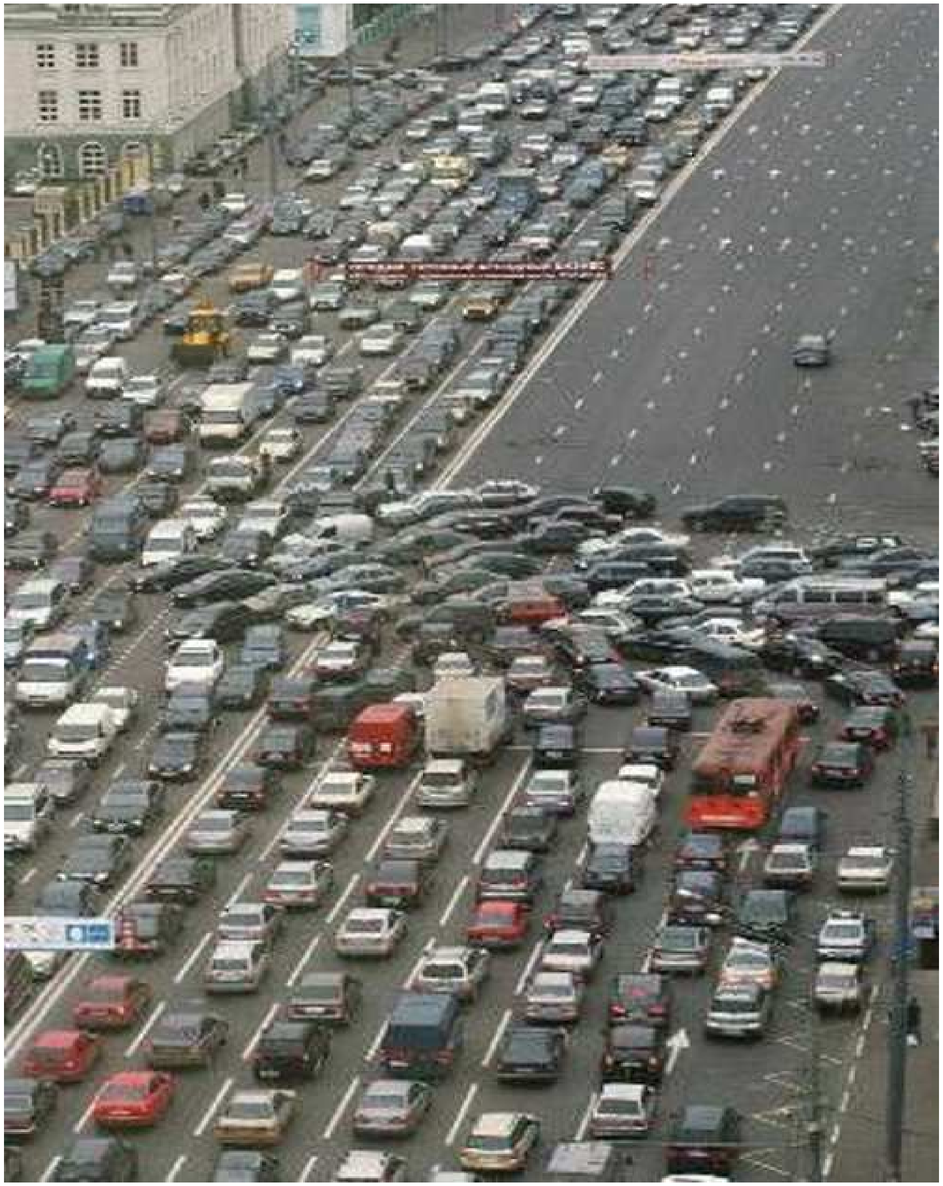
Intelligent Traffic Signal Control

Dave McKenney
Carleton University
School of Computer Science

Presentation Layout

- Problems caused by traffic
- Outline of a traffic model
- Previous Work
- Initial algorithm, tests, and results
- Moving to SUMO traffic simulator
- Modeling a real world example
- Improvement ideas
- Future Work

Why is Traffic a Problem?



http://spearfishnews.com/wp-content/uploads/2009/04/spearfish-traffic-jam.jpg

Problems Caused by Traffic

- Economic, Social, Environmental, Safety issues
- 37 Million hours spent commuting daily in Thailand[1]
- Avg. time spent commuting in Toronto area has increased 16% in the last 10 years[2]
- 21 Million hours spent a day commuting in UK (£226 Million working time lost)[3]
- Wasted fuel has large economic and environmental impact

Traffic Model

- Several things can be controlled/observed:
  - Traffic Signals
  - Vehicle Routes
  - Driver/Vehicle Behaviour
  - Roads/Streets/etc.

Traffic Signals

- Can store known information
- Can receive/calculate traffic information
- Each light implements a signal plan, composed of cycles of the light phases (green, yellow, and red)
- Global (entire traffic network) optimization is difficult

Vehicle Routes

- Real-time vehicle routing is now extremely common (GPS devices)
- Correct strategies could route drivers efficiently:
  - Balance traffic flow
  - Alleviate traffic jams
  - Avoid accidents and other road blockages
  - Take the fastest path, not just the shortest

Driver/Vehicle Behaviour

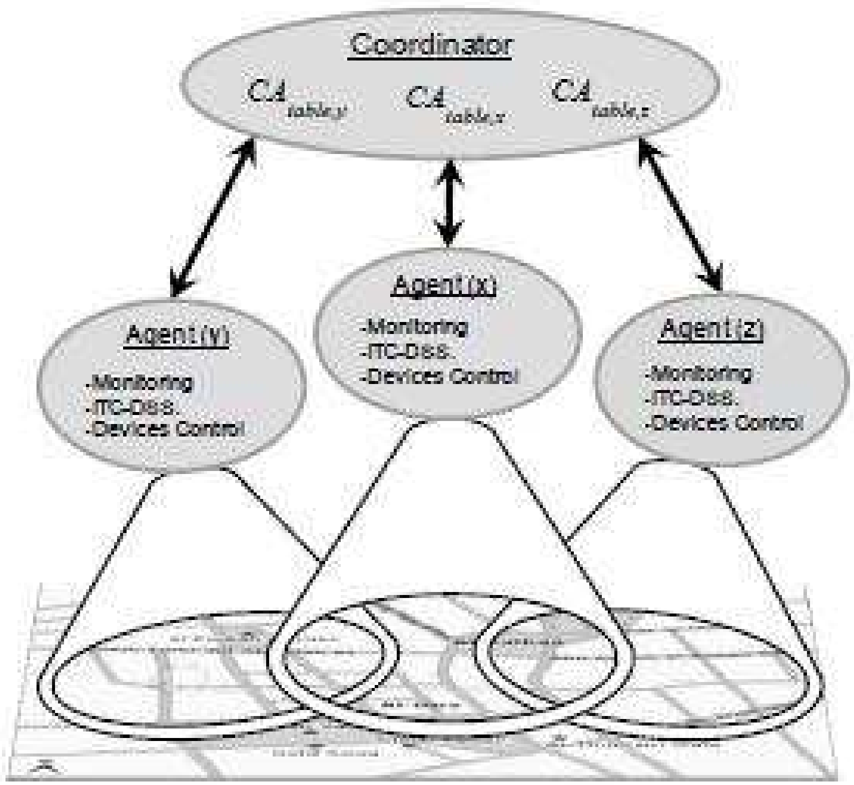
- Real world drivers act differently
- A traffic model must represent this in some way

Roads/Streets/Etc.

