CBPRS: A City Based Parking and Routing System

J.L. Boehlé, L.J.M. Rothkrantz and M. van Wezel

ERIM REPORT SERIES *RESEARCH IN MANAGEMENT*	
ERIM Report Series reference number	ERS-2008-029-LIS
Publication	May 2008
Number of pages	20
Persistent paper URL	http://hdl.handle.net/1765/12467
Email address corresponding author	mvanwezel@few.eur.nl
Address	Erasmus Research Institute of Management (ERIM) RSM Erasmus University / Erasmus School of Economics Erasmus Universiteit Rotterdam P.O.Box 1738 3000 DR Rotterdam, The Netherlands Phone: + 31 10 408 1182 Fax: + 31 10 408 9640 Email: info@erim.eur.nl Internet: www.erim.eur.nl

ERASMUS RESEARCH INSTITUTE OF MANAGEMENT

REPORT SERIES
RESEARCH IN MANAGEMENT

ABSTRACT AND KEYWORDS	
Abstract	Navigational systems assist drivers in finding a route between two locations that is time optimal in theory but seldom in practice due to delaying circumstances the system is unaware of, such as traffic jams. Upon arrival at the destination the service of the system ends and the driver is forced to locate a parking place without further assistance. We propose a City Based Parking Routing System (CBPRS) that monitors and reserves parking places for CBPRS participants within a city. The CBPRS guides vehicles using an ant based distributed hierarchical routing algorithm to their reserved parking place. Through means of experiments in a simulation environment we found that reductions of travel times for participants were significant in comparison to a situation where vehicles relied on static routing information generated by the well known Dijkstra's algorithm. Furthermore, we found that the CBPRS was able to increase city wide traffic flows and decrease the number and duration of traffic jams throughout the city once the number of participants increased.
Free Keywords	dynamic routing, information systems, computer simulation
Availability	
Classifications	

CBPRS: A City Based Parking and Routing System

Reducing congestion by integrating routing, adaptivity and parking place allocation.

J.L.Boehlé*, L.J.M.Rothkrantz†, M.van Wezel*

* Econometric Institute
School of Economics
Erasmus University Rotterdam
P.O.Box 1738
3000 DR Rotterdam
The Netherlands
Phone: +31 10 4081264
Fax: +31 10 4089167
Email: JLBoehle@planet.nl, mvanwezel@few.eur.nl

† Man Machine Interaction group
Department of Information Technology and Systems
Delft University of Technology
Mekelweg 4
2628 CD Delft
The Netherlands
and
Technical Sciences/SEWACO
Netherlands Defense Academy
Den Helder
The Netherlands

Phone: +31 15 2787504
Fax: +31 15 2787141
Email: L.J.M.Rothkrantz@ewi.tudelft.nl

Abstract - Navigational systems assist drivers in finding a route between two locations that is time optimal in theory but seldom in practice due to delaying circumstances the system is unaware of, such as traffic jams. Upon arrival at the destination the service of the system ends and the driver is forced to locate a parking place without further assistance. We propose a City Based Parking Routing System (CBPRS) that monitors and reserves parking places for CBPRS participants within a city. The CBPRS guides vehicles using an ant based distributed hierarchical routing algorithm to their reserved parking place. Through means of experiments in a simulation environment we found that reductions of travel times for participants were significant in comparison to a situation where vehicles relied on static routing information generated by the well known Dijkstra's algorithm. Furthermore, we found that the CBPRS was able to increase city wide traffic flows and decrease the number and duration of traffic jams throughout the city once the number of participants increased.

Keywords – Dynamic Routing, Information Systems, Computer Simulation.

1. INTRODUCTION

The emergence of modern communication technology has enabled city councils to revise their transportation and parking policy in ways that were infeasible a few years ago. Through the use of these technologies new services have been developed to combat urban congestion and generate a more optimal use of road capacity. One example of these services are the electronic traffic loops embedded into road surfaces that display congestion information to drivers on large matrix panels in and around cities. The most striking example is the toll zone introduced in the city of London that requires drivers to pay a fee when entering the city center and enforces this through TV cameras monitoring the entrance of all vehicles.

The manner in which modern technology is employed by city councils to combat urban congestion is still of a static nature: Services only gather information and relay it to specific locations or discourage access to a certain part of the city. This results in an increase in traffic pressures in other parts of the city. The application of modern technology in a more dynamic manner by collecting traffic data on a large scale, using the vehicles themselves, and distributing it directly to those who require it instead of at selected locations along the roads, could combat inner city congestion more efficiently.

Besides congestion, another common problem in large cities is the lack of sufficient parking places at shopping centers and other commercial venues. Drivers circulate in the vicinity of their point of destination in search of a parking place, and by doing so they hinder other traffic. Dutch city councils have opted to combat this problem by creating large parking and rest facilities at the edges of cities and increasing the cost of inner city parking in the hope that people will park their vehicle at one of these facilities and use public transport to enter the city (Reader Kennisatelier 11 "Bereikbaarheid van steden" 2003). While this solution in theory alleviates traffic pressures, there is no guarantee it does so in practice since it introduces the hassle of traveling via one more public transport services.

Navigational systems are another example of modern technology commonly used by drivers: These are able to assist the driver in reaching his destination as soon as possible by calculating a route that, in theory, represents the shortest route from source to destination. However,most of these devices are still unable to take into account factors like inner city traffic congestions, accidents and road maintenance and they rely solely on static routing algorithms. (Although recently, TomTom launched the first navigation system, called TomTom HD traffic, based on a dynamic routing and tracking system of cellphones of cardrivers.) Furthermore, the driver himself is required to search for suitable parking place, which can cause considerable delays and hampers the flow of other traffic within the city.

To alleviate these problems we propose the City Based Parking and Routing System (CBPRS). CBPRS is a distributed system that comprises two complementary services. A parking service allows participating vehicles to reserve a parking place based on the drivers preferences (monetary cost, distance to original destination, etc). The second service, the routing part of CBPRS, constantly gathers traffic data from participating vehicles. This data is then used to determine time optimal routes from the vehicles' current position to the destination parking place. The information exchange between vehicles and the CBPRS is facilitated by means of `intelligent lampposts' located at each intersection in a city and near conglomerations of parking places under CBPRS control. Section 2 describes the manner in which the CBPRS functions in detail.

The purpose of this study is to determine the impact of the CBPRS within the bounds of a city environment. The null hypothesis in this study is based on the assumption that drivers currently use either their static routing navigational system or their own local road knowledge to lead them to their point of destination. The route preferred by these drivers is, for the purpose of this study, taken to be the shortest path in time between origin and destination, computed using Dijkstra's (Dijkstra 1959) well known static routing algorithm. Once drivers reach their destination they will need to search for a parking place near their destination.

Our hypothesis is that drivers that opt for usage of CBPRS will achieve significant travel time benefits as opposed to drivers in the null hypothesis situation. There are two potential reasons for these benefits: First, we suspect that the use of dynamic routing information instead of static routing shortens the timespan needed to arrive at the destination. Second, CBPRS alleviates the hassle of finding a parking place by reserving one upfront. We think that this reduces overall travel time CBPRS participants. In order to verify our hypothesis we measure the effects of the CBPRS in comparison to the null hypothesis through a series of experiments conducted in a simulation environment specifically developed for this study.

It is worth noticing that CBPRS's beneficial effects may extend beyond CBPRS users. The travel time benefits for participating drivers could lead to a better distribution of traffic within the city allowing for increased traffic flows throughout the city. This general increase should be observed when comparing this hypothesis with the city traffic flows as resulted from the situation described for the null hypothesis.

This paper has two main contributions: (1) It introduces CBPRS and studies its effects through simulation, and (2) it introduces and describes a simulation environment that is easily extensible and available in the public domain. The remainder of this paper is structured as follows: Section 2 gives a global overview of CBPRS's components and their interaction. Section 3 elaborates on the routing algorithm that is employed by CBPRS. This `ant based' routing algorithm is both adaptive, distributed and robust. We will argue that these properties make it well suited for application in the CBPRS. Section 4 describes the simulation environment we used in this study. Section 5 describes the experiments we performed and the results that were obtained. Finally, Section 6 gives some conclusions and recommendations for further research.

