Journal of Intelligent Transportation Systems

ISSN: 1547-2450 (Print) 1547-2442 (Online) Journal homepage: http://www.tandfonline.com/loi/gits20

Using Genetic Algorithms to Design Signal Coordination for Oversaturated Networks

Montty Girianna & Rahim F. Benekohal

To cite this article: Montty Girianna & Rahim F. Benekohal (2004) Using Genetic Algorithms to Design Signal Coordination for Oversaturated Networks, Journal of Intelligent Transportation Systems, 8:2, 117-129

To link to this article: http://dx.doi.org/10.1080/15472450490435340

Published online: 24 Jun 2010.

Intelligent Transportation Systems, 8:117–129, 2004

ISSN: 1547-2450 print / 1547-2442 online
DOI: 10.1080/15472450490435340

Taylor & Francis
Taylor & Francis Group

Using Genetic Algorithms to Design Signal Coordination for Oversaturated Networks

MONTTY GIRIANNA
National Development Planning Agency (BAPPENAS), Jakarta, Indonesia

RAHIM F. BENEKOHAL
Department of Civil and Environmental Engineering, University of Illinois at Urbana Champaign, Newmark Civil Engineering Laboratory, Urbana, Illinois, USA

This article presents an algorithm to generate optimal (real-time) signal timings that distribute queues over a number of signalized intersections and over a number of cycles on any signalized intersection. A discrete-time signal-coordination model is formulated as a dynamic optimization problem and solved using Genetic Algorithms (GA). Signal timings for all intersections in the network during congested periods are decision variables and are represented in the individual GA candidate solutions. The algorithm is applied to a one-way arterial network with 20 signalized intersections. Depending on the traffic demand's variation and the position of critical signals, the algorithm intelligently generates optimal signal timing (offsets) along individual arterials. If critical signals are located at the exit points, the algorithm sets the optimal signal timing that protects them from becoming excessively loaded. If critical signals are located at the entry points, the algorithm ensures that queues are reduced or cleared before released platoons arrive at a downstream signal system. In this article, the simple genetic algorithm (SGA) with multiple epochs is used to solve the signal coordination problem. When a serial SGA 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. Master-slave SGA executed in a parallel computing machine is then used to reduce the execution time. We found that with a master-slave parallelism, SGA can be efficiently executed with significant speed-up, allowing the opportunity to implement the algorithm on real-time signal control systems.

Keywords Signal coordination; Genetic algorithms; Oversaturated

INTRODUCTION

While much research has been devoted to the development of signal control algorithms under normal traffic conditions, little information is available on methodologies that deal with oversaturated networks. Oversaturation occurs when queues of vehicles fill entire approaching streets and interfere with the performance of adjacent upstream intersections. None of the existing algorithms for signal coordination explicitly takes into account such oversaturation. As a result, implementation of the algorithms for congested networks has caused undesirable outcomes. SCOOT, for example, has performed well in moderate traffic conditions but has shown major deficiencies in oversaturated and highly fluctuating conditions (Yagar and Dion, 1996). In addition, SCATS proved more effective at reducing delay during low volume periods than high. This article presents the development of algorithms to coordinate signalized intersections on oversaturated networks. It describes a GA-based procedure used to solve the optimization problem and demonstrates the application of the procedure on an arterial network with 20 intersections. This

Address correspondence to Montty Girianna, the National Development Planning Agency, BAPPENAS, JI. Taman Suropati, Jakarta, Indonesia. E-mail: girianna@bappenas.go.id

article also shows the application of parallel genetic algorithm to improve the computation time. In the following section, the formulation of signal coordination is briefly described. Next, the basic feature of genetic algorithms is described. After the results are examined, the article concludes with findings and recommendations for future works.

SIGNAL COORDINATION

For oversaturated conditions, the traditional control policies, such as delay minimization, become secondary due to their ineffectiveness and inapplicability. The formation of queues and blockages are inevitable during oversaturation. Removal of queues and blockages must be the prime objectives (Roess et al., 1998). Note that this removal does not explicitly eliminate delay or the number of stops, as they are secondary exhibitions of the basic problems. Therefore, the effectiveness of signal coordination on oversaturated networks depends on controlling the formation and dissipation of queues at signals because queue length determines the control action and the available storage on intersection approaches. The queue dissipation objective requires a strategy that sets signal timing such that the number of vehicles released at every signal in congested networks is maximal (Girianna and Benekohal, 2002a; Lieberman et al., 2000; Abu-Lebdeh and Benekohal, 1997). After queues are cleared, resulting in undersaturated conditions, signal timings must provide offsets that generate traffic progression (green-bandwidth) for certain destinations.

Objective Function

Let $G = (N, L, P)$ denote a traffic signal network consisting of a set of signals N, a set of directional streets L, and a set of coordinated paths, P. Also let L_p be a set of streets along coordinated paths, and N_p is a set of signalized intersections situated on coordinated paths. A path for signal coordination starting from signal i to signal j is represented by $p_{ij} = \{s_i^h, \ldots, s_j^h\}$, where s_i^h denotes signal i with coordinated phase h, and s_j^h is signal j with coordinated phase h. For a network with two or more coordinated paths, a set of these paths is represented as $P = \{p_{ij}/s_i \in S_o \text{ and } s_j \in S_d\}$, where S_o is a set of signals from which traffic progression starts, and S_d is a set of signals at which traffic progression terminates. Different objectives of signal coordination may be made for different arterials. For queue dissipation purposes along coordinated arterials, offsets between two adjacent signals must provide an early queue dissipating opportunity at the downstream approach before a released platoon for upstream signals arrives. In addition, green time must be utilized effectively so any wasted green time, not used to process vehicles (new arrivals or queue), must be avoided. Two or more coordinated arterials can be intersected, run in parallel, or form open/close loops. The intersection between two or more coordinated arterials can be a critical intersection for the operation of signal timing for the arterials. A single coordinated arterial that crosses several coordinated arterials can be a critical arterial for the whole arterial system. A critical signal or a set of signals along the critical arterial determines the optimal signal timings for the entire network because, unlike the rest of signals, signals along the critical arterial serve two conflicting coordinated movements that compete for early green times to dissipate queue. When offsets and green time allocation fulfill these movements, signal timings for the remaining signalized intersections follow.

Equation (1) shows the objective function for signal coordination. The first term of the equation is the number of vehicles released at signalized intersections weighted by the ratio of the distance traveled to the maximum length of street in a network. The second term represents disutility function that penalizes the occurrence of queue at the end of green time along coordinated arterials. Departure rates are calculated for every multiple integer of a sample time. Queue is evaluated for the same time interval, but only queue at the beginning of cycle's green time is used in the equation. The objective function, represented as Z, is to maximize (over a set of green time) the net effect of released vehicles and the disutility function, both of which are calculated for the entire duration of an oversaturated period.

$$\text{Max} \quad Z = \sum_{t}^{T} \sum_{(i,j)\in L} \sum_{h}^{H} \frac{d_{i,j}}{d_{\max}} D_{i,j}^{h}(t) - \sum_{k}^{K} \sum_{(i,j)\in L_p} \delta_{i,j}(k) \max \begin{pmatrix} 0 \\ q_{i,j}^{h*}(k) - q_{i,j}^{\max} \end{pmatrix} \times \delta_{i,j}(k), d_{i,j}, d_{\max} > 0 \qquad (1)$$

$D_{i,j}^{h}(t)$ symbolizes departure flows (veh/sec) of phase $h \in H$ at signal j serving flows from signal i over a period of $[t\Delta T, (t+1)\Delta T]$. H is the total phase number, ΔT is a sample time interval (say 2, 3, 4, or 5 or more seconds), and $t = 1,2, \ldots, T$ is a discrete time index. $d_{i,j}$ is the distance from signal i to j, and $d_{\max}$ is the maximum length of streets in the network. $q_{i,j}^{h*}(k)$ is the number of vehicles in queue approaching signal j coming from signal i at the beginning of the downstream coordinated green

phase, h^*, in cycle k. h with a star indicates a coordinated phase. Depending on the signal plan and traffic movements served by coordinated phase, $q_{i,j}^{h*}(k)$ may refer to the left-turning or right-turning plus through movements, or total. $\delta_{i,j}(k)$ is a non-negative disutility factor whose values are determined based on a queue management strategy. In this article, $\delta_{i,j}(k) = 1$ for $(i, j) \in L_p$ and $k \in K$. K is the period of oversaturation in a cycle number, and T is the period of oversaturation in a sample time. Note that the second term (disutility function) of Eq. (1) is only for coordinated arterials, $(i, j) \in L_p$. Queue on all streets along both coordinated and noncoordinated paths must be less than the storage capacity during the oversaturation periods. The disutility function ensures that queue is less than storage capacity along coordinated arterials and the length of streets bounds the length of queue along noncoordinated arterials.

Constