Route Assignment for Autonomous Vehicles

Nick Moran and Jordan Pollack

Brandeis University, Waltham MA 02453, USA,
nemtiax@brandeis.edu,
pollack@brandeis.edu,
http://www.demo.cs.brandeis.edu/

Abstract. We demonstrate a self-organizing, multi-agent system to generate approximate solutions to the route assignment problem for a large number of vehicles across many origins and destinations. Our algorithm produces a set of mixed strategies over the set of paths through the network, which are suitable for use by autonomous vehicles in the absence of centralized control or coordination. Our approach combines ideas from co-evolutionary dynamics in which many species coordinate and compete for efficient navigation, and ideas from swarm intelligence in which many simple agents self-organize into successful behavior using limited between-agent communication. Experiments demonstrate a marked improvement of both individual and total travel times as compared to greedy uncoordinated strategies, and we analyze the differences in outcomes for various routes as the simulation progresses.

Keywords: Swarm Intelligence, Vehicle Routing, Autonomous Vehicles, Multi-Agent Systems, Co-evolution, Coordination Games

1 Introduction

As autonomous vehicles become a significant portion of road traffic, the routing decisions made by those vehicles will have a strong impact on the congestion and efficiency of the road network. At present, it is acceptable for any autonomous or autonomously-routed vehicle to simply greedily select the most effecient route from its origin to its destination. However, once these vehicles represent the majority (or even a large minority) of traffic, a problem with this greedy approach arises: if all vehicles take the apparent shortest route, that route will quickly become overloaded and travel will slow to a crawl. Such a situation could be avoided by more intelligent coordination between the vehicles - if they were to spread themselves out across a variety of routes, congestion could be avoided and travel times improved. This problem mirrors the class of coordination games in which players must simultaneously select from a set of choices, where the utility of a choice decreases as more players choose it, such as the famous "El Farol Bar Problem"[1]. In these problems, the solution is for each player to utilize an equilibrium mixed strategy over the set of possible options. That is, each player assigns a probability distribution to the set of choices, and no player can

improve their expected utility by unilaterally deviating from that probability distribution.

The problem of vehicle route selection is further complicated by the fact that travel time depends not only on the route selection of other vehicles travelling from the same origin to the same destination, but is also affected by the choices made by vehicles travelling between other origin/destination pairs, even those which share neither the origin nor destination. Additionally, the domain of autonomous vehicles gives particular importance to the requirement that strategies be in equilibrium. If a particular traveller believed that a better route choice was available, he or she would simply direct the vehicle to use that route instead. This prevents arrangements in which a small subset of vehicles are assigned poor routes with the goal of alleviating congestion for the majority of vehicles. This concept of equilibrium is based on early work by Wardrop in which he defines "user equilibrium" on a travel network as a state in which every individual agent is acting in the way which minimizes its own travel time.[2]

In this paper, we present a multi-agent simulation from which a set of equilibrium mixed strategies for route selection may be gleaned. Our algorithm begins from the greedy assignment of vehicles to the shortest route, and uses a combination of exploration of alternative routes, and reinforcement of successful, faster routes, to arrive at a set of routing strategies which are more efficient, and from which no individual agent can profitably deviate.

1.1 Related Work

The study of route selection for vehicles and its effect on traffic congestion is not new. It has been studied since at least the 1950s to assist in the planning of extensions to the road network and the prediction of their effects on traffic flow. The problem has traditionally been modeled as optimization under constraints, in which a variety of simplifying assumptions are made about traffic dynamics to reduce the problem to an instance of convex non-linear programming which can be approximated through gradient descent methods such as the Frank-Wolfe Algorithm[3]. Further work has found an assortment of improvements[4] and alternative algorithms[5][6] within this paradigm.[7][8]

More recently, there have been a variety of attempts to apply both evolutionary algorithms and swarm intelligence[9] algorithms to various traffic problems, including signal control[10][11], network layout[12], and scheduling[13]. There have also been efforts to adopt ant colony optimization[14] approaches to both the signal control[15] and vehicle routing problems.