The importance of our study from a policy perspective was recently illustrated by the Dutch State Secretary of Economic Affairs (Mr. Frank Heemskerk), who declared that the Dutch government would find it desirable if car navigation manufacturers would offer parking place information in order to combat congestion, pollution and accident rates. (Wierenga 2008). However, it is unclear what the effects of such a service would be in practice -- Our study helps to answer that question.

2. SYSTEM MODEL

As stated earlier, the CBPRS is an interconnected distributed system that provides both parking allocation and routing support. To this end, the CBPRS uses four component types: an on board device located in the vehicle of the user, intelligent lampposts spread throughout the city, parking place sensors that monitor the availability of parking places and a parking place scheduling service.

To facilitate communication between its components the CBPRS is active within three layers within the city environment as shown in Figure 1. The CBPRS employs radio for wireless communication between vehicles and other components such as intelligent lampposts and parking place sensors, while the power line infrastructure is used for communication among lampposts and between lampposts and the parking place scheduling service. The city road network dictates the distribution of the intelligent lampposts and thereby the layout of the network on the wireless and power line layer.

The CBPRS bases its routing decisions on real-time traffic information gathered by the participating vehicles. In order to collect this information the CBPRS needs to communicate with the vehicles on a frequent basis. The communication between vehicles and the CBPRS could be established via various methods, the most obvious of which seems to be the mobile phone. However, using a mobile phone network has many drawbacks, e.g., potential network overload, central network components going down, high usage costs, and the proprietary nature of the networks and the generated data.

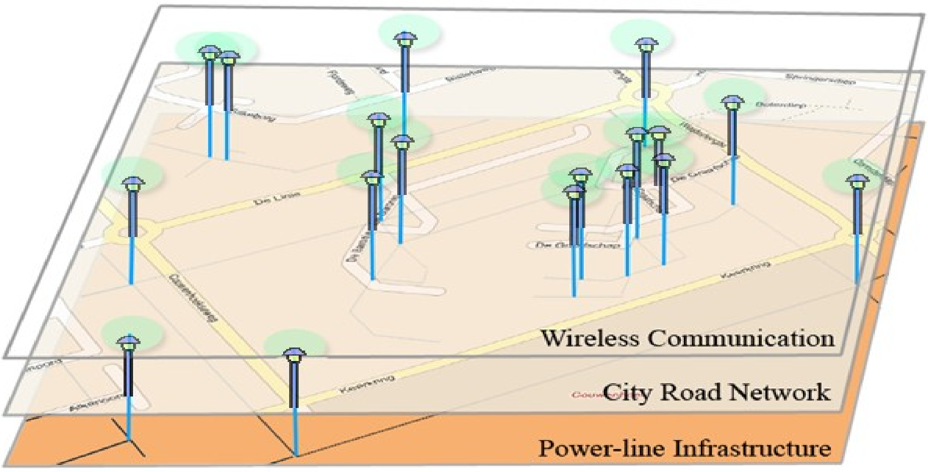

Figure 1: Communication infrastructure of the CBPRS

Our solution remedies these drawbacks and is highly cost effective. We require that each user of the CBPRS possesses a navigational device. In addition, users receive a small on board device that is able to interface with their navigational system with a unique identity, containing a wireless transmitter and receiver and a processing unit to interpret and exchange messages with the CBPRS. Through the interface with the users navigational device the on board device is able to query the user for information and display driving instructions received from the CBPRS that lead towards the reserved parking place. Communication between the CBPRS and vehicles occurs every time the vehicle passes an intelligent lamppost: The vehicle then supplies the lamppost with traffic data, gathered by measuring the time between intelligent lampposts.

The intelligent lampposts thus collect the traffic data and distribute it within the CBPRS. They are placed at each intersection in the city and near conglomerations of parking places and communicate through the (necessarily existing) power line infrastructure. Each lamppost further contains a device that allows it to communicate with vehicles and parking places, and a small processing unit to process routing information and maintain routing tables. These routing tables, used to route vehicles, are constantly updated using the most current information. Updating is done by the ant based routing algorithm, as is discussed below.

The large number of intelligent lampposts and their distributed operation makes the network highly robust against sabotage or failure of one or more intelligent lampposts: Such an event merely decreases the quality of the routing somewhat but it does not bring the system down.

When the CBPRS reserves a parking place near the user's destination the parking place scheduling service determines the optimal parking place, reserves it, and notifies the user. The optimal parking place is not only determined by the users preferences but strives for the most optimal distribution of vehicles over the available parking places from the perspective of the collective of CBPRS users. Each user that reserves a parking place with the CBPRS specifies his point of destination, the maximum distance the parking place is allowed to be from the destination and the maximum fee he is willing to pay for the parking place. These factors are then weighed against the collective interest of all other users to determine the optimal parking place. Existing city council parking policy ordainments can be further refined by adjusting the weighting factors used by the scheduling service in such a manner that certain areas of the city are more expensive during peak hours or by offering cheaper places at other locations in the city in combination with a (free) public transport fare.

Currently, the parking place scheduling services is a centralized component that communicates with the lampposts via the power line infrastructure. The lampposts will forward these messages to parking place sensors or vehicles within the city. One could also envision a more robust decentralized system where each city area has its own controller.

Parking place sensors are small, power efficient devices that monitor the status of a single parking place. They are positioned at every parking place assigned to the CBPRS and function to determine the availability of a parking place and to gather data to support accurate billing. Groups of nearby parking place sensors are controlled by a single lamppost via wireless communication, and are powered through wireless power transfer.

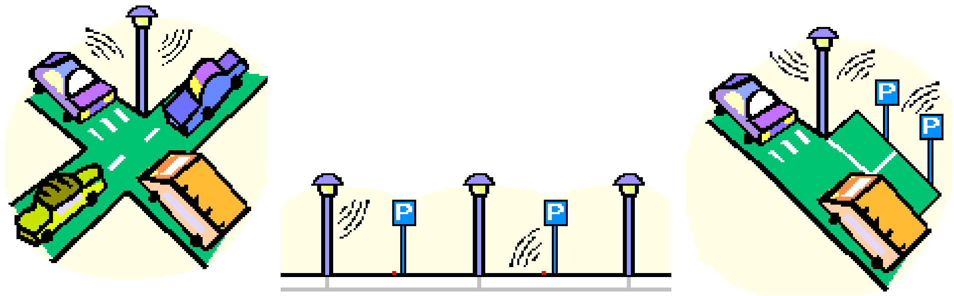

Figure 2: Interaction between the various components of the CBPRS

Figure 2 shows how the components of the CBPRS interact. On first contact with a lamppost, a vehicle transmits its destination and request a parking place. The lamppost transmits the request to the scheduling service via the power line infrastructure and gives the vehicle routing directions to its current destination. Once the scheduler has determined the parking place it transmits this to the next lamppost on the route of the vehicle, where the vehicle is informed. Simultaneously, the vehicle with its unique identity transmits its traffic data to the CPBRS via the lamppost. The vehicle receives updated directions at every lamppost. As soon as the vehicle parks, the parking place sensor detects this and informs the parking place scheduler. When the vehicle leaves, the sensor again informs the scheduler. (This data allows the vehicle to be billed for parking through an associated billing service.) The CBPRS guides the departing vehicle to the most optimal exit out of the city.

The technology required for all components of the CBPRS is widely available and produced in high quantities. The infrastructural requirements imposed on a city are low and the abundance of lampposts within Dutch cities – one about every 25 meters on alternating sides of the road – provides more than significant density to create the required coverage of wireless signals for the CBPRS to operate. The cost of implementing the CBPRS can be recouped rapidly if additional services are associated with the CBPRS. The intelligent lampposts could e.g., provide information concerning points of interest within their vicinity and allow wireless Internet access. It is also worth mentioning that CBPRS requires no additional user input over a normal routing system and is therefore very user friendly.

