Genetic Approach to Optimize Traffic Flow by Timing Plan Manipulation

Halina Kwasnicka and Michal Stanek
Wroclaw University of Technology, Institute of Applied Informatics
Wybrzeze Wyspianskiego 27, 50-370 Wroclaw, Poland
halina.kwasnicka@pwr.wroc.pl, michal.stanek@student.pwr.wroc.pl

Abstract

It is proved that the traffic control systems are one of the most profitable solutions of transportation problems. In this paper we present computer system, which consists of micro-simulation module and genetic based optimization module. Its aim is to improve traffic conditions in signalized intersections network. We focus on minimization of time losses and maximization of average vehicles speed during ride to guarantee the maximally smooth vehicles movement inside optimized area. We present assumptions of optimization methods as well as obtained results.

1 Introduction

With the growth of modern cities and the reliance of many of their populations on personal automobiles for the primary mode of transport, we need to use better traffic management systems [3]. There are substantial benefits to be derived from improved traffic flow. By improving the policies that control traffic lights, traffic flow can be improved and it can be done for considerably less cost than other infrastructural improvements such as increasing the capacity and number of roadways or adding public transit systems. A signal optimization method based on a genetic algorithm supported by microscopic simulation program is proposed.

In the next section we focus on modelling and controlling traffic flows methods where we described optimization process. Third part describes used genetic algorithm, fitness function and operators as well as chromosome structure. At the end we present obtained results and discussion.

2 Modelling and Controlling Traffic

In this section we focus on using information technology in transportation. Because of optimization of traffic flow demands effectively minimization of average traveling (and waiting) times for all cars crossed investigated area, hence it is our main goal.

A common tool for analyzing and predicting changes in traffic conditions is simulator [12, 6]. Traffic light optimization, in general, is a very complex problem and even for a single junction there might be no obvious optimal solution [4].

2.1 Controlling Car Flow

In road networks where each intersection has its own signalization the most effective car controlling method is changing the timing plans [4]. We define here main terms. An *intersection cycle* is defined as a time period in which all lighting groups get a green light at some point. *Lighting group* can assemble several entrances in order to they receive green light in the same time. The *split time* determines for how long the lights should stay in each state. Busy roads can get preference adjusting the split time. The *cycle time* is the duration of a all splits. In crowded traffic, longer cycles lead to better performance. The *offset of the cycle* defines the starting time of a cycle relative to other traffic lights. Offset can be adjusted to let several lights cooperate for example to create green waves. There is also possibilities to have influence on lights order.

The number of controlled variables leads us to statement that the optimization process can be very complex even for a single junction and grow exponential with the number of controlled junctions. Due to many difficulties, most researches in traffic light control focus on adapting the duration or the order of the control cycle. In our approach we propose a control method, which manage all timing plans variables, that can lead to more accurate traffic control. Of course, our approach requires information about the current traffic situation, which can be obtained by using different sensors (detectors) or communication systems.

2.2 Optimization Process

Before we start optimization process we need to collect traffic data from road detectors. This data are required to make decision if current conditions are sufficient or they

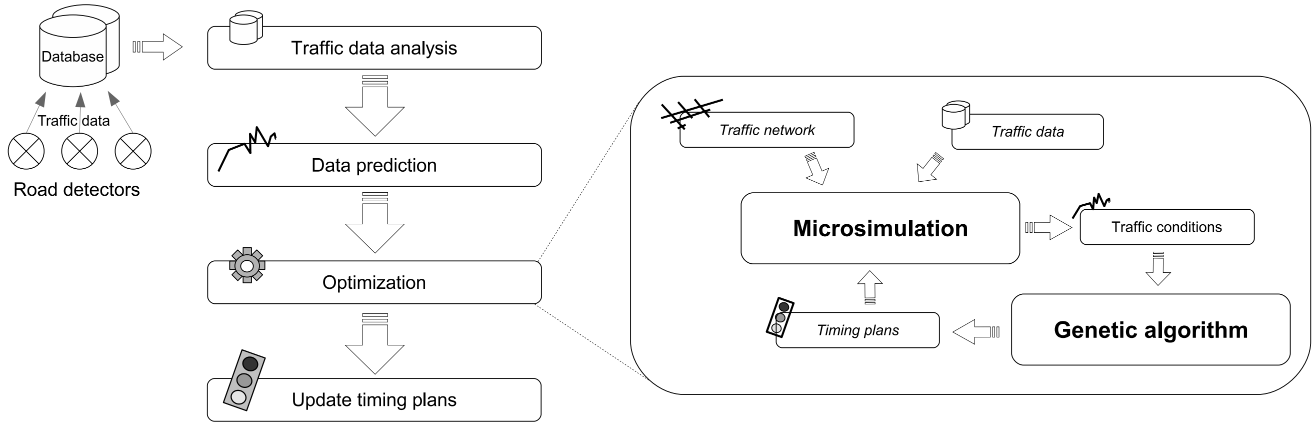


Figure 1. Scheme of traffic optimization process using data collected from road detectors