- Each roadway has specific attributes
- These attributes are generally static

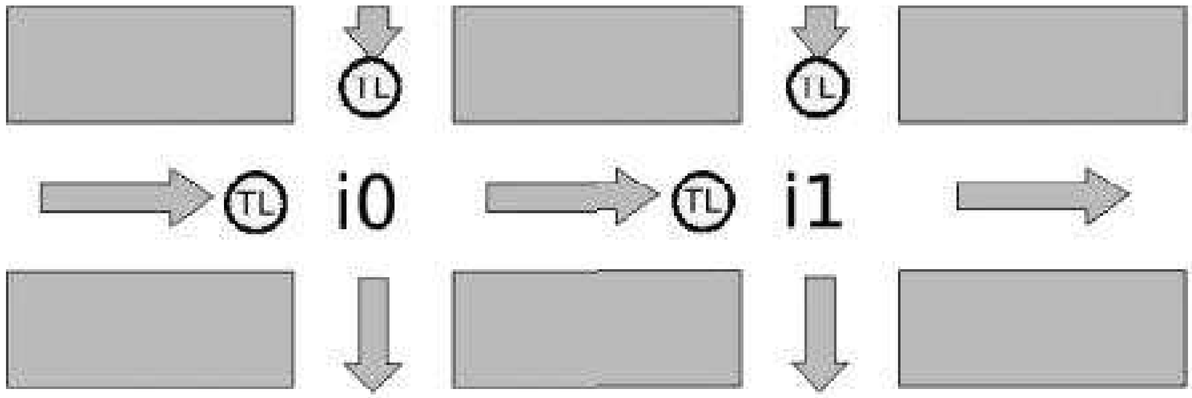
Hierarchical/Coordinator Approach

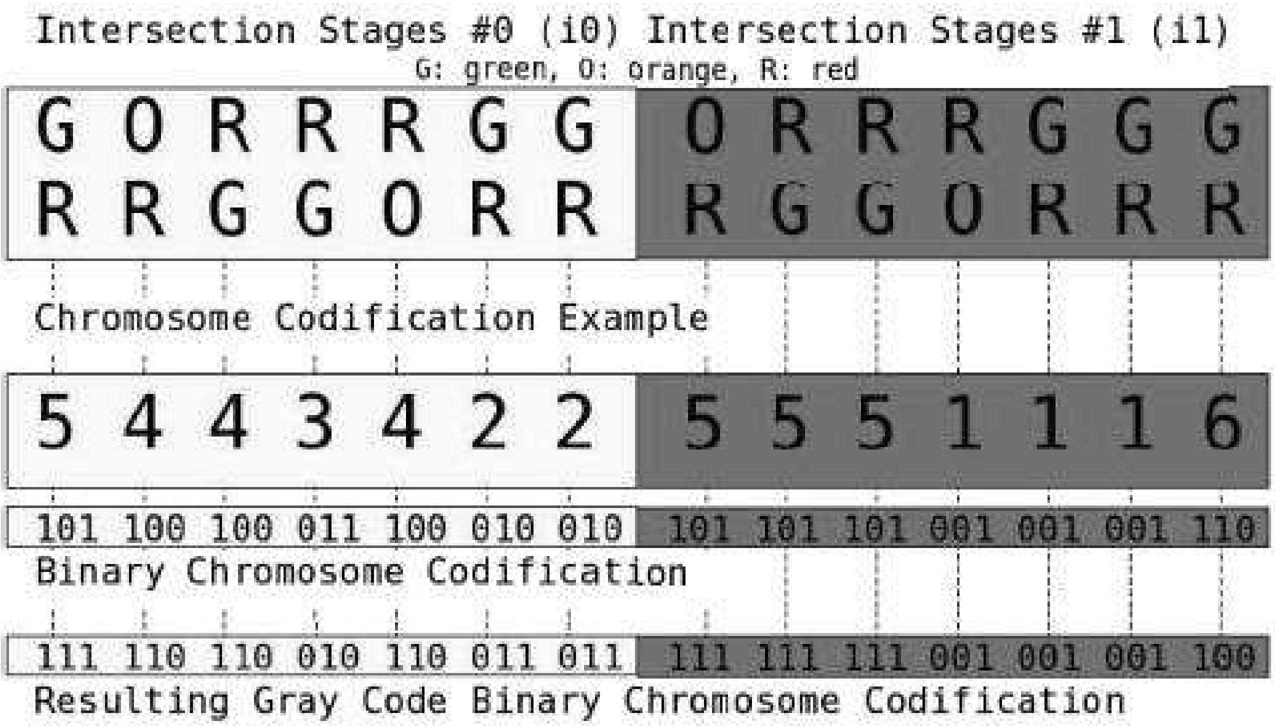
- Almejalli et. al[4] implemented a system which uses a coordinator agent to help find a global optimum among all agents within the system



An Intelligent Multi-agent Approach for Road Traffic, Almejalli et al.

Genetic/Evolutionary Approach[5]

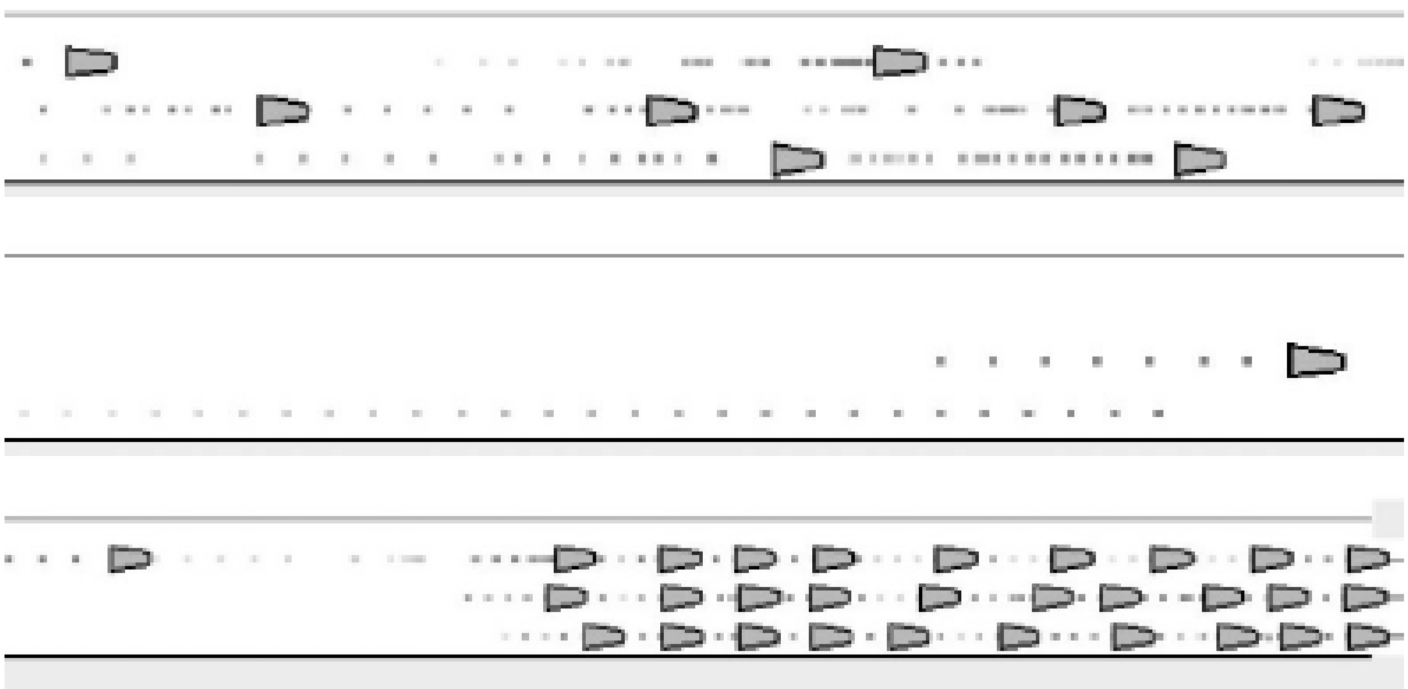




Applying a Traffic Lights Evolutionary Optimization Technique to a Real Case, Sanchez et al.

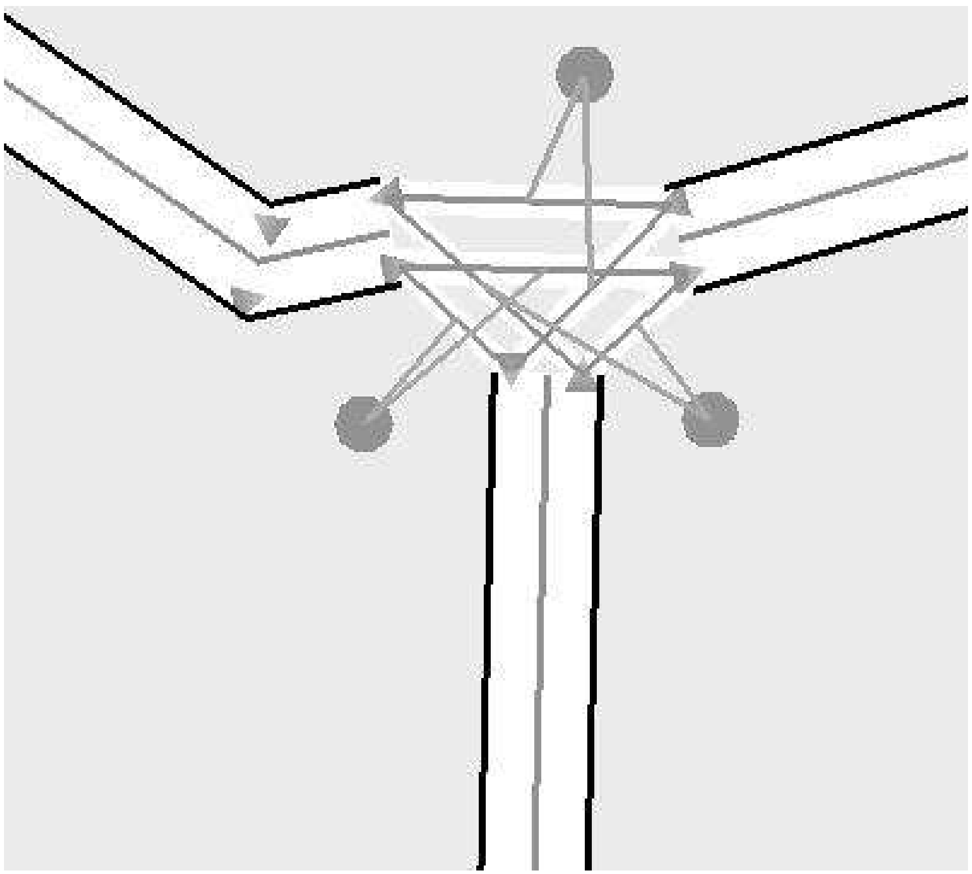
SuRJE Traffic Simulation[6]