3. ANT BASED ROUTING

CPBRS constructs its routing tables using the ant based routing algorithm (Di Cario 1998, Dorigo 2002, 2006, 2007; Schoonderwoerd 1996, Tatomir 2004). This algorithm mimics the food searching behavior of Argentinean ants. Ants traveling between a food source and their colony deposit a pheromone (a scent-molecule) trail that marks the trail they have taken and which evaporates over time. At each obstacle encountered, an ant decides whether to go left or right. Each path is taken with a probability proportional with the strength of the pheromone trail. Thus, initially approximately fifty percent of the ants will choose to divert by going left around an obstacle while the others will go right. If, however, the right route is shorter than the left route, the pheromone trail of the ants following that route creates a denser pheromone trail than the ants traveling via the left route. The higher density will entice other ants in following the right route, which thereby reinforces itself until the trail on the left route has evaporated. In the end, only a single trail around all obstacles remains. It has been shown that, under certain conditions, the path marked by this trail is the shortest path from food to colony.

In traffic networks, obstacles are formed by congestions, road works or accidents. Avoidance of these obstacles is paramount when attempting to reach the destination in the shortest time possible. The ant based routing algorithm enables human drivers to cooperate in a manner equal to ants in order to form time optimal shortest paths throughout the city, circumventing the obstacles. Vehicles provide the system with a constant stream of up-to-date data concerning obstacles. These data enable intelligent agents called ants to approximate the shortest routes through a city. This routing information is then relayed real time to vehicles . The following subsections describe the infrastructure of the ant-based algorithm in relation to the CBPRS and present the algorithm in more detail.

3.1 Road Network Hierarchy

Routing vehicles via time optimal paths towards their destination involves the determination of vehicular traffic flows and densities on the roads within the network. Low vehicular traffic flows combined with high vehicular densities often indicate the presence of an obstacle and should therefore be avoided in favor of roads with a high vehicular traffic flows and low vehicular densities. While this approach functions in theory, in practice certain negative side effects can occur that are unacceptable in modern cities. The most common example is when a main road leading into a city is congested: The ant based algorithm then favors roads that lead through the densely populated area residential area next to the main road. While in terms of optimal time routing this path is valid, residents will soon start to complain and undesirable effect from the city councils point of view. In order to prevent the negative side effects from occurring we have opted to impose a hierarchy on the road network within the city (Tatomir 2004).

A second reason for imposing a hierarchy on the road network is the relation between nodes and ants within the ant based algorithm. The ant based algorithm requires that each node periodically sends an ant to all other nodes within the network, as will become clear below. Thus, when the number of nodes increases the number of ants required for proper function of the algorithm grows quadratically. This inevitably leads to an increase in ant processing time which can cause to the formation of suboptimal routes within the network and overall degenerated performance of the ant based algorithm (Kassabalidis 2002).

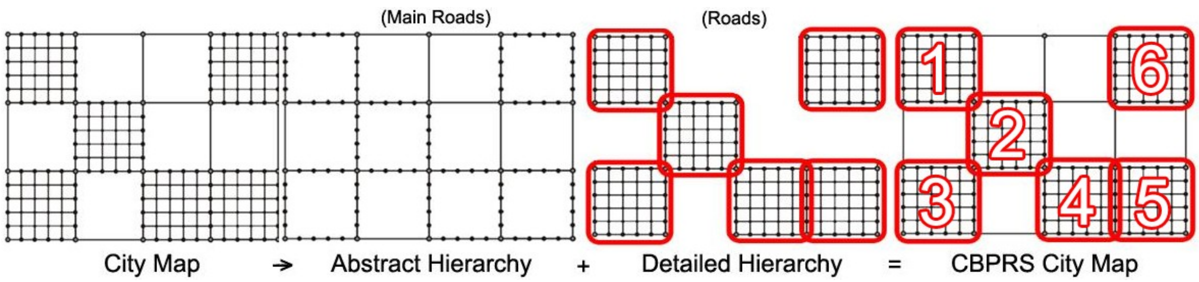

Figure 3: Transformation from city map to CBPRS network hierarchy

Road networks within cities display a 'natural hierarchy' when considering throughput capacity. The high capacity roads (main roads) serve to quickly transport vehicles from the outskirts of the city to the city center and back, while roads with a lower capacity enable drivers to access the main roads within the city. The hierarchy we have designed consists out of two hierarchical layers that exploit this 'natural hierarchy' as shown in Figure 3. The main road layer consists out of the main roads within a city and so forms an interconnected network for transporting large amounts of vehicles between the sectors of the detailed layer. The detailed layer consists of the lower capacity roads within the city, and they are clustered into sectors.

We created these sectors by subtracting the main road layer from the normal city map creating independent groups of roads that form the basis of the sectors. To enable travel between sectors we linked each sector to the nearest main road layer intersections. Each sector than has an `entry point' that we termed virtual node. Vehicles traveling between sectors use the virtual node associated with their destination sector as destination in the main road level and once they enter the sector they are routed towards their final goal.

3.2 Routing tables

Our ant based algorithm uses separate routing tables for both hierarchical levels described in the previous section. The global routing table serves to route vehicles between sectors over the main roads, while the local routing table serves to route vehicles within a sector. The routing table configuration, as shown in Table 1, contains an entry for each destination to which the node is allowed to route vehicles. This entry consists out of the address of the destination D_i and a series of probability values $P_{1\to i} \dots P_{n\to i}$ one for each neighbor $1 \dots n$. $P_{j\to i}$ represents the strength of the pheromone trail for destination i via the neighbor *j*. The entry further contains an average delay μ_i representing the average time in seconds required to travel from the current intelligent lamppost towards the destination.

Table 1: Node routing table basic layout

Destinations	Neighbor 1	…	Neighbor n	Average Delay
D_i	$P_{1\to i}$		$P_{n\to i}$	μ_i

Valid destinations for the local routing table of a given node in a certain sector are those nodes that reside in the same sector. The neighbors listed in the local routing table are those nodes that reside in the same sector and can be reached by following an outbound lane from the current node. A node has one local routing table for each sector it is part of. Valid destinations for the global routing table are the virtual nodes in the main road level. Table 2 presents an illustration of the information stored within the global and local routing tables. Note that the sum of probabilities for each destination over all neighbors always equals one.

Table 2: Examples of a global and local routing table

Destination	Neighbor 1	Neighbor 2	Average Delay
Virtual node 1	0.25	0.75	631s
Virtual node2	0.05	0.95	1138s

Destination	Neighbor 1	Neighbor 2	Average Delay
Node 5	0.65	0.35	60s
Node 8	0.12	0.88	135s

3.3 Traffic Condition data

The ant based algorithm requires a constant stream of up-to-date travel time information to generate optimal routes. In the CBPRS the participants provide this information through the additional hardware placed in their vehicle. This device monitors the time required to travel between intelligent lampposts and communicates this perceived delay d_r to the first intelligent lamppost it encounters, which in turn makes an estimate of the average delay M_r encountered on road in question r. M_r can be seen as an edge weight in the road graph used in the shortest path computation, and it is updated using the following equation:

$$M_r = M_r + \omega . (d_r - M_r) .$$

Here, the weight $\omega \in [0;1]$ determines the influence that a newly reported delay d_r has on the measurement and how long such a reported delay influences this measurement. We use the value $\omega = 0.3$ which implies that the latest ± 17 delays reported by vehicles influence this measurement. The number of weights that really influence the measurement can be approximated by $5(1/\omega)$ (Di Cario 1998). This relatively small amount of delays that influence the average delay allows the routing algorithm to react quickly to ever changing traffic conditions. The intelligent lamppost stores the delay information in this manner for each road with incoming lanes on the intersection it is monitoring. This information is then used by the route finding system, described in the next section, to determine the most time optimal routes within the network.