2 Algorithm

Our algorithm works by modeling the road network as a graph, a set of vehicles, and set of origin-destination pairs of nodes between which vehicles travel. The model is denoted as a tuple (I, L, R, V), where I is a set of nodes in the graph representing intersections, L is a set of directed edges in the graph, representing

road links between intersections, R is a set of ordered pairs of nodes (o, d), representing that vehicles travel from o to d, and V is a set of vehicles. Each vehicle is assigned to a particular element of $r \in R$, and has a path $P \subseteq L$ from R_o to R_d. The number of vehicles (load) travelling on a praticular route r is designated r_l. Each r_l remains constant for the duration of the simulation. For simplicity, the set of all vehicles travelling between a particular origin-destination pair is referred to as a *species*.

The algorithm proceeds by simulating the flow of these vehicles across the network. When a vehicle completes its assigned route, it informs another randomly selected vehicle of the same species of its success, and the route it took. The second vehicle will then adopt that strategy for its next trip, with a small chance of alteration through the addition of a detour. In this way, successful strategies spread throughout the population, and new strategies are explored through the occasional addition of detours.

The time it takes for a particular vehicle to pass through an edge is determined by a congestion equation, which calculates the delay due to other vehicles on the same edge. Each edge has a length which defines the base travel time, and a capacity, which defines the number of vehicles an edge can carry before congestion effects take over. In this work, we use a slight modification of the standard Bureau of Public Roads congestion equation[16]. The travel time on a link is determined by the equation

$$S = t * (1 + 0.15 * (v/c)^4)$$

Here S is the actual travel time on a link, t is the base free-flow travel time (that is, the travel time experienced with no other vehicles on the road), v is the current number of vehicles on the road, and c is the capacity of that particular road. This differs from the standard BPR equation in that v and c are actual vehicle counts, whereas the BPR equation deals with average rates of flow instead. The form and parameters of the equation remain unchanged.

Conceptually, this equation means that congestion has only a minor impact while $v < c$; the travel time at capacity is only 15% longer than at complete free-flow. On the other hand, once the number of vehicles surpasses the capacity, travel times quickly become exponentially longer.

A known difficulty with this equation is that the exponential form predicts unrealistic travel times in the case of severe congestion[17]. Although this does not present a problem for some applications in which these sorts of traffic loads do not arise, it threatens to trap simulated vehicles in an eternal traffic jam in our simulation. To avoid this, we model congestion with queueing once the traffic load passes a certain threshold (2.5 times the capacity of the link in our experiments). If a vehicle attempts to enter a link which is at the specified threshold, it instead enters the back of a queue for that link. Whenever a vehicle exits a link, if its queue is not empty, the vehicle at the front of the queue is removed and enters the link. This accomodation is necessary for our congestion model, but is not integral to the proposed algorithm. Different congestion models may obviate the need for queueing.

An overview of the core algorithm is given in Algorithm 1. Details about each component of the simulation can be found in the following sections.

Algorithm 1 A pseudo-code overview of the simulation algorithm.

```
For each r ∈ R, initialize r_l vehicles, each with a path leading from r_o to r_d.
Place each vehicle at the start of the first edge of its path. Compute the travel time along that edge.
Insert each vehicle into a priority queue keyed by their time of arrival at the end of their first edge.
loop
    Select the vehicle v from the priority queue which will arrive at a node first
    if The node it arrives at is it's destination then
        if v is marked for replacement then
            Remove v, and set its replacement v_r at the start of its path. Calculate v_r's travel time along its first edge, and add it to the queue.
        else
            Return v to the start of its path. Calculate its travel time along its first edge, and add it to the queue.
            Select a random vehicle, w travelling between the same origin-destination pair. Mark w for replacement, and set its replacement, w_r to be a possibly mutated copy of v.
        end if
    else
        Move the vehicle to the next edge on its path
        Calculate travel time along that edge
        Insert vehicle back into priority queue.
    end if
end loop
```

Note that this algorithm simulates all vehicles simultaneously. Although each individual step of the simulation updates only a single vehicle, those updates are interleaved such that every vehicle is progressing through the network. Unlike many evolutionary models in which the entire population is updated together in a series of discrete generations, vehicles in our formulation change and spread their strategies asynchronously.

