

## Traffic Signal Control with Swarm Intelligence

David Renfrew, Xiao-Hua Yu

*Department of Electrical Engineering, California Polytechnic State University  
San Luis Obispo, CA 93407, USA*

### Abstract

*Traffic signal control is an effective way to regulate traffic flow to avoid conflict and reduce congestion. The ACO (Ant Colony) algorithm is an optimization technique based on swarm intelligence. This research investigates the application of ACO to traffic signal control problem. The decentralized, collective, stochastic, and self-organization properties of this algorithm fit well with the nature of traffic networks. Computer simulation results show that this method outperforms the conventional fully actuated control, especially under the condition of high traffic demand.*

### 1. Introduction

With the ever-increasing traffic demand, congestion has become a serious problem in many major cities around the world. ATMS (advanced traffic management system) is a systematic effort toward the design of an integrated transportation system with new technologies. By regulating the traffic demand at each intersection in the network, the goal is to avoid traffic conflicts and shorten the queue length at a stop line.

At a signalized intersection, traffic signals typically operate in one of three different control modes, namely, pre-timed control, semi-actuated control and fully actuated control. Pre-timed control is an open-loop control strategy, in which all the control parameters are fixed and pre-set off-line. It is easy to implement and is well suited for predictable traffic patterns. In actuated control, the control signal is adjusted in accordance with real-time traffic demand obtained from detectors. In general, actuated control performs better than the pre-timed control.

Traffic signal control problem has been studied by many researchers over years. Some major conventional traffic signal control systems, such as TRANSYT (traffic network study tool) [1], SCOOT (split, cycle and offset optimization technique) [2], and SCATS (Sydney coordinated adaptive traffic system) [3], select

the best pre-calculated off-line timing plan based the current traffic conditions on the road. Some latest developments on traffic signal control employ artificial intelligence technology, such as neural networks [4] and fuzzy logic [5]. Algorithms using Petri nets [6] and Markov decision control [7] have also been investigated in recent years.

Ant colony algorithm is a meta-heuristic approach for solving computationally hard combinatorial optimization (CO) problems [8] [9] [11]. Inspired by the behavior of the ants in real world, ant colony algorithm is a multi-agent system, in which each single agent is called an artificial ant. It is one of the most successful examples of swarm intelligent systems and has been applied to solve many different types of problems, including the classical traveling salesman problem, path planning and network routing.

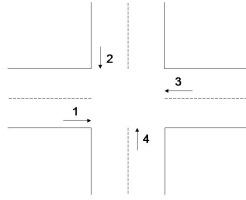
In nature, when searching for food, real ants may wander randomly until they find food. As an ant returns to the colony with food, it deposits pheromone, a chemical used for communication. These pheromone trails guide other ants as they continue their search for food. As more pheromone is deposited, the ants' paths become less random and are biased toward the paths with higher pheromone concentration.

In the ant colony algorithm, artificial ants search the solution space probabilistically to create candidate solutions. These candidate solutions are then evaluated and updated, based on the pheromone associated with each one of them. It should be noticed that over time, certain amount of pheromone concentration may evaporate. Finally, the one with the highest value of pheromone is considered to be the optimal solution of the problem.

In this research, a new approach to find the optimal signal timing plan for a traffic intersection is investigated using ant colony optimization algorithm. Rolling horizon algorithm is also employed to achieve real-time adaptive control. Computer simulation results indicates that this new approach is more efficient than traditional fully actuated control, especially under the conditions of high, but not saturated, traffic demand.

## 2. The dynamic model of a traffic intersection

Modeling traffic dynamics and optimizing the control signals are two interrelated problems. Consider a typical four-lagged isolated traffic intersection with four external approaches, as shown in Fig. 1. For the sake of simplicity, only through movements are considered. The traffic flows move along two directions (east/west or north/south) and thus only two sets of traffic control signals (green for east/west or green for north/south) are considered.



