Intelligent Signal Coordination on Congested Networks Using Parallel Micro Genetic Algorithms

Montty Girianna and Rahim F. Benekohal
University of Illinois at Urbana-Champaign

Abstract

This paper presents the application of Parallel Genetic Algorithms (GA) on signal coordination for networks with congested intersections. When a serial GA is applied to solve traffic control problems, its performance in terms of computation time diminishes as the size of signal networks increases, or the duration of congestion lengthens. This paper uses master-slave parallel micro GA (PMGA) to solve signal coordination, formulated as a dynamic optimization problem. The elapsed time per generation reduces as the number of processors involved increases. When the number of processors equals the number of individuals, for the network described in this paper, the master-slave PMGA reaches the highest speed-up. Although the master-slave PMGA provides a lower bound speed-up and its efficiency degrades as the number of processors increases, this study demonstrates the potential benefits of using parallel GA in solving signal coordination problems.

Introduction

In traffic control application, as in many other applications, the search for optimal solution requires use of simulation to calculate the objective or fitness function and constraints. This simulation takes a significant fraction of optimization time. GA works with a population of independent solutions, which makes it easy to distribute the computational load among several processors. The easiest way is to distribute the evaluation of fitness among several processors while one master executes the operation of genetic algorithms (selection and crossover). This master-slave parallel GA is important for several reasons. It explores the search space in an exact manner to a serial GA, it results in significant improvements in performance, and it is easy to implement.

This paper shows the application of parallel GA to solve signal coordination for congested networks. Signal coordination is formulated as a dynamic optimization problem (Girianna, 2002). In the following section, the formulation of signal coordination is briefly described. Next, the parallel micro GA (PMGA) is described. After the results are examined, the paper concludes with findings and recommendations for future works.

Signal Coordination Problems

Let $G=(N,L)$ denote a traffic signal network consisting of a set of signals $N=20$ and a set of directional (one-way) streets $L=49$ (one-way including exit and entry links), as shown in Figure 1. All signals work with a two-simple phase plan. Suppose that during an oversaturated period, one tries to coordinate the signals along a northbound arterial and those along arterials running E-W, shown by directed thick lines. For this given network, the task is to find optimum signal timings (green times, cycle length, and offsets) that can release the largest number of vehicles out of the signal network. The signal coordination problem described below is an extension of similar work by Abu Lebdeh (Abu-Lebdeh, 1997) on congested signal systems along single arterials.

Objective function
Equation (1) is the objective function, and it represents the number of vehicles processed by a signal network minus a network disutility function, which represents the occurrence of queue on signal approaches along coordinated arterials. The overall goal is to maximize the net effect.

$$Max \quad Z=\sum_{k}^{K}\sum_{i,j\in L} l_{i,j}\, D_{i,j}(k)-\sum_{k}^{K}\sum_{(i,j)\in L_P} \delta_{i,j}(k)\, q_{i,j}(k) \qquad \delta_{i,j}(k)>0 \quad l_{i,j}>0 \tag{1}$$

K is the period of oversaturation, and $D_{i,j}(k)$ is the departure flow at signal j coming from signal i at cycle k. $l_{i,j}$ is the distance between signal i and j. $q_{i,j}(k)$ is the number of vehicles in queue, approaching signal j coming from signal i at the beginning of cycle k, and $\delta_{i,j}(k)$ is a non-negative disutility factor whose values are determined based on a queue management strategy. In this paper, $\delta_{i,j}(k)=1$, for all k. L_p is a set of streets carrying coordinated movements.