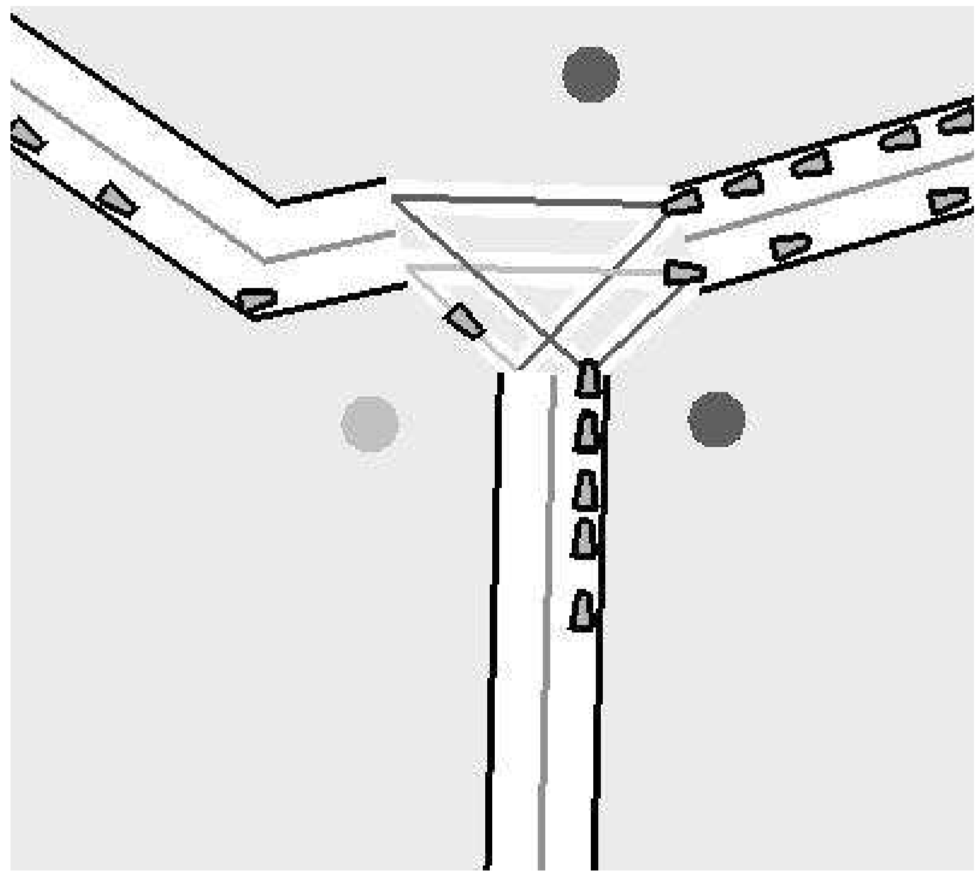
- Uses a swarm model to simulate traffic
- Vehicles leave pheromone as they drive
- Pheromone detection is used to chose actions



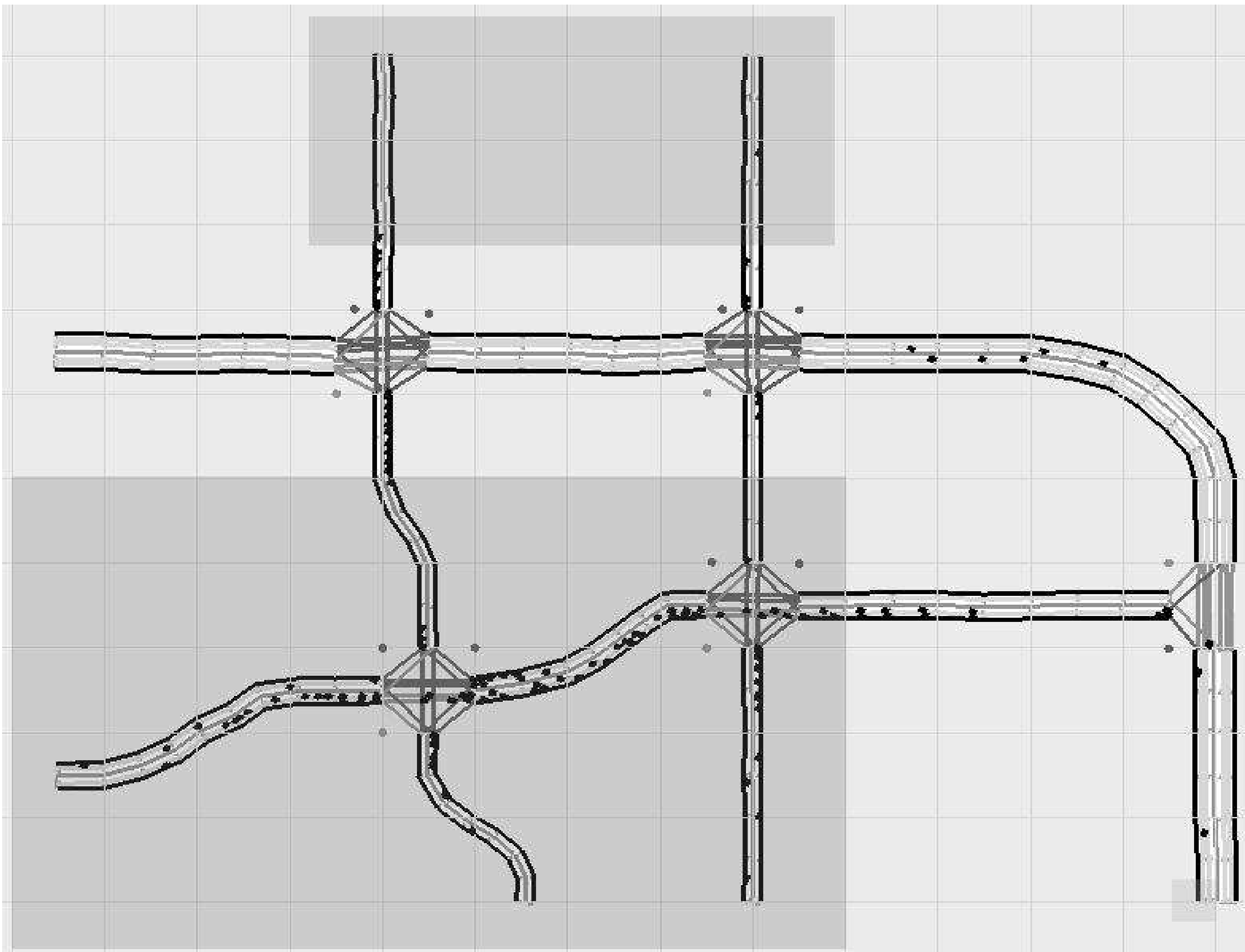
Swarm-based Traffic Simulation with Evolutionary Traffic Light Adaptation, Penner and Hoar

