and is constructed based on real time traffic data. Instances are uniformly picked at random. Two indicators are used for comparisons a) the average running time and b) quality of nondominated solutions obtained from each method with respect to the multicriteria label setting algorithm applied a posteriori (i.e. after fixing a scenario and having the exact travel time values); in this case, the optimal Pareto set is obtained. The quality of solutions is computed by dividing the hypervolume indicator value of the Pareto front resulting from each method over the true Pareto front resulting from the multicriteria algorithm applied a posteriori. That is we measure the average GAP between an approximate set of solutions and the optimal Pareto front using the hypervolume indicator. We show in the equation below how the average GAP is computed. HI_{opt} refers to the value of the hypervolume indicator of the optimal Pareto front; HI_{best} accounts for the value of hypervolume indicator of an approximate Pareto front (several executions have been performed and the best value is considered for comparison). It is obvious that smaller the GAP is, better it is.

$$GAP = ((HI_{opt} - HI_{best})/ HI_{opt}) * 100)$$

The hypervolume indicator has been chosen as a performance indicator to assess approximate approaches since it reflects the quality of an approximation based on two criteria: diversity and closeness to the optimal Pareto front. It is worth to mention that a reference point Z_{ref} (”anti-optimal” or nadir point) which refers to the worst possible point has to be used in order to compute this indicator. Besides, normalizing criteria is a crucial operation for this performance indicator.

B. Experimental Analysis

Results in Table 2 show that better itineraries are obtained using the proposed MA and the pure GA in comparison with the deterministic approach (applied a priori). While the average GAP to the optimality of the proposed MA does not exceed 3% and 8% when the pure GA is considered, the average GAP of the deterministic approach may increase to 15 %. Analyzing results clearly prove that both MA and GA are able to provide more robust solutions than the deterministic algorithm. When it comes to the Hill Climbing, results indicate that its average GAP to the optimality may reach 25% and 29% in the worst case. Therefore, it can be said that solutions provided when using the HC approach are very poor and not robust enough to cope with travel time changes.

Table 2. Experimental results

Approach	Running time(s)		GAP (%)	
	average	worst	average	worst
(a) DA-(a priori)	180	300	15	21
(b) Genetic Algorithm	0.13	0.16	8	12
(c) Hill Climbing	0.12	0.14	25	29
(d) DA-(a posteriori)	180	300	0	0
Memetic Algorithm	15	19	3	6

As can be easily noticed, comparing results proves that the proposed MA results in better performance in terms of quality of solutions in comparison with the pure GA and HC. Consequently, it can be concluded that the proposed MA is an efficient approach in terms of quality of solutions to solve the studied routing issue. This can be explained by the fact that MA exploits the advantages of both approaches (GA an HC) to efficiently visit the search space and thereby find better solutions. From one side, the MA benefits from the capacity of GAs in exploring a wide search space, and take the advantageous of using a local search procedure inside its genetic operations in order to better exploit interesting regions in the search space from another side.

Since an archive is used in metaheuristics, its size can also be a good indicator for comparison. More the size of the archive is high, more a metaheuristic is capable of providing nondominanded solutions. Experimentations have shown that the proposed MA succeeded in average at obtaining 15 nondominated path, whereas that decreases to 7 in the GA and to 4 in the HC. This can be explained by the fact that the MA integrates efficient diversification mechanisms in the mutation and other genetic operators. However, that is not the case for the GA that uses a random mutation operator and for the HC that rapidly gets trapped at local minima.

When it comes to the evaluation of the running time performance, results show that deterministic algorithms suffer from high computational time. The ~ running time of the deterministic approaches may increase to 3 minutes and 5 minutes in the worst case. This can be explained by the fact that deterministic approaches exhaustively visit the whole search space in order to find all nondominated solutions and to guarantee that no other solutions exist.

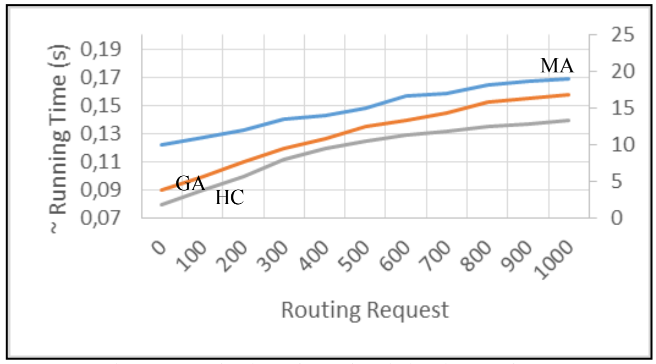

Figure 7: Evaluation of Metaheuristics time performance

When it comes to the heuristic approaches, results indicate that the ~ running time of the proposed MA does not exceed 19 seconds while it decreases to 135 milliseconds for the pure GA and to 120 for the Hill climbing. The high computational time of the proposed MA in comparison with the GA and HC can be explained by the fact that additional time is required in the proposed MA to enhance initial solutions and to perform the mutation via the local search. The low computational time of the HC can be explained by the fact the approach rapidly converges to a local minimum. The time performance of metaheuristics is shown in Figure 7. Only running time performance of metaheuristics is plotted since the running time of other approaches is very high.

C. Sensitivity Analysis

We study in this section the impact of different parameters on the performance of the proposed MA. We proceed with the probability of crossover and mutation operators. These two parameters may highly affect the performance of a population-based metaheuristic.

As in most traditional GA schemes, the crossover rate is always high, while the mutation probability is very low. However, in the proposed MA, the mutation and crossover rates used are high. Experimental results (see Table 3) have shown that the best crossover and mutation rates for the studied problem using the aforementioned data instances are 0.9. Since the crossover is a convergence operation, which is intended to pull the population towards a local minimum/maximum, performing the crossover over all individual will lead to premature convergence. On the other side, results indicate that using less than 0.9 as a crossover probability will affect the final quality of solutions. That is, the average GAP to the optimality will increase as the crossover rate decreases.

Results in Table 3 have also indicated that decreasing the mutation rate will prevent the algorithm from converging towards interesting regions within the search space. This can be explained by the fact the mutation is a divergence operation and is usually intended to occasionally break one or more members of a population out of a local minimum/maximum space and potentially discover a better minimum/maximum space. On the other hand, decreasing the mutation rate to less than 0.9 will lead to premature convergence. Thus, the quality of final solutions found will be degraded.

Table 3. Effect of crossover and mutation rates

Crossover rate	Mutation rate	Running time(s)		GAP (%)	
		~	worst	~	worst
0.1	0.1	7	9	12	17
	0.5	8	11	10	15
	0.9	9	12	8	15
0.5	0.1	8	13	11	16
	0.5	10	14	8	15
	0.9	11	14	7	13
0.9	0.1	12	15	8	10
	0.5	14	16	5	8
	0.9	15	19	3	6

The last parameter we want to study is related to the strategy chosen while moving from a solution to one of its neighbors in the Hill Climbing approach. As previously explained, the best neighbor strategy has been used in this paper. Other strategies can be used such as the first enhancing neighbor or a random enhancing neighbor such as in the stochastic HC.

Table 4. Performance of several moving strategies

Approach	Running time(s)		GAP (%)	
	average	worst	average	worst
HC-first	0.11	0.12	27	32
HC-stochastic	0.19	0.28	23	28
HC-best	0.12	0.14	25	29
MA-(HC-first)	13	25	5	8
MA-(HC-stochastic)	20	32	2.9	5.7
MA-(HC-best)	15	19	3	6

Repeating experimentations after varying the moving strategy has indicated that the best neighbor represents the best compromise between the convergence time and solutions' quality for both the HC when it is solely used and when it is used inside MA (see Table 4). The average GAP has been slightly enhanced when selecting a random solution (HC-stochastic) from the list of solutions that are better than the initial solution. However, that little improvement in the solutions' quality caused a remarkable increase in the computational effort. On the contrary, selecting the best enhancing neighbor did significantly increase the running time, however, a decrease in the quality of solutions have been noticed. Therefore, we chose the best neighboring as a moving strategy to be used inside the local search.

VII. Conclusions

We have proposed in this paper a memetic algorithm for computing shortest paths in a stochastic multimodal network. This latter has been modeled so that the dynamic and stochastic aspects of the transport system are taken into consideration. The proposed MA that consists of a combination between a GA and a local search procedure shows a good performance in terms of quality of solutions and computational time in comparison with other approaches. As can be noticed from the experimental phase, several parameters have to be tuned in order to achieve the best performance. This issue of tuning will undoubtedly be an essential step for applying the proposed approach to solve other real world optimization problems.

Finally, several works have been planned to be done in the future. Firstly, we will try to distribute the proposed MA by making subpopulations independently evolve and exchange information about the search space when necessary. Such technique will undoubtedly enhance the efficiency of the whole evolution process. Besides, using other local search procedures such as Tabu Search or Simulated Annealing may also enhance the approach's performance. It is worth finally to mention that this work will be soon integrated within a real world routing system that help passengers to accomplish their trips in the city of Paris and its suburbs.

Acknowledgment

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] Neri, Ferrante, and Carlos Cotta. "Memetic algorithms and memetic computing optimization: A literature review." Swarm and Evolutionary Computation 2 (2012): 1-14.

[2] Skiena, S. "Dijkstra's algorithm." Implementing Discrete Mathematics: Combinatorics and Graph Theory with Mathematica, Reading, MA: Addison-Wesley (1990): 225-227.

[3] Kirchler, Dominik. Efficient routing on multi-modal transportation networks. Diss. Ecole Polytechnique X, 2013.

[4] Zhang, Min-Ling, and Zhi-Hua Zhou. "A review on multi-label learning algorithms." IEEE transactions on knowledge and data engineering 26.8 (2014): 1819-1837.

[5] Bast, Hannah, et al. "Route planning in transportation networks." Algorithm Engineering. Springer International Publishing, 2016. 19-80.

[6] Goel, Nidhi, et al. "Optimal routing algorithm for large scale multi-modal transit networks." (2017).

[7] [Ardakani, Mostafa K., and Madjid Tavana. "A decremental approach with the A∗ algorithm for speeding-up the optimization process in dynamic shortest path problems." Measurement 60 (2015): 299-307.

[8] Mohamed, Cheikh, Jarboui Bassem, and Loukil Taicir. "A genetic algorithms to solve the bicriteria shortest path problem." Electronic Notes in Discrete Mathematics 36 (2010): 851-858.

[9] Dib, O., et al. "An advanced GA–VNS combination for multicriteria route planning in public transit networks." Expert Systems with Applications 72 (2017): 67-82.

[10] Miller-Hooks, Elise D., and Hani S. Mahmassani. "Least expected time paths in stochastic, time-varying transportation networks." Transportation Science 34.2 (2000): 198-215.

[11] Randour, Mickael, Jean-François Raskin, and Ocan Sankur. "Variations on the stochastic shortest path problem." International Workshop on Verification, Model Checking, and Abstract Interpretation. Springer Berlin Heidelberg, 2015.

[12] Mendoza, Jorge E., Louis-Martin Rousseau, and Juan G. Villegas. "A hybrid metaheuristic for the vehicle routing problem with stochastic demand and duration constraints." Journal of Heuristics 22.4 (2016): 539-566.

[13] Juan, Angel A., et al. "A review of simheuristics: Extending metaheuristics to deal with stochastic combinatorial optimization problems." Operations Research Perspectives 2 (2015): 62-72.

[14] Niksirat, Malihe, Mehdi Ghatee, and S. Mehdi Hashemi. "Multimodal K-shortest viable path problem in Tehran public transportation network and its solution applying ant colony and simulated annealing algorithms." Applied Mathematical Modelling 36.11 (2012): 5709-5726.