# Ant Colony Optimization applied to Route Planning using Link Travel Time predictions

Rutger Claes
*DistriNet Labs*
*Katholieke Universiteit Leuven*
*Leuven, Belgium*
*rutger.claes@cs.kuleuven.be*

Tom Holvoet
*DistriNet Labs*
*Katholieke Universiteit Leuven*
*Leuven, Belgium*
*tom.holvoet@cs.kuleuven.be*

***Abstract*—Finding the shortest path in a road network is a well known problem. Various proven static algorithms such as Dijkstra and A* are extensively evaluated and implemented. When confronted with dynamic costs, such as link travel time predictions, alternative route planning algorithms have to be applied. This paper applies Ant Colony Optimization combined with link travel time predictions to find routes that reduce the time spend by travels by taking into account link travel time predictions. The presented algorithm is evaluated using simulations in real world traffic networks.**

***Keywords*-Mobile agent, Ant Colony Optimization, Traffic control**

## I. Introduction

Techniques such as online simulation or machine learning can be used to provide vehicles with link travel time predictions. These predictions forecast network link traversal times based on current traffic situations, historic data and real-time information provided by vehicles.

Using this forecast data to calculate shortest routes poses several difficulties. The first difficulty is the dynamic nature of the link travel time predictions. As these values are only predictions, they can change over time as new information becomes available. The link travel time predictions $ltt_{ij}\left(t_1\right)$ describing the traversal time of the link between nodes $i$ and $j$ at time $t_1$ calculated at time $t_2$ can be different from the $ltt_{ij}\left(t_1\right)$-value calculated at time $t_3$.

A second difficulty is the increase in search space. Because the link travel time predictions describing a link $(i, j)$ are a function of time, the search space becomes an order of magnitude bigger.

Finally, the third difficulty is the distributed nature of link travel time predictions. Many of the mechanisms used to process link travel time predictions result in the predictions being distributed across the network, making it impossible for vehicles to use all forecast data locally to calculate a route. The problem of making the data available to the vehicle is made worse by the dynamic nature of the forecast data.

The paper describes an Ant Colony Optimization (ACO) based algorithm that can plan routes taking into account the dynamic and distributed link travel time predictions. Ant Colony Optimization is inspired by the foraging behavior of ants and their use of pheromones to guide other colony members to food sources. In ACO, a number of virtual ants constructs solutions to a certain problem. The ants do so by exploring a graph-like representation of the problem domain and by exchanging information about their individually found solution with other ants. ACO has been applied in a variety of problem domains such as graph coloring, protein folding and the traveling salesman problem. Bonabeau in [1] predicts that ACO, in general, is a good candidate solution to dynamic problems, and thus in our approach we use ACO to tackle the problem of calculating a route based on dynamic $ltt_{ij}$ values.

Algorithms based on Ant Colony Optimization are usually centralized algorithms, i.e. they assume all information is known locally on one device. Ant Colony Optimization based algorithms can be modified to work in a distributed search environment. The graph representing the problem domain is then distributed across a number of devices connect by a network. When constructing solutions, ants no longer just traverse a graph, they now also travel between devices. As the ants travel between devices, the carry partial information gathered from previously visited graph locations on previously visited devices. The ants thus construct coherent solutions based on the information that is scattered across a network of devices.

It is precisely this characteristic of Ant Colony based algorithms that make it a suitable candidate for the route planning using link travel time predictions problem. The virtual ants search a distributed graph of link travel time predictions for fast routes from one location to another. The graph describing the travel network can be distributed across a distributed infrastructure that is embedded within the actual road network. This allows techniques such as embedded online simulation to be used to generate the link travel time predictions.

Earlier work in [2], [3] already describes the use of ants as a mechanism to harvest the information scattered across a number of road side computers. These approaches, however,

1530-2075/11 $26.00 © 2011 IEEE
DOI 10.1109/IPDPS.2011.173

are not true to the Ant Colony Optimization meta heuristic as described by Dorigo et al. in [4]. Instead of constructing solutions, the ants used in these previous approaches only evaluate solutions found using traditional search techniques such as $A^*$ on local data. The approach taken in this paper goes beyond the evaluation of solutions constructed using static data and uses an ACO based algorithm to construct solutions directly from the distributed data.

The algorithm presented in this paper is based on two variants of Ant Colony Optimization: the original Ant System algorithm described in [5] and an improvement on this original variant called Ant Colony System (ACS) described in [6]. Applying ACO in a distributed setting and using it to construct routes in large scale graphs requires some modifications. Elements from both ACS and AS are combined together with novel techniques to adapt the algorithm to traffic route planning. The resulting algorithm is described in Section II. The differences between the algorithm described in this paper and ACS and AS are also described in that section.

ACO algorithms such as Ant Colony System are influenced by a large number of parameters. Together, these parameters define the balance between exploration and exploitation of the algorithm. The higher the exploration, the more new routes the algorithm will discover that are unrelated to the ones already known. The higher the exploitation of the algorithm, the more it will try to improve the results it knows, without looking for routes that are significantly different. Because of the importance of this balance in the algorithm presented here, it is discussed in Section III.

A number of experiments where conducted to evaluate the presented approach. The experiments all use real-world traffic networks. The results obtained in these experiments are outlined in Section IV. Other approaches to the route planning problem in distributed and dynamic environments using heuristic approaches are described in Section V. Finally, a conclusion is presented in SectionVI.

## II. ACO FOR ROUTE PLANNING

This section describes an Ant Colony Optimization algorithm capable of planning a route in a large scale urban traffic network. The algorithm described here draws inspiration from Ant System and Ant Colony System, two ACO variants. First the graph representation of the traffic network is briefly explained. Then, the details of the approach are outlined together with a description of the algorithm.

The traffic network is described as a unidirectional graph in which edges $(i, j)$ connect vertices $i$ and $j$. All edges represent existing roads, all vertices describe existing junctions. Edges have a length $l_{ij}$ and a speed $v_{ij}$. The length $l_{ij}$ corresponds with the actual length of the road and is always larger than the euclidean distance between $i$ and $j$, noted as $\|i, j\|$. At every vertex $i$, the set of outgoing edges is denoted as $O(i)$.

The link travel time, or $ltt$, is the time a vehicle needs to traverse a link in the traffic network. Different traffic links will have different characteristics and thus different values for $ltt$. The link travel time for the road represented by edge $(i, j)$ is noted as $ltt_{ij}$. Because traffic densities vary over time, so do link travel times. The link travel time for an edge $(i, j)$ thus becomes a function over time: $ltt_{ij}(t)$ where $t$ is the moment the vehicle enters the link. For values of $t$ that are ahead of the current time, estimates for $ltt_{ij}(t)$ are predicted. Such a prediction is noted as $\overline{ltt_{ij}}(t)$.

### A. Exploration of the traffic graph

The virtual ants that explore the traffic graph in order to construct routes between an origin and a destination are called *exploration ants*. This is to distinguish them from a second kind of ant introduced in the subsequent section.

Exploration ants explore the traffic graph. They can travel between vertices only if an edge connects those vertices. If the two vertices are located on separate devices, traveling between the vertices also implies traveling between the devices. This algorithm assumes that, as exploration ants traverse an edge $(i, j)$, they can request predictions $\overline{ltt_{ij}}(t)$ for any value of $t$ that interests them.

Exploration ants start out at the origin of the route. The ants construct a route from that origin to their vehicles' destination by simulating a vehicle driving over the traffic network. Exploration ants keep track of a virtual time as they simulate the vehicle driving over the network. Starting at the origin vertex $o$, the exploration ant chooses an edge $(o, i) \in N(o)$ and traverses it. As the ant traverses $(o, i)$, it requests $\overline{ltt_{oi}}(t_v)$ where $t_v$ is the current virtual time of the exploration ant. The exploration ant now assumes that it reaches vertex $i$ at $t_v + \overline{ltt_{oi}}(t_v)$ and updates its virtual time accordingly. This exploration is repeated until the exploration ant reaches the destination vertex $d$. At that point, the value $t_v$ that was kept by the ant represents an estimate arrival time at the junction represented by $d$ if a vehicle were to choose the route constructed by the exploration ant. This process is described in the algorithm below.

```
1: $t_v \leftarrow current\_time$
2: $route \leftarrow []$
3: $l \leftarrow o$
4: while $l \neq d$ and $length\,(route) < MAX\_HOPS$ do
5:     $(l, n) \leftarrow choose\_edge\,(N(l))$
6:     $t_v \leftarrow t_v + \overline{ltt_{ln}}(t_v)$
7:     $\tau_{ln} \leftarrow (1 - \varphi)\tau_{ln} + \varphi \cdot \tau_0$
8:     $route\,[] \leftarrow (l, n)$
9:     $l \leftarrow n$
10: end while
```

All edges $(i, j)$ in the graph have pheromone level $\tau_{ij}$. The pheromone levels for all edges are initialized at $\tau_0$. In the approach taken in this paper, the pheromone levels are limited between $\tau_{min}$ and $\tau_{max}$, as in the $\mathcal{MAX}$–$\mathcal{MIN}$ Ant System described in [7].