3.4 Ant Based Routing Table Construction

In order to compute the shortest paths based on the most recent delays (edge weights) we use the Ant based routing algorithm, which has desirable optimality and robustness properties (Bonabeau et.al. 1999, Dorigo et.al. 2004, Schoonderword et.al. 1996). The method uses intelligent agents called ‘ants’. There are four types of ants, each with a specific task. Table 3 describes these ant types from a percept/action/goal perspective. (This is a common perspective to consider intelligent agents in the field of Artificial Intelligence.)

In table 4 you put dots after every item in the second column

Table 3: Intelligent agent capability overview

Agent Type	Percepts	Actions	Goals	Hierarchy
Forward Ant	- Source node Id. - Destination (virtual) node Id. - Current node Id. - Current node routing tables.	- Store route information. - Determine optimal next node on route. - Remove cycles in route. - Transform to backward Ant.	Travel to destination node along the most optimal path and store the node ids of the visited nodes and the required travel time between each pair of visited nodes on the route.	The ant operates solely on the detailed hierarchal level within the bounds of its origin sector.
Backward Ant	- Source node id. - Destination node id. - Current node id. - Current node routing tables.	- Alter routing tables. - Terminate the agent on completion of route.	Travel backwards along the path followed by the forward ant and alter probabilities in the routing tables to reflect the discovered optimal route.	The ant operates solely on the detailed hierarchal level within the bounds of its origin sector.

Forward Exploring Ant	- Source node Id. - Source virtual node id. - Destination virtual node id. - Current node Id. - Current node global routing tables.	- Store route information. - Determine optimal next node on route. - Remove cycles in route. - Transform to backward exploring ant.	Travel to destination virtual node along the most optimal path and store the node ids and virtual node ids of the visited nodes and the required travel time between each pair of visited nodes on the route.	The ant operates solely on the nodes present on the main road level.
Backward Exploring Ant	- Source node id. - Source virtual node Id. - Destination node id. - Destination virtual node id. - Current node id. - Current node global routing tables.	- Alter routing tables. - Terminate the agent on completion of route.	Travel backwards along the path followed by the forward exploring ant and alter probabilities in the routing tables to reflect the discovered optimal route between the virtual nodes.	The ant operates solely on the nodes present on the main road level.

Ants are electronic entities that are transmitted between adjacent lampposts via the power line infrastructure They should not be confused with cars. (Since the power line network and the road network do not necessarily coincide, an ant's route between two adjacent nodes may be different from a car's route. However, this is irrelevant for the car routing algorithm.) The forward ant and the forward exploring ant, each one operating at its own hierarchy level, are responsible for discovering the optimal routes by examining routing and traffic data at each intermediate node. Periodically, each lamppost spawns a forward ant to a random destination within its hierarchy level. This ant is then routed to its destination using the routing tables in the intermediate nodes. On its route, the ant memorizes the latest traffic delay data from the nodes on its path. Upon arrival, a forward (exploring) ant is transformed into a backward (exploring) ant which retraces the route taken and updates the routing tables int the intermediate nodes in a manner that will be described shortly.

The routing tables are updated as follows. Upon arrival at lamppost *i*, the backward ant calculates the average delay T_{id} between the current lamppost and the destination lamppost d by summing up the individual average delays T_{jk} for the roads between the lampposts on the path between i and d :

$$T_{id}=\sum T_{jk},$$

where $j,k\in i\rightarrow d$. (This notation means that edge *j,k* is on the path taken from *i* to *d*.) The average delay is then used to alter the average delay for the route from the current lamppost to the destination as stored by the lamppost:

$$\mu_d=\mu_d+\eta(T_{id}-\mu_d),$$

where the parameter $\eta\in[0;1]$ is used as a weight to limit the influence of each delay on the average delay. We use $\eta=0.1$. Next, the virtual delay T_{id} for the current route is divided by average delay μ_d multiplied with a scaling factor $c\in[1;2]$ to determine the strength of the reinforcement λ given to the probability P_{dn} :

$$\lambda=\begin{cases}\dfrac{T_{id}}{c.\mu_d} & if\ \dfrac{T_{id}}{c.\mu_d}<1\\[2ex] 1 & if\ \dfrac{T_{id}}{c.\mu_d}\geq 1\end{cases}$$

Here, n represents the next lamppost on the path towards the destination. We use $c=1.1$. If the virtual delay for the route between intelligent lampposts i and d is less than the average delay the probability eventually receives a positive increment, otherwise the probability is left unaltered.

Once the reinforcement strength is known, the backward adjusts the probability values for each valid neighbor V_n from the set of neighbors N_n of the current lamppost. The set of valid neighbors is a subset of the set of neighbors for a certain intelligent lamppost; the members of this set are those neighboring intersections to which traffic rules at the intersection allow a vehicle to move. The probability value P_{dn} for neighbor n on the shortest path towards to destination intelligent lamppost d receives a positive stimulus using

$$P_{dn}=\alpha.(1-\lambda)(1-P_{dn}).$$

Here $\alpha\in[0;1]$ is used as a scaling factor to dampen oscillation caused by the probability updating process (Kassabalidis 2002). (We use $\alpha=0.1$.) The probabilities P_{dr} of the other intelligent lamppost r ???out of the set of valid neighbors V_n are decreased by

$$P_{dr}=-(1-\lambda).P_{dr},$$

where $dr\neq dn, dr\in V_n, V_n\subset N_n.$ The decrease in probability values for these intelligent lampposts reflects their current status as sub-optimal routing solutions for the CBPRS routing service.

After alteration of the probabilities, the values are normalized and clipped between 0.05 and 1. The lower value of 0.05 is set to prevent ants from ignoring a possible route towards the destination via an alternative intelligent lamppost. Intelligent lampposts between which the route should be disabled can set their probability to zero for that specific neighbor; this prevents ants from inspecting that route. The backward ant repeats this process on every intermediate node between *s* and *d*. Once the backward ant arrives in *s* – the source of the forward-ant - the backward ant is destroyed and the path between the source and destination nodes is updated according to current information available to the algorithm.

4. SIMULATION ENVIRONMENT

Needless to mention, a field experiment with the proposed system was beyond our reach. Therefore, we resorted to simulation. For the purpose of experimenting with the CBPRS we required a simulation environment that:

1. Facilitates the modeling of city environments in a detailed manner enabling the recreation of actual city environments.

2. Provides ‘realistic’ vehicular behavior in terms of acceleration and deceleration, the formation and dissipation of congestions within the city environment and lane changing.

3. Features a complete implementation of the CBPRS according to the model of Section 2.

4. Contains a graphical user interface to enable visual study of formation of pattern and the effects of the CBPRS on traffic conditions within the city.

Initially we planned to modify an existing microscopic vehicular simulator but after studying a number of alternatives (Tatomir 2004, Protuberant 2001, Randolph 1993) we found them lacking in vehicular behavior and extendibility. So, we implemented a new simulation environment, which we will briefly describe. A more elaborate description can be found in (Boehle 2007). The simulator is in the public domain and it can be found at http://mmi.tudelft.nl/dynamic_routing/.

The simulation environment was implemented in the object oriented C# programming language and based on design patterns described by (Gamma 1995). To ensure flexibility and extendibility, we divided the functionality of the simulator into five groups of classes. The *simulation* group provides classes for conducting and regulating the flow and speed of the simulation as well as basic data logging services. The *infrastructure* group contains classes that define a road network such as intersections, roads, lanes and a default implementation of a vehicle. The *routing* group provides classes for experimenting with different routing algorithms. The ant based and Dijkstra routing algorithms that are provided by default. The *CBPRS* group contains a full implementation of the CBPRS as proposed earlier including extended data logging services that monitor both vehicles and the CBPRS itself. The *GUI* group contains classes and functionality for providing the graphical display of the road network and vehicles. The GUI group is the most coupled of all groups since it combines functionality of all other group in providing a uniform overview to the user. A screenshot of the GUI is shown in Figure 4.