2.1 Representation

Each vehicle stores its path as a simple list of nodes through which it will travel from its designated origin to its designated destination. For any particular vehicle, the origin and destination are fixed, but the path to be taken between them can be altered, either by copying the successful route of another vehicle, or by copying that route with the addition of a detour. Each vehicle also tracks whether its path has been marked for replacement, and, if so, the replacement path that it will use in the future. At any given time, a vehicle is travelling along a particular edge, and stores the time at which it will arrive at the end of that edge, as calculated by the congestion equation.

2.2 Spread of Strategies

When a vehicle reaches its designated destination, it has the potential to either spread it's current path to another vehicle of its species (that is, a vehicle with

the same origin and destination), or to replace its own path with one it has received from another vehicle. Upon reaching its destination, a vehicle returns to its starting node, and checks whether it has received a replacement path. If so, it adopts that path, and follows it through the network. If that vehicle has not received a replacement path, then it instead communicates its current path to another vehicle of the same species.

The exact procedure for communicating the path of a successful vehicle is as follows. Let v be the vehicle that has just successfully completed its route. Randomly select another vehicle w of the same species as v. Mark w as having received a replacement path, and set its replacement path to the path of v. With a small random chance (in our experiments, a chance of 5% was used), modify the w's new replacement path with a detour, as explained in the next section. Note that it may be the case that w was already marked for replacement. In this case, its previous replacement route is simply overwritten with the new one it receives from v. Also note that w will continue on its current route, and will not actually adopt the new replacement route until it reaches its destination and begins a new journey.

The motivation for allowing w to complete its current journey, rather than simply immediately returning it to its origin to begin its new path, is to ensure that traffic remains consistently distributed across the network. If vehicles chosen for path replacement returned immediately to the start, then vehicles would tend to be found near their origins, and only a few vehicles would be found near their destinations, leading to an unrealistic distribution of congestion effects.

It is also important to note that a vehicle's performance has no bearing on its chances of being selected to receive a replacement path. Vehicles which are making efficient trips from their origin to their destination are equally likely to receive a replacement path as vehicles which are stuck on circuitous, congested, or otherwise inefficient routes.

The result of this system of spreading successful strategies is that the average rate at which a particular path increases its share of the population is dependant on the speed at which vehicles following that path complete their route as compared to the average speed across all vehicles of that species. A path which allows vehicles to complete their journey faster than this average speed will tend to increase its share of the population. As its share of the population increases, congestion effects will slow it, until eventually vehicles travelling along it are no longer faster than the average, and its share of the population levels off.

Eventually, all paths used by a particular species will take roughly the same amount of time. In this case, each will spread at the same rate, and population will be stable (the random nature of path replacement and detours will of course allow some amount of variation to remain). Thus, the population will have found a set of equilibrium strategies for navigating the network - no vehicle could improve its time by switching, because the feasible paths all take approximately equal time.

It is also important, however, to note that an equilibrium state does not depend only on the behavior of vehicles within a particular species. The congestion

along a path also depends on how heavily members of other species make use of the links along that path, and the efficiency of a particular route will change as other species move towards their own equilibria. Ultimately, an equilibrium assignment of traffic must balance the concerns of these many interacting species.

2.3 Detours

It is not sufficient to only increase the frequency of efficient paths within the population. We also must explore new routes that are not currently in use. It may be that one of those is more efficient than any route being currently taken. To accomplish this exploration, we apply a detour (with a small random chance) to paths that are passed from one vehicle to another.

A detour is formed by randomly selecting two nodes from the current path to serve as the start and end of the detour, call them D_s and D_e. It must be the case that D_s occurs before D_e in the path. Next, a random node is selected from the set of all nodes in the graph to serve as the midpoint of the detour, call this node D_m. We then replace the section of the original path from D_s to D_e with the concatenation of the shortest (in distance, not apparent travel time) route from D_s to D_m followed by the shortest route from D_m to D_e. This shortest route may be computed by any standard path-finding algorithm, such as A*.[18]

Despite the name, the detour operator does not only create more and more circuitous routes. Because the paths between the start, midpoint and end of the detour are the shortest paths available, a detour operator can also make a path more direct. If D_m lies on the shortest path from D_s to D_e (or even a shorter path than is currently being taken), then the effect of this detour will actually be to simplify the route.