As ants traverse an edge $(i, j)$, they reduce the local pheromone level $\tau_{ij}$ by a factor $\varphi$, in a way that is similar to *local pheromone update* in Ant Colony System. This pheromone decrease is to encourage other ants to choose different edges, thus improving the exploration factor of the search. The constant $MAX\_HOPS$ limits the ants movements. The function $choose\_edge(S)$ uses a modified version of the stochastic method used in Ant System. $choose\_edge(S)$ selects an edge $(i, j) \in S$, where every edge has a probability $p_{ij}$ of being selected given by:

$$p_{ij} = \frac{(1-\gamma)\,\tau_{ij}^{\alpha} \cdot \gamma\,\eta_{ij}^{\beta}}{\sum_{(i,n)\in S} (1-\gamma)\,\tau_{in}^{\alpha} \cdot \gamma\,\eta_{in}^{\beta}}. \tag{1}$$

The parameters $\alpha$, $\beta$ and $\gamma$ together determine the relative importance of the pheromone level versus a heuristic $\eta_{ij}$ given by:

$$\eta_{ij} = \frac{\|i\,d\|}{\|i\,j\| + \|j\,d\|}, \tag{2}$$

where $d$ is the destination to which the route has to lead.

Only when an exploration ant reaches the destination $d$ before it reaches $MAX\_HOPS$ does it backtrack to increase the pheromone levels on the edges in $route$.

```
Require: $route$
Require: $l = d$
1: while $l \neq o$ do
2:     $(n, l) \leftarrow pop\,(route)$
3:     $\tau_{nl} \leftarrow [\tau_{nl} + \Delta\tau]_{\tau_{min}}^{\tau_{max}}$
4:     $l \leftarrow n$
5: end while
```

The $[\tau_{nl} + \Delta\tau]_{\tau_{min}}^{\tau_{max}}$ notation in the previous algorithm means that the result will be constrained by $\tau_{min}$ and $\tau_{max}$. $\Delta\tau$ is calculated as follows:

$$\Delta\tau = \theta \frac{tts_{min}}{tts_{route}},$$

where $tts_{min}$ is the minimal travel time spend by a vehicle to get from origin $o$ to destination $d$ given by $\|o\,d\|/v_{max}$ with $v_{max}$ the highest allowed speed in the traffic network and where $tts_{route}$ is the travel time spend by a vehicle that would follow $route$, as constructed by the exploration ant. $tts_{route}$ is the difference between the current time and the virtual time $t_v$ of the vehicle when it arrives at $d$. The factor $\theta$ is a weighing factor.

In the pheromone increase step, our approach differs from both the Ant System and Ant Colony System approach. Both AS and ACS work in well defined iterations. At the end of each iteration the pheromone levels $\tau_{ij}$ are updated by decreasing them with a parameter $\rho$ and increasing them based on the quality of the solution found in that iteration. For Ant System this is means that

$$\tau_{ij} \leftarrow (1-\rho)\cdot\tau_{ij} + \sum_{k=1}^{m} \Delta\tau_{ij}^{k},$$

meaning a relative reduction by factor $\rho$ and an increase corresponding with all the $\Delta\tau_{ij}$-values for all ants in that iteration. Ant Colony System uses a different approach and only updates edges that are a part of the best solution:

$$\tau_{ij} = \begin{cases} (1-\rho)\cdot\tau_{ij} + \rho\cdot\Delta\tau_{ij} & \text{if } (i,j) \text{ belongs to best tour} \\ \tau_{ij} & \text{otherwise} \end{cases}$$

The algorithm presented in this paper does not decrease pheromone levels in one global update, as is done in both AS and ACS. Decreasing pheromone levels is only done locally by passing ants. The algorithm presented here does not require global pheromone updates at all. Structuring ant explorations in iterations becomes harder when confronted with a distributed setting. Having to globally update all pheromone levels would require synchronization and more communication.

## B. Preparation of the traffic graph

Ant System, Ant Colony System and most Ant Colony Optimization inspired techniques assume that ants will construct solutions easily. As soon as one solution is found, a pheromone trail is formed and other ants will start constructing variations on this solution. In a traffic graph, the assumption that ants will quickly find a solution doesn't hold. Even with the inclusion of the heuristic $\eta$ in the $choose\_edge()$-function, the chance of ants getting caught in loops or dead-ends is too big.

This problem is solved by *priming* the traffic graph. Based on static data known about the links in the network, i.e. the length $l_{ij}$ and max speed $v_{ij}$, a solution, $route_{stat}$, is calculated using $A^{*}$. The solution thus obtained would be optimal if $ltt_{ij}(t)$ was always equal to $l_{ij}/v_{ij}$.

