Evolutionary Swarm Traffic: *If Ant Roads Had Traffic Lights*

Ricardo Hoar **Joanne Penner** **Christian Jacob**
Department of Computer Science, University of Calgary
2500 University Drive N.W., Calgary, T2N 1N4, CANADA
hoarr | pennerj | jacob@cpsc.ucalgary.ca

Abstract — **Traffic congestion has become a major concern for many cities throughout the world. Consequently, simulations can provide helpful tools for engineers to plan traffic systems. In this paper we present SuRJE, a *swarm-based* traffic simulation system. All interactions among individual drivers are implemented through the dropping and sniffing of pheromones on a 2-dimensional map. Furthermore, SuRJE optimizes traffic light turn and time sequences using an evolutionary algorithm and *swarm voting*, which helps to speed up the adaptation process for the traffic light settings.**

I. TRAFFIC SIMULATION

Traffic congestion decreases the standard of living for many people. Cars stuck in traffic add to pollution by idling needlessly. People value their time, and being stuck in traffic can be emotionally stressful; drivers may act more aggressively and impatiently, or may become tired and distracted. The varying attitudes of drivers along with the close proximity of cars in congested traffic lead to an increasing amount of accidents. By making traffic systems more efficient as a whole, each person who relies on some form of automobile transportation, as well as entire communities, will benefit.

A. *Emergent Traffic Dynamics*

The design of infrastructures for automobile traffic poses many challenging problems. One of the key issues is that the individual behaviors of drivers of automobiles are not as easily manageable and predictable as for mass transportation systems, such as public buses and trains. Drivers act as individual agents, orchestrated only through traffic rules, signs, and traffic lights, in particular. This makes predictions of the overall traffic dynamics challenging, as most traffic control as well as its short- and long-term planning today rely on models of global dynamics derived from statistical properties of observed traffic. Most of the current models, however, do not allow to simulate traffic dynamics as emergent patterns arising from the interactions of a large number of individual drivers.

B. *Swarm Traffic and SuRJE*

We propose a decentralized, agent-based approach to traffic simulation and planning, which allows to develop traffic systems from the bottom up: starting from a collection of individual drivers, that follow specific behavioral rules, the interaction of these individual agents are simulated on a road map, thus leading to emergent patterns of traffic dynamics. *SuRJE* (*S*warms *U*nder *R&J* using *E*volution) is a swarm-based traffic simulation environment designed for use in many areas of traffic system development, from the layout of road infrastructures to evolutionary optimization of the direction of traffic flow.

In *SuRJE*, the traffic swarm exists in a two-dimensional graphical environment, where the user can interactively manipulate control parameters of the individual car agents that comprise the swarm. A road map can be interactively built and navigated through multiple distribution zones. Roads are connected through various types of intersections, enhanced by traffic signs and lights (Fig. 1). The traffic swarm drives through this environment, with each agent obeying traffic laws and avoiding collisions. Based on statistics, collected from the car agents, evolutionary design and swarm voting is used to adjust the timing sequences of the traffic lights, in order to avoid congestion and minimize the drivers' overall waiting times.

II. SWARM INTELLIGENCE

The key idea of swarm intelligence is based on emergent behavior patterns observed among highly parallel and decentralized systems, such as social insects. Societies of social insects survive as a group by following simple, local rules by which they interact with their environment. Colonies of ants, for instance, exhibit exceptional teamwork and problem-solving abilities [2], [3]. Interestingly enough, their communication occurs only by relatively simple means, through *stigmergy*. That is, individual ants leave signs in the environment, e.g. in the form of pheromones, which other ants can use for their individual decision making. Although in such a decentralized system each agent acts independently, behaviors do emerge as though there is a collective intelligence, seemingly orchestrated through a central control instance [5].

A. *Cars Like Ants*

The cars in *SuRJE* are agents with given start and destination neighborhoods, which react to local circumstances and traffic laws on their way towards their destinations (Fig. 1). Each car attempts to travel at its desired speed ratio s_{ratio}, but adjusts to the various cars around it. Therefore, local rules allow the cars to act independently while following a simple algorithm to arrive at their destination (Fig. 2).