In some cases, the selection of the detour nodes may result in a path with a loop. For example, if D_m or another node on the $D_s \rightarrow D_m$ or $D_m \rightarrow D_e$ paths is already in the original path, the detour will create a path which passes through that node twice. Such a route is clearly inferior to the same route without the loop, and is therefore not worth exploring. To avoid this, we perform a process of loop removal before finalizing a detour. Loop removal is acheived by passing over the new path, checking for a repeated node. If one is found, the section of path between the two occurances of that node is deleted. This process is repeated until no repeated nodes remain.

2.4 Initialization

It's necessary to select a starting set of routes for the vehicles in the simulation. In our experiments, we have each vehicle begin by taking the shortest path from its origin to its destination. However, other starting configurations are also admissible. For example, initial routes may be randomized via the mutation operator, or, if available, may be determined by data about current travel patterns in the situation to be simulated.

In any case, the simulation begins with a winding-up period in which no mutation and reproduction occurs - vehicles simply travel along their paths,

and return to the start once complete. This allows the initial traffic jam caused by starting all vehicles simultaneously to disperse, and spreads link usage out across the network. The exact duration of this winding-up period will necessarily depend on the network being simulated, but an appropriate length can be determined by waiting for the times taken across a particular route to stabilize. In our experiments, we use a preset winding-up length which is more than long enough to stabilize travel times.

At the end of this initiailization period, evolution of strategies can begin, with spread and replacement of paths of occuring as normal.

2.5 Termination

Just as it is necessary to allow traffic patterns to stabilize at the beginning of the simulation, it is also desirable to allow them to stablize at the end. Once sufficient evolution has occurred (as determined by a lack of further improvement across the simulation), evolution is stopped to allow the new traffic flow to reach equilibrium.

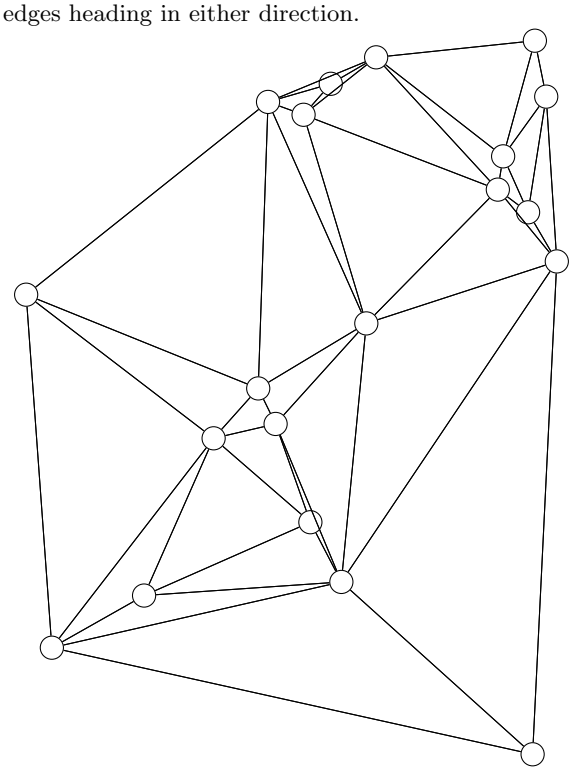
After the simulation ends, the routes in use can be examined to read off a mixed strategy for each origin-destination pair. Given an origin-destination pair, take the set of all paths from the origin to the destination which are in use by any vehicle at the end of the simulation. The mixed strategy for navigating from that origin to that destination is then determined by the relative frequency of each path in use. For example, if, for a particular origin-destination pair, 30 vehicles follow path A, and 20 vehicles follow path B, then the mixed strategy is 60% path A and 40% path B.