Constraints
The first constraint is offsets between two signals along coordinated arterials. Offsets between signal i and j, $\phi_{i,j}(k)$, must satisfy equation (2). $\alpha_{i,j}(k)$ is the time required for the tail to start moving, and is calculated by dividing the queue length at the approach of signal j, $q_{i,j}(k)$, by the starting shock-wave speed. $\tau_{i,j}(k)$ is the time required for the first vehicle in the released platoon from signal i to join the tail of the downstream platoon (as the tail has reached its desired speed), is determined by the acceleration rate, the existence of queues at the approach of signal i, and the availability of an unoccupied space along the street connecting the two signals. Thus, when the unoccupied space is long enough that it provides the platoon's first vehicle a sufficient distance to reach a free flow speed, $\tau_{i,j}(k)$ is simply the ratio of the space to the free flow speed.

$$\phi_{i,j}(k)=\tau_{i,j}(k)-\alpha_{i,j}(k) \tag{2}$$

$$g_i(k)\le g_j(k)+\phi_{i,j}(k)+\beta_{i,j}(k) \tag{3}$$

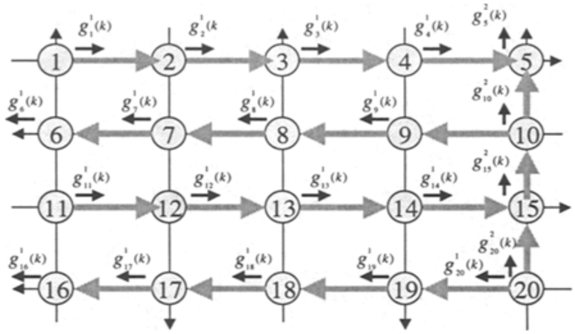

Figure 1. A one-way arterial network with 20 signals

De facto red exists when the signal is green but traffic cannot proceed because of backed-up traffic on a receiving street. To avoid this situation, the effective green time for the upstream signal, $g_i(k)$, should be less than the sum of the effective green time for the downstream signal, $g_j(k)$, the offset between the two signals, $\phi_{i,j}(k)$, and the time it takes for a stopping shock wave to propagate upstream, $\beta_{i,j}(k)$. This mechanism is formulated by equation (3).

Equation (4) provides a formulation to calculate effective green times at a downstream signal serving movements that cross coordinated arterials, $g_j^c(k)$, where Δ denotes loss time.

$$g^{c}_{j}(k) = ext\,\phi_{i,j}(k) + \phi_{i,j}(k+1) - (g_{j}(k) + \Delta) - \Delta \quad (i,j) \in L_p \tag{4}$$

The extended offset, $ext\ \phi_{i,j}(k)$, formulated by equation (5), is a measure of neighboring intersection linkage over time, and it reveals the relationship between a cycle length at an upstream intersection, $C_i(k)$, and the ideal offset at the next cycle, $\phi_{i,j}(k+1)$.

$$ext\,\phi_{i,j}(k) = C_{i}(k) + \phi_{i,j}(k+1) \quad (i,j) \in L_p \tag{5}$$

The sum of offsets and green times around any loop of the network is equal to an integer multiple of the cycle time (Gartner, 1972), and it is formulated in equation (6). $N(l)$ is a set of nodes of directed loop l, $F(l)$ is the set of forward links in loop l, where traffic flow moves in the same direction as the loop's direction, and $R(l)$ is the set of reverse links in loop l, where traffic flow moves in the opposite direction as loop's course. $g_{j,l}\ (k)$ is the green time of signal j at cycle k serving movements in loop l. n is an integer number and P is the number of loops in the network.

$$\sum_{i,j\in F(l)} \phi_{i,j}(k) - \sum_{i,j\in R(l)} \phi_{i,j}(k) + \sum_{j\in N(l)} g_{j,l}(k) = \sum_{k,j\in N(l)}^{k+n} C_{j}(k) \quad \forall l \in P \tag{6}$$

To avoid blockage on coordinated arterials, queue along non-coordinated arterials must be less than the storage capacity of approach links. This is formulated in equation (7) where L_s is a set of streets carrying non-coordinated movements.

$$q_{i,j}(k) \leq \max q_{i,j} \qquad (i,j) \in L_s, k = 1, \ldots, K \tag{7}$$

The queue at intersection j at the beginning of cycle $k+1$, $q_{i,j}(k+1)$, is calculated using equation (8), where $A_{i,j}(k)$ is the number of vehicles arriving during that cycle.

$$q_{i,j}(k+1) = q_{i,j}(k) + A_{i,j}(k) - D_{i,j}(k) \qquad \forall j \in N, \quad (i,j) \in L \tag{8}$$