Creating a SuRJE Simulation

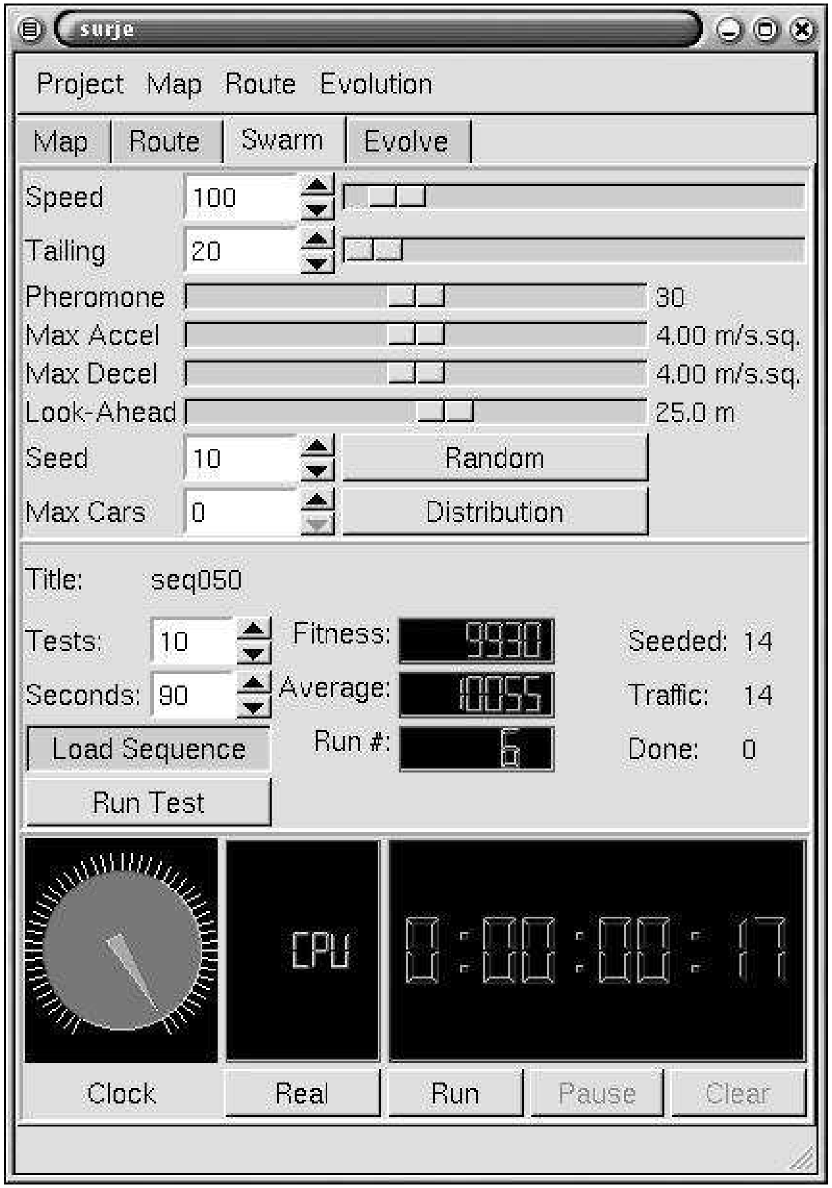


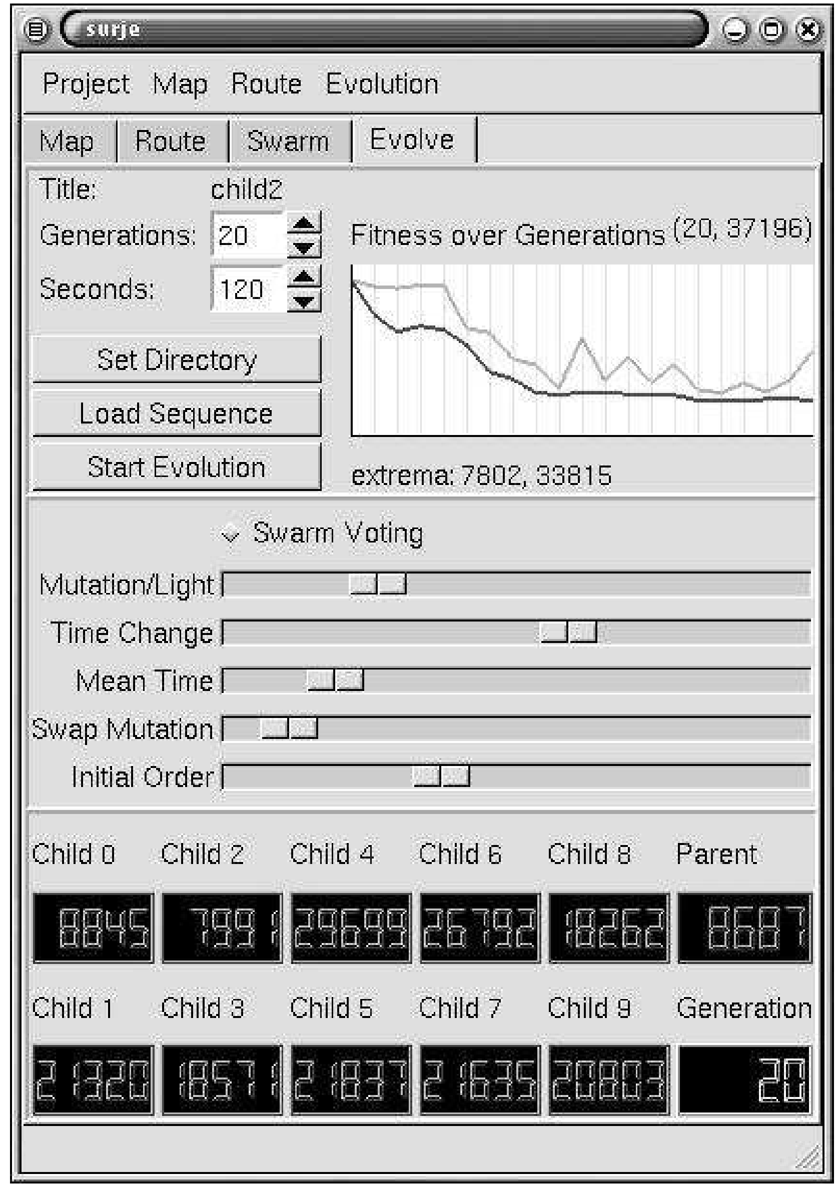


Creating a SuRJE Simulation



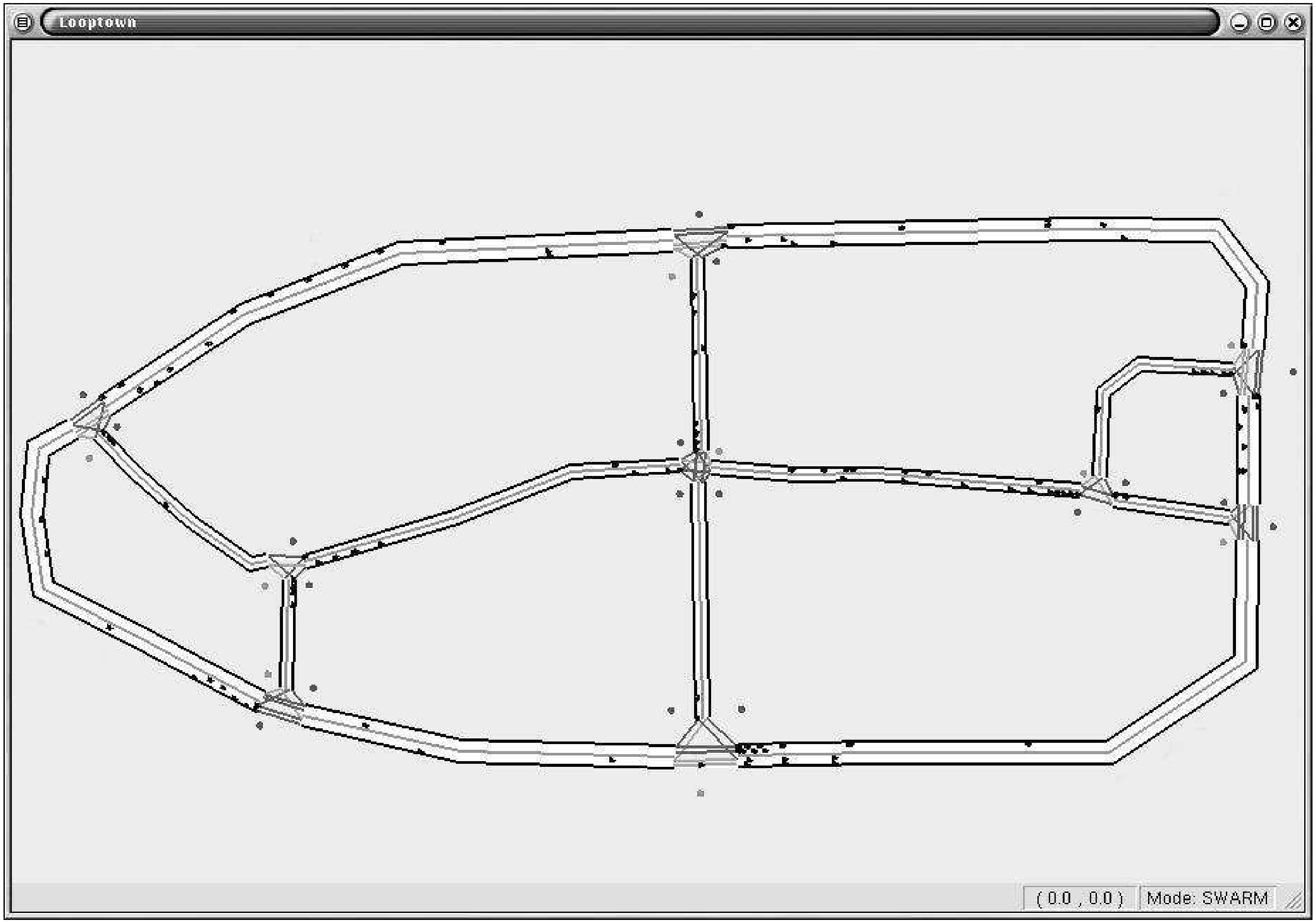
Running a SuRJE Simulation



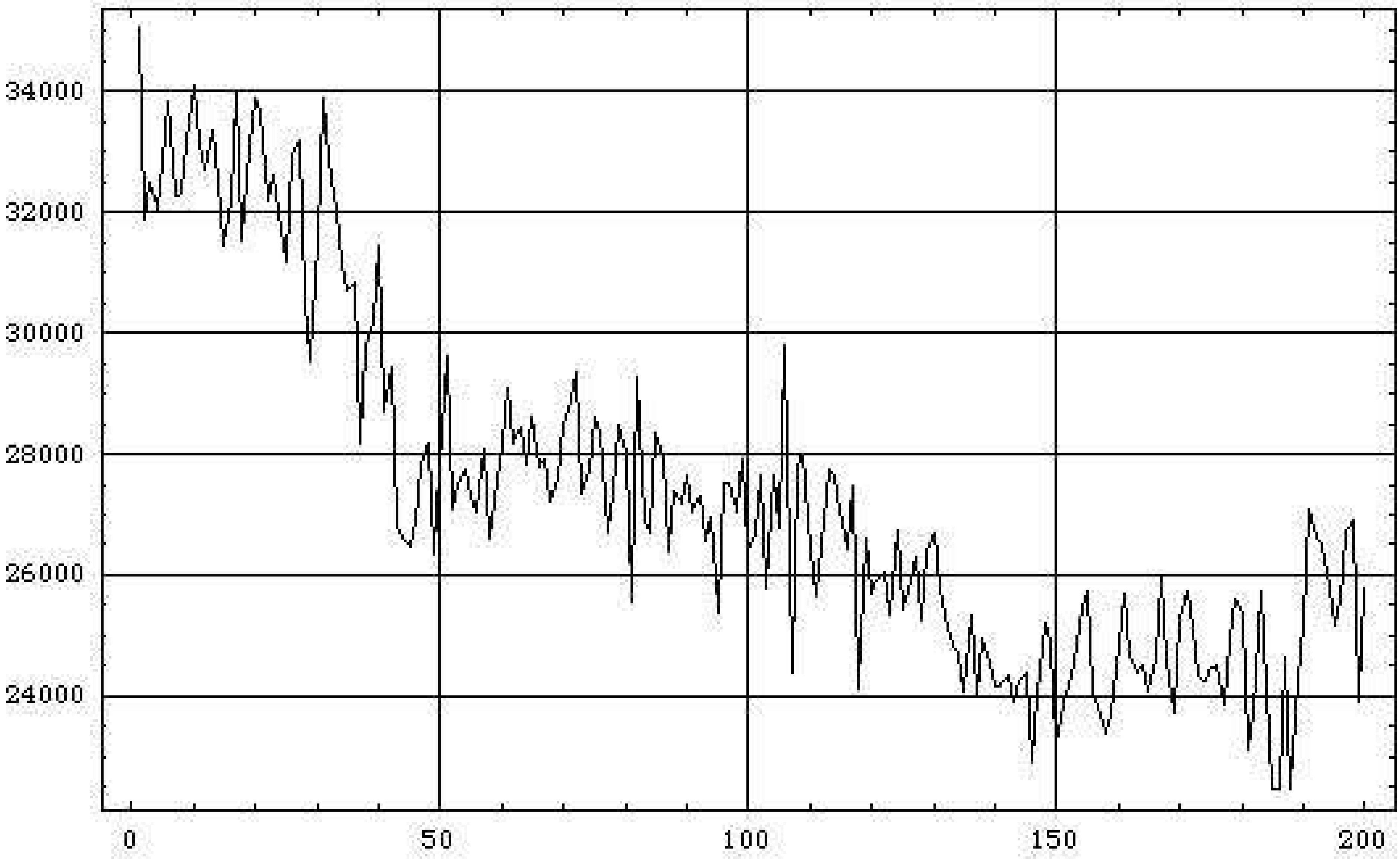


Swarm-based Traffic Simulation with Evolutionary Traffic Light Adaptation, Penner and Hoar

SuRJE Simulation Results



SuRJE Simulation Results



Swarm-based Traffic Simulation with Evolutionary Traffic Light Adaptation, Penner and Hoar

Bazzan and de Oliveira[7] - Traffic Signal Group Formation

- Treat each intersection as a social insect
- Each 'insect' attempts to optimize their signal plan