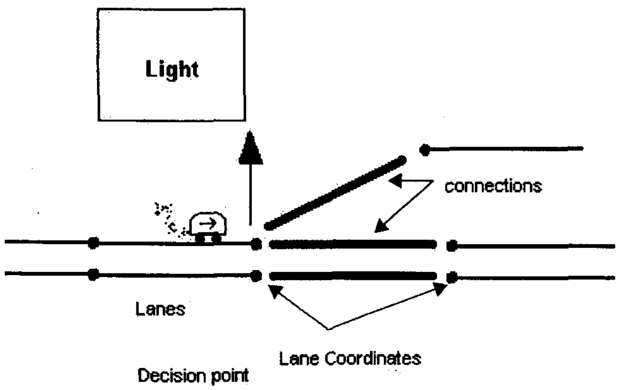

Fig. 1. Environment visible to a car on a lane and intersection.

```
procedure Car::Drive()
  int smell // value of strongest pheromone in front of the car
  int distance // distance to upcoming intersection
begin
  if ¬arrived then smell := sniffPheromone(pos_head, Lane)
  if smell > P_tail then
     DEC := true; s_goal := s_curr - (s_curr / (MAX_SCENT - smell))
  else if smell < P_tail and s_curr < (s_ratio * Lane.speedlimit()) then
     ACC := true; s_goal := s_ratio • Lane.speedlimit()
  if distance < D_check then
     if ¬decided then
        MakeDecision(upcoming intersection); decided := true
   if the upcoming rule requires me to stop then
     DEC := true; Sgoal := 0; Δ_dec := s_curr / distance
  if s_curr > Lane:speedlimit() * MIN_SPEED_RATIO
     then DropSinglePheromone(pos_tail, Lane)
     else DropPheromoneTrail(pos_tail, Lane)
  Proceed()
end

procedure Car::Proceed()
  int dx, dy // x,y displacement of car
begin
  if DEC then
     s_curr := s_curr - Δ_dec
     if s_curr ≤ s_goal then DEC := false
  else if ACC then
     s_curr := s_curr + Δ_acc
     if s_curr ≥ s_goal then ACC := false
  dx := Lane.vector()[0] * s_curr
  dy := Lane.vector()[1] * s_curr
  UpdateCarPosition(dx,dy)
end
```

Fig. 2. Pseudo code for a car agent. ACC/DEC: accelerating/decelerating; Δ_{acc} / Δ_{dec}: ac-/deceleration rate; s_{curr}: current speed; s_{goal}: goal speed; s_{ratio}: preferred speed ratio (per speedlimit); P_{tail}: preferred pheromone strength for tailing; D_{check}: preferred rule checking distance; decided: decision on which connection to take at an intersection; pos_{head}/pos_{tail}: car head/tail position; MAX_SCENT: strongest pheromone a car drops; MIN_SPEED_RATIO: minimum speed ratio.

B. *Coordination Through Pheromones*

We have adopted the stigmergy paradigm of communication through pheromones for *SuRJE* [1], [2], [3]. Each car in the system drops and smells pheromones. Using a swarm system approach provides the ability of agents to communicate both directly and indirectly. Some agents may have information that will aid other agents in their decision-making. Constraint interactions between agents in a local area are by far more frequent than interactions on a more global scale. For modeling traffic, local communication between agents is important for avoiding collisions in traffic situations, including tailing, changing lanes, and crossing intersections.

In *SuRJE*, all communication between agents is done through dropping scents (pheromones) on the road, which fade over time and are detectable by all cars (Fig. 3(a)). As a car's speed varies, so does its trail of pheromones. Faster cars leave longer, more spread-out trails, whereas slow cars leave shorter trails. Additional pheromone dropping behavior has been added for circumstances where cars do not leave sufficient trails behind, such as when a car is stopped, decelerates quickly, or changes lanes (Fig. 3(b), (c)). This is analogous to real drivers using signal and brake lights to inform other drivers about their actions, and performing checks before changing lanes or turning. This method of communication allows a reduction of the computation costs for collision detection by localizing the area in which a car sniffs pheromones. Since a car is always either in a lane or in an intersection, scents can be dropped and managed there, whereas cars remain free to examine signs and messages in their local environment.