have to be improved. Starting costly traffic optimization process we will need to predict conditions which appear at the end of it. Furthermore, if the optimization process took 5 minutes we need to estimate traffic conditions 5 minutes ahead. Predicted data go to the optimization module. In this module new timing plans for all intersections are produced. At the end, the signalization is updated by the best of developed timing plans. Optimization process is presented in Figure 1.

Every traffic plan proposed by optimization algorithm should be checked from the optimization criteria point of view. To do that we use microsimulation module.

2.3 Microsimulation

We decide to use a microsimulation model to evaluate signalization timing plans. Unlike macromodels [4], where only information about whole traffic flow can be obtained, microsimulation allows us to track routes of all cars moving in simulated area [2, 6, 8, 9, 10]. To reduce computation cost we decide to use cellular automate and rules proposed by Esser and Schreckenberg [2, 10]. Four rules presented below make the core of the simulation process:

1. $v_i = min(v_i + 1,\ v_{max})$ with probability R,
2. $v_i = max(v_i - 1,\ 0)$ with probability Q,
3. $v_i = min(v_i,\ d_i)$,
4. $x_i = x_i + v_i$,

where x_i is a current car position, v_i is a number of cells in which car will be moved in the next automate step (we define it as a car velocity), and v_{max} represents the maximum car velocity. d_i is a number of cells between cars i^{th} and $i^{th} + 1$. The probability of acceleration R should be close to 1 and probability of deceleration Q should be approximately 0.25.

We have extended the above model to adjust it to better simulate urban traffic [9]. We had to reduce two things, time period between each simulation event, and a size of cells. Our model the has resolution of one meter and one second per simulation event. We introduce also four types of vehicles differ in length, maximum speed, and acceleration ratio. To precisely simulate car movement along junctions we introduce *collision points* where cars from different entrances can meet during movement.

Microsimulation model concerned in optimization module (Figure 1) requires data about road infrastructure (traffic network), current traffic data which include cars number on each entrance, flow structure (car types), and timing plans for traffic lights situated at all junctions.

Traffic conditions are the result of simulation and enter to the optimization algorithm (a genetic algorithm) which has access to data such as average waiting time, and average technical speed (average traveling speed exclude waiting time) in order to optimize timing plans for intersections.

2.4 Traffic Network

The considered in our study traffic network has only one way roads and six intersections which have two lighting groups, one for each traffic directions. Figure 2 presents the traffic network structure. There are two main roads with equal flow density and five side roads. Flow densities on all side roads are five times smaller than in main roads. Required flow density is assured by generators which place cars on roads with a given probability in each simulation step (on main road 45%, on side road 9%). It is possible to obtain information about total number of generated vehicles even if some of them cannot reach target road because of it is full.

Vehicle with certain probability chooses the target road

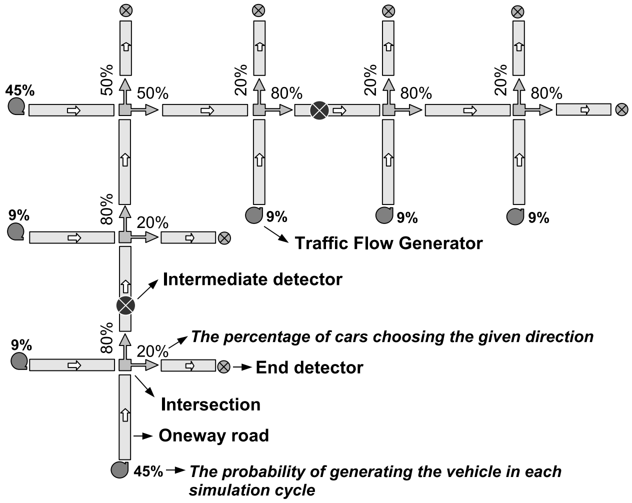


Figure 2. Structure of traffic network used in optimization process

on every intersecton. The probability of enter main road is equal 80%, the side road 20% (with except intersection of two main roads where the probabilities of the choice the two target roads are equal).