- Pheromones left by vehicles motivate change in the behaviour of the 'insect'
- Goal is to create non-stop flow of traffic through sections of road in a certain direction

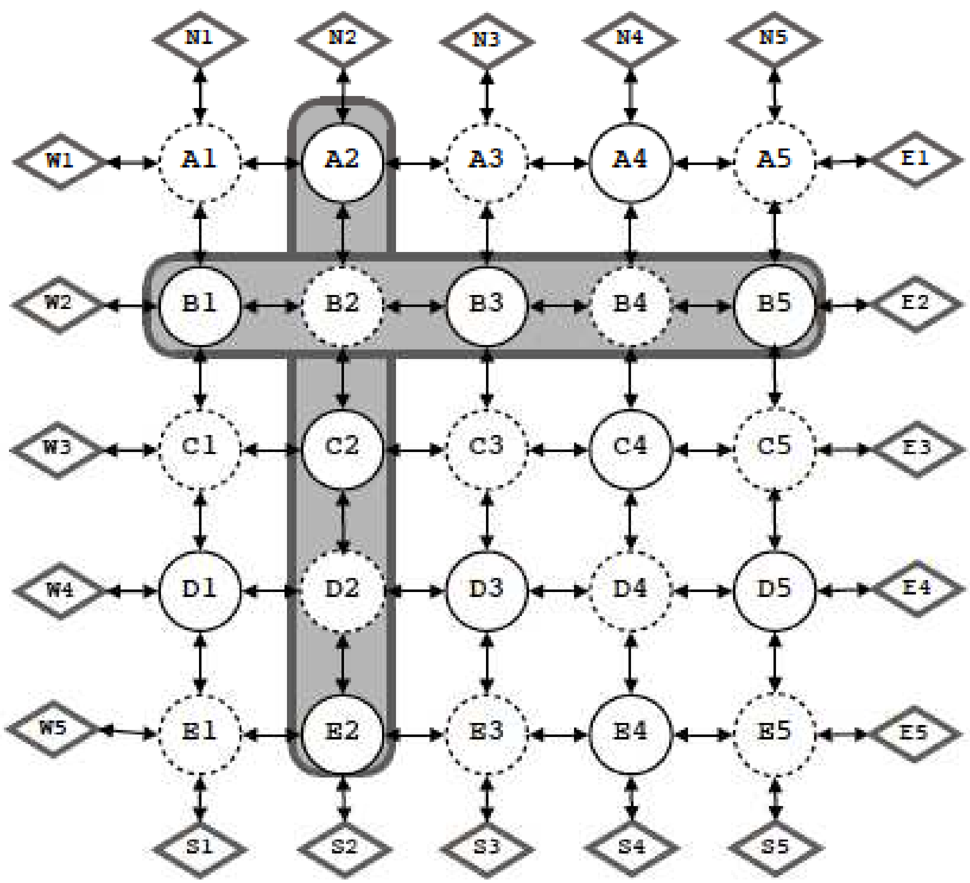
Traffic Signal Group Formation

- All vehicles waiting at an intersection drop pheromone at every time step
- The amount of pheromone in a given lane *l* at time *t* is given by the following equation:

$$d_{l,t} = \frac{\sum_{i=1}^{w} \frac{w-i}{\beta}(d_{l,t-i})}{\sum_{i=1}^{w} \frac{w-i}{\beta}}$$

Traffic Signal Group Formation

- Below is a representation of the area visible to an intersection.