**Fig. 1. A typical traffic intersection**

At a given time  $t$ , the queue length on movement  $i$  can be denoted as  $q^i(t)$ , where  $i$  represents the index of a movement. Thus, the queue length at the whole intersection can be denoted as:

$$q(t) = [q^1(t), q^2(t), q^3(t), q^4(t)] \quad (1)$$

Similarly, the number of vehicles leaving movement  $i$  during a time interval  $(t_1, t_2)$  can be denoted as  $q_{out}^i(t_1, t_2)$ . It is a function of the signal choice and the queue length at  $t_1$ . When  $u(t_1, t_2) = \text{green}$ , we have:

$$q_{out}^i(t_1, t_2) = \min \left[ q^i(t_1), 1 + \text{Int} \left( \frac{t_2 - t_1}{hw} \right) \right] \quad (2)$$

where  $hw$  is the headway between vehicles as they leave the intersection,  $u(t_1, t_2)$  is the signal during the time interval  $(t_1, t_2)$  and  $\text{Int}(\cdot)$  gives the integer part of the input. Obviously, when  $u(t_1, t_2) = \text{red}$ ,  $q_{out}^i(t_1, t_2) = 0$ .

The number of cars arriving during a time interval  $(t_1, t_2)$  can be denoted as  $q_{in}(t_1, t_2)$ . It has been supported by the results of many field tests that under most circumstances, the arrival of vehicles for the external movements follows the Poisson distribution [9]. Therefore,

$$P(n) = \frac{(\lambda \Delta t)^n e^{-\lambda \Delta t}}{n!} \quad (3)$$

where  $n$  is a positive integer for number of arrivals,  $\lambda$  is the average vehicle arrival rate in vehicles per hour and  $\Delta t$  is the duration of time period.

From the above, the dynamic equation of traffic flow can be described as:

$$q(t) = q(t-1) + q_{in}(t) - q_{out}(t) \quad (4)$$

## 3. The ant colony optimization algorithm

The principle of swarm intelligence is based on the studies of social interactions between biological insects in nature. In contrast to the global, centralized traditional approach, it offers an alternative way to design an intelligent system based on the collective, decentralized behavior of many self-organized sub-systems.

A swarm intelligent system typically contains a population of simple agents which only interact locally with each other and the environment. That means, each individual agent in the system only follows simple rules and may not have the knowledge of the overall system. However, the local interactions between such agents can lead to the emergence of a very sophisticated and complicated group behavior. Some of the examples of biological swarm intelligent systems include ant colonies, bird flocking, fish schooling, bacterial growth, etc.

The ant colony optimization (ACO) algorithm was first developed by M. Dorigo in 1992 in his Ph. D. dissertation. It is a meta-heuristic approach for solving computationally hard combinatorial optimization (CO) problems; in other words, it is an ‘‘approximate’’ algorithm which can be used to obtain ‘‘good enough solutions’’ in a reasonable amount of computation time [9]. Inspired by the foraging behavior of the biological ants in real world, artificial ants are introduced and employed as a novel computational intelligence tool. In fact, it is a stochastic search algorithm based on a parameterized probabilistic model called the pheromone model.

Consider a solution space in which each node represents a possible solution for an optimization problem. The major steps of ACO can be summarized as follows:

- 1) Initialization. The pheromone values on each node are set to a constant value.
- 2) Solution construction. Each ant begins on a start node and moves to one of its neighboring node based on the pheromone values. In general, ants move from node  $i$  to node  $j$  with the following probability (also

called the proportional rule, or the transition probability):

$$p_{ij} = \begin{cases} \frac{\tau_{ij}^\alpha \eta_{ij}^\beta}{\sum_{l \in N_i} \tau_{il}^\alpha \eta_{il}^\beta} & \text{if } j \in N_i \\ 0 & \text{if } j \notin N_i \end{cases} \quad (5)$$

where  $N_i$  is the set of the neighborhood nodes of  $i$  that an ant has not visited yet, which includes all possible nodes that an ant can move to when at node  $i$ .  $\tau_{ij}$  is the pheromone value between node  $i$  and  $j$ ; and

$\eta_{ij}$  represents the heuristic information (which is available a priori – for example, in the famous traveling salesman problem, the reciprocal of the distance between two different cities  $i$  and  $j$  is usually chosen to be  $\eta_{ij}$ ). The values of  $\alpha$  and  $\beta$  are usually application dependent; they weigh the importance of the pheromone and heuristic values, respectively. Note that there are potentially many different ways of choosing the transition probabilities; however, Eq. (5) was introduced in the first ACO algorithms, and is still used most often in ACO literature nowadays mainly due to this historical reason [9].