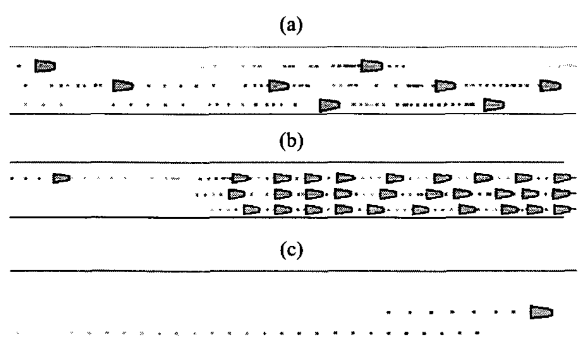

Fig. 3. Car interactions by pheromones: (a) cars on a 3-lane road, dropping pheromones; (b) cars stopped at a light, emitting pheromones behind them; (c) a car emitting pheromones while changing lanes.

III. EVOLUTIONARY DESIGN

The goal of the evolution is to minimize the average waiting time of all cars. The lights in a map are operated based on timing sequences for the colors red, green, and yellow (Fig. 4).

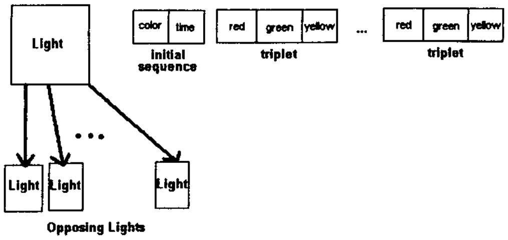

Fig. 4. Data structure for traffic lights.

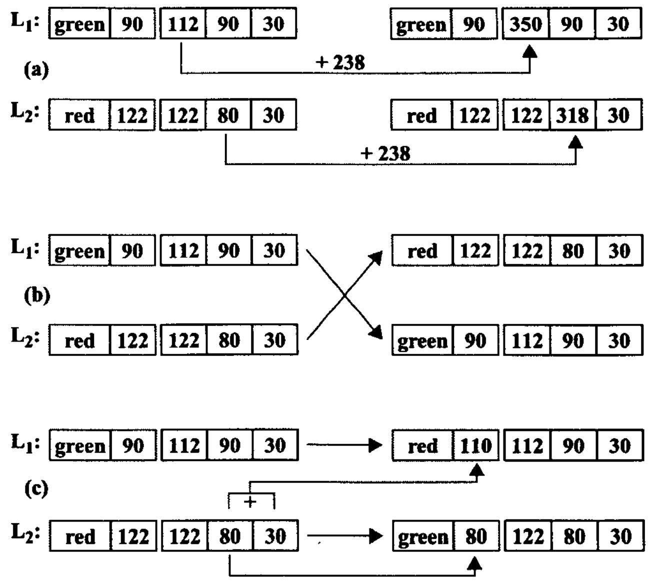

Fig. 5. Mutations on the light sequences.

Each traffic light is attributed with an initial color, its duration, and a sequence of red-green-yellow timings. Traffic lights are also aware of any "opposing" lights that have to be kept in synchronization with each other, such as the two corresponding pairs of lights at a 4-way intersection.

A. *Fitness Function*

Each car keeps track of two things throughout its journey from its seed point to its destination: the total driving, d_i^{tot}, and waiting times, w_i^{tot}. At the end of a simulation cycle, the fitness measure σ_i for each car i is calculated as

$$\sigma_i = \frac{w_i^{tot}}{d_i^{tot} + w_i^{tot}}.$$

Depending on the traffic light timings, the overall traffic congestion is then evaluated by $\Delta_{wait} = \sum_{i \in Cars} \sigma_i$, where Δ_{wait} should be minimized.

B. *Mutation Operations*

The goal of the evolutionary search is to minimize the overall waiting time by adjusting the timing sequences of the traffic lights. Currently, we are using three evolutionary mutation operations on the traffic light data structures (Fig. 4), which are discussed in the following sections. It is important to note that any mutation also has to take all the opposing lights into account, in order to ensure that no two lights will be green at the same time at a shared intersection.

Extend / Decrease Green Time

This mutation is the major operator and causes the most significant changes (Fig. 5(a)). In the example, the red time, $\Delta_{red} = 112$, of light L_1 is increased by $\Delta = 238$, such that the new duration for red is $\Delta_{red} + \Delta = 350$. To prevent both lights from turning green at once, the green timing for the opposing light, L_2 is adjusted to $\Delta_{green} = \Delta_{green} + \Delta = 318$.

Swap Two Timing Sequences

This operation implements a crossover of timing sequences between two lights, with an increased probability for lights that control the same intersection (Fig. 5(b)).

Reassign the Starting Sequence

The reassignment operator changes the initial order in which the lights turn green while leaving all the timings unchanged. However, the timings of opposing lights have to be adjusted accordingly. In our example (Fig. 5(c)), light L_2 would change to start off with an initial green light, instead of red. Therefore, light L_1 must show a red light during the green and yellow phase of L_2. This mutation allows lights to synchronize together by changing their lighting order.

C. *Swarm Voting*

Swarm voting is introduced to initiate specific adaptations more quickly by forcing the cars to vote for lights that might stop them on their way. A vote is cast at each time step for an upcoming light for each car that is stopped. This means that lights with a higher number of votes cause more delays than lights with only a few votes. This weighting system assigns higher probabilities for mutation to lights with the most votes to (1) increase their green period or (2) to reduce the green period for one of their opposing lights. With swarm voting, adaptations to changing traffic dynamics occur more quickly, thus the optimization results are more refined and achieve better timings (see Section V).

IV. WORKING WITH *SuRJE*

SuRJE provides a design environment to build, test, and optimize traffic scenarios interactively. The user may adjust and control parameters that immediately affect the behavior of the traffic seeding, density, and flow, as well as the evolution of the light sequences.

Map Building. In map building mode, the user can create and edit maps with multi-lane roads, connections, lights and various signs. Road control points (Fig. 1) can be placed anywhere in 2D space, and are then linearly interpolated to form segments. The user creates roads with multiple points and approximate curves. Traffic control lights are then designated to one or more lanes interactively. The user can model a dynamic range of traffic light configurations similar to those in real-world setups.

Car Seeding. An essential feature of *SuRJE* is the ability to specify where and when cars begin and end their journey. This is important, as it enables accurate simulation of a city's unique traffic patterns. *SuRJE* allows the user to create and select highlighted regions on the map, and input the rate at which cars are seeded into the system. The user can also define a list of possible destination regions with associated probabilities, which seeded cars use to determine their destination. This system allows city planners to input statistical data when modeling real situations to test the throughput of traffic flows.

Lights Setting. The user can initialize the lights to a default timing sequence or load a timing file. When the simulation is started cars are seeded according to the aforementioned distribution and travel down the roads following the posted speeds and obeying the posted rules and lights.

Evolutionary Adaptation. Once a map and distribution are input, traffic light sequences can be evolved for that specific scenario. The user loads a predefined sequence for the lights or randomly creates new settings, following a $(\mu + \lambda)$ evolution strategy (ES) [4]. That is, $\lambda = 10$ offspring traffic light settings are generated using the mutation operations described in Section III(B). For each new sequence, its fitness is evaluated by running the traffic simulation for $N = 1200$ time steps, resulting in a particular average waiting time Δ_{wait} for the cars. Finally, among the $\mu = 1$ parent and the λ offspring light sequences, the one that causes the least average waiting time survives into the next generation, where it serves as the new parent solution.

Throughout the simulations the user retains control of all parameters and settings. The speed of the simulation can be adjusted interactively as well as parameters relating to the cars' driving behavior. While evolving a sequence, the user may re-adjust the probabilities of mutation operators, change mutation stepsizes, as well as enable or disable swarm voting.

V. RESULTS

A. SWARM VOTING VS. NORMAL MUTATION

The fitness function used by the evolution engine determines the winner of each round, however, the manner in which the mutations occur is based on random changes to the system as a whole. In order to make more efficient use of the swarm, each car in *SuRJE* casts a vote for the nearest upcoming light whenever it is idling. These votes then determine the probabilities by which lights are mutated. Lights with a large percentage of the votes are more likely to increase their green or decrease their opposing lights' green time, and are also more likely to mutate their initial color and timing.

Both swarm voting and regular mutation cause the timing adaptations to improve over time. In the case of swarm voting, stabilization of the light adjustments occurs after a relatively short number of generations, which is particularly important for an interactive evaluation of the results (Fig. 6). Swarm voting also allows the system to reach a better final sequence, as it results in mutations focussing on lights that, on average, cause the most traffic delays.