The detectors situated in the range of studied area enable us to get movement statistics of crossing them vehicles. Vehicle movement data are reset after crossing intermediate detectors. Every detector can have different priority used in fitness function (see equation 7). In presented traffic network all detectors placed outside the main roads have priority three times smaller than one placed on the main roads (the intermediate detectors are situated on the main roads).

Intermediate detectors are also treated as generators where numbers of generated and detected cars are equal. This assumption is required to assure correctly values of function given by equation 9.

3 Genetic Algorithm

In this section we present used evolutionary approach [7, 11]. The main described problems are: chromosome structure, fitness function, and genetic operators.

3.1 Chromosome Structure

All intersections timing plans are coded into a single chromosome. We assume that the cycle time is the same for all intersections, but other setting can be different for each one. Chromosome length depends on a total number of intersections and a number of lighting groups inside each one.

If we assume that N is the total number of intersections and the function $G(i)$ returns the number of lighting groups inside i^{th} intersection, than the total length of the chromosome is given by the equation 1:

$$L = 1 + 2 \cdot \sum_{i=1}^{N} G(i) + N \tag{1}$$

Other setting, such as split time s, order o of lighting groups and offset f, creates group for each intersection in the chromosome structure (Figure 3).

To code the total cycle time T_c we use integer representation. Setting value have to satisfy equation:

$$T_c \geq T_{min} \ \wedge \ T_c \leq T_{max}, \tag{2}$$

where T_{min} and T_{max} is the minimum and maximum cycle time respectively.

To code split time and offset value we also use integer representation with two constraints for the split time:

$$\begin{gathered} \forall i,j \in \aleph \ . \ (1 \leq i \leq N \wedge 0 \leq j \leq G(i)) \Rightarrow \\ \left(\left\lceil \tfrac{S_{min}}{T_c} \cdot 100 \right\rceil \leq s_j^{(i)} \leq 100 \right) \end{gathered} \tag{3}$$

and

$$\forall i \in \aleph \ . \ (1 \leq i \leq N) \Rightarrow \left(\sum_{j=i}^{G(i)} s_j^{(i)} = 100 \right), \tag{4}$$

where S_{min} is the shortest time of each split. Offset value for i^{th} intersection must satisfy the condition:

$$\forall i \in \aleph \ . \ (0 \leq i \leq N) \Rightarrow \left(f^{(i)} \geq 0 \wedge f^{(i)} < T_c \right) \tag{5}$$

To code order of lighting groups we use ordinal representation [7] with following constraints:

$$\begin{gathered} \forall i,j \in \aleph \ . \ (1 \leq i \leq N \wedge 0 \leq j \leq G(i)) \Rightarrow \\ \left(1 \leq o_j^{(i)} \leq (G(i) + 1) - j \right) \end{gathered} \tag{6}$$

3.2 Operators and Selection Method

We use standard genetic operators [7, 11]. The probabilities of use crossover as well as mutation operators are fixed during whole evolution process.

The crossover operator chooses randomly three points, by one in each parameters group.

Mutation operator draws value Δs from the assumed range $[min; max]$ with equal probability. Then Δs is added to the current parameter value. Min and max are defined separately for every parameter, and they are changed during evolution process – they exponentially follow to *zero* in successive generation ($min \to 0$ and $max \to 0$). In effect, at

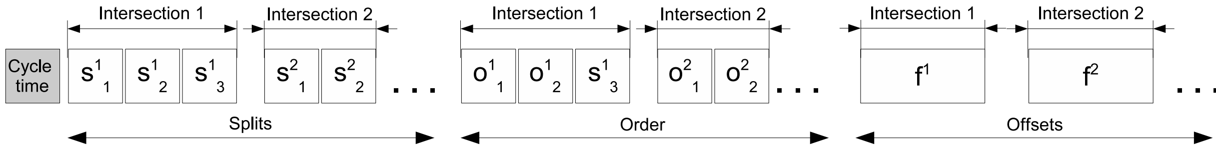


Figure 3. Chromosome structure with three parameters groups

the end of evolution changes caused by mutation are very small, what should enable tuning found solutions.

After use any operator we check conditions given by equations from 2 to 6, and in case of necessity, we repair the individual to fulfill them.

We use the ranking selection method [7] in which only the highest rated individuals can become parents. After each generation all individuals are sorted by fitness function. Predefined number of the best individuals become parents, where a number of children depends exponentially on the ranking position. The best individual in all evolution process is stored.

3.3 Fitness Function

The fitness function needs to minimize time losses during ride, as well as to guarantee the maximally smooth vehicles movement inside the optimized area. It is defined by equation 7:

$$Q(conf) = J(conf)^4 \cdot H(conf), \tag{7}$$

where $conf$ is the current timing plans setting. Function $J(conf)$ guarantees that with given flow density maximally a lot of vehicles will travel through the studied area, while function $H(conf)$ is responsible for fluency of cars movement. Both functions are defined by equations 8 and 9:

$$H(conf) = \sum_{i=1}^{N_D} \left(Prt^{(i)} \cdot {N_v}^{(i)} \cdot \left(\bar{v}^{(i)} + \frac{1}{\bar{s}^{(i)}} \right) \right), \tag{8}$$

where N_D is the total number of detectors, $Prt^{(i)}$ is the priority of the i^{th} detector, $N_v^{(i)}$ – a total number of vehicles recorded by i^{th} detector, $\bar{v}^{(i)}$ – the average technical speed of the vehicles detected by i^{th} detector, and $\bar{s}^{(i)}$ is the average waiting time of the vehicles register by i^{th} detector.

$$J(conf) = \frac{\sum_{i=1}^{N_D} {N_v}^{(i)}}{\sum_{j=1}^{N_G} {N_g}^{(j)}}, \tag{9}$$

where N_G is the total number of generators and ${N_g}^{(i)}$ is a number of cars generated by j^{th} generator. Function J returns values from the range $[0;\ 1]$.

4 Results

The genetic algorithm has to optimize traffic network described in section 2.4. We used in our experiments two network configurations, the first without, and the second with using intermediate detectors. Results are presented and discussed in this section.

4.1 Evolution Process

The timing plans were optimized with the following parameters of the genetic algorithm: population size – 100, number of generations – 70, mutation rate – 12%, crossover rate – 60%, simulation time – 1000 events, 40% best rated individuals in each generation became parents.

Taking into account random nature of traffic events, we perform the simulation process for each chromosome three times to minimize random factors. Chromosome fitness is the average value of Q (equation 7) obtained after simulations.

Evolution process for both considered network configurations is rather similar. Fitness results observed during the first 70 generations are presented in figure 4. Usually after 6 generations the average adaptation value stabilize. In Figure 5 we can see the best values of main traffic parameters used by function Q during successive generations of evolution.

We can see that the algorithm prefers solutions where more cars reached detectors (Fig. 5 b). The algorithm tries also to maximize average speed and minimize waiting time (Fig. 5 a).

The weakest individual fitness, especially in the latest generations, indicates that even small changes of settings in signalization timing plans can lead to large changes in traffic conditions. The limitation of optimized parameters number to two, for example – cycle time and offset, would limit search space very significantly but setting fixed split times in many ways can lead to inefficient solutions. Evolution process for traffic network without intermediate detectors looks very similar.

4.2 Discussion

Resulting timing plans for both tested network configurations give very good traffic conditions with fluently cars movement where no traffic jams are created. Queue on