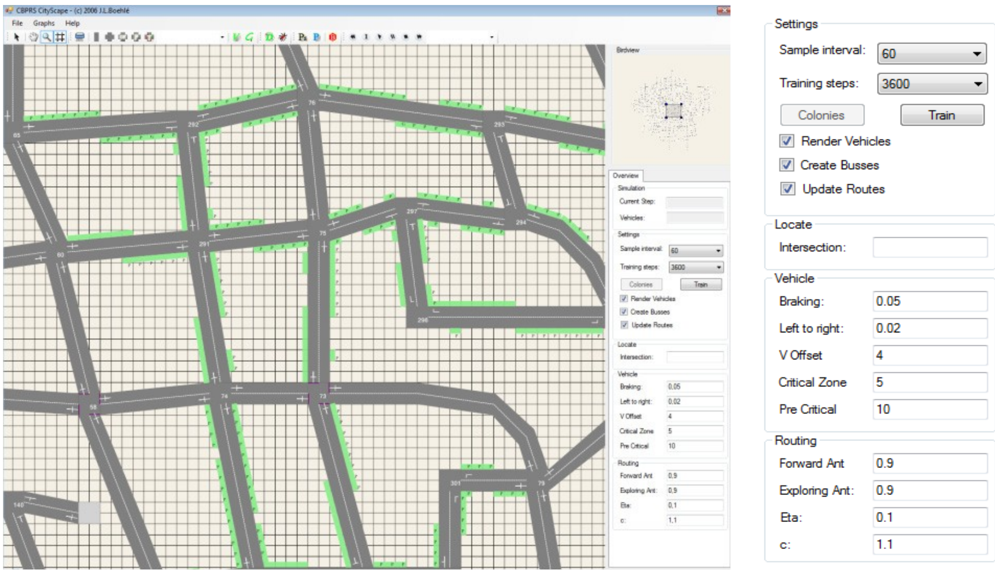

Figure 4: The simulation environment graphical user interface

The simulation of vehicular movements in a city environment is a complex and potentially computational heavy task, certainly when modeling in a microscopic simulation environment with a large set of parameters that influence the behavior of vehicles and drivers. Trade offs exist between the natural behavior of vehicles and the running time of the simulation and graphical representation thereof.

During our study into realistic behavior we came upon the concept of the cellular automata (Taub 1963). This principle enabled us to create 'realistic' vehicular movement and behavior without sacrificing simulation execution speed. When applying the concept of cellular automata to road networks, the driving lanes of the roads and intersections are divided into blocks of equal length. Each block can be occupied by one vehicle at a time and is 7.5 meters long – the space required by a vehicle standing still in a traffic jam –although it is possible to shorten the block length in order to simulate lower speeds.

At every time step of one simulated second all vehicles are moved to their new positions based on the rule set in use. In our simulation environment, we have opted for the vehicular movement rule set defined by Nagel and Schrekenberg (Nagel 1992) extended with ideas presented in (Wagner 1996) in order to produce 'realistic' accident free driving and lane changing behaviors in vehicles. Parameters within the rule set enable us to influence the tendency to change between lanes, traffic flow on driving lanes and chance that a vehicle will not properly assess the current traffic situation by braking in excessive manner. We extended the model with two additional parameters that influence the lane changing tendency before intersections to ensure vehicles arrive at intersection in a valid driving lane for their current route. The incorporation of visual signs such as turn lights and breaking lights, as shown in Figure 5, provided new means for simulating the reactions of specific types of behavioral driving models within the simulation environment.

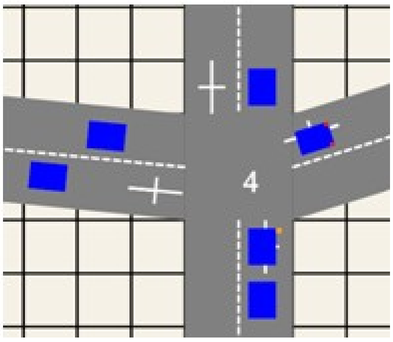

Figure 5: Traffic behavior at a precedence intersection

The infrastructure within the simulation environment consists out of a relatively small number of classes that, when linked together, enable the user to create complex road networks. Users are able to manually position intersections and roads as well as to determine the number of lanes on each road. Extended configuration parameters on each object further allow the user to increase the realism of the simulation although this can become quite time consuming. Table 4 contains an overview of the infrastructural classes and their capabilities.

Table 4: Simulation environment class descriptions

Class	Capabilities	Description
Lane	- Supports vehicle lane changing. - Enables complex vehicular movement through cellular automata structure. - Enables turning restrictions for vehicles at Intersections.	One-directional part of a road divided into blocks of 7.5 meters that allow a vehicle to move according to the global traffic rules. Turning restrictions {right, left, ahead, etc} at intersection can be used to stimulate complex presorting behavior of vehicles.
Road	- Supports multiple lanes. - Maximum vehicle speed. - Supports placement of Parking places / Garages.	Connection between two intersections that supports multiple interconnected lanes in either direction, facilitates lane changing for vehicles and supports the placement of parking places and intelligent lampposts.
Intersection	- Enables collision free crossing of multiple vehicles simultaneously. - Supports traffic lights. - Supports precedence rules . - Contains an intelligent lamppost.	Intersections function on the basis of a messaging system that works in a manner similar to traffic light detection loops in roads. The vehicle on the loop automatically requests a crossing then intersection then reports when the vehicle can. Intersections support the crossing of multiple vehicles from different lanes at the same time. Contains an intelligent lamppost that can be queried by vehicles for routing directions and the reservation of a parking place.
Generator	- Creates different types of vehicles at user defined periodic intervals based on. - Enables vehicles to leave the simulation environment.	The generator enables natural flow of vehicles within city by creating entry/exit points that generate different types of vehicles based on user preferences. Generators connect are connected to the city infrastructure via roads and can so be used to represent highway connections to other city, dense residential area's that serve traffic focal points during peak traffic hours.

Vehicle	- Supports different driver behaviors. - Supports collision free traffic movement - Supports uniform congestion behavior based on global rule-set. - Supports pre-sorting at intersection in correct drive lanes - Maximum vehicle speed. - Vehicle length {car, bus, truck}. - Communication with intelligent lampposts at intersections and parking places. - Supports routing via different routing algorithms {Ant-based routing, Dijkstra}. - Supports parking and roaming behavior.	As the most important class within the simulation environment the vehicle enables the realistic creation and resolving of congestions in combination with different vehicular behaviors as found in human drivers. The vehicle can be specialized by adjusting parameters and overwriting different behavioral functions. Specialization of this vehicle present in the simulation environment are the ant vehicle, ant parking vehicle, Dijkstra Vehicle, Dijkstra parking vehicle, bus and truck. Vehicles prefixed with ant are users of the CBPRS where Dijkstra Vehicles route themselves using Dijkstra's algorithm. The bus is used to provide additional information to the CBPRS concerning a fixed route through the city.
Parking place	- Supports parking for a single vehicle. - Hourly tariff specification. - Tracks and reports parking place status . through incorporated parking place sensor.	Parking place located at the side of a road and accessible via the outermost most lane of a road. Parking places can be queried for their status {free, occupied, reserved} and track their own occupancy rate and revenue.
Parking garage	- Supports parking 100 to 500 vehicles. - Hourly tariff specification. - Tracks and reports parking place status per parking place.	Conglomeration of parking places that all have a uniform hourly tariff. The individual parking places within the parking garage operate in the same manner as the parking places described above.

The manner in which intersections are modeled in a simulation environment has a great impact on the patterns of vehicular movement that emerge during a simulation. Intersections that allow only a single vehicle to pass at each time step unnecessarily delay the progress of other vehicles causing traffic jams and delays to appear on roads where they would not appear in reality (Schadschneider 2000, Giridhar 2006). Therefore, certain precautions have to be taken in order to insure that multiple vehicles can cross the intersection at the same time without leading to collisions and disobedience of the precedence rules applicable to the intersection.

Each vehicle that approaches an intersection notifies the intersection of its arrival and the edge it intends to follow when crossing the intersection. The intersection then verifies if the edge that the vehicle wants to follow is available – meaning that there are no conflicting routes in use at current – and applies precedence and traffic light rules. The intersection should then formulate a response to the vehicle stating whether it is allowed to proceed onto the intersection. In the next time step, the vehicle determines if it is allowed to continue and enter the intersection or to break to a halt before entering the intersection. This type of messaging system ensures collision free traffic movement over intersection and reduces the computational overhead.

We stress the fact that we are using a random traffic generator. Our results are context sentive, this means depending of the capacity of the roads and total amount of generated cars. The impact of those parameters is beyond the scope of this study.

5. EXPERIMENTS

In order to verify the hypothesis concerning reduced travel times for CBPRS participants and increased traffic flows in the city, we performed experiments with cities ranging from hamlet to metropolis size. Here, we only show the results for the two largest environments (city and metropolis) since these yield the most interesting results. For small environments the effects of the CBPRS were negligible (Boehle, 2007).

5.1 The City Environment

The city environment can be distinguished from the smaller environments through the fact that traffic volumes are reasonable high and remain high for almost the entire duration of the day sometimes stretching well into the evening. The city environment for this experiment is modeled after the Dutch city of Apeldoorn that has a ring road along its periphery. A further characteristic is the presence of multiple distinct neighborhoods within the city and the clustering of commercial activity within a certain neighborhood of the city. The ring road allows vehicles to move at an increased maximum speed of 80 km/h giving vehicles the opportunity to travel quickly from one side of the city to another. This prevents overcrowding of the roads within the inner city that have a lower maximum speed. While such a ring road has certain benefits, it can also be the source of congestions within a city.

Participating vehicles can be characterized along two dimensions. The first is the routing algorithm employed, which can either be Dijkstra's algorithm or the ant based algorithm. (We do not simulate vehicles that do not employ routing.) Since Dijkstra's algorithm is nonadaptive we expect it to perform inferior when many traffic jams occur. The second dimension concerns parking behavior: Vehicles can either park or not, and if they do, they can use the CBPRS parking service or they can search for a place randomly. This leads to six possible vehicle types. We performed five experiments with various relative frequencies for these vehicle types, as shown in Table 5. Each experiment involved 7500 vehicles.

Table 5: City environment vehicular distributions per experiment. Each experiment involved 7500 vehicles

	Dijkstra nonparking	Dijkstra random parking	Dijkstra CBPRS parking	Ant nonparking	Ant CBPRS parking
Exp. one	90 %	10 %	0 %	0 %	0 %
Exp. two	90 %	0 %	0 %	0 %	10 %
Exp. three	0 %	0 %	0 %	90 %	10 %
Exp. four	40 %	0 %	0 %	50 %	10 %
Exp. five	90 %	0 %	10 %	0 %	0%

Experiment one served as our benchmark situation where all vehicles were routed via Dijkstra's algorithm. Experiment two measured the effectiveness of the CBPRS when 10 percent of all vehicles, the minimal participation limit set in this study, used the CBPRS services. Experiment three shows the maximum possible level of effectiveness of the CBPRS when all vehicles participate in the system. Other experiments illustrate upper and lower limits of effectiveness found in this simulation environment.

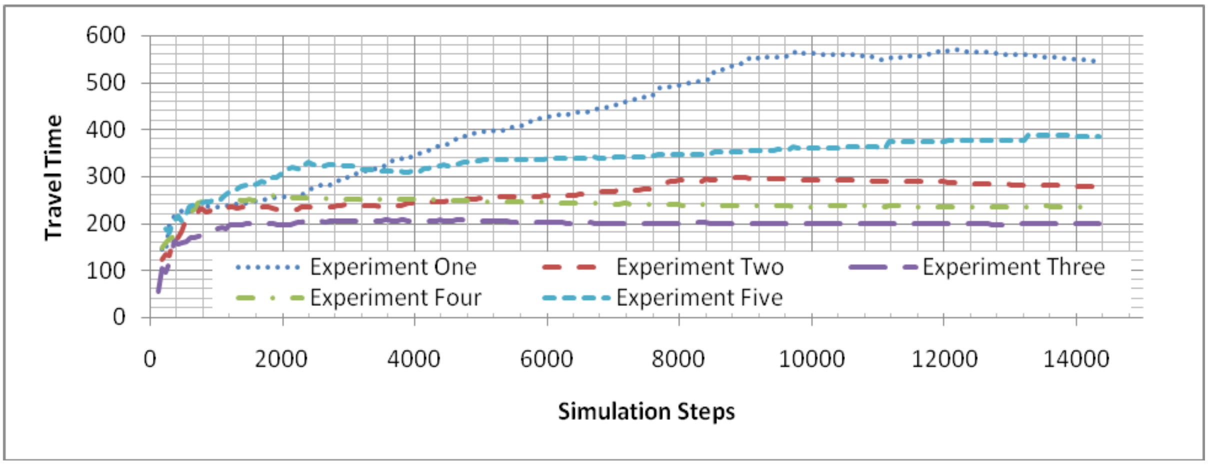

Figure 6: City environment average travel time for parking vehicles per experiment

Figure 6 depicts the development of travel times for parking vehicles in the experiments discussed in this section. A number of things are worth noticing. First, experiments one and three show that the average travel time in the case where all vehicles use Dijkstra's algorithm approaches 600 time units, whereas when all vehicles use CBPRS, this is reduced to 200 units. This is a huge reduction. Second, experiment two shows that a small proportion of CBPRS participants (10%) already has a large influence on total traffic flow in the city because it reduces travel times with app. 50%. Third, experiment four shows that when the CBPRS participation grade approaches 60%, the travel times approach the optimal situation. Fourth, experiment five shows that when only Dijkstra routing is used and this is combined with the CBPRS parking service, this reduces travel times by approx. 30%, which is only half of the total savings that were achieved in experiment three.

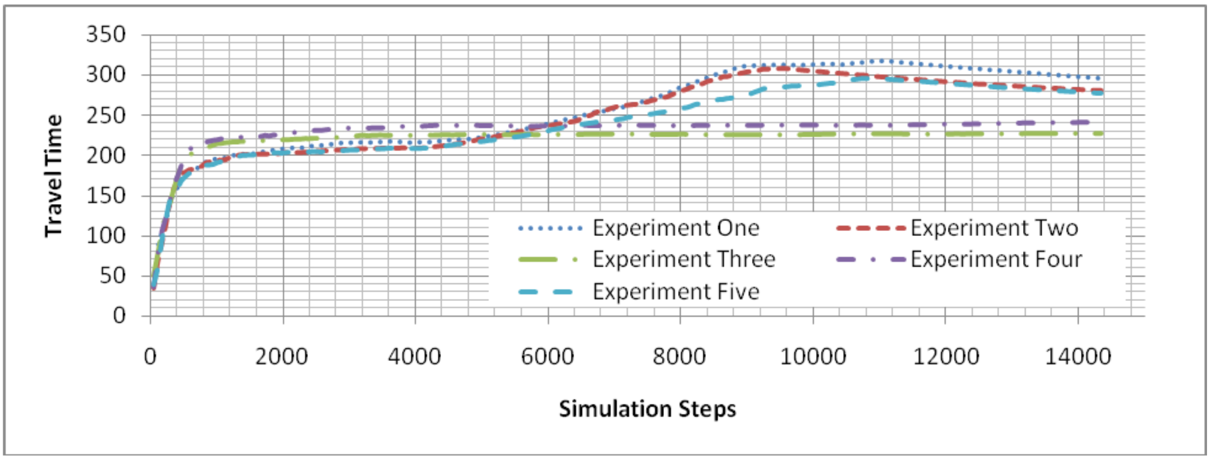

Figure 7: City environment average travel time for non-parking vehicles per experiment

Figure 7 shows the development of average travel times for nonparking vehicles. Experiments one, two and five all made use of Dijkstra oriented vehicles. These vehicles while initially slightly outperforming those of experiment three, but show deteriorating travel times from simulation step 6000 onward. The reason for these deteriorating travel times is found in the low vehicular distribution during these experiments and the resulting traffic jams. Experiment three shows the optimal situation when all nonparking vehicles are participants in the CBPRS. After an initial period of around 3000 simulation steps the ant based algorithm has acquired enough information to route the vehicles to their destination in the most optimal manner. Experiment four shows the same trend as experiment three however the nonparking vehicles used here are a mixture of Dijkstra oriented vehicles and ant based vehicles. While the Dijkstra vehicles have a slightly negative impact on the average travel time, this vehicle mixture outperforms those of experiments one and two significantly.

Table 6: City environment traffic jam statistics.

	Traffic jams	First appearance	Average time
Exp. 1	655	331	125
Exp. 2	388	419	105
Exp. 3	19	2648	38
Exp. 4	165	251	48
Exp. 5	378	628	129

Table 7: Average number of roads traveled per vehicle.

	Parking vehicles	Nonparking vehicles
Exp. 1	18.7	10.4
Exp. 2	10.1	10.4
Exp. 3	9.6	11.5
Exp. 4	10.2	11.5
Exp. 5	16.4	10.4

Table 6 and Table 7 further illustrate the beneficial effects of the CBPRS. The former table shows that with CBPRS, far less traffic jams appear. The latter table shows that the number of roads traveled is much smaller with CBPRS. Thus, CBPRS benefits traffic flows, travel times and it reduces pollution.

5.2 The Metropolis Environment

In the next series of experiments we performed we increased the city size to metropolis level. We modeled the environment after the Dutch city of Rotterdam, which possesses a clearly defined highway network around the city supplemented by an extensive network of main roads. The natural division of the city caused by the river Meuse provides another interesting challenge to the CBPRS in terms of routing between the northern and southern parts of the city using the few available connections. While the basic layout of the city of Rotterdam is incorporated into the environment, the modeled city itself should by no means be considered a realistic representation. Using this environment, six experiments were carried out of which Table 8 shows the vehicular distributions. Each experiment involved 10000 vehicles.

Table 8: Metropolis environment vehicular distributions. Each experiment involved 10000 vehicles

	Dijkstra nonparking	Dijkstra random parking	Dijkstra CBPRS parking	Ant nonparking	Ant CBPRS parking
Exp. one	90 %	10 %	0 %	0 %	0 %
Exp. two	90 %	0 %	0 %	0 %	10 %
Exp. three	0 %	0 %	0 %	90 %	10 %
Exp. four	60 %	0 %	0 %	30 %	10 %
Exp. five	40 %	0 %	10 %	50 %	10%
Exp. six	90 %	0 %	10 %	0 %	0%

The metropolis has two major characteristics – beyond size – that set it apart from the previous simulation environments. The first and most obvious is the extensive high-speed highway network running around the city. The capacity of the highway network enables vehicles to travel quickly around the metropolis without the chance of becoming stuck in inner city traffic. The second is the large number of multi lane main roads running through the metropolis. This combination of both a highway network being able to divert vehicles from the inner city as well as an extensive inner city main road network that allows the CBPRS to reroute vehicles via a myriad of paths provides benefits to both participants and non-participants as Figure 8 and Figure 9show.

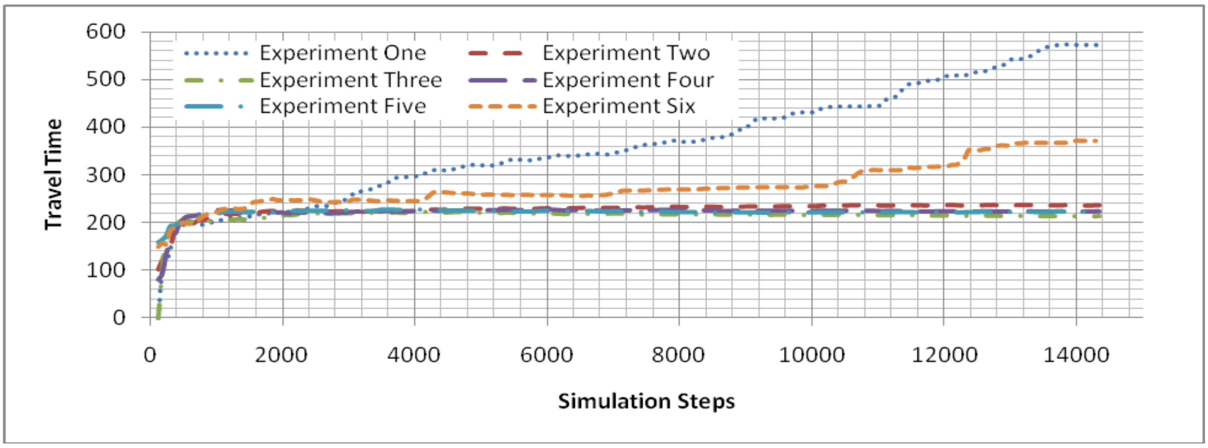

Figure 8: Metropolis environment average travel time for parking vehicles per experiment

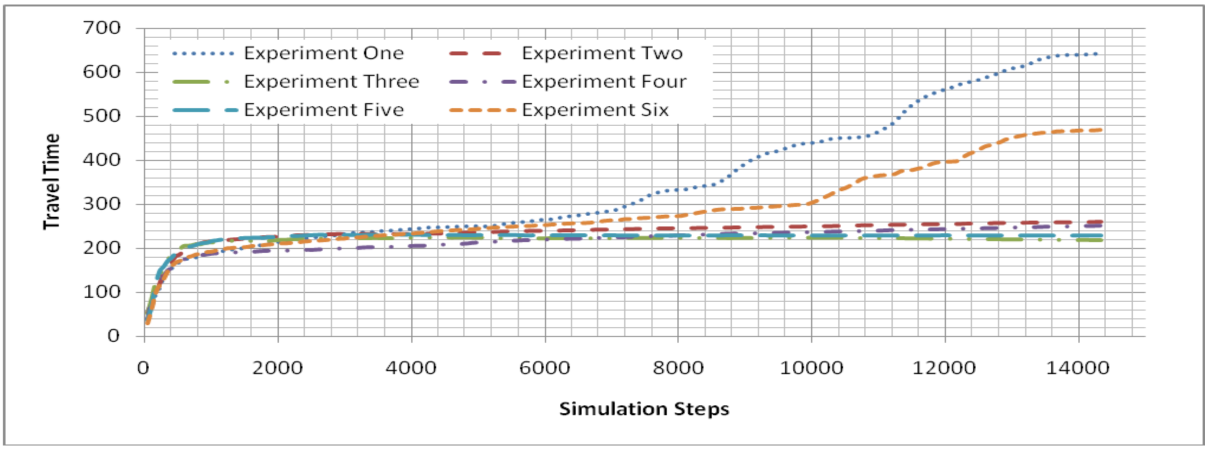

Figure 9: Metropolis environment average travel time for non-parking vehicles per experiment.

These results are in the same line as the results that were previously obtained for the city environment. However, there are two points worth noticing. First, all experiments employing CBPRS vehicles yield a low average travel time of app. 200 time units. The proportion of CBPRS participants does not seem to matter as much as in the city environment. Second, the non-parking Dijkstra vehicles seem to suffer more in this environment than in the previous one. Probably they have a higher probability to get caught in a traffic jam.

Table 9: Metropolis environment traffic jam statistics.

	Traffic jams	First appearance	Average time
Exp. 1	475	907	563
Exp. 2	155	1536	216
Exp. 3	33	8785	147
Exp. 4	53	5019	187
Exp. 5	39	7015	164
Exp. 6	389	434	411

Table 10: Average number of roads traveled per vehicle.

	Parking vehicles	Nonparking vehicles
Exp. 1	21.5	14.2
Exp. 2	14.6	15.9
Exp. 3	13.9	15.6
Exp. 4	14.1	14.5
Exp. 5	14.2	16
Exp. 6	17.5	14.5

Tables 9 and 10 show traffic jam data and the average amount of roads traveled per vehicle for all experiments. The same image arises as in the city experiments, although it seems to be more pronounced here.