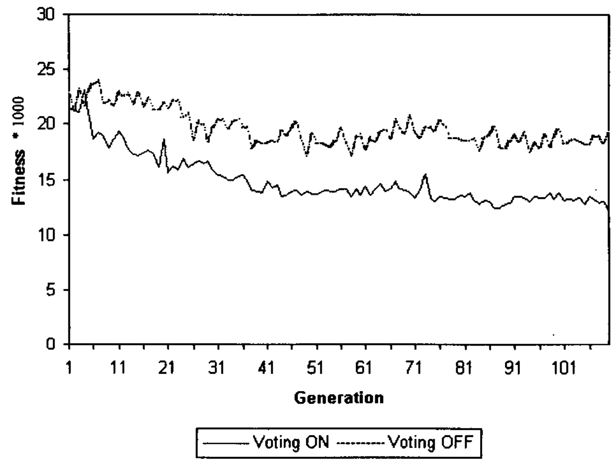

Fig. 6. Normal mutation vs. mutation with swarm voting for the *Swarm Boulevard* traffic scenario (see Section D).

B. SEED REGION PLACEMENT AND SEED RATE

A car seed region contains a list of all the control points within its rectangular bounds. Different strategies of region placement produce different results as cars are seeded randomly at these points within a region. Larger regions covering many points produce distributions similar to those of large neighborhoods. Conversely, a region containing only a single point is excellent for seeding a controlled flow of cars onto a road.

The introduction of these variances might lead to unlikely distributions that favor a specific light timing sequence on a single occasion. Therefore, a plus evolution strategy, $(1 + 10)$-ES, is used (see Section IV), where the best light timing sequence of the previous generation, the parent solution, must recompete with it's offspring solutions. Each time a light sequence is run, different random numbers are used. This strategy leads to better overall timing sequences, because the lights adapt to dynamic traffic flow rather than rely on a predetermined flow pattern.

C. *THE STRAIGHT ALLEY TESTBED*

The *Straight Alley* map in Fig. 7 is designed as an easy testbed scenario consisting of a two-lane straight alley, which intersects three times with another two-lane road. Cars do not need to make decisions, as they only follow their road, starting from their seed points marked as A, B, C, and D. Traffic lights are at positions 1 through 6, where lights 1, 3, and 5 control the straight lane traffic, and lights 2, 4, and 6 regulate the traffic flow on the winding side road.

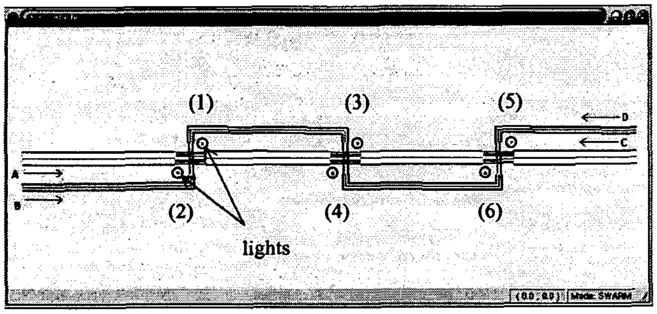

Fig. 7. *Straight Alley* map: 200 cars are seeded into the system in both directions on each road with the rates as in Fig. 9. Cars are seeded into the system at points A, C (straight alley) and B, D (winding road).

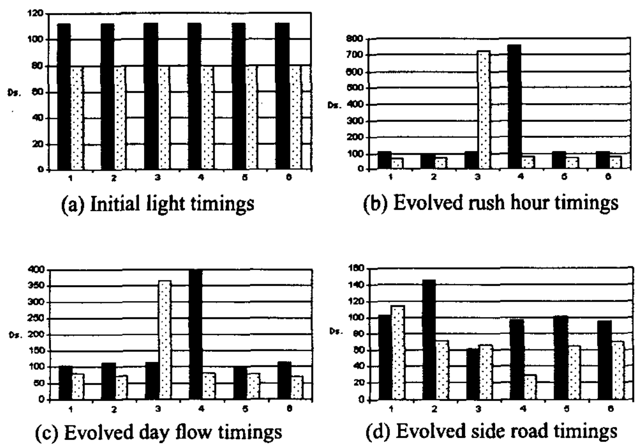

Fig. 8. Initial and evolved timings for the 6 traffic lights in *Straight Alley* with car seedings as described in Fig. 9

We have tested evolutionary light adaptations for three different traffic flow scenarios (Fig. 9). The first scenario simulates a normal day-time traffic distribution, where we assume to have 90 percent of the traffic flow in both directions along the main alley (seed points A and C). Initially, the timings are the same for all traffic lights (with a green/red ratio of 80/112) (Fig. 8(a)), which causes extremely uneven traffic flow on both roads. After 15 generations, the timings for lights 3 and 4, which both control the second intersection, have evolved towards an almost four times longer green phase for the main alley traffic (light 3). Consequently, the traffic flow on the main alley between intersection one and three is increased, because there is less delay at light 3. All the other lights have kept their lighting sequences almost unchanged.

A similar light sequence adaptation is observed for the rush hour scenario, where 60 percent of the traffic enters the system from seed point A, which is designed to test whether the system also can deal with non-symmetric traffic flows. The light timing for green is increased seven times, compared to the initial settings (Fig. 8(c)).

In the third scenario, 60 percent of the traffic originates from locations B and D, travelling on the winding side road. This causes the system to evolve a 40 percent increase of the overall traffic flow.

Flow	Car Seedings (rate / sec.)				Δ_{wait} Gen. 0 – 15		α
	A	**B**	**C**	**D**			
Day	0.45	0.05	0.45	0.05	44.0	14.8	66.4%
Rush	0.62	0.15	0.20	0.03	38.5	20.5	46.6 %
Side	0.20	0.30	0.20	0.30	28.9	17.5	39.4 %

Fig. 9. Overview of car seedings for the *Straight Alley* scenario for (a) normal day traffic, (b) rush hour traffic, and (c) increased traffic on the side road. Δ_{wait} denotes the overall waiting time of all the cars in the system. α is the improvement compared to the initial light timing, after 15 generations.

D. *SWARM BOULEVARD*

The *Swarm Boulevard* scenario, like *Straight Alley*, contains three intersections with eight traffic lights (Fig. 10(a)). There are three lights at intersections (1) and (2), and two lights at intersection (3). These intersections are more complicated, as cars have to make decisions whether to go straight or turn (Fig. 10(b)).

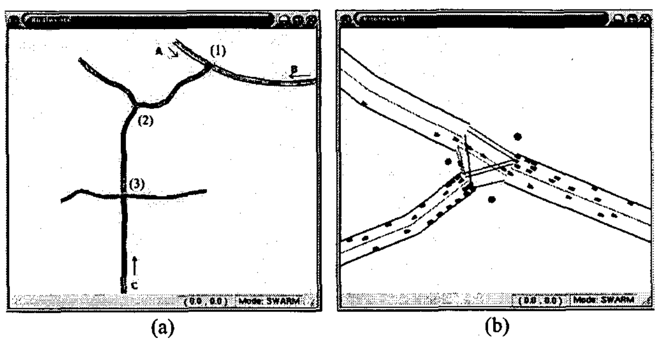

Fig. 10. *Swarm Boulavard* map: about 200 cars are seeded into the system at locations A, B, and C with the following rates per second. (A) 0.45, (B) 0.45, (C) 0.10.

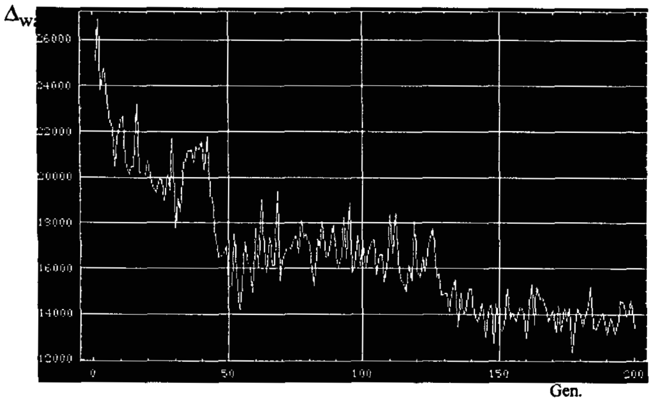

Fig. 11. Decrease of the overall waiting time Δ_{wait} for *Swarm Boulevard.*

With 90 percent of the cars originating from seed locations A and B, a heavy North to South flow of traffic is generated. The fitness dynamics over 200 generations of the evolutionary traffic light adaptations are shown in Fig. 11.

E. LOOPTOWN

The *Looptown* scenario is considerably more complicated than the previous two test cases. *Looptown* has 9 intersections with 28 lights. The distribution of cars simulates main traffic flow on the outer ring road with all seeded cars originally starting in a counter-clockwise direction. As cars are free to decide where to go at each intersection, traffic flow eventually emerges in both directions on the ring road. Additional traffic is finally also building up at the inner roads and intersections.

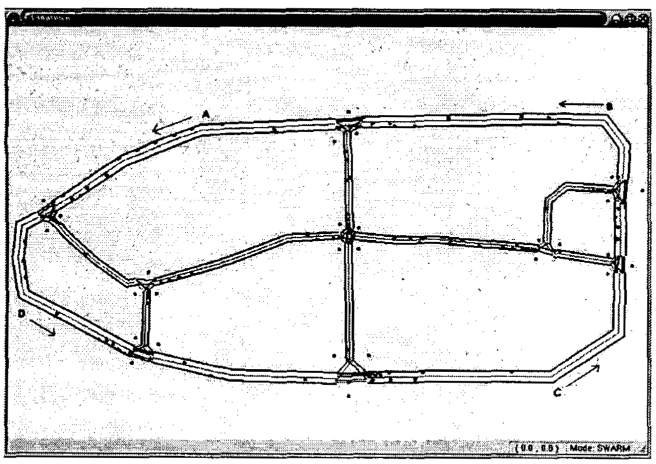

Fig. 12. *Looptown* map (28 lights at 9 intersections): 300 to 500 cars are seeded into the system at four locations with the following rates per second: (A) 0.23, (B) 0.31, (C) 0.23, (D) 0.23.

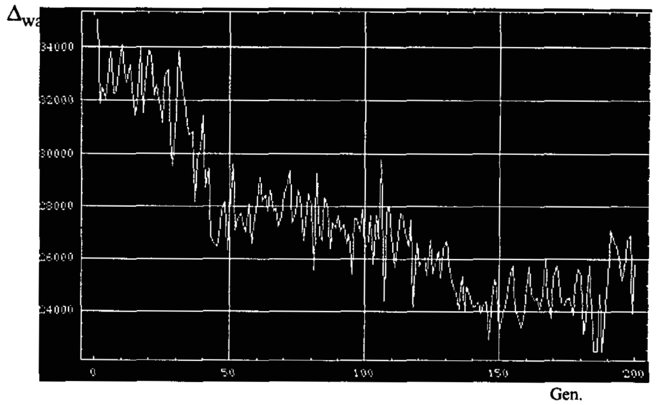

Fig. 13. Decrease of the overall waiting time Δ_{wait} for *Looptown.*

As evident from the fitness graph in Fig. 13, adaptation of improved traffic light coordinations is more demanding, but still leads to an improvement of 26 percent for the overall waiting time Δ_{wait}, after evolution over 200 generations.

VI. CONCLUSION AND FUTURE WORK

Based on communication through pheromones, *SuRJE* provides a surprisingly accurate model of car traffic dynamics, including congestion, lane changes, yielding, and free flowing traffic. The evolutionary engine supports real-time adaptation of traffic light timings. The next version of *SuRJE* will take the following aspects into account: (1) increased intelligence for each car, to make more accurate decisions where and how to navigate through the road map; (2) the database of car behaviors will be extended by rules for right turns on a red light or turns against oncoming traffic, for example; (3) both the light data structures and the set of mutation operators will be extended to allow for variations closer to rules that human traffic designers and planners use; (4) we will explore a parallel version of *SuRJE* over a distributed network and, finally, (5) the possibility of modelling real city traffic.

REFERENCES

[1] Bonabeau, E., et al., *From Natural to Artificial Swarm Intelligence*, Oxford University Press, Oxford, 1999.

[2] Gordon, D., *Ants at Work*, The Free Press, New York, 1999.

[3] Hölldobler, B., and Wilson, E. O., *The Ants*, Harvard University Press, Cambridge, MA, 1990.

[4] Jacob, C., *Illustrating Evolutionary Computation with Mathematica*, Morgan Kaufmann, San Francisco, 2001.

[5] Resnick, M., *Turtles, Termites, and Traffic Jams*, MIT Press, Cambridge, MA, 1994.