3 Experiments

This algorithm was tested on a randomly-generated network consisting of 20 nodes and 98 edges, which was embedded in 2D space. The layout of the experiment network can be seen in figure 1. The base free-flow travel times for links were determined by the distance between the start and end of that link. The capacity of a link is proportional to its length. The origin-destination pairs used were all pairs of non-adjacent nodes on the convex hull of the network, giving 40 different origin-destination pairs. Adjacent nodes were omitted because they tend to lack feasible alternative routes, and therefore are not subject to meaningful evolution. In real-world applications, travel between adjacent nodes may be better modeled as a static feature of the network (i.e. by adjusting the base vehicle count of the link in question).

The number of vehicles placed on each origin-destination pair was equal to the total capacity of the shortest path between those nodes. Although this would allow for relatively uncongested traffic flow if there were only a single origin-destination pair, many pairs have overlapping shortest routes, leading to significant congestion in need of relief. In total, over 1,100 vehicles were simultaneously simulated.

Fig. 1. The network layout used in experiments. Each edge in the image represents a pair of directed edges heading in either direction.



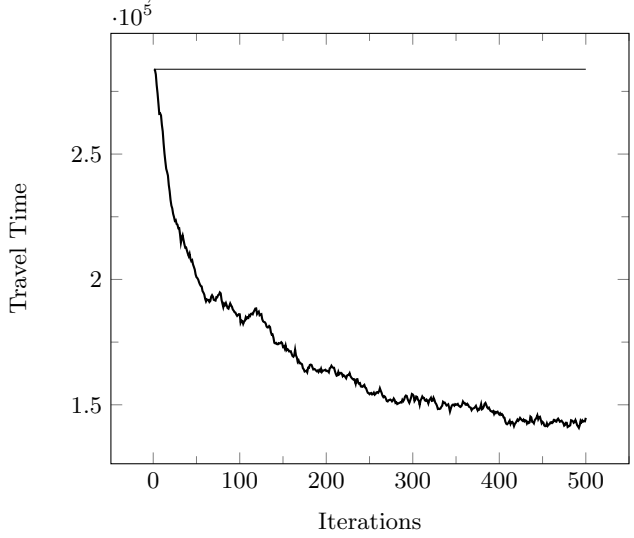
Each experiment ran for five million iterations (that is, five million simulated edge traversals), with an additional five hundred thousand iterations each of initilization and termination periods to ensure stable starting and ending conditions. Data points about the current travel times in the network were collected every ten thousand iterations. Travel times were calculated based on the length of each vehicle's last completed trip.

We conducted ten experiments in which we measured the total average travel time experienced across the network, and the average travel times experienced for each individual origin-destination pair. Additionally, we examined the average time of each individual path between a single origin-destination pair to determine whether travel times were approximately equal, which would be the expected outcome if the simulation had reached equilibrium.

4 Results

Over the course of the simulation, the average total travel time fell from roughly 310,000 simulation ticks to roughly 158,000 ticks, a reduction of nearly half, as seen in Figure 2. A steep initial improvement occurs as the most severely

Fig. 2. Total travel time across all routes as the simulation progresses. The horizontal line represents the baseline travel time in which vehicles always take the shortest distance path. The vertical axis shows units of simulation ticks, while the horizontal axis shows simulation iterations, in tens of thousands.

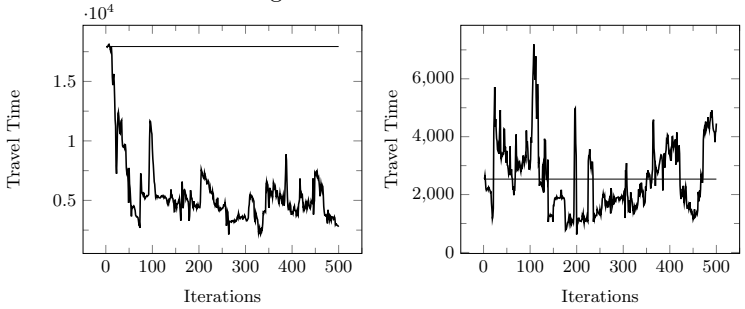


congested routes clear out, followed by a period of slowing, somewhat unsteady improvement as vehicles settle into efficient equilibria.