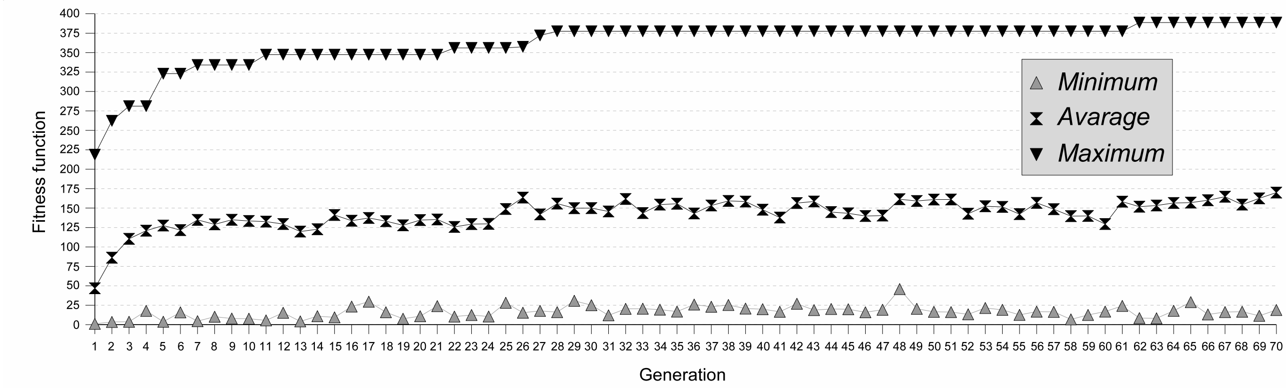
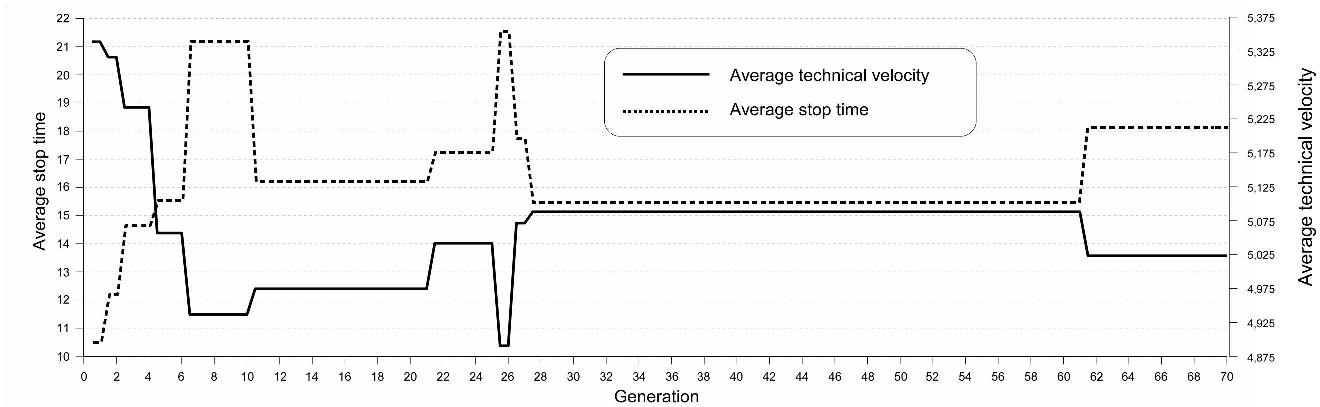
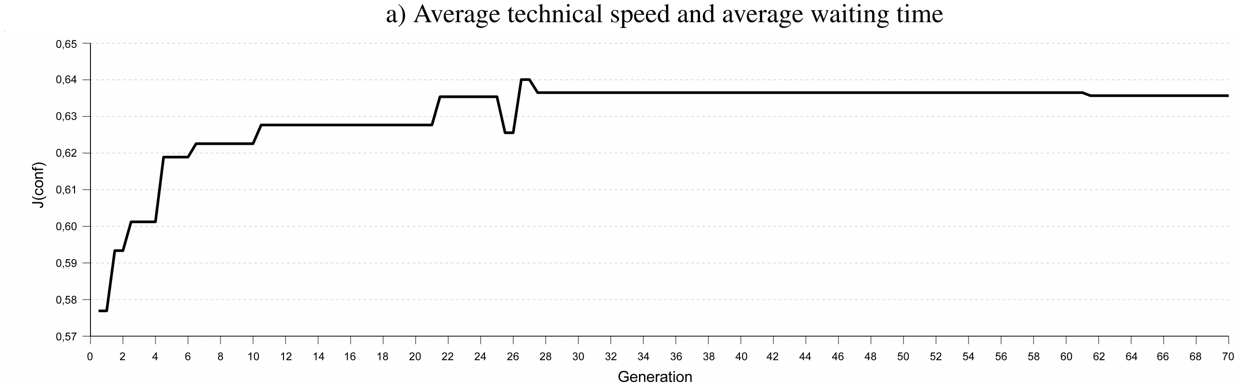


Figure 4. Evolution of timing plans in traffic network with intermediate detectors



a) Average technical speed and average waiting time



b) Values of J(conf) function (equation 9)

Figure 5. Best parameters obtained during optimization traffic network with intermediate detectors

every entrance is unloaded quickly and situations in which next vehicles cannot enter to the studied area does not appear. A little bit more fluently conditions in meaning of cars stops are observed in configuration with intermediate detectors. The average time of green light on every entrance is approximately equal to the queue unloading time. Green wave is created on the main roads and the way of grouping vehicles on entrances assure later they smooth ride through the studied area. Once again the solution using intermediate detectors gives better results.

It is very hard to compare obtained results with different algorithm because of universal testing sets are not exist.

Despite algorithm does not possess explicit traffic engineering knowledge, it is able to create very good solutions. We observed green weaves on the main roads, and very good queue unload time. Proposed algorithm is very scalable and can be used to optimize arbitrarily complicated traffic network even if there exist intersections without traffic lights.

5 Conclusion

The results of experiments indicate that optimization based on genetic algorithms and microsimulation module is suitable solution for evolving timing plans. Proposed algorithm is very scalable and can be used to optimize any transportation network. Managing all timing plans parameters give us opportunity to better adopt them to the current traffic conditions. This is significant achievement in comparison with many algorithms which optimize only one or two parameters [3, 4]. Use intermediate detectors inside traffic network can lead to even more accurate results without significantly growth of computation time.

The considered problem can be treated as multiobjective one, where we must maximize the capacity of the road configuration as well as the comfortable drive conditions. It is worth mentioning that the proposed in our system objective (fitness) function allows us to reduce problem to single objective optimization. Full realization of genetic algorithm and microsimulation was made. Environment was implemented in Java – it means, the complete framework and testing module for the traffic networks. Application screenshot is presented in Figure 6.

Proposed algorithm can be used in real time control systems because of constant optimization time. It is also very suitable to run in parallel mode where each individual in a population can evolve in different CPU.

References

[1] E. Delight, B. Provido, M. A. Ramires, W. C. Tabares, and P. R. Saldana. Computer simulations of traffic on road intersections. *not published*.

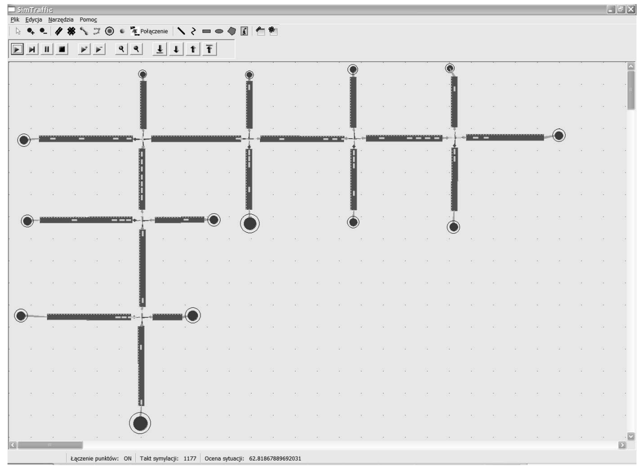


Figure 6. Screenshot of created environment with running simulation process

[2] J. Esser and M. Schreckenberg. Microscopic simulation of urban traffic based on cellular automata. *International Journal of Modern Physics C*, 8(5):1025–10036, January 1997.

[3] T.-T. expert group on ITS for road traffic management. Deployment of intelligent transport systems on the trans-european road network. *Technical Report, European Commision*, 2002.

[4] Z. Komar and C. Wolek. *Inzynieria ruchu drogowego. Wybrane zagadnienia*. Wydawnictwo Politechniki Wroclawskiej, Wroclaw, 1994.

[5] H. L. Liu, J.-S. Oh, and W. Recker. Adaptive signal control system with on-line performance measure. *In 81st Annual Meeting of the Transportation Research Board*.

[6] F. Miaoqing and W. W. Foong. Investigate traffic flow using cellular automata. *An USC3001 report 2003/2004*.

[7] Z. Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs third edn*. Springer, 1996.

[8] K. Nagel and M. Schreckenberg. Traffic jam dynamics in stochastic cellular automata. *95ATS089*, 1997.

[9] K. Nagel and P. M. Simon. Simpliffied cellular automaton model for city traffic. *Phys. Rev. E*, 58:1286, 1997.

[10] K. Nagel, D. E. Wolf, P. Wagner, and P. M. Simon. Two-lane traffic rules for cellular automata: A systematic approach. , *Phys. Rev. E*, 58(2), Aug. 1998.

[11] I. Rechenberg. Evolution strategy: Nature's way of optimization. in bergmann, editor, methods and applications, possibilities and limitations. *Lecture notes in Engineering*, pages 106–126.

[12] K. Tavladakis and N. C. Voulgaris. Development of an autonomous adaptive traffic control system. *In ESIT '99 - The European Symposium on Intelligent Techniques*.