3) Update pheromone. Pheromone update can be implemented in different ways, depending on the specific algorithm being studied; but they all follow a general form. Over time, pheromone evaporates:

$$\tau_{ij} \leftarrow (1 - \rho)\tau_{ij} \quad (6)$$

where  $\rho \in (0, 1]$  is the evaporation rate. The pheromone on some of the paths is then updated by:

$$\tau_{ij} \leftarrow \tau_{ij} + \Delta\tau_{ij} \quad (7)$$

where  $\Delta\tau_{ij}$ , the pheromone update, is determined by the specific algorithm.

4) The above solution construction and pheromone update procedures (i.e., step 2 and 3) are repeated until a stop criterion is met.

ACO has been successfully applied to solve many different types of problems, including the classical traveling salesman problem, task assignment, path planning and routing in telecommunication network, etc. Many different ACO algorithms have been proposed, including the original Ant System algorithm (AS), Elitist Ant System, and MAX-MIN Ant System (MMAS). In fact, the initialization and solution construction procedures are the same for different ACO algorithms; only the ways to update pheromone (i.e., step 3) are different. In this research, we consider two different ACO algorithms, namely, the Ant System (AS) and the Elitist Ant System (EAS) algorithm.

In the Ant System algorithm, after all  $m$  ants have constructed their own solutions and the pheromones on all edges/arcs evaporate based on Eq. (6), the pheromones are updated by:

$$\tau_{ij} \leftarrow \tau_{ij} + \sum_{k=1}^m \Delta\tau_{ij}^k \quad (8)$$

where  $\Delta\tau_{ij}^k$ , the pheromone deposited by ant  $k$  when moving from  $i$  to  $j$ , is defined by:

$$\Delta\tau_{ij}^k = \frac{1}{C^k} \quad (9)$$

where  $C^k$  is the associated cost or reward. Otherwise (i.e., ant  $k$  doesn't move to node  $j$  from node  $i$ ), there is no pheromone deposit, i.e.,  $\Delta\tau_{ij}^k = 0$ .

In Elitist Ant System (also called elitist strategy for ant system) algorithm, extra weight is given to the best-so-far solution. As in the Ant System algorithm, pheromone evaporate first (Eq. (6)), then can be updated by:

$$\tau_{ij} \leftarrow \tau_{ij} + \sum_{k=1}^m \Delta\tau_{ij}^k + e\Delta\tau_{ij}^{bs} \quad (10)$$

where  $e$  is a weighting parameter. The additional term  $\Delta\tau_{ij}^{bs}$  reinforces the best-so-far solution and can be defined as the following (if ant  $k$  moves from  $i$  to  $j$ ):

$$\Delta\tau_{ij}^{bs} = \frac{1}{C^{bs}} \quad (11)$$

where  $C^{bs}$  is the total cost/reward (from the start of the algorithm) associated with the best-so-far solution (including the transition from  $i$  to  $j$ ). This term can also be viewed as the pheromone deposited by an additional ant called the best-so-far ant.

#### 4. Traffic signal optimization using ant colony algorithm

A typical optimization problem is defined on  $(S, f, \Omega)$ , where  $S$  is the set of candidate solutions (or the search space),  $f$  is the objective function, and  $\Omega$  is a set of constraints. The goal is to find a globally optimal solution  $s^* \in S$  such that  $f$  is maximized or minimized.

One of the most important goals of traffic signal control is to minimize vehicle waiting time at intersections. In this research, the amount of pheromone deposited by artificial ant is directly related with this performance criterion. As we know, the green time duration for each signal phase can be any value bounded between a minimum and a maximum value

(called the minimum green and maximum green time). The ACO algorithm determines the optimal green time duration to minimize the total vehicle waiting time, which includes the actual waiting time of the vehicles already in the current queue, and the estimated waiting time of vehicles that may just arrive during this time duration.

The inputs of the ACO controller include the current traffic queue (available from sensor measurements) and a prediction of vehicle waiting time; the output of ACO controller is the optimal signal switching time (or optimal time duration of the signal phase). The prediction of future arrivals and the associated waiting time are critical to the controller and are discussed in detail below.