A *primer ant* is send out over this $route_{stat}$ before exploration ants are send out. The primer ant will locally increase the pheromone level of all edges in $route_{stat}$. This pheromone increase will guide the exploration ants to the statically optimal solution so they can start to construct variations to this solution that take into account the dynamic nature of the $ltt_{ij}(t)$ values. The following algorithm describes the movement of the primer ants:

```
Require: $route_{stat}$
1: while $l \neq d$ do
2:     $(l, n) \leftarrow shift\,(route_{stat})$
3:     $\tau_{ln} \leftarrow \tau_p$
4:     $l \leftarrow n$
5: end while
```

where $\tau_p$ determines how high the pheromone increase is. In the experiments presented in this paper, the value for $\tau_p$ is chosen at $\tau_{max}$, the maximum allowed pheromone level.

As the exploration ants use the pheromone trail laid by the primer ants, parameter $\varphi$ causes the pheromone levels to decrease and converge back to their original $\tau_0$-values. By then, interesting variations on $route_{stat}$ will have higher pheromone levels.

## III. Exploration versus Exploitation

While searching the traffic graph for valid routes between the origin and the destination, exploration ants have to balance the desire to improve upon the current known solutions and the desire to find new and unrelated solutions that could be better than the currently known routes. Getting this balance right is crucial to the success of the Ant Colony Optimization method applied to route planning. If ants only exploit the current known solutions, they will never divert from $route_{stat}$. If ants focus too much on exploration, they will disregard information learned by their predecessors and the chances of them stumbling on a good solution are minimal.

A number of parameters determine this balance. This section analyzes the parameters $\alpha$, $\beta$, $\gamma$, $\theta$ and $\varphi$ used in the algorithms presented in the previous section. The use of probabilities to choose the edge in the $choose\_edge()$ function ensures the explorative nature of ants. As long as $\tau_{min}$, the minimal pheromone level, is non-zero the possibility of an ant choosing a previously unknown edge exists.

*Power of the pheromone ($\alpha$):* The parameter $\alpha$ will determine the chance of an edge to be chosen dependent on its pheromone level compared to the other alternatives. The larger the value of $\alpha$, the more the $choose\_edge()$ function is likely to return the edge with the largest $\tau$ value. As such, increasing the value of $\alpha$ increases the exploitative nature of the ants, as they will strongly favor edges that have been traversed before and resulted in good solutions. Decreasing the value of $\alpha$ will increase the explorative nature of the ants. In the extreme case of $\alpha = 0$, the exploration ants will completely disregard $\tau$ values.

*Power of the heuristic ($\beta$):* Parameter $\beta$ is very similar to the $\alpha$ parameter. It determines the importance exploration ants will give to the relative difference in heuristic values when deciding on which edge to traverse. The larger the value of $\beta$, the more exploration ants will favor edges with good $\eta$ values. Increasing the value of $\beta$ thus increases the exploitative nature of the exploration ants.

*Importance of the heuristic ($\gamma$):* The values of the heuristic $\eta$ in (1) will all fall between 0 and 1 because of the heuristic used in equation (2). The pheromone levels $\tau$ in equation (1) will lie between $\tau_{min}$ and $\tau_{max}$ because of the pheromone restrictions imposed by our algorithm. The $\theta$-parameter is used to determine the relative importance of $\tau^{\alpha}$ and $\eta^{\beta}$ in equation (1). The value of $\theta$ does not influence the balance between exploitation and evaporation directly. In exploration ants with very exploitative behavior, large values of $\theta$ will cause the ant to exploit routes with good $\eta$ values.

*Pheromone increase factor ($\theta$):* This parameter determines how much the pheromone levels $\tau$ are increased when exploration ants backtrack over the solution they have created. The larger the $\theta$ value, the more prominent already constructed solutions will be imprinted into the network, and the more likely future exploration ants will take into account existing solutions. Increasing the $\theta$ parameter makes the exploration ant do more exploitation.

*Pheromone reduction factor ($\varphi$):* As the value of $\varphi$ is higher, pheromone trails representing previously constructed solutions will disappear from the graph more quickly. Thus, a higher value of $\varphi$ will increase the explorative nature of the exploration ants. The reduction of the pheromone level $\tau$, when an ant traverses a link makes the link less likely for future ants. This leads future exploration ants to construct routes that are totally different from previously examined ones.

The pheromone reduction also filters out edges that lead to dead-ends. Ants starting on such an edge will reduce the edges pheromone value, but because the ants will never backtrack, the pheromone level will not be increased.

