Applying meta-heuristic algorithms with a parallel computation framework to simulation-based dynamic traffic assignment

Mostafa Ameli, Jean-Patrick Lebacque, Ludovic Leclercq

Applying meta-heuristic algorithms with a parallel computation framework to simulation-based dynamic traffic assignment

Mostafa AMELI [a,b], Jean Patrick LEBACQUE [a] and Ludovic LECLERCQ [b]

[a] University of Paris-Est, IFSTTAR, GRETTIA, France
mostafa.ameli@ifsttar.fr, jean-patrick.lebacque@ifsttar.fr
[b] Univ. Lyon, IFSTTAR, ENTPE, LICIT, F-69518, Lyon, France
ludovic.leclercq@ifsttar.fr

Extended abstract submitted for presentation at the hEART 2019 7th Symposium
Sept. 4–6, 2019, Budapest, Hungary

Word count ≃ 2350 words (excluding the references)
February 15, 2019

Extended abstract

The Dynamic Traffic Assignment (DTA) refers to the procedure of assigning trips to paths in a given transportation system considering the Origin Destination pair (OD) flow demand and the network dynamic traffic states. The main output of DTA is path flow distribution over all feasible paths for all OD pairs. Travelers in the traffic network usually attempt to minimize their own travel time (cost). The solution of the assignment problem which is based on Wardrop's first principle is called User Equilibrium (UE) (Wardrop, 1952). Calculating trip-based dynamic network equilibrium with is a challenging problem. The goal of this study is computing UE solutions in a simulation-based DTA process.

Many researches have shown that this problem can be represented as a fixed-point problem (Wang *et al.*, 2018). In order to solve the fixed-point problem, the iterative process is used to reassign a fraction of the users at each step. The goal is to monitor whether the reassignment process improves the solution or not. In other words, the algorithm consists, at each iteration of the calculation of the equilibrium, in reassigning the part of the users who have chosen a non-optimal path to a more efficient alternative. In a large-scale and trip-based setting it is not possible to guarantee that fixed point algorithms converge towards the optimal solution because of the step size and because there is no exact method for determining the step size in the literature (Szeto & Lo, 2005). There are some criteria such as the total gap to see how far the solution is from the optimal solution. Total gap is the total delay of users with respect to the shortest path travel time. It often happens that the total gap stops improving after some iterations because the step size is small and prevents the solution from being improved further (Levin *et al.*, 2015). From a computational point of view, the main drawback of these methods for addressing DTA on large-scale networks is that they can not be parallelized. This is because the existing algorithms need to know the last iteration results to determine the next best path flow for the next iteration. Indeed, they need the state of the network before carrying out the next iteration. Therefore, all of the existing methods work as serial algorithms to find the UE.

The goal of this study is to overcome the drawbacks of serial algorithms by using meta-heuristic algorithms that are known to be parallelizable. The meta-heuristics are iterative methods; they can be classified into two categories: single solution and population-based (Talbi, 2009). The single solution methods start with an initial solution, apply a process to improve the candidate solution to achieve the best solution by following a trajectory in the solutions space. The second class is the population-

based; the methods of this class aim to improve a set of solutions (population) by applying a specific process.

This study proposes two new solution methods based on two categories of meta-heuristic algorithms. The new extension of the Simulated Annealing (SA) method from the first category and adaptive Genetic Algorithm (GA) from the second category to solve trip-based network equilibrium problem. This study uses parallel simulations in order to better explore the solution space. The algorithms are developed generally to solve traffic assignment with parallel computation in order to consider more than one path distribution per iteration. It is obvious that with parallel simulation, the algorithm is going to run more simulations in comparison with existing methods but it is expected to carry out a better exploration of the solution space and consequently achieve better solutions in terms of quality and closeness to the optimal solution. Moreover, with parallelizing the framework, the algorithm could solve the problem with better computation time in comparison with classic methods.