Let's consider the situation when the length of a green signal is  $(t_2 - t_1)$ , where  $t_1$  is the starting time, and  $t_{min\_green} \leq (t_2 - t_1) \leq t_{max\_green}$ . Let  $q$  be the queue length (number of vehicles) at time  $t_1$ , and  $q \neq 0$ . As shown in Fig.1, only through traffic movements are considered here; thus the green signal for movement 1 and 3 implies the red signal for movements 2 and 4, and vice versa.

When all vehicles in the initial queue are released, that is, when  $(t_2 - t_1) \geq (q - 1)hw$ , the total expected waiting time for a traffic movement under green signal (from  $t_1$  to  $t_2$ ) can be written as:

$$J_{1green}(t_1, t_2) = \frac{q(q-1)hw}{2} + \sum_{n=1}^q (t_1 - at_n) + \frac{\lambda((q-1)hw)^2}{2} + \frac{\lambda((q-1)hw)[\lambda((q-1)hw) - 1]hw}{2} \quad (12)$$

where  $at_n$  is the arrival time of vehicle  $n$ . The first and second terms are the total waiting time of the initial queue, the third term is the expected waiting time of vehicles that arrive during the time interval  $(t_1, t_2)$  when the initial vehicles are released, and the fourth term is the expected time that takes to release these new arrivals in  $(t_1, t_2)$ .

When  $(t_2 - t_1) < (q - 1)hw$ , not all the vehicles in the initial queue can be released. The total expected waiting time for this case is:

$$J_{2green}(t_1, t_2) = \frac{q_{out}(q_{out}-1)hw}{2} + \sum_{n=1}^q (t_1 - at_n)$$

$$+ (q - q_{out})(t_2 - t_1) + \frac{\lambda(t_2 - t_1)^2}{2} \quad (13)$$

The first term is the waiting time of the released vehicles in  $(t_1, t_2)$ , the second term is the waiting time of the initial queue before  $t_1$ , the third term is the waiting time of the initial vehicles not being released in  $(t_1, t_2)$ , and the fourth term is the expected waiting time of estimated arrivals in  $(t_1, t_2)$ .

During the red phase, no vehicle can be released; in addition,  $\lambda(t_2 - t_1)$  vehicles are expected to arrive. Therefore, the total queue at  $t_2$  becomes  $q + \lambda(t_2 - t_1)$ . The expected total waiting time is:

$$J_{red}(t_1, t_2) = q(t_2 - t_1) + \sum_{n=1}^q (t_1 - at_n) + \frac{\lambda(t_2 - t_1)^2}{2} \quad (14)$$

The first and second terms are the waiting time of the initial queue, and the third term is the expected waiting time for new vehicle arrivals in  $(t_1, t_2)$ .

For a traffic intersection as shown in Fig. 1, a complete signal cycle contains two signals and can be written as  $T = T_1 + T_2$ . For example, if  $T_1$  is the time duration of the green signal for movement 1 and 3 (and red signal for movement 2 and 4), then  $T_2$  should be the time duration of the red signal for movement 1 and 3 (and green signal for movement 2 and 4). To minimize the vehicle waiting time during the entire signal cycle, both  $T_1$  and  $T_2$  are determined by the ACO algorithm at the beginning of each signal cycle. Note that  $T_1$  and  $T_2$  are actually determined based on the estimated delay for each movement. Due to the random nature of traffic flow, this estimation may not be accurate and needs to be updated whenever it is possible. Therefore, in this research, a rolling horizon approach is employed. After one signal phase is implemented, all the queues at intersection are checked by sensors and all the information is updated to generate a better estimation for the next signal phase. In this case, for each signal cycle, the ACO algorithm is called twice to find the optimum signal length for each of the two phases.

## 5. Simulation results

The proposed algorithm is tested and compared with the conventional fully actuated control by computer

simulation. In fully actuated control algorithm, both the cycle length and the green time for every phase of the intersection can be varied. At every time step, the fully actuated controller checks whether an arrival has occurred on any lane of the intersection. If an arrival has occurred, then the phase is given an extension if it has a green indication. If the phase does not have a green, a call is registered for that phase. To determine the signal indication of next phase, all the calls need to be taken into account. The phase sequence of fully actuated control is fixed; however, certain phases in the cycle may be skipped if there is no demand detected by detectors.