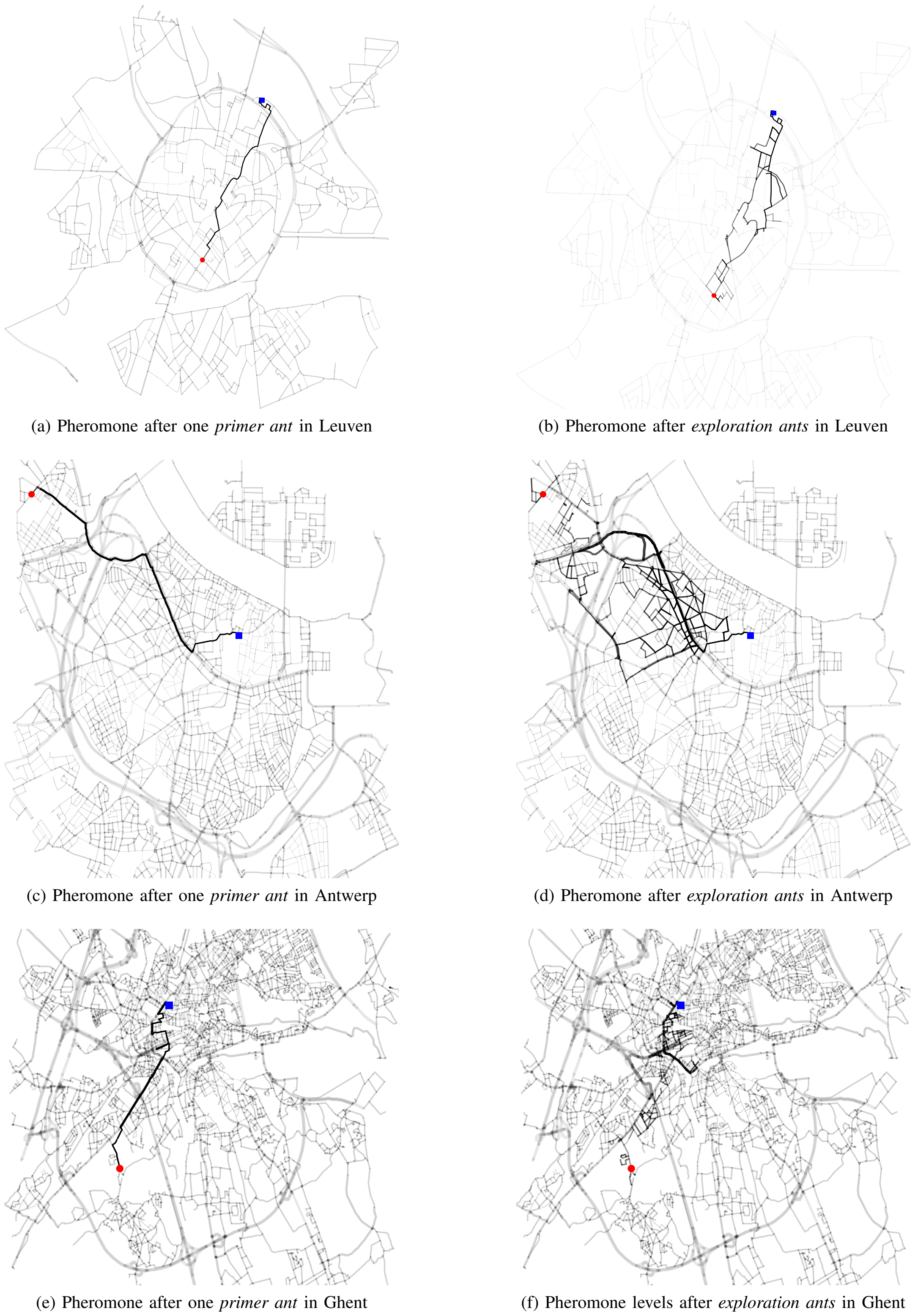
## IV. Evaluation

In this section we evaluate the proposed approach. First pheromone trails are depicted and analyzed. These trails give a good indication of the explorative power of the approach. The more pheromone trails in the environment, the more unique solutions the ants have constructed. The locations of the pheromone trails will also give a first indication of the exploitative nature of the ants. If the pheromone trails are centered across the axis defined by the static solution, they will likely be variations to this static solution. Next, a more in-depth analysis of the various parameters and their influence on the exploitation and exploration balance are presented.

The algorithm described in Section II has a number of parameters that affect the outcome. Finding a good set of parameters is very difficult and time consuming. The set of parameters used in the remainder of this section was obtained by running the algorithm with various values for $\alpha$, $\beta$, $\gamma$, $\theta$ and $\varphi$ and comparing the quality of the results.

### A. Pheromone trails

The algorithm is tested in three urban city networks, namely Leuven, Antwerp and Ghent. All three traffic network graphs are based on OpenStreetMap data. Figure 1 shows the pheromone trails in these three cities. Figures 1a, 1c and 1e on the left show the pheromone trail of the primer ant. The figures 1b, 1d and 1f on the right show pheromone trails after 1000 ants traversed the network. The spread of the pheromone indicates that ants manage to construct solutions that are variations on the statically constructed route.