Swarm Intelligence Applied to Traffic Lights Group Formation, Oliveira and Bazzan

Traffic Signal Group Formation

- The stimulus *s* of a signal plan j is then computed based on:
  1) The pheromone levels within the lanes during each phase
  2) The plans being used by visible neighbours
  3) The proportion of the entire time of cycle k that the light is green (Δk), given by:

$$\Delta_k = (time_{end} - time_{begin}) / time_{cycle}$$

Traffic Signal Group Formation

- The equation used to calculate stimulus is:

$$s_j = \alpha \sum_{k=0}^{n} (d_{in_{k,t}}) \Delta_k + (1-\alpha) \frac{a_j}{A}$$

- The probability of a signal plan *j* being implemented by an intersection *i* is then given by:

$$T_{\theta ij}(s_j) = \frac{s_j^2}{s_j^2 + \theta_{ij}^2}$$

Traffic Signal Group Formation

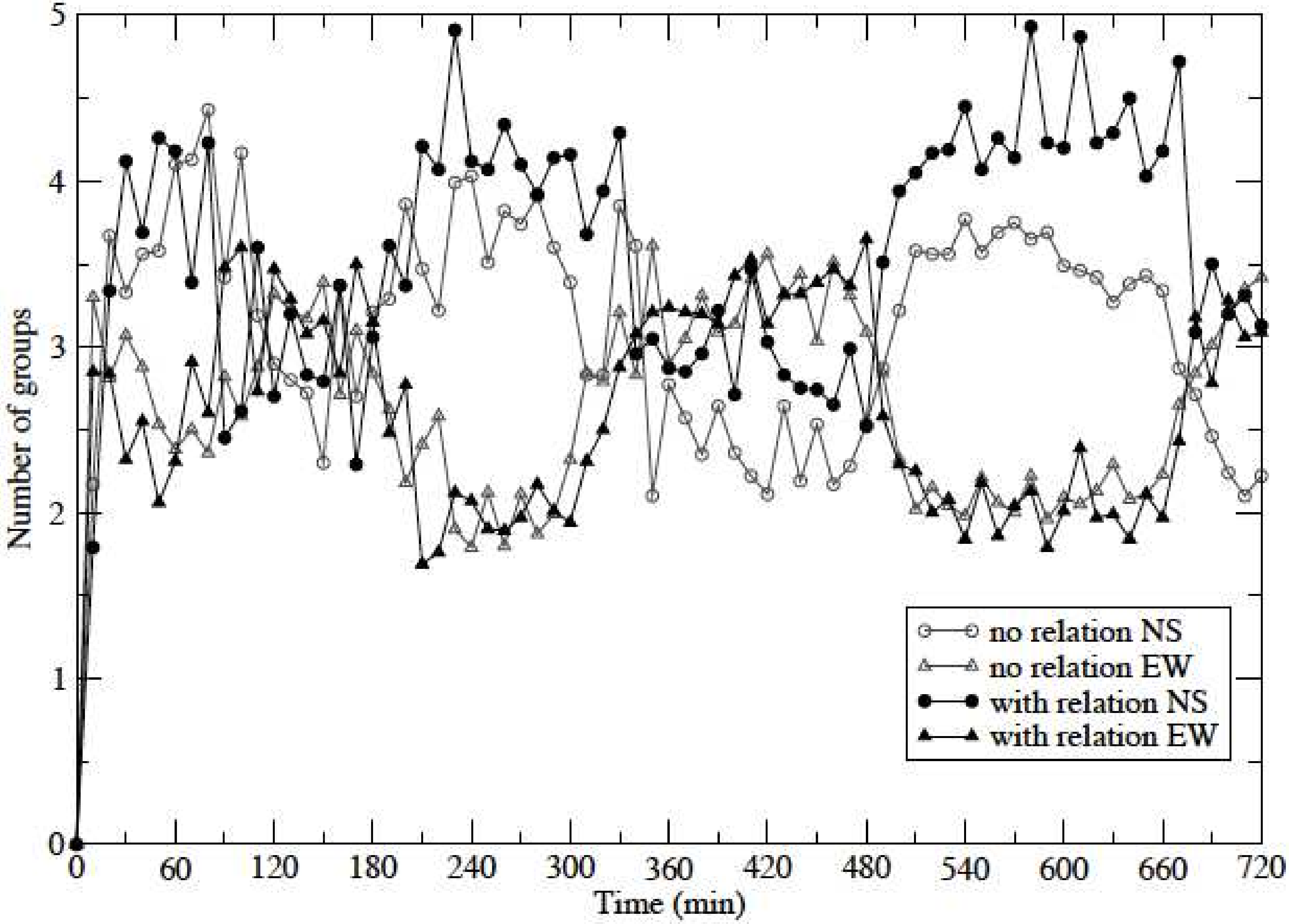
- θ_{ij} is reinforced at the end of every time interval δ_t (set at 10 minutes for their tests) using the following formula:

$$\theta_{ij} = \theta_{ij} - l\,\delta_t$$

- Where *l* is a learning coefficient calculated with:

$$l = 1 - 2\sigma$$

Traffic Signal Group Formation



Swarm Intelligence Applied to Traffic Lights Group Formation, Oliveira and Bazzan

Gershenson - Self-Organizing Traffic Lights[8]

- Defined 3 methods of controlling traffic lights:
  - SotlRequest
  - SotlPhase
  - SotlPlatoon
- Claims no communication between lights is necessary, but fail to explain a method in which the information required is passed

Gershenson - Self-Organizing Traffic Lights: Sotl-Request

- Each traffic light keeps a counter K_i which is reset to 0 every time the light switches
- K_i is increased by number of approaching vehicles at each time step
- When K_i reaches a specified threshold (θ), the lights at the intersection switch (red->green, green->red)
- Problem: Fast switching

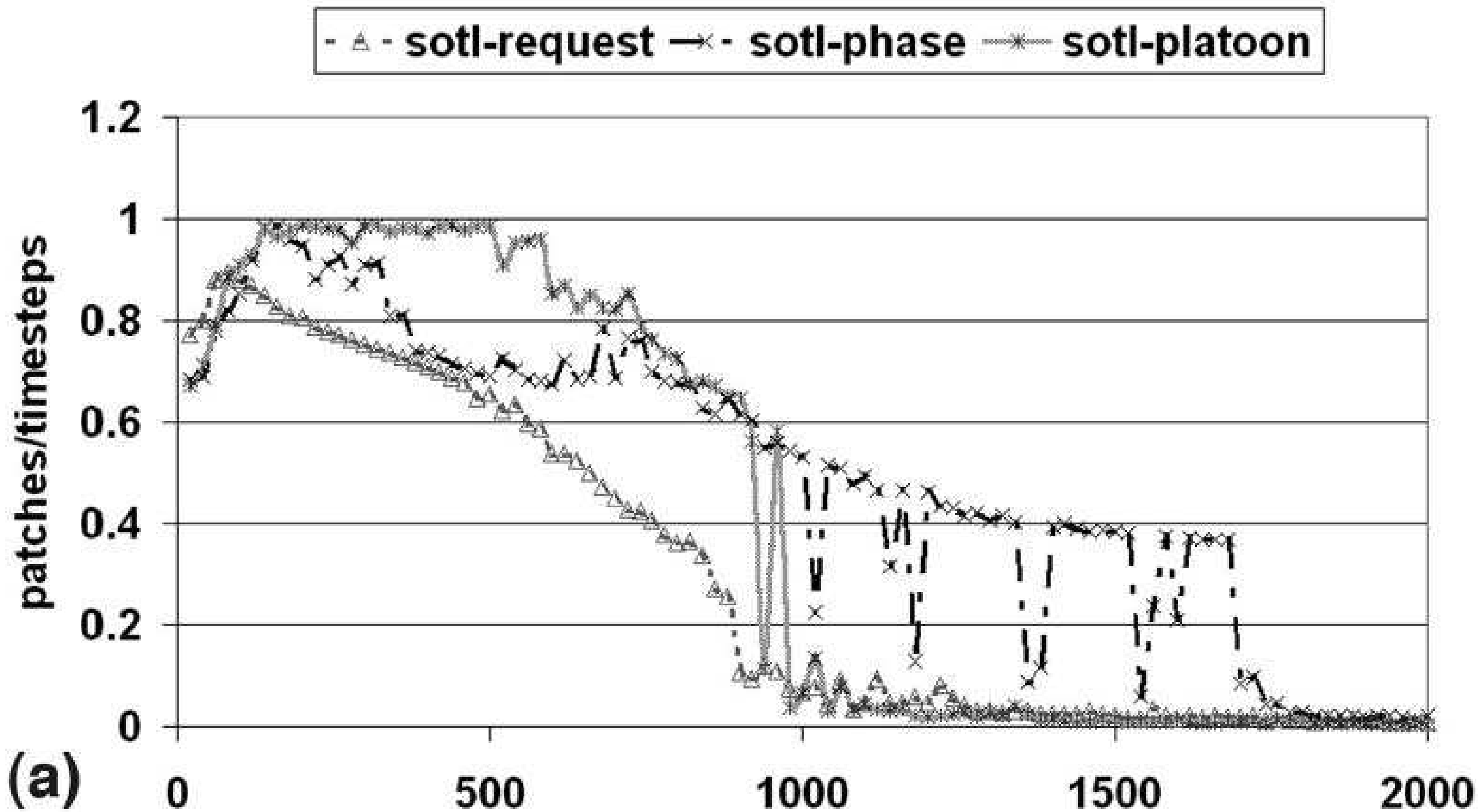
Gershenson - Self-Organizing Traffic Lights: Sotl-Phase