6. SUMMARY, CONCLUSIONS AND DISCUSSION

In this paper, we have presented a city based parking and routing system (CBPRS) that routed participants through a city environment using distributed hierarchical algorithm based on the Ant colony meta-heuristic. Through a series of computer-simulated experiments, we have been able to show that providing a guaranteed parking place at or near the driver's point of destination in combination with a dynamically determined optimal route toward the parking place can lead to significant benefits for participants and non-participants in the CBPRS. These benefits are twofold: Both the travel time and the number of roads traveled are reduced by up to 70%. This has important economical and environmental consequences.

During the experiments we found that the size of the environments has a significant impact on the performance of both the Dijkstra guided vehicles and the CBPRS vehicles. The benefits of CBPRS increase with the size and complexity of the environment. We also found that the benefits increased with the participation grade for the CBPRS. However, even with very modest CBPRS participation grades of approx. 10%, the beneficial effects were substantial -- In the case of a metropolis environment the benefits approached those of full participation. Given this observation, it may be feasible to deploy a CBPRS like system successfully in reality without having to acquire a large user base. As mentioned we are using a random traffic generator. In the future we will use a traffic generator based on realistic traffic streams.

The CBPRS has a number of further advantages: It can be implemented with cheap, of the shelf hardware. The CBPRS is robust against failure of any of its components because of its distributed nature. When component become defective this merely deteriorated systems performance slightly. Thus, the CBPRS exhibits `graceful degradation' . Finally, the CBPRS does not require any additional user interaction or input over a traditional navigational system and it is thus very user-friendly. In our view, this study justifies further research and development into the CBPRS system.

7. REFERENCES

Bonabeau, E., Dorigo, M. and G. Theraulaz, *Swarm Intelligence*, Oxford University Press, 1999

Boehle, J.L., *City Based Parking and Routing System,* M.Sc. thesis. Man-machine interaction group Delft University of Technology. Available from http://www.kbs.twi.tudelft.nl/Publications/MSc/2007-Boehle-Msc.html (2007)

Di Cario G., Dorigo M., *AntNet: Distributed Stigmergetic Control for Communication Networks,* IEEE Journal of Artificial Intelligence Research, pp. 217-365 (1998)

Dijkstra E.W., *A note on two problems in connection with graphs*, Numerische Mathematik, pp. 269-271 (1959)

Dorigo M., Birattari M., Stützle T., *Ant Colony Optimization Artificial Ants as a Computational Intelligence Technique.* IEEE Computational Intelligence Magazine, (2006)

Dorigo M., Socha K., *An Introduction to Ant Colony Optimization,* In: Gonzalez T.F. (ed) *Approximation Algorithms and Metaheuristics,* CRC Press (2007)

Dorigo M., Stützle T., *The Ant Colony Optimization Metaheuristic: Algorithms, Applications, and Advances, Handbook of Metaheuristics, (*2002).

Dorigo M., Stützle T., *Ant Colony Optimization,* MIT Press, 2004

Gamma E., Helm R., Johnson R. and Vlissides J., *Design Patterns Elements of Reuseable Object Oriented Software,* Addison Wesley (1995).

Giridhar A., Kumar P.R., *Scheduling Automated Traffic on a Network of Road,* IEEE Transactions on Vehicular Technology, Vol. 55, pp. 1467 T 1474 (2006)

Kassabalidis I., ElK Sharkawi M.A., Marks II R.J., Arabshahi P., Gray A.A., *Adaptive-SDR: Adaptive Swarm-based Distributed Routing*, IEEE Proceedings of the 2002 International Joint Conference on Neural Networks, pp. 351-354 (2002)

Nagel K., Schrekenberg M., *A Cellular Automaton model for Freeway Traffic,* Journal Physics France, pp. 2221-2229 (1992)

Randhawa S.U., White S.J., Burhanuddin S., *A SimulationA based Analysis of Parking System Performance*, IEEE Proceedings of the 1993 Winter Simulation Conference (1993)

Rothkrantz L.J.M., Ehlert P.A.M., *Microscopic traffic simulation with reactive driving agents,* IEEE Intelligent Transportation Systems Conference Proceedings, pp. 860-865 (2001)

Schadschneider A., *Statistical Physics of Traffic Flow*, Physica A, v. 285, iss. 1-2, p. 101-120 (2000)

Schoonderwoerd R., Holland O.E., Bruten J.L., Rothkrantz L.J.M., *Ant based Load Balancing in Telecommunication networks*, Adaptive Behavior, pp. 169-207 (1996)

Tatomir B., Boehlé J.L., Rothkrantz L.J.M., *Dynamic routing in traffic networks and MANNETs using Ant Based Algorithms,* Proceedings of the 7-th International Conference on Artificial Evolution (2005)

Tatomir B., Dibowsky H., L.J.M. Rothkrantz, *Hierarchical routing in traffic networks*, 16th Belgian-Dutch Conference on Artificial Intelligence (BNAIC 2004), pp. 75-82 (2004)

Tatomir B., Kroon R., Rothkrantz L.J.M., *Dynamic Routing in Traffic Networks Using AntNet*, Ant Colony Optimization and Swarm Intelligence ANTS 2004, ISBN 3-540-22672-9, Lecture Notes in Computer Science 3172, pp. 334-342 (2004)

Tatomir B., Rothkrantz L.J.M., *Hierarchical routing in traffic using swarm-intelligence*, The 9th International IEEE Conference on Intelligent Transportation Systems, IEEE press, pp. 230-235,(2006)

Taub, A. H. (ed). 1961-63. *John von Neumann: Collected Works*, Pergamon Press, Oxford (UK)

Wagner P., Nagel K., Wolf D.E., *Realistic Multi Lane Traffic Rules for Cellular Automata*, Physica A, pp. 687 - 698 (1996)

Wierenga, Q., *Staatssecretaris: TomTom moet parkeerplekken zoeken.* Elsevier's Weekblad. Available from http://www.elsevier.nl/nieuws/internet_en_gadgets/artikel/asp/artnr/201662/index.html Accessed may 15-th, 2008 (2008)

Publications in the Report Series Research* in Management

ERIM Research Program: "Business Processes, Logistics and Information Systems"

2008

An Evolutionary Framework for Determining Heterogeneous Strategies in Multi-Agent Marketplaces
Alexander Babanov, Wolfgang Ketter and Maria Gini
ERS-2008-002-LIS
http://hdl.handle.net/1765/10972

Choosing between Auctions and Negotiations in Online B2B Markets for IT Services: The Effect of Prior Relationships and Performance
Uladzimir Radkevitch, Eric van Heck and Otto Koppius
ERS-2008-004-LIS
http://hdl.handle.net/1765/11288

Key Issues in Expansion of End-User Mobile Communication in China
Sunanda Sangwan, Guan Chong and Louis-Francois Pau
ERS-2008-011-LIS
http://hdl.handle.net/1765/11762

Some Comments on the Question Whether Co-Occurrence Data Should Be Normalized
Ludo Waltman and Nees Jan van Eck
ERS-2008-014-LIS
http://hdl.handle.net/1765/11810

Bibliometric Mapping of the Computational Intelligence Field
Nees Jan van Eck and Ludo Waltman
ERS-2008-015-LIS
http://hdl.handle.net/1765/11811

Demand Management Opportunities in E-fulfillment: What Internet Retailers Can Learn from Revenue Management
Niels Agatz, Ann Campbell, Moritz Fleischmann, Jo van Nunen and Martin Savelsbergh
ERS-2008-021-LIS
http://hdl.handle.net/1765/12244

Time Slot Management in Attended Home Delivery
Niels Agatz, Ann Campbell, Moritz Fleischmann and Martin Savelsbergh
ERS-2008-022-LIS
http://hdl.handle.net/1765/12245

Mobile Service Affordability for the Needy, Addiction, and ICT Policy Implications
L-F Pau
ERS-2008-023-LIS
http://hdl.handle.net/1765/12246

CBPRS: A City Based Parking and Routing System
J.L. Boehlé, L.J.M. Rothkrantz and M. van Wezel
ERS-2008-029-LIS
http://hdl.handle.net/1765/12467