(a) Pheromone after one *primer ant* in Leuven (b) Pheromone after *exploration ants* in Leuven

(c) Pheromone after one *primer ant* in Antwerp (d) Pheromone after *exploration ants* in Antwerp

(e) Pheromone after one *primer ant* in Ghent (f) Pheromone levels after *exploration ants* in Ghent

Figure 1. Pheromone trails for random origin destination pairs in the city networks of Leuven, Antwerp and Ghent. The origin and destination are depicted as a circle and square respectively. The intensity of a link is proportional to the pheromone level on that link, the width of a link is proportional to the speed limit on that link.

| parameter | | value |
|---|---|---|
| Pheromone power | $\alpha$ | 2.5 |
| Heuristic power | $\beta$ | 16 |
| Heuristic importance | $\gamma$ | 0.1 |
| Pheromone increase | $\theta$ | 8 |
| Pheromone decrease | $\varphi$ | 0.3 |

Table I
PARAMETER VALUES USED IN THE EVALUATION

### B. *Parameter sensitivity*

As explained in Section III, all parameters have a slightly different influence on the exploitation versus exploration tradeoff and thus the quality of the results. To analyze these effects, the sensitivity of two quality indicators regarding changes to the parameter is analyzed. The quality indicators chosen are: *relative quality of the best route* and *number of improvements found*. The first indicator, *relative quality of the best route*, or $q_{best}$ is defined as:

$$q_{best} = \frac{min_{k \in R}\,(tts_k)}{tts_{stat}}, \tag{3}$$

where $R$ is the set of all routes found by the ants, $tts_k$ is the total time spend by the vehicle if it was to use route $k$ and $tts_{stat}$ is the total time spend by the vehicle if it was to choose the result obtained using $A^*$.

The second indicator, *number of improvements found*, is the number of routes found by the ants having $tts_k < tts_{stat}$. Meaning the number of routes that would be an improvement regarding total time spend by the vehicle when choosing that route instead of the route obtained using $A^*$.

In these sensitivity analysises, one of the parameters in Table I is varied over a range around the value in the table. The rest of the parameters are kept constant. While this ignores any interactions between the parameters, it can still give a better understanding of the effects that parameter has.

The box plots in Figure 2 shows the performance indicators as parameters change. The plots on the left show the sensitivity of $q_{best}$ to changes in the parameters. It is worth noting that for the parameter-values chosen, the $q_{best}$-value can be as low as $0.7$. Meaning a 30% reduction of time spend for the vehicles. The figures on the right show the number of routes found that have a score that is better than that of the static solution. Together with Figure 1, these graphs show that the ants manage to explore a large part of the network graph and still find solutions that are qualitative.

## V. RELATED WORK

Ant Colony Optimization algorithms are often applied to routing in communication networks [8] and in vehicle routing problems [9]. Despite the similarity in name, the vehicle routing problem does not resemble the problem tackled in this paper. Vehicle routing problems involve the routing of one or more service vehicles to a number of customers. VRP problems are more related to travelling salesman problems, than to vehicle route calculation.

Many Ant Colony Optimization based algorithms used in communication networks bare close resemblance to the algorithm presented in this paper. AntNet [8] even serves as inspiration for a traffic routing algorithm, Ant Based Control (ABC) by Tatomir et al. [10] where an ACO inspired algorithm is used to route traffic through a network. In the ABC approach, vehicles send their perceived link travel times to a central service. The information stored in this service is used by ants to build up a probabilistic routing table in every node of the traffic network. Ants, called *forward ants*, travel from random origins to random destinations. These forward ants use information stored in the nodes of the graph to navigate to their destination. At every node, a routing table contains probabilities that, given the forward ants destination, determine how probable an outgoing edge is. The route choice of the forward ants is, like in the approach presented here, probabilistic. As forward ants traverse their randomly chosen path, they collect information on the total time spend. When a forward ant reaches its destination, it transfers all of its information to a *backward* ant. This backward ant then traverses the route and updates the information in the routing table. Vehicles, when trying to find a route to their destination, use the probability in the routing tables build by the ants.

Together, the forward and backward ants behavior is the equivalent of the exploration ants in our approach. The ants traverse a virtual graph representing the environment, picking up information known locally to the roads. This information is aggregated in the ants and is used to construct routes through the network. In both our approach and that of Tatomir, information about the constructed solutions is stored in the environment in the form of pheromones using the principle of stygmergy.

The way vehicles and ants interact in the work of Tatomir differs from the interaction between vehicles and ants in our approach. In [10], ants are send out between random points in the network graph. The information about the constructed solutions is not reported back directly to the vehicles, instead, it is stored in the network in the form of probabilistic routing tables. These routing tables are there for the benefit of all vehicles. In later work, Tatomir allows a vehicle to dispatch ants on routes that are relevant to the vehicle, thus ensuring that the relevant entries in the routing tables are present instead of leaving this up to the random ant movements.

The scope of the problem tackled in [10] differs from the scope of the problem solved here. All vehicles in the network benefit from the ants in Ant Based Control, while in our approach, only the vehicle that controls the ant uses the information brought back by the ant. Our approach assumes ants and pheromones from different vehicles don't interact.

In Ant Based Control, ants from different vehicles cooperate to build one globally relevant routing table. This global routing table is later divided in an hierarchical network [11].

Other heuristic search algorithms, such as genetic algorithms, and fuzzy approaches [12]–[16] can be used to generate what is called "dissimilar" routes, i.e. routes that are sufficiently different from each other to offer drivers meaningful alternatives. Li et al. argue in [13] that because "dissimilar" is a fuzzy concept, it benefits from fuzzy approaches. Genetic algorithms by nature are well suited to construct multiple solutions in a parallel fashion. The principles of cross-over and mutation ensure that the resulting solutions resemble each other. While genetic algorithms can easily construct solutions in parallel and can handle dynamic problems well, they are less suited for problems in which the fitness evaluation would require information that is distributed, as is the case in our problem.

## VI. Conclusion

This paper presents an Ant Colony Optimization based approach to route planning taking into account dynamic and distributed link travel time predictions. Applying the ACO metaheuristic to this problem required some modifications to the well known Ant System and Ant Colony System algorithms. Beside the introduction of a new parameter $\gamma$ to balance heuristic and pheromone values, the main modification is structural: The algorithm presented in this paper no longer operates in well defined iterations. The consequences of this modification is that applying global updates to pheromone levels becomes infeasible. Instead, the algorithms presented in this paper only use local updates by exploration or primer ants. This restriction makes the algorithms more easily deployed in distributed environments.

The tradeoff between exploration and exploitation that is crucial to the success of our algorithm, and Ant Colony Optimization algorithms in general, is discussed. The parameter space formed by the most important parameters of the algorithm is explored using a Monte Carlo simulation and a suitable set of parameters was found. The exploration versus exploitation balance is illustrated using pheromone trails in three urban networks. Pheromone trails in these networks show that the ants in our algorithm balance exploitation and exploration. The sensitivity of the results to parameter changes is analyzed and shown graphically. Finally, we show that when the link travel time predictions differ from the theoretical link travel times, using our approach outperforms static routing using an $A^*$ algorithm.

## Acknowledgment

This research was funded by the IWT - SBO project 'MASE' (project no. 060823) and by the Interuniversity Attraction Poles Programme Belgian State, Belgian Science Policy, and by the Research Fund K.U.Leuven.

## References

[1] E. Bonabeau, M. Dorigo, and G. Theraulaz, *Swarm intelligence: from natural to artificial systems*. Oxford University Press, USA, 1999.

[2] R. Claes and T. Holvoet, "Maintaining a distributed symbiotic relationship using delegate multiagent system," in *Proceedings of the 2010 Winter Simulation Conference*, B. Johansson, S. Jain, J. Montoya-Torres, J. Hugan, and E. Yücesan, Eds., 2010.

[3] R. Claes, T. Holvoet, and D. Weyns, "A decentralized approach for anticipatory vehicle routing using delegate multiagent systems," *Intelligent Transportation Systems, IEEE Transactions on*, 2010.

[4] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," *IEEE Computational Intelligence Magazine*, vol. 1, no. 4, pp. 28–39, 2006.

[5] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: optimization by a colony of cooperating agents," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 26, no. 1, pp. 29–41, 2002.

[6] M. Dorigo and L. Gambardella, "Ant colony system: A cooperative learning approach to the traveling salesman problem," *Evolutionary Computation, IEEE Transactions on*, vol. 1, no. 1, pp. 53–66, 2002.

[7] T. Stutzle, H. Hoos *et al.*, "Max-min ant system," *Future Generation Computer Systems*, vol. 16, no. 8, pp. 889–914.

[8] G. Di Caro and M. Dorigo, "Antnet: Distributed stigmergetic control for communications networks," *Journal of Artificial Intelligence Research*, vol. 9, no. 1, pp. 317–365, 1998.

[9] L. M. Gambardella, L. M. Gambardella, É. Taillard, and G. Agazzi, "Macs-vrptw: A multiple colony system for vehicle routing problems with time windows," *NEW IDEAS IN OPTIMIZATION*, pp. 63–76, 1999.

[10] B. Tatomir and L. Rothkrantz, "Dynamic traffic routing using ant based control," in *Systems, Man and Cybernetics, 2004 IEEE International Conference on*, vol. 4. IEEE, 2005, pp. 3970–3975.

[11] B. Tatomir and L. J. Rothkrantz, "Hierarchical routing in traffic using swarm-intelligence," in *Intelligent Transportation Systems Conference, 2006. ITSC'06. IEEE*. IEEE, 2006, pp. 230–235.

[12] J. Inagaki, M. Haseyama, and H. Kitajima, "A genetic algorithm for determining multiple routes and its applications," in *Circuits and Systems, 1999. ISCAS'99. Proceedings of the 1999 IEEE International Symposium on*, vol. 6. IEEE, 2002, pp. 137–140.

[13] Y. Li, R. He, L. Liu, and Y. Guo, "Genetic algorithms for dissimilar shortest paths based on optimal fuzzy dissimilar measure and applications," *Fuzzy Systems and Knowledge Discovery*, pp. 312–320, 2005.

[14] B. Chakraborty, "Ga-based multiple route selection for car navigation," *Applied Computing*, pp. 76–83, 2004.

[15] S. Nanayakkara, D. Srinivasan, L. Lup, X. German, E. Taylor, and S. Ong, “Genetic algorithm based route planner for large urban street networks,” in *Evolutionary Computation, 2007. CEC 2007. IEEE Congress on*. IEEE, 2008, pp. 4469–4474.

[16] L. Lup and D. Srinivasan, “A hybrid evolutionary algorithm for dynamic route planning,” in *Evolutionary Computation, 2007. CEC 2007. IEEE Congress on*. IEEE, 2008, pp. 4743–4749.

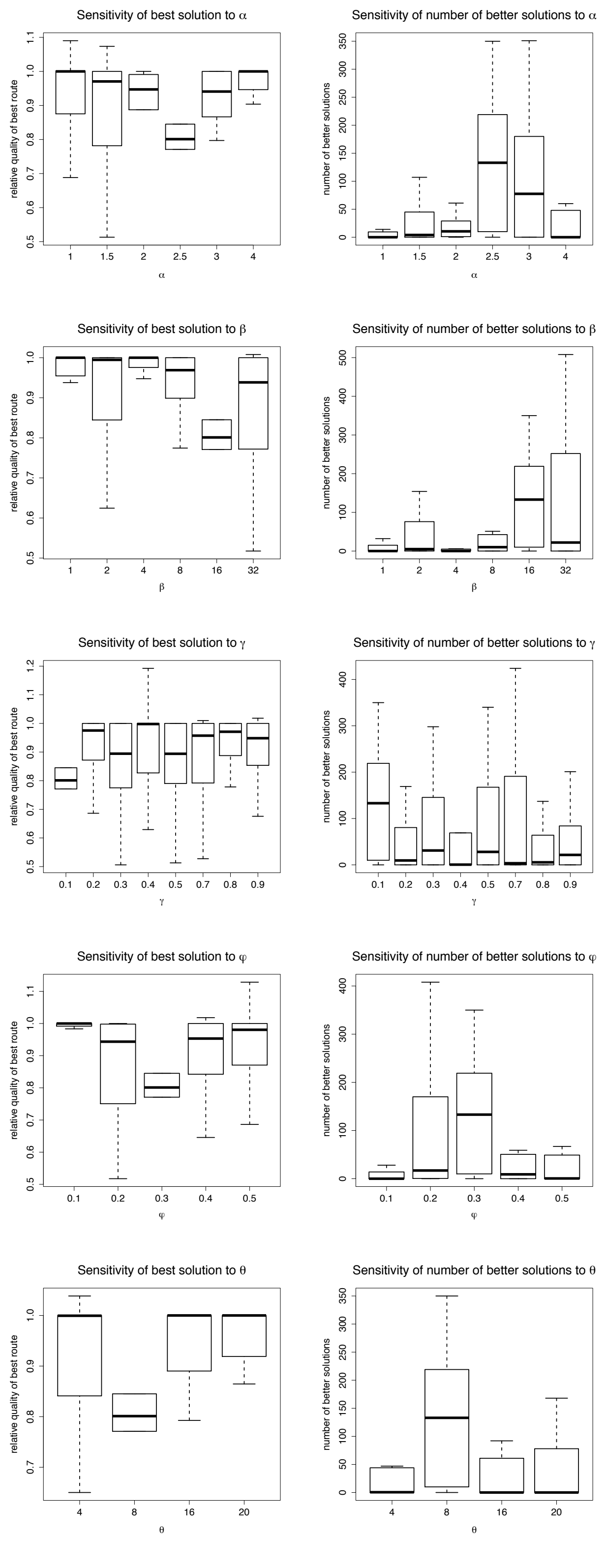


Figure 2. Sensitivity of the best solution and the number of better solutions regarding changes to parameters $\alpha$, $\beta$, $\gamma$, $\varphi$ and $\theta$.