Memetic algorithm for solving one-to-one shortest path problem

Omar DIB[1], Alexandre CAMINADA[2], Marie-Ange MANIER[3]

[1] Technological Research Institute SystemX, 91120 Palaiseau, France
{omar.dib}@irt-systemx.fr

[2] OPERA– UTBM, 90010 Belfort Cedex France
{Omar.dib, alexandre.caminada, marie-ange.manier }@utbm.fr

Keywords*: Shortest path, exact algorithms, meta-heuristics, optimization*

1 Introduction and motivations

The shortest path problem (SPP) is widely applied in different fields of application such as transportation, telecommunication and networks. It aims at determining the path between two points with the minimum associated/allocated resource (distance, time, cost…) in a directed or undirected graph. SPP has been well studied over the last 60 years, graph theory experts still endeavor to extract new variants of the basic problem in order to model and provide efficient solutions for several real life issues. Important forms of the SPP may be considered such as finding the shortest paths in a dynamic graph where the arcs' weights vary as functions of time. Moreover, finding shortest paths subject to some additional constraints like fixing the maximum number of visited nodes or precising the nodes' types, colors, etc.

To deal with the SPP and its different variants, several algorithms have been developed since 1956. The standard solution to the one-to-one shortest path problem, which is the target of our study, is Dijkstra's algorithm [1]. Its running time depends on the selected data structure. For the average case, one can get an O (|E|+|V|*log|V|) (|V| #Vertices, |E| #Edges). To deal with negative edge weight, we have the Bellman-Ford algorithm. SPP was also solved using MILP technique but its running time is usually higher than standard algorithms. MILP is widely used for solving many variants of SPP when standard algorithms fail for a lack of flexibility to cope with additional constraints but this usually leads to a significant execution time.

In practice, several speed-up techniques have been proposed to reduce the search space such as bidirectional search, A* algorithm, preprocessing techniques, etc. Although, those techniques are efficient and fast enough, they become less performant when the size of the network becomes very large or when additional constraints are added such as dynamic arcs weight, multi-criteria paths optimization. To overcome such shortcomings, we believe that meta-heuristics such as GA, local search may be efficient candidates to give near optimal solutions within acceptable time.

2 The Memetic algorithm and experimental results

Our method is actually based on hybridization between a genetic algorithm and a local search heuristic. Indeed, Genetic Algorithms (GAs) usually start with a population of solutions and try to enhance their quality by applying some operations such as crossovers and mutations. However, local search heuristics proceed with a single solution and try to move from solution to another one by applying local changes until they get stuck at a local optimum. One of the challenges in this kind of algorithms is to find one or a set of initial solutions (the qualities of initial solutions play an important role in the efficiency of the method).

In our case, the initial solutions are a set of paths generated intelligently using a double search technique. The idea is to simultaneously run a forward search from the origin point and a backward search from the target point. Thanks to this technique, we can reduce the amount of time to get a set of feasible paths at the algorithm's initial state. After getting initial solutions, we have repeatedly applied, with different probabilities, crossover and mutation operations in order to enhance our solutions. To do the crossover, we choose any two paths from the population and we try to find two common edges to be our crossover points. Once we detect those latters, we take parts from each initial path and we form then new solutions to the next generation. By doing this, our algorithm will have the chance to visit new regions in the search space. For the mutation operator, we have used a local search heuristic based on two-neighborhood structures [2]. The construction of the neighborhoods is done thanks to a preprocessing step accomplished during the generation of the graph. Each time our algorithm is stuck at a local minimum, we change the structure of the neighborhood. By following this technique, our algorithm will exploit and explore wide regions of the search space.

We have done experimentations on 50 different graphs instances (from small to large size) generated randomly thanks to a random graph generator that we developed using Java. We compared our approach with two other exact algorithms (Dijkstra and MILP). We implemented Dijkstra using a priority queue and we solved the MILP generated using Cplex 12.6 solver. The results showed that the running time of Dijkstra is highly better than MILP. However, our method outperforms Dijkstra and MILP with 5% average gap to the optimality. The average speed of our method is 5-times faster comparing to Dijkstra and more than 20-times comparing to MILP. Regardless the efficiency of our approach, we remarked that in special instances of graphs, it fails to outperform Dijkstra since the crossover and mutation operations of the GA may become less efficient. However, and contrary to conventional algorithms, our method is flexible and adaptable to be applied to any kind of SPP. It can take into account any kind of additional constraints by modifying the conditions of crossovers and mutations operators.

3 Conclusions and perspectives

We have addressed in our study the challenge of computing the one to one SP in a static graph. We proposed a memetic algorithm and we compared it with other exact algorithms. The results showed that meta-heuristics are good candidates to solve the SPP. In addition, they have more flexibility to cope with additional constraints. The next step in our research is to apply such efficient combination of meta-heuristics for computing multi-criteria shortest paths in a multimodal transportation system that has the capacity and the ability to evolve and change over time.

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] E. Dijkstra, "A Note on Two Problems in Connection with Graphs." Numerische Math. 1, 269-271, 1959.

[2] P. Hansen, N. Mladenovic, "Variable Neighborhood Search: Principles and application", European journal of Operational Research, 130(2001) 449-467.