It is assumed that the intersection is “clear” when the simulation starts (i.e., zero initial conditions, or no queue at the beginning), and each traffic movement is independent. It is also assumed that the number of vehicles at the intersection is known, i.e., video-camera type detectors are available at the intersection. We choose the maximum and minimum green time to be 50 seconds and 5 seconds, respectively. Both arrival and departure headway are 2 seconds. Loss time (human reaction time) is 0 second; and all red phase time is 3 seconds. With Poisson arrival pattern, the test is performed on different vehicle arrival rate, from 200 (vehicles per hour per movement) to 800 (vehicles per hour per movement). The results are shown in Fig. 2, where the x-axis shows the vehicle arrival rate and y-axis represents the average delay (seconds per vehicle per movement). Two different ACO algorithms are considered, i.e., the Ant System (represented by squares in the plot) and the Elite Ant System (represented by diamonds in the plot). The average delay of fully actuated control is shown in triangle. It is shown that fully actuated control performs well when the traffic demand is light (i.e., arrival rate at 200 – 600 vehicles per hour per movement); however, both ACO algorithms yield better results when the traffic demand is heavy (i.e., arrival rate at 700 and 800 vehicles per hour per movement).

## 6. Conclusion

ACO (Ant Colony) algorithm is a new optimization technique based on swarm intelligence. In this paper, two different ACO algorithms are applied to control signals at traffic intersection to reduce the vehicle waiting time. Initial test results show this method outperforms the conventional fully actuated control under the situation of high traffic demand. Further

evaluation and testing on this approach will be performed.

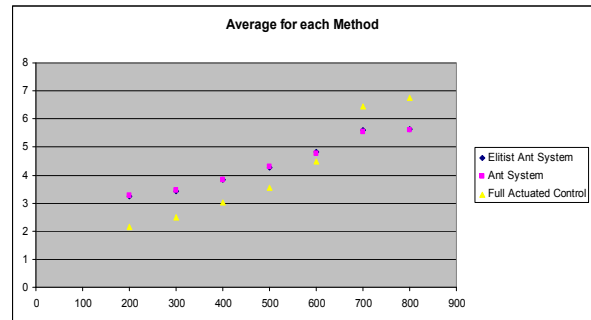


Fig. 2. Computer simulation results

## References

- [1] Transportation research center, Traffic network study tool: TRANSYT-7F software summary, University of Florida, 1987.
- [2] P. Hunt, and D. Robertson, “The SCOOT on-line traffic signal optimization technique”, *Traffic engineering and control*, April 1982.
- [3] P. Lowrie, “The Sydney coordinated adaptive control systems – principles, methodology, algorithms”, IEE conference publication, vol. 207, 1982.
- [4] S. Chien, Y. Ding, “Dynamic Bus Arrival Time Prediction with Artificial Neural Networks,” *J. Trans. Engrg.*, vol. 128, 2002.
- [5] W. Wei, et. al., “Traffic signal control using fuzzy logic and MOGA,” *Proceedings of IEEE conference on systems, man, and cybernetics*, 2001.
- [6] G. List, and M. Cetin, “Modeling traffic signal control using Petri nets”, *IEEE Transaction on intelligent transportation systems*, vol. 5, no. 3, 2004.
- [7] X.-H. Yu, and A. Stubberud, “Markovian decision control for traffic signal systems”, *Proceedings of the 36th IEEE conference on decision and control*, 1997.
- [8] M. Dorigo, and T. Stutzle, *Ant Colony optimization*, The MIT Press, 2004.
- [9] M. Dorigo, and C. Blum, “Ant colony optimization theory: A survey”, *Theoretical Computer Science*, vol. 344, 2005, pp. 243 – 278.
- [10] R. Wilshire, R. Black, et al., *Traffic control systems handbook*, FHWA-IP-85-12, 1985.
- [11] M. Dorigo, M. Birattari, and T. Stutzle, “Ant colony optimization - Artificial ants as a computational intelligence technique”, *IEEE Computational Intelligence magazine*, vol. 1, issue 4, 2006, pp. 28-39.