All control parameters should be within reasonable ranges. Equation (9) formulates the allowable green time where g_{min} is the lower bound and g_{max} is the upper bound, and equation (10) defines the initial values for queues at k=0.

$$g_{\min} \leq g_j(k) \leq g_{\max} \qquad \forall j \in N, k = 1, .., K \tag{9}$$

$$q_{i,j}(k) = q_0 \qquad k = 0; \quad \forall (i,j) \in L, k = 1, \ldots, K \tag{10}$$

For a given cycle, traffic volume that departs from signal j along coordinated arterials is determined by equation (11), where s_j is saturation flow rate.

$$D_{i,j}(k) = \min \left\{ \begin{array}{l} q_{i,j}(k) + A_{i,j}(k) \\ q_{i,j}(k) + \left(\dfrac{A_{i,j}(k)}{g_i(k)} \right) x \left(g_j(k) - \dfrac{q_{i,j}(k)}{s_j} \right) \end{array} \right\} \qquad \forall j \in N \tag{11}$$

Two conditions can occur when non-ideal offsets are applied (along non-coordinated arterials): wasted green and blockages. The duration of wasted green is a period of unused green time after the last vehicle in a queue is cleared until new arrivals pass intersections. Blockages, on the other hand, can occur if traffic is prevented from leaving upstream intersections due to the presence of queue on the downstream intersections. Blockages form when the stopping shock wave, caused by the mid-block traffic, reaches upstream intersections and ends when the blockage is cleared. During this time, no vehicles will be processed. A combination of the occurrence of wasted green time and blockages determines the departure flows along non-coordinated traffic movements.

Parallel Genetic Algorithm (PGA)

To design a good parallel GA, one needs to examine the execution time during one generation as well as the efficiency of the parallel GA. The derivation of theoretical elapsed time per generation and the efficiency described in this paper follows Cantu-Paz (Cantu-Paz, 1999). The execution time in parallel GA depends on the time required to evaluate one individual solution (T_f) and the time required for one processor to communicate to each other (T_c). When $P=S+1$ processors are used (S is the number of slaves) with n individuals in the population, the time required for the evaluation is $(nT_f)/(P)$ and the time required for the communication is $(P)(T_c)$. Thus, the elapsed time for one generation of the parallel GA is estimated by equation (12).

$$T_p = PT_c + \frac{nT_f}{P} \tag{12}$$

This equation exhibits that as more processors are used, the computation time decreases as expected, but the communication time increases. The optimal number of processors is derived by taking the first derivative of the equation equaling to zero and solving for P to obtain the optimal number of processors.

$$P^* = \sqrt{\frac{nT_f}{T_c}} \tag{13}$$

which can be expressed in a more compact form as P=√(n γ), where $\gamma = (T_f)/(T_c)$. To ensure that the master-slave PGA provides an efficient run, the following relationship must hold:

$$\frac{T_s}{T_p} = \frac{nT_f}{\frac{nT_f}{P} + PT_c} = \frac{n\gamma}{\frac{n\gamma}{P} + P} > 1 \tag{14}$$

The ratio T_s/T_p is called the parallel speed-up of the master-slave GA. Solving for γ results in the necessary condition for the PGA performs better than serial GAs.

$$\gamma > \frac{P^2}{n(P-1)} \tag{15}$$

Another parameter to measure the performance of PGA is the efficiency of using more processors. The efficiency of PGA, E, is formulated below.

$$E = \frac{T_s}{T_p P} \tag{16}$$

Ideally, E equals one or $T_s=(T_p)(P)$, and the parallel speed-up, T_s/T_p, equals the number of processors. In practice, due to the cost of communication among processors and whether PGA fully utilizes processors, the ideal conditions may not occur, and the efficiency, E, measures the deviation from the ideal conditions.

Application of PMGA

Genetic solutions, S, consists of all green times serving coordinated movements, i.e., the northbound (signal 20 to signal 5) and all arterials running E-W (see Figure 1). At signal 20, green times allocated for all movements are included in the solution.

There are m=21 decision variables included in S for one cycle. If the duration of oversaturation is K=15 cycles, then there are $(m)(K)$= (21)(15) = 315 variables for all cycles. These green times will determine the number of vehicles processed by the network, accumulated queues, offsets, and all remaining green times.

Every possible value of green times defined in the genetic solutions is first coded in a fixed-length of binary string l, allowing each variable to have 2^l different possible values. The length of a substring is decided by the precision needed in a variable. When five seconds is chosen to be the smallest unit of green time (accuracy) and 80 and 20 seconds are the maximum and minimum green time, respectively, the total number of possible green time values must be (80-20)/5=12. This number can be set equal to 2^l and l can be computed as $\log_2$ (12), and l should equal at least four bits of binary string. For $(m)(K)$ variables, the total string length is $(4)(m)(K)$ = (4)(21)(15) = 1260 for each individual.

For a given candidate solution (individual), the value of fitness function is calculated following the procedure below.

a) Define genetic solutions, S
b) $k = 0$ (cycle 0)
c) Calculate departure flows at signals along the coordinated arterial from signal 20 to 5. This is called the "closing path" because it locks in all other E-W offsets
d) Calculate crossing green times at signals along the closing path
e) For all coordinated arterials that cross the closing path, calculate departure flows and crossing green times
f) Calculate all departure flows along non-coordinated arterials
g) If $k = K$ proceed to step h); otherwise $k = k+1$, and proceed to step c)
h) Calculate the total number of vehicles exiting from each signal
i) Calculate the GA's fitness function, as formulated in equation (1)

The application of genetic algorithms to the signal coordination problem defined in Figure 1 is made for six cases with different population sizes. The maximum population size is selected to be the optimum size for this problem, that is, the square root of the size of individuals (Abu-Lebdeh, 1999; Harik, 1997), $n=\sqrt{1260}=36$. The first case uses n=6 individuals and the second uses n=12 individuals, and so on until it reaches the maximum population size with the increment of 6 individuals. Applied using SGI Origin 2000, for each case, the master-slave PMGA is executed for ten epochs (about 300 generations). Micro GA (MGA) developed by Krishnakumar (Krishnakumar, 1989) is used. MGA is different than Simple GA (Goldberg, 1989) in that MGA only uses selection procedures and crossover genetic operators. In addition, MGA applies an elitism strategy where at least one good individual of the previous generation is kept as one of the candidate solutions in the current generation.

For each case, one processor is first used, and with an increment of one processor, the case is subsequently executed until the number of processors equal the population size. Thus, for example, for $n=30$, PMGA is executed using p=2, 3, 4, and so on until p=30 processors. The load balancing among processors requires that, for a given population size, all slaves and the master evaluate the same number of individuals. If n/p is an integer, for each generation the master and a slave each evaluate n/p individuals. When n/p is not an integer number, some processors evaluate as many as n/p individuals plus one, and some evaluate only n/p. Hence, for n=30 and p=12, some processors evaluates three individuals, say processors one to six, and the remaining evaluates only two.

Figure 2 shows the elapsed time per generation for different population sizes and different number of processors. The elapsed time per generation for a serial MGA, p=1, with 36 individuals requires about one second, and it reduces to about 0.4 seconds when p=6 processors are used, or about 2.5 speed-up (see Figure 3). Further noticeable reduction of the elapsed time is observed until p=20, beyond which an additional processor does not significantly reduce the elapsed time. When p=36, which is the highest number of possible processors used in this 36-individual MGA, the speed-up is about 7.5. A serial typical 36-individual GA solves a signal coordination problem for a network with 20 signals in about one hour. Thus, applying PMGA to a similar problem reduces the computation time to 60/7.5=8 minutes.

When $p=n$ processors are assigned to evaluate n individuals, the largest speed-up is reached when n=36. As shown in Figure 3, speed-up for n=6 is about 2.5, for n=12 is 4, for n=18 is 5.3, for n=24 is 5.8, for n=30 is 7.0, and for n=36 is 7.5. However, it does not look efficient to assign a large number of processors only to gain a marginal speed-up. Elapsed time per generation for 36-individual serial MGA is T_s=0.976624 seconds, and that for 36-individual PMGA with p=36 is T_p=0.130905 seconds. The efficiency for 36-individual PMGA, $E=T_s/(T_p)(p)$ = 21 percent, which is a marginal efficiency.

Conclusion and Recommendations

Applied using two or more processors to solve signal control problems, MGA can be executed faster. Even though the master-slave PMGA does not provide a linear speed-up, it reduces the elapsed time per generation considerably, and for a large population the reduction is very significant. The master-slave PMGA only provides a lower bound speed-up, and the illustration in this paper demonstrated the potential benefits expected from parallel GA in general. To gain a considerable efficiency of using more processors, further investigation is needed on using more robust parallel GA that allow fewer communications than the master-slave PMGA uses.

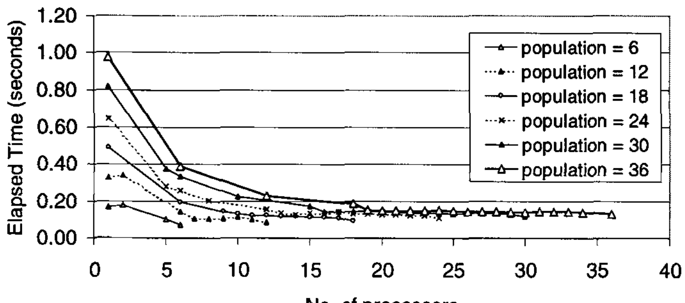

Figure 2. Elapsed time (seconds) per generation for PMGA

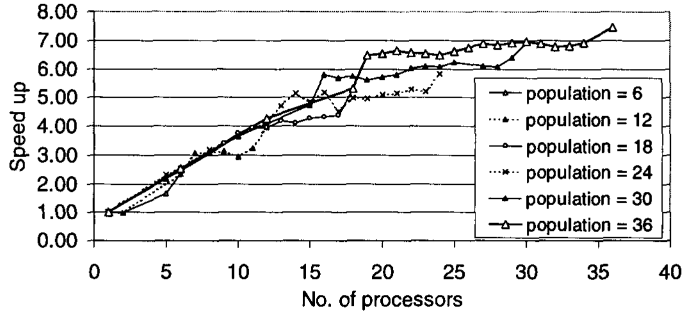

Figure 3. Speed up per generation for PMGA

References

1. Abu-Lebdeh, G., Benekohal, F. (1997). Development of Traffic Control and Queue Management Procedures for Oversaturated Arterial, *Transportation Research Record 1603*, pp.119-27.
2. Abu-Lebdeh, G., Benekohal, F. (1999). Convergence Variability and Population Sizing in Micro-Genetic Algorithms, *Computer-Aided Civil and Infrastructure Engineering*, vol. 14, pp. 321-34.
3. Cantu-Paz, E. (1999). *Designing Efficient and Accurate Parallel Genetic Algorithms*, IlliGal Report No. 99017.
4. Gartner, N.H. (1972). Constraining Relations Among Offsets In Synchronized Signal Networks, *Transportation Science, Vol. 6*, 1972, pp.88-93.
5. Girianna, M., Benekohal, R. F. (2002). Dynamic Signal Coordination for Networks with Oversaturated Intersections, *Transportation Research Record* (forthcoming).
6. Goldberg, D.E. (1989). *Genetic Algorithm in Search, Optimization and Machine Learning*, Addison Wesley Longman, Inc.
7. Harik, G., Cantu-Paz, E., Goldberg, D.E., and Miller, B. (1997). The gambler ruin problem, genetic algorithm, and the sizing of populations, in Bäck, T. (Ed.), *Proceedings of the IEEE Int. Conference on Evolutionary Computation*, pp. 7-12.
8. Krishnakumar, K. (1989). Micro-genetic Algorithms for Stationary and Non-stationary Function Optimization, in Proceedings of SPIE, Vol. 1196: *Intelligent Control and Adaptive Systems*, Philadelphia, Pa., pp. 289-96.