Simulated Annealing (SA) method

SA algorithm is inspired by annealing in metallurgy. The basic simulated annealing algorithm is presented in Kirkpatrick *et al.* (1983). This study redesigns and adapts the classic SA to simulation-based traffic assignment. Figure 1 presents the SA algorithm of this study:

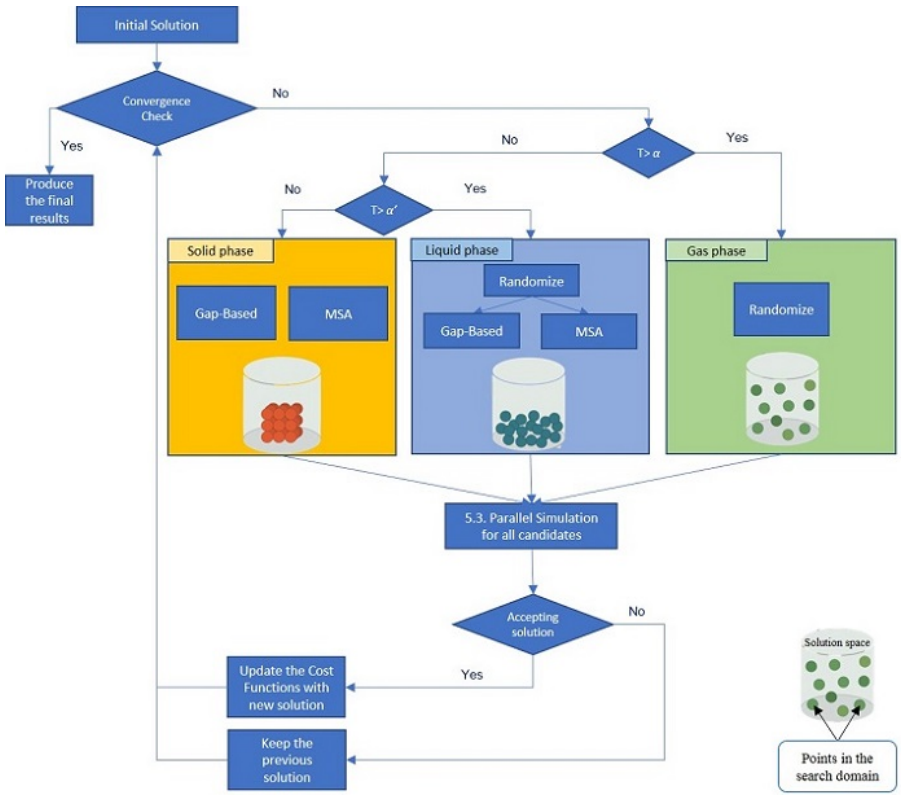


FIG. 1 – *SA solution algorithm flowchart*

The algorithm starts with an initial solution which is generated randomly. The next solution is generated with respect to the current one based on the temperature (T) of the current iteration. The current phase of the iteration depends on the temperature of the process. Inspired by the physics of matter, this study distinguishes three different methods to generate a neighbor solution, gas, liquid and solid; these methods represent the states of matter in nature. When the temperature is high ($T > \alpha$ where α denotes the boiling temperature), the *gas method* is applied. During the SA algorithm, by decreasing the temperature the algorithm enters the liquid phase ($\alpha > T > \alpha'$ where α' denotes the melting temperature) and then the *liquid method* is applied. When the temperature is quite low ($T < \alpha'$), the *solid method* is applied.

In the gas phase, we explore the solution space without limitation of any step size. Therefore, the candidates for neighbor solution correspond to random path flow distribution with respect to the demand value for each OD pairs (feasible OD-assignment). In the liquid Phase, we target exploring the solutions space randomly and also apply step size methods. First, we apply a randomizing process on the current solution, Then we optimize it by applying the Method of successive Average (MSA) (Robbins & Monro, 1951) to get the first solution and the Gap-Based method (Lu *et al.*, 2009) to get the second solution. In the solid phase, we execute the same process as liquid phase but without randomization. It means the two solutions are generated based on the current solution.

Afterward, the algorithm runs parallel simulations to update the network based on new different path flows that are obtained from the previous step. For a new solution s', the total gap $TGap(s')$ between the users' travel time and the shortest path travel time is calculated and corresponds to the energy of solution (E) compared to the current solution s. The last step consists in making a decision about accepting one of the best new solutions based on $TGap$ compared to the current solution of the algorithm. The acceptance decision is made by the binary test. If $P_s \geq R_s$, the new solution will be accepted. Here, $P_s = e^{\frac{-\nabla E}{T}}$, $\nabla E = TGap(s') - TGap(s)$ and R_{tr} is the random number $(0 < R_{tr} < 1)$. The quality of the solution is evaluated in the convergence check step. At the end of each iteration the temperature is decreased ($T = \frac{T_0}{ln(k+1)}$ where T_0 denotes the initial temperature and k denotes the iteration index) and the algorithm iterates until converging to the optimal solution or the lowest temperature (T_{min}) is reached.

