Evolutionary Traffic Signals

Mike McClurg
Rose-Hulman Institute of Technology
5500 Wabash Ave. CM 762
Terre Haute, IN 47803
mcclurmc@rose-hulman.edu

ABSTRACT

While modern traffic signals are increasingly sensor based, many older traffic signals are controlled by predetermined timings. These timings are rarely perfect and often cause unnecessary delays in traffic and increased congestion in cities. Converting existing traffic signals to use sensors will increase the efficiency of the traffic signal, but this option is often too expensive to consider. Updating signals with better timings could be an inexpensive way to increase traffic signal efficiency. There is no good method for choosing signal timings and evolutionary algorithms are a promising alternative to setting them by hand. Because of the dynamic nature of vehicle traffic, swarm intelligence might provide a good alternative to the probabilistic traffic models that are often used in traffic simulations.

Categories and Subject Descriptors

J.2 [**Physical Sciences and Engineering**]: Engineering; H.1 [**Information Systems**]: Models and Principles

General Terms

Experimentation

Keywords

Traffic Signals, Evolutionary Algorithms, Swarm Intelligence

1. INTRODUCTION

While modern traffic signals use sensors and can be centrally controlled, smaller cities with older infrastructure often do not have the funds to upgrade from "dumb" traffic signals that have no sensors and are controlled only by predetermined timings. Sensorless traffic signals are only as good as their programmed timings — they have no way to adjust dynamically to changes in traffic if they have not already been programmed by an engineer who has anticipated changing traffic patterns. For instance, a traffic signal at an intersection that is busy only in the daytime may be programmed to flash yellow and red after midnight so that late night drivers do not have to wait at an empty intersection.

Poorly timed traffic signals are inefficient because they cause drivers to wait longer than they have to, causing unnecessary delays and increasing traffic congestion. Increased congestion slows down commuters and business traffic and can increase the cost of transportation through a city. Congestion can also lead to more traffic accidents because traffic signals changing from green to red at inappropriate times can encourage people to run red lights.

These problems might be avoided by making traffic signals more effective. One way to do this is to install sensors in older intersections, but this solution is expensive and disrupts traffic through the intersection while it is being upgraded. A less expensive solution that does not involve construction is to find better timings for these dumb traffic signals. This solution would allow improvements to be implemented inexpensively as soon as they were found, without requiring expensive construction that delays traffic. Evolutionary computation and swarm intelligence are two fields which can be applied to the problem of finding effective traffic signal timings.

2. EVOLUTIONARY COMPUTATION

Evolutionary computation is a branch of artificial intelligence that uses Darwinian evolution to find solutions to optimization problems. An evolutionary algorithm works by creating a population of random solutions to the problem; these solutions are known as individuals. An individual may consist of subparts that can be referred to as chromosomes; each chromosome may have subparts that can be referred to as genes. Each of these individuals is evaluated against the problem and assigned a fitness value based on how good of a solution it is.

Once all individuals are evaluated the evolutionary algorithm ranks them based on their fitness values. The fittest individuals are selected for recombination. This step is analogous to sexual reproduction in biology: a new individual, the child, is produced from a pair of individuals, the parents, so that the child contains some information from each of its parents. These children replace the least fit individuals in the population. After this step, some individuals from the population are selected to be randomly mutated, in order to introduce new information into the population. Each of these cycles is called a generation. After a large number of

GECCO'06, July 8–12, 2006, Seattle, Washington, USA.

Phase time gene:	24.5	12.2	3.2	16.7
Phase order gene:	3	1	2	4

Figure 1: Example representation of a chromosome for an intersection with four phases.

generations, a good solution to the problem may emerges.

3. SWARM INTELLIGENCE

Swarm intelligence is a branch of artificial intelligence that mimics the behavior of swarming or flocking animals to solve problems. One of the most common animals to model swarm intelligent systems after is the ant; this class of problems is particularly well suited for graph based problems such as the traveling salesperson problem. In this approach, a simple intelligent agent is created that mimics the some of the basic activities of ants: searching for food, laying pheromones and bringing food back to the nest. The "food" could be located on one or more nodes on the graph or nonexistent, depending on the problem. The act of laying pheromones is called stigmergy, which is a form of communication that modifies the environment in order to transmit information between agents. A number of these agents are released on the problem space; their interaction with each other as they perform the activities that they were programmed to do eventually presents the solution.

4. COMBINING EVOLUTION AND TRAFFIC SIGNALS

The proposed approach to optimizing traffic light timings uses a combination of both evolutionary computation and swarm intelligence. An evolutionary algorithm will generate a population of individuals which consist of information for each of the intersections in the city being simulated.