However, not all origin-destination pairs experience the same smooth reduction in travel time as seen in the overall average. Figure 3 shows two representative graphs which demonstrate the different types of behavior an origin-destination pair can display. On the left we have the data for vehicles travelling from Node 12, in the upper left corner of the network, to Node 4, near the lower left. Here we see the same steep initial improvement as in the overall average, followed by slower improvement marked by significant spikes in travel time. This occurs when the vehicles on other origin-destination pairs make an adjustment to their route choices such that more vehicles occupy the links being used to travel from 12 to 4. Once the congestion becomes severe, alternative routes are found to once again lower travel times.

On the right side of Figure 3, we have the data for vehicles travelling from Node 4 to Node 3, both in the lower left corner of the network, separated by only two links. Here, instead of a sharp initial improvement, we see a brief, small improvement followed by a severe spike. Fluctuations occur throughout the rest of the simulation, but the travel time ultimately ends up worse than how it started. This is because the links used by vehicles travelling from Node 4 to Node 3 were not used by many other species at the start of the simulation. However, as those other species explored alternative routes through the network, these little-used links were discovered and exploited. The 4-to-3 species was unable to ultimately improve its travel time because few viable alternative routes exist.

Fig. 3. Average travel time between a single origin-destination pair, from Node 12 to Node 4 on the left and from Node 4 to Node 3 on the right. The horizontal lines represent the baseline travel time in which vehicles always take the shortest distance path and do not alter their strategies.



Thus we see that an overall improvement in travel time will not necessarily correspond to uniform improvement across all origin-destination pairs.

As an example of the routes generated for a particular origin-destination pair, consider the set of routes from node 3 to node 15 in the experiment network seen in Figure 4.

Fig. 4. Routes from node 3 to node 15 at the end of an experiment run, showing the relative frequency of each route and the time taken in simulation ticks.

$3 \Rightarrow 14 \Rightarrow 8 \Rightarrow 17 \Rightarrow 1 \Rightarrow 11 \Rightarrow 15$	39.4%	6025 ticks
$3 \Rightarrow 14 \Rightarrow 8 \Rightarrow 5 \Rightarrow 1 \Rightarrow 10 \Rightarrow 15$	36.4%	6266 ticks
$3 \Rightarrow 13 \Rightarrow 4 \Rightarrow 11 \Rightarrow 15$	18.2%	6205 ticks
$3 \Rightarrow 0 \Rightarrow 8 \Rightarrow 1 \Rightarrow 10 \Rightarrow 6 \Rightarrow 12 \Rightarrow 15$	6.1%	7320 ticks

The first two routes take similar paths, differing only in two respects - their route from node 8 to node 1, and their route from node 1 to to node 15. Although a direct edge exists from node 8 to 1, it is often the location of significant congestion due to its central location, and so these strategies make use of detours to avoid it. The third strategy takes an entirely different route around the edge of the network, but still arrives in a similar amount of time to the others. Only the fourth strategy, which has a very low share of the population, lags behind, taking significantly longer due to its use of congested links and a circuitous detour to node 12 near the end. This strategy likely represents a relatively recent mutation which has not yet been competed out of the population.

The fact that the travel times are roughly equal (aside from the small outlier) is to be expected - this represents an equilibrium state in which all three strategies will reproduce at approximately the same rate. The equality is only approximate, however, because of the discrete nature of the simulation and on-

going changes in the co-evolutionary fitness landscape up until the end of the simulation.

5 Future Work

Our algorithm offers several directions for potentially fruitful future work. In this paper we use a standard model for congestion, the BPR equation. This model is ultimately a simplification of real traffic behavior, made in order to allow efficient computation and provide mathematically convenient properties (in particular, a nicely concave shape for gradient descent approaches as well as full independence of delays along different links). Our algorithm does not rely on these properties of congestion modelling, and any desired model could be used instead, including those which account for interdependence between the congestion on links (such as models which account for the delay at intersections caused by cross-traffic[19], or cascading congestion across multiple links[20]). Further work is needed, however, to determine the performance of our algorithm under these more detailed congestion models.