Genetic Algorithm (GA)

The genetic algorithm was proposed by Holland (1992), it is inspired by the natural genetic variation and natural selection; selection, crossover, and mutation are the main operators of this approach. In the genetic algorithm, we use terms of chromosomes and genes to refer the different segments of an individual; in our implementation for the genetic algorithm, we consider our solution and the $TGap$ as an individual (DNA) with fitness value, our ODs-assignment as chromosomes and the paths flow as the genes. Finally the set of individuals constitutes a population. Figure 2 illustrates the solution structure for GA.

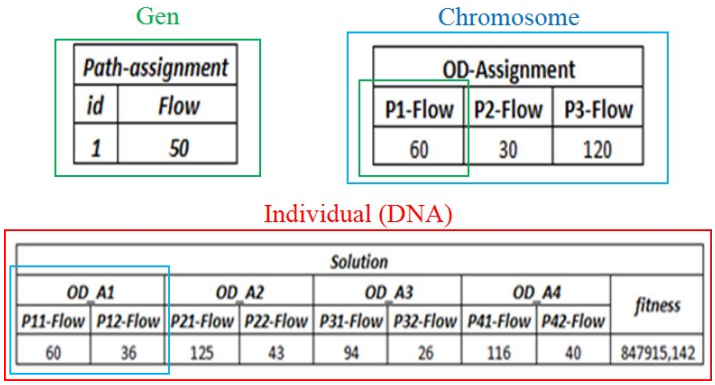


FIG. 2 – *Solution structure in the GA case*

Figure 3 presents the application of GA to the DTA problem. The GA process starts by generating the initial population (Initial set of individuals). In this study, we generate a random population. This study designs a two layer GA process in order to search solution space by changing the path flows in inner GA and overcome the draw back of OD impact by consider different combination of OD assignment in GA-operators. In other words, the classic fixed-point algorithms plus a random method is applied in GA inner and the GA operators in one upper level generate different combinations of OD assignments in order to improve the population. The steps of GA applied to traffic assignment are as follows:

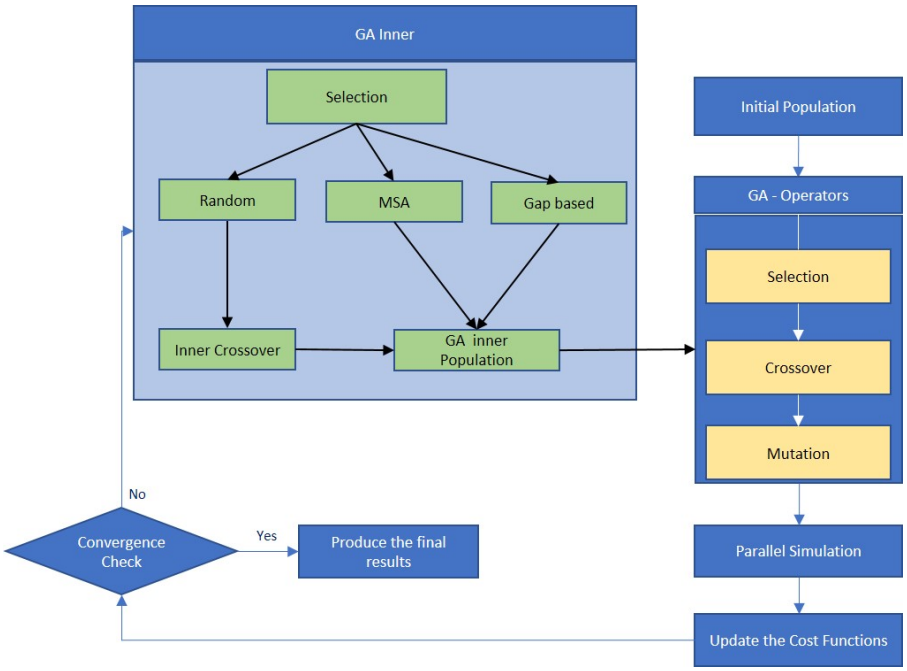


FIG. 3 – *GA solution algorithm flowchart*

- Selection: We use a random selection based on the crossover rate (Cr) and population size (PS) in order to compute the number of selected solution for crossover process:

$$SS = PS \times Cr \tag{1}$$

- Crossover: We apply a non-uniform crossover by using a bit-vector mask method (Maini *et al.*, 1994). We select two different solutions (parents) from the selected solutions set; we apply the crossover between each pair of solutions. As result, we will have new solutions; the two new solutions will have a part of each parent.
- Mutation: We apply the mutation operator for a set of selected solutions; by replacing one OD assignment (Chromosome) of the solution by another chromosome from another solution, the aim of this operator is to increase the quality of the worst solution.
- Parallel simulation: All new solutions obtained from previous steps are simulated in parallel in order to calculate the fitness function which is the $TGap$ in this study.
- Replacement: After applying the different GA operators and parallel simulation, the size of the evaluated population set is increased. In order to keep it as a fixed value (PS), we apply the selection operator and we use keeping best solutions as a replacement strategy.
- Convergence check: The algorithm converges when the maximum iteration is reached or the algorithm tends to a stagnation. In order to check the stagnation, we use an indicator Stagnation Factor (SF) when SF tends to zero, our process tends to stagnation, that means the quality of population solution is not changed. The Stagnation Factor is presented as follow:

$$SF = 1 - \frac{T\hat{G}ap_j}{T\hat{G}ap_{j-1}} \tag{2}$$

Where $T\hat{G}ap_j$ denotes the total gap of population of iteration j. If algorithm does not converge, the iteration index is increased and the inner layer of GA is applied in order to search the solution space by gen modifications inside the chromosomes of selected solutions.

- Ga Inner: The Initial Population, Selection, Crossover, Mutation, and Replacement are the basics of GA. In this study, we extend the GA algorithm, by adding a new operator. We called it GA Inner, the aim of this operator is to create the diversity in the current population (diversity inside the solution ODs). This method is applied by three different methods: the MSA, the Gap-based methods like SA and addaptive random method.
  - Random method: the foundation of this method is based on the genetic algorithm. We consider the selection and crossover operators for this method.
    - Inner Selection: Select a set of solutions (worst ones) from the main population. The aim of this selection is to increase the quality of population, by improving the solutions (especially worst ones).
    - Inner Crossover: We consider the OD assignment (Chromosome) as an individual and the flow of each path (Gen) as a chromosome in order to apply the crossover by the same way as in GA in previous layer of optimizer. By applying the crossover, we risk to have non-feasible OD assignment with respect to the demand constraint.To fix this problem, we use a following process to keep only the feasible solutions:
      - Step 1: Put zero for the flow of the worst path (wo) which has maximum travel time.
      - Step 2: Apply the crossover on the other paths of current OD.
      - Step 3: Compute R; the remaining flow of current OD pair:

$$R = \sum_{i \neq wo} X_i - D \tag{3}$$

        Where X_i denotes the flow of path i and D denotes the demand of current OD pair. If $R \leq 0$ then we put the rest of the flow on the worst path ($X_{wo} = R$) else we reject this chromosome.
    - The MSA assignment method and Gap-based method are also applied to the chromosomes of the selected solutions in GA inner.

- The new solutions are injected into the main population and the algorithm iterates while GA converges.

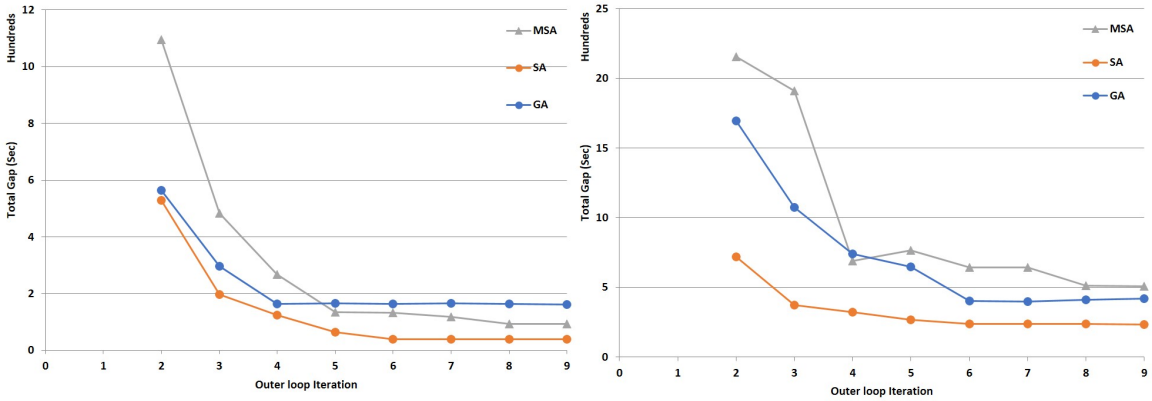
Numerical experiments

In the general iterative process of calculating the UE for trip-based DTA, there are two elements have to be determined: the path set between all origin-destination pairs and the optimal path flow distribution. The most advanced solution methods solved the problem with two-layers, the outer and the inner loop, that tackle each element respectively (Ameli *et al.*, 2019). The outer loop is finding the paths and the inner loop is solving the UE problem with fixed path set.

In order to compare the performance of the new methods, we use the same outer loop component with different methods in inner loop. This study evaluates the algorithms in order to compare the methods with MSA method and then apply the method to DTA problem for the large-scale test case. In this work, we use Symuvia[1] as a trip-based simulator for calculating the needed variables in the network. It is a microscopic simulator based on a Lagrangian discretization of the LWR model (Leclercq *et al.*, 2007). In the small-scale, the methods are applied to a 5×5 grid network. The primary results for 3 and 6 OD pairs with the given demand of 50 users per OD are presented in figure 4. All three methods converge very fast at the beginning of the process and in order to present three figures at the visible scale for comparison the total gap figures are presented from second outer loop. The results show that the SA method dominates the MSA method even in the small-scale and the results of GA

[1]Symuvia is an open source simulator that is available from winter 2018 (http://www.licit.ifsttar.fr).

are good for larger number of OD pairs when common links between different OD pairs are increased. We are currently running simulations on a large-scale network (Lyon 6e + Villeurbanne: 1,883 Nodes, 5,935 Links, 94 Origins, 227 Destinations, 54,190 trips) with dynamic trip-based implementation and the results are very promising.



(a) (b)

FIG. 4 – *Total gap in* 5×5 *grid network; (a): results for 3 ODs. (b): results for 6 ODs.*

Acknowledgement

This work is supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program. (Grant agreement No 646592 – MAGnUM project)

References

Ameli, M., Lebacque, J.-P., Delhoum, Y. & Leclercq, L. (2019). Simulation-based user equilibrium: improving the fixed point solution methods. *The 98th Transportation Research Board Annual Meeting (TRB)*.

Holland, J. H. (1992). *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press.

Kirkpatrick, S., Gelatt, C. D. & Vecchi, M. P. (1983). Optimization by simulated annealing. *science*, 220(4598):671–680.

Leclercq, L., Laval, J. A. & Chevallier, E. (2007). The lagrangian coordinates and what it means for first order traffic flow models. In *Transportation and Traffic Theory 2007. Papers Selected for Presentation at ISTTT17*.

Levin, M. W., Pool, M., Owens, T., Juri, N. R. & Waller, S. T. (2015). Improving the convergence of simulation-based dynamic traffic assignment methodologies. *Networks and Spatial Economics*, 15(3):655–676.

Lu, C.-C., Mahmassani, H. S. & Zhou, X. (2009). Equivalent gap function-based reformulation and solution algorithm for the dynamic user equilibrium problem. *Transportation Research Part B: Methodological*, 43(3):345–364.

Maini, H., Mehrotra, K., Mohan, C. K. & Ranka, S. (1994). Knowledge-based nonuniform crossover.

Robbins, H. & Monro, S. (1951). A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407.

Szeto, W. & Lo, H. K. (2005). Non-equilibrium dynamic traffic assignment. In *Transportation and Traffic Theory. Flow, Dynamics and Human Interaction. 16th International Symposium on Transportation and Traffic TheoryUniversity of Maryland, College Park.*

Talbi, E.-G. (2009). *Metaheuristics: from design to implementation*, vol. 74. John Wiley & Sons.

Wang, Y., Szeto, W., Han, K. & Friesz, T. L. (2018). Dynamic traffic assignment: A review of the methodological advances for environmentally sustainable road transportation applications. *Transportation Research Part B: Methodological*.

Wardrop, J. G. (1952). Road paper. some theoretical aspects of road traffic research. *Proceedings of the institution of civil engineers*, 1(3):325–362.