Each intersection is treated as a chromosome of the individual, with the phase orderings and phase timings as genes. An intersection's phases are the combination of lights that are green at the same time; for instance, in a four way intersection the east and west lights will be green at the same time, while the north and south lights will be green at the same time. The order of the phases is the order in which the phases turn green.

Figure 1 above shows an example of the representation of individuals in this system. The phase timing gene is a string of real values representing the number of seconds that each phase is active in an intersection. The phase ordering gene is a string of integers representing the order in which the phases will become active.

To evaluate each individual's fitness, the evolutionary algorithm will simulate traffic through the city using the timings from each individual. The fitness of each individual will be set to the average amount of time that each vehicle in the simulation had to wait for a traffic light to turn green; the lower the fitness value the fitter the individual.

The vehicles in the city will be modeled as swarm agents. Each agent will have a set of simple rules to follow, such as "follow the speed limit within a factor of S" and "don't get within $T \cdot SpeedLimit$ feet of the next vehicle". "Food" represents someplace that the agents should go: in the morning, areas where people work will have "food" but at night residential areas will have "food". Agents will drop pheromones to alert other agents to roads that are less congested: if there are multiple paths that will take an agent to its destination, it will choose the one with fewer pheromones, which is the less congested one. In this simulation, swarm intelligence is not being used to directly solve the problem; it is being used to create a more dynamic simulation than probability based models can.

5. PREVIOUS WORK

Much work has been done on computer simulation of vehicle traffic, though only a small subset of that work relates to evolutionary computation or swarm intelligence; much of this work only incorporates one of these technologies but not both. Balmer et al. [2] describe a system for agent-based traffic simulation. Their system uses XML to describe agents and can simulate traffic networks with several million agents, though it does not attempt to discover efficient traffic signal timings. Paruchuri et al. [4] describe a similar system for agent-based traffic simulation. Both papers demonstrate the effectiveness of using swarm agents to simulate vehicle traffic.

Vogel et al. [5] propose a system for using evolution strategies, a form of evolutionary computation, to discover efficient traffic signal timings. Their algorithm operates on timings for traffic signal phases, where a phase denotes a set of lanes in an intersection which are allowed through an intersection at the same time (whose lights are green). Yi and Silberstein [7] describe an algorithm for automatically generating sets of phases. This algorithm will be described in more detail in section 6.2.

Hoar et al. [3] have developed a system that combines both evolutionary computation and swarm intelligence to optimize traffic signal timings. Their application, SuRJE (*S*warms *U*nder *R&J* using *E*volution), uses swarm agents to model traffic and a combination of evolution strategies and swarm voting to discover more efficient traffic signal timings. Instead of operating over signal phases, SuRJE operates on the time that each individual light in an intersection is green and red. Both [5] and [3] demonstrate the effectiveness of using evolutionary computation to optimize traffic signal timings.

6. CURRENT PROGRESS

The system described in section 4 is nearing completion. As of now the code that implements the network in the simulation is mostly complete and needs testing. The evolutionary algorithm will be implemented with GAlib [6], a free C++ library of genetic algorithm tools.

6.1 Implementation Details

The system is written in C++ on a Linux cluster. The evolutionary algorithm will be implemented using GAlib; the fitness evaluation function, which runs the swarm-based traffic simulation on each of the traffic signal timings, will be

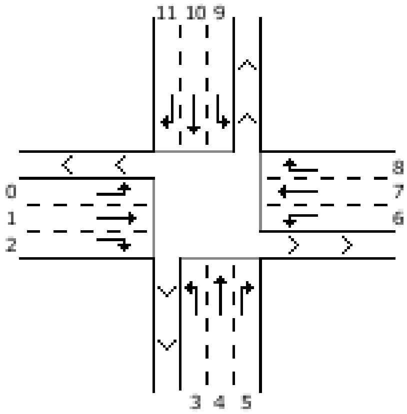

Figure 2: Example representation of a four way, three lane intersection. A possible complete set of legal phases for this intersection is ((0 1 2 11) (2 3 4 5) (5 6 7 8) (8 9 10 11)).

written using MPI (Message Passing Interface, an API for writing distributed applications) to perform the simulation on each of the nodes of the cluster. The traffic network will be described in XML and parsed by by Xerces [1], a free XML parser written by the Apache XML Project. The algorithm for finding traffic signal phases is written in Scheme and will probably be rewritten into C++.

6.2 Representation of Intersections

This representation of intersections was first described in Yi et al. [7]. Each intersection is represented as a graph $G = (V, E)$, where each lane entering and leaving the intersection is a vertex. An edge is placed between each pair of vertices whose lanes do not conflict; a pair of lanes conflicts if it is illegal for them to have right of way through the intersection at the same time, such as a lane with a green left turn signal and an opposing straight green light.

All legal signal phases are then maximal cliques of this graph. Yi et al. give two algorithms: one for finding all maximal cliques C_k of a given intersection graph and another for finding all combinations of cliques such that $\bigcup C_k = V$ for each combination. Each combination of maximal cliques represents a set of phases for the signals in the intersection. One of these sets of phases is chosen to represent the intersection in the simulation; the evolutionary algorithm can then operate on phase times and phase orders.

Figure 2 shows a standard four way, three lane intersection. Each incoming lane is numbered; these numbers correspond to the numbers in the example legal phase set.

6.3 Work to be Done

The first thing that needs to be finished is the simulation. Most of the network code is in place but very little of the agent code has been written. The agent code is small in comparison to the network code, though, so there should not be much effort required to implement the swarm agent code.

Once the simulation is coded and tested the evolutionary algorithm can be implemented. Because of the way it is written, extending GAlib to farm out the fitness evaluation using MPI will not take any more effort than configuring GAlib to run in this application, since GAlib requires the user to write his or her own fitness evaluation function anyway. While it is arguably the most important part of the system, the evolutionary algorithm actually makes up a small portion of the total number of lines of code in the application.

7. EXPECTED CONCLUSIONS

Once the application is built and tested, it can be used to run experiments. The system will be run with different setting on the simulator in order to represent different traffic conditions. There will be at least four different traffic conditions tested in the simulator:

1. Heavy morning traffic
2. Medium mid–day traffic
3. Heavy evening traffic
4. Light late night traffic

These traffic conditions will produce different sets of timings: a traffic signal that stays green for a whole minute in one direction and only ten seconds in the other direction is much more efficient at 2:00am than at 7:00am. The results from each of these different simulation settings would be combined when they were implemented so that a city's traffic signals will work optimally at each time of the day.

Test networks for the simulation will be constructed from subsets of Terre Haute, Indiana. The simulation will be run independently from the evolutionary algorithm with timings that are similar to Terre Haute's timings; the results from this simulation run will serve as the base to which later experiments can be compared.

The first set of experiments involving the evolutionary algorithm will fix the phase orderings; even though phase orderings are represented in the structure of the chromosomes, they will not be operated on by the evolutionary algorithm. This will hopefully simplify the evolutionary algorithm in such a way that it will converge more quickly on a solution than with the full implementation of the chromosome. The next set of experiments will enable the phase orderings; this set of results will then be compared to the first.

8. ACKNOWLEDGMENTS

I would like to thank Dr. Larry Merkle, my advisor and friend, for encouraging me to pursue a thesis and for all his help along the way. I would also like to thank Dr. Mike Wollowski, whose class on Swarm Intelligence inspired this research.

9. REFERENCES

[1] APACHE. Xerces. Xerxes is an XML parser for C++ developed by the Apache XML Project. It can be found at http://xml.apache.org/xerces-c/.

[2] Balmer, M., Cetin, N., Nagel, K., and Raney, B. Towards truly agent-based traffic and mobility simulations. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems* (2004), vol. 1, ACM SIGART, IEEE Computer Society, pp. 60–67.

[3] Hoar, R., Penner, J., and Jacob, C. Evolutionary swarm traffic: If ant roads had traffic lights. In *Proceedings of the 2002 Congress on Evolutionary Computation* (2002), vol. 2, IEEE, pp. 1910–1915.

[4] Paruchuri, P., Pullalarevu, A. R., and Karlapalem, K. Multi-agent simulation of unorganized traffic. In *International Conference on Autonomous Agents* (2002), ACM, pp. 176–183.

[5] Vogel, A., Goerick, C., and von Seelen, W. Evolutionary algorithms for optimizing traffic signal operation. In *Proceedings of European Symposium on Intelligent Techniques* (2000), ESIT.

[6] Wall, M. Galib: A c++ library of genetic algorithm components, 2005. This is a C++ library for implementing genetic algorithms. It can be found at http://lancet.mit.edu/ga/.

[7] Yi, K., and Silberstein, A. Modeling traffic intersections to optimize flow. *http://www.cs.duke.edu/ yike/cps300paper.ps* (2005).