Our experiments in this paper demonstrate the effectiveness on a moderately-sized, randomly-generated network, but more work is needed to study the behavior of this approach on a real-life road network. It may be necessary to alter the method of generating detours when working on a very large-scale network - perhaps by favoring local detours over distant ones.

6 Conclusion

We have demonstrated an algorithm by which large number of autonomous agents can arrive at a set of equilibrium strategies for navigating a network. Our algorithm significantly outperforms a baseline greedy approach in which each agent independently selects its apparent fastest route. Further work is needed to adapt this approach to real-world road networks and vehicle behavior.

References

1. Arthur W.B.: Inductive Reasoning and Bounded Rationality. The American Economic Review 84: 2 (1994)
2. Wardrop J.G., Whitehead J.I.: Correspondence. Some theoretical aspects of road traffic research. Proceedings of the ICE 1:5 (1952)
3. Frank, M., Wolfe, P.: An algorithm for quadratic programming. Naval Research Logistics Quarterly 3: 95 (1956)
4. Fukushima, M.: A modified Frank-Wolfe algorithm for solving the traffic assignment problem. Transportation Research Part B: Methological 18: 2 (1984)
5. Dafermos, S. C., Sparrow, F. T.: The traffic assignment problem for a general network. Journal of research of the National Bureau of Standards. B, Mathematical sciences. 73B (1969)

6. Larsson T., Patriksson M.: Simplical decomposition with disaggreagted representation for the traffic assignment problem. Transportation Science 26 (1994)
7. Jayakrishnan R., Tsai Wei T., Prashker J.N., Rajadhyaksha S.: A Faster Path-Based Algorithm for Traffic Assignment. University of California Transportation Center. UC Berkeley: University of California Transportation Center. (1994)
8. Bar-Gera H.: Origin-based algorithm for the traffic assignment problem. Transportation Science 36: 4 (2002)
9. Beni, G., Wang, J.: Swarm Intelligence in Cellular Robotic Systems, Proceed. NATO Advanced Workshop on Robots and Biological Systems, Tuscany, Italy (1989)
10. Turky, A. M., Ahmad M.S., Yusoff M.Z.M., Sabar N.R.: Genetic Algorithm Application for Traffic Light Control. UNISCON (2009), LNBIP 20, pp 115-120, (2009)
11. Medina, J.J.S., Moreno, M.J.G., Royo, E. R.: Evolutionary Computation Applied to Urban Traffic Optimization, Advances in Evolutionary Algorithms (2008)
12. Schweitzer F., Ebeling W., Rosé H., Weiss O.: Optimization of road networks using evolutionary strategies. Evolutionary Computation 5: 4 (1997)
13. Teodorovic D., Lucic P.: Schedule synchronization in public transit by fuzzy ant system. Transportation Planning and Technology 28, (2005)
14. Dorigo, M.: Optimization, learning and natural algorithms. PhD thesis, Dipartimento di Elettronica, Politecnico di Milano, Italy (1992)
15. D'Acierno L., Gallo M. Montella B.: An Ant Colony Optimisation algorithm for solving the asymmetric traffic assignment problem. European Journal of Operational Research 217:2 (2012)
16. US Department of Commerce Bureau of Public Roads: Traffic Assignment Manual (1964)
17. Spiess H.: Conical volume-delay functions. Transpotation Science 24: 2 (1990)
18. Hart, P. E.; Nilsson, N. J.; Raphael, B. "A Formal Basis for the Heuristic Determination of Minimum Cost Paths". IEEE Transactions on Systems Science and Cybernetics SSC4 4 (2) (1968)
19. Jeihani, M., Lawe, S., Connolly JP.: Improving Traffic Assignment Model Using Intersection Delay Function. 47th Transpotation Research Annual Forum (2006)
20. Ji, Y., Geroliminis N.: Modelling Congestion Propagation In Urban Transportation Networks. Swiss Transport Research Conference (2012)