- Uses the same strategy as Sotl-Request, only with a predefined minimum phase length (φ_{min})
- Intersections now keep another counter (φ_i) which represents the time steps since the last traffic light change
- Even when $K_i > \theta$, lights will not switch until $\varphi_i > \varphi_{min}$

Gershenson - Self-Organizing Traffic Lights: Sotl-Platoon

- Adds two more restrictions to Sotl-Phase
- Before switching lights, Sotl-Platoon checks for platoons crossing the intersection
- If a car is within ω patches of the intersection, the lights will not switch
- This restriction is not taken into account if there is more than μ vehicles approaching the intersection

Gershenson - Self-Organizing Traffic Lights: Results



Decentralized Traffic Signal Control With Intersection Communication

- Traffic state can be determined through communication between intersections
- Sensors at each intersection detect traffic state
- Pieces of information can be sent to neighboring intersections
- Intersections use local observation and information from neighbors to generate signal plans

Decentralized Traffic Signal Control With Intersection Communication

- The algorithm begins by calculating a green length for a single direction as follows:

$$P_w = \frac{AC_w}{(AC_w + AC_N)}$$

$$TG_1 = P_w \times CL$$

- Where AC_w and AC_N are the average cars in the west/north approaching roads over the time window and CL is the cycle length

Decentralized Traffic Signal Control With Intersection Communication

- To facilitate coordination and prevent extremely volatile phase lengths, the neighbor weight (NW) and current time weight (CTW) parameters are introduced:

$$TG_2 = (1 - NW)TG_1 + NW\left(\frac{\sum_{i \epsilon X} TG_i}{|X|}\right)$$

$$TG_3 = (1 - CTW)TG_2 + (CTW \times CGT)$$

- Where X is the set of all valid neighbors, and CGT is the amount of green time currently allotted in this direction

Decentralized Traffic Signal Control With Intersection Communication

- Finally, the amount of green time is bound so a minimum amount of time (MT) is allowed in each direction during each cycle using the following equations:

$$TG_4 = max(TG_3, MT)$$
$$TG_F = min(TG_4, (CL - MT))$$

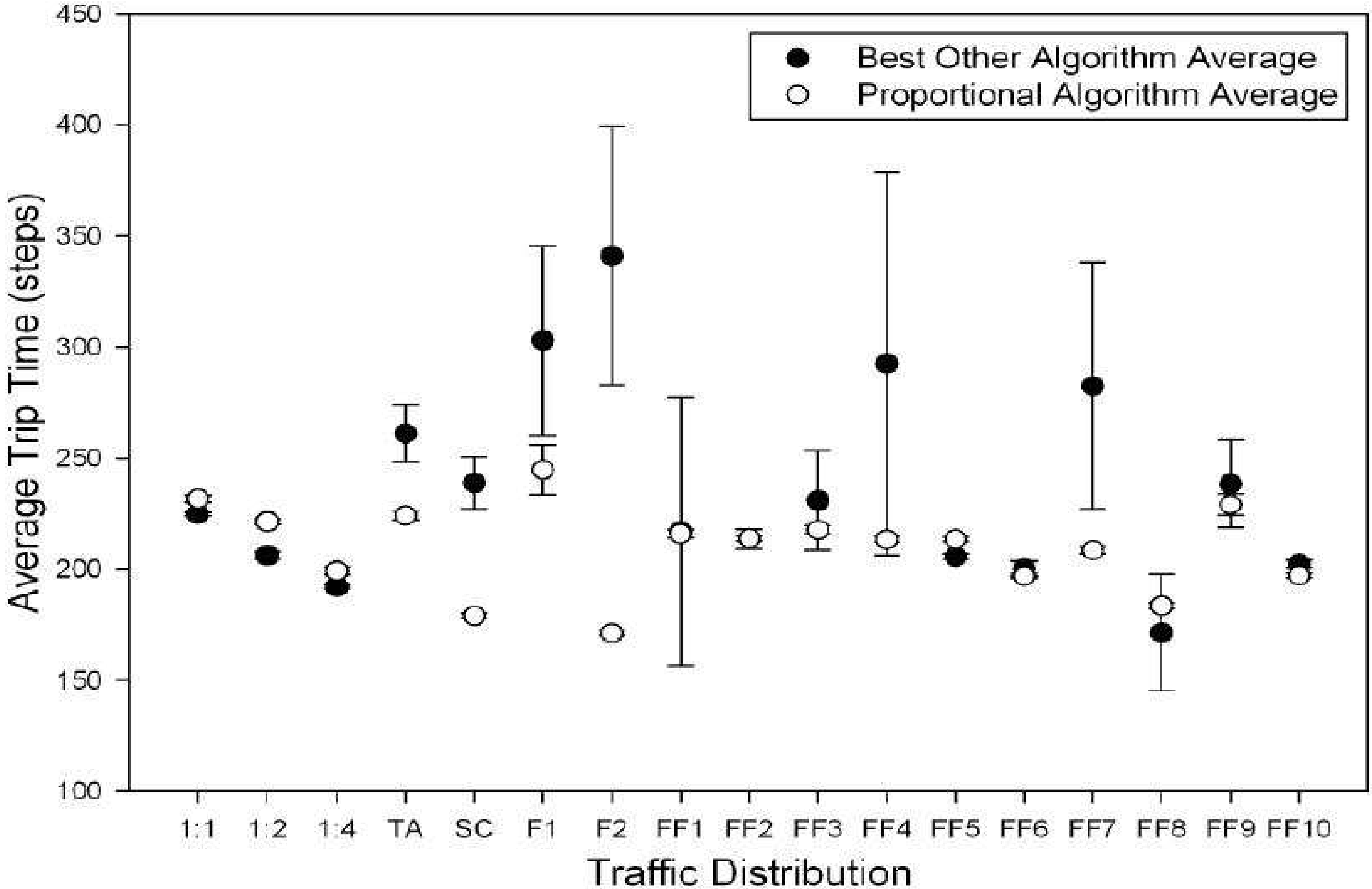
Decentralized Traffic Signal Control With Intersection Communication

- The performance of this algorithm was tested and compared to results found using 3 fixed signal plans (30/30, 40/20, 20/40), as well as the task allocation approach outlined above
- Each strategy was tested on 17 different distributions, averaging the results found over 25 runs on each distribution

Decentralized Traffic Signal Control With Intersection Communication

Distribution	30/30 Avg	30/30 SD	40/20 Avg	40/20 SD	20/40 Avg	20/40 SD	TA Avg[1]	TA SD	Prop Avg	Prop SD
Fixed Even	224.80	0.69	Failed	Failed	Failed	Failed	Failed	Failed	231.67	1.56
Fixed 2:1	281.06	34.79	206.15	1.64	Failed	Failed	Failed	Failed	221.36	1.21
Fixed 4:1	Failed	Failed	192.10	0.93	Failed	Failed	Failed	Failed	199.23	1.81
TA	261.15	12.63	Failed	Failed	Failed	Failed	Failed	Failed	224.13	1.88
Sin/Cos	238.80	11.70	Failed	Failed	Failed	Failed	Failed	Failed	179.07	1.05
F1	302.95	42.52	Failed	Failed	Failed	Failed	Failed	Failed	244.76	11.35
F2	341.09	57.99	514.40	29.67	Failed	Failed	Failed	Failed	171.24	0.95
FF1	367.64	59.83	217.03	60.48	Failed	Failed	Failed	Failed	215.88	1.88
FF2	213.60	4.34	Failed	Failed	312.59	23.31	Failed	Failed	213.77	1.02
FF3	230.94	22.42	685.37	59.73	Failed	Failed	Failed	Failed	217.63	1.86
FF4	292.61	86.30	356.38	23.84	Failed	Failed	Failed	Failed	213.34	1.26
FF5	312.12	57.31	205.67	1.29	Failed	Failed	Failed	Failed	213.47	1.07
FF6	200.17	3.43	264.15	20.16	772.78	81.58	380.27	62.51	196.92	0.96
FF7	282.59	55.59	410.92	41.53	Failed	Failed	Failed	Failed	208.47	1.52
FF8	195.31	1.91	417.11	29.03	221.67	11.68	171.41	26.18	183.52	1.26
FF9	238.38	19.69	Failed	Failed	Failed	Failed	Failed	Failed	229.05	4.79
FF10	381.49	122.85	202.38	1.57	Failed	Failed	Failed	Failed	197.14	0.86
Bests	2		3		0		1		11	
Failures	1		6		14		15		0	

Decentralized Traffic Signal Control With Intersection Communication



Moving to a Traffic Simulator

- The traffic model within NetLogo is very basic
- A real traffic simulator allows for much more realistic behavior
- Many open source traffic simulators are available (SUMO[9] is used here)
- The algorithm and traffic model were moved to SUMO for testing in a more realistic environment

Adding Intersection Offsets to the Algorithm

- The addition of offset values allow intersections to further coordinate
- The main goal is to have a traffic light turn green as a group of cars approaches, allowing them to travel through without stopping
- Each intersection has times within the cycle at which it will turn green in the W/E or N/S direction (these values are calculated using known/observed data)

Adding Intersection Offsets to the Algorithm

- Steps to creating an offset value

  1) Determine which neighbor to offset with

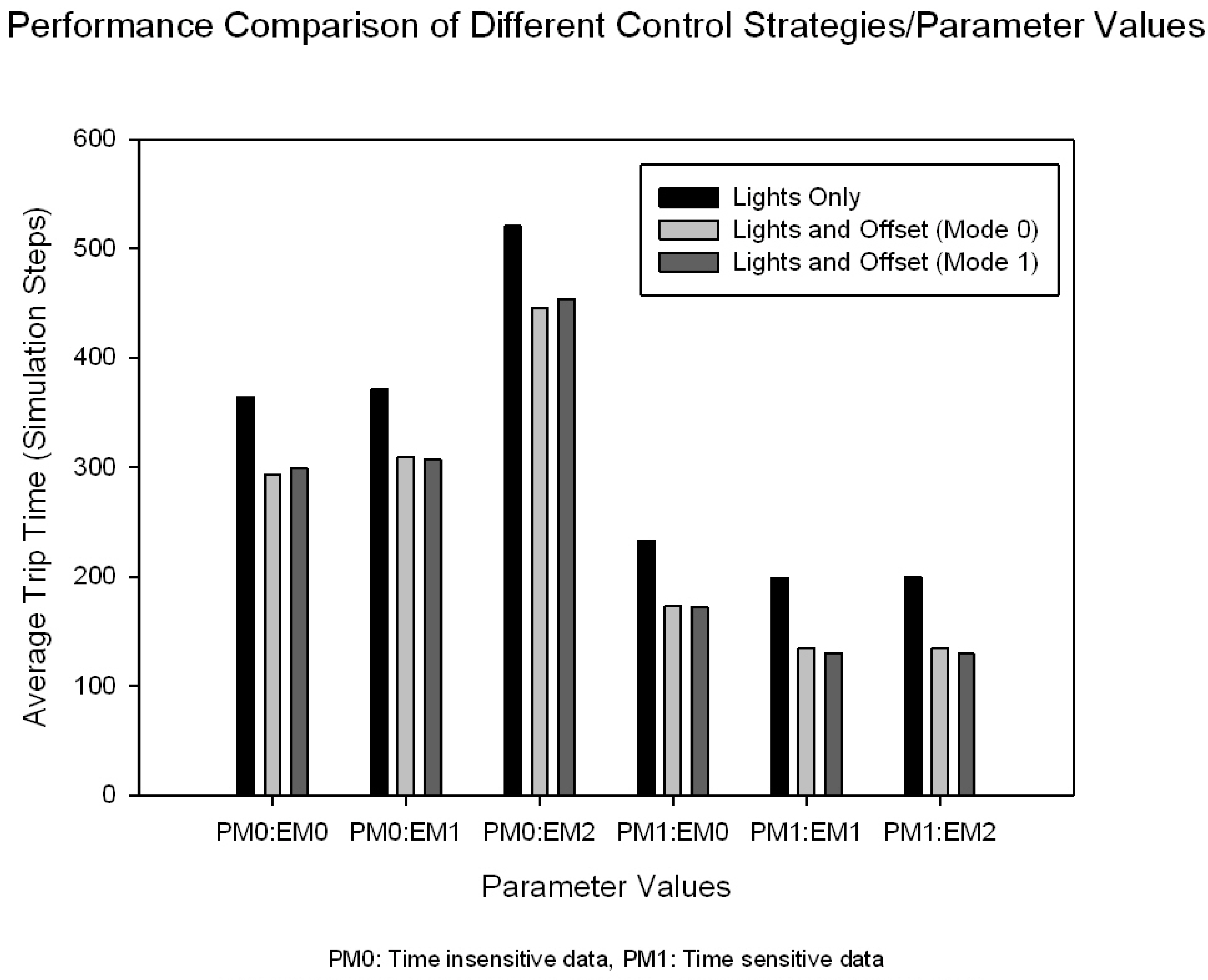
  2) Determine the estimated trip time from that neighbor to the intersection in question (distance/speed)

  3) Set the intersections green switch time to the sum of the neighbor’s green switch time and the estimated trip time (taking cycle length into account)

New Parameters/Operations

- Two different methods for calculating average cars were implemented (straight average and time-weighted average)
- Three different methods for counting vehicles on a road were implemented (number of vehicles per unit of road, number of vehicles on the road, and number of stopped vehicles)
- Two offset methods (average over all neighbors, offset with neighbor in most saturated direction)

Results from Simulation in SUMO



Testing with Real World Data

- Data supplied by the City of Ottawa has allowed for a real world area to be modeled within SUMO
- Includes a realistic road network (7x9 blocks)
- Hourly traffic counts have been supplied and integrated into the model
- Turning rates have been calculated using the given data, allowing for vehicle route generation
- Signal plans currently used within the City are also available to test against

Real World Difficulties

- Using real world data complicates signal generation
- Example #1: Offsetting lights is much more difficult
- Example #2: More complicated intersection logic (e.g. turning lanes, advanced green lights, etc.)

Real World Difficulties

- Several things need to be addressed by an algorithm within the real world:
  - All possible offsets need to be considered
  - Different network structures must be taken into account
  - Different signal logics may be required

Possible Improvements to the Algorithm

- Prediction of future traffic volumes
- Different strategies for different volume levels (e.g. offset vs. clearing)
- Dynamic cycle lengths
- Improved information propagation
- Dynamic speed limits

Future Work

- Improvements in traffic modelling (perhaps some sort of standard?)
- Improvements in traffic simulation
- Inclusion of pedestrians into models
- More testing on real data
- Dynamic traffic routing

References

[1]http://www.worldmapper.org/display.php?selected=141

[2]http://www.thestar.com/News/article/285168

[3]http://www.personneltoday.com/articles/2009/11/09/52908/over-21-million-hours-a-day-spent-commuting-by-uk-workers.html

[4] Almejalli, K., Dahal, K., and Hossain, A. (2009). An intelligent multi-agent approach for road traffic management systems." 18th IEEE International Conference on Control Applications, IEEE. 825-830.

References

[5] Sánchez, J., Galán, M., and Rubio, E. (2008). Applying a traffic lights evolutionary optimization technique to a real case: “Las Ramblas" area in Santa Cruz de Tenerife." *IEEE Transactions on Evolutionary Computation, Vol. 12, IEEE.* 25-40.

[6] Penner., J., Hoar, R., , and Jacob, C. (2002). Swarm-based traffic simulation with evolutionary traffic light adaptation." IASTED International Conference Applied Simulation and Modeling. 467-472.

References

[7] de Oliveira, D. and Bazzan, A. L. C. (2007). Swarm intelligence applied to traffic lights group formation." VI Encontro Nacional de Inteligencia Artificial, ENIA, 1003-1012.

[8] Gershenson, C. Self-organizing Traffic Lights. Complex Systems, 2005, 16, 29-53

[9]http://sourceforge.net/apps/mediawiki/sumo/index.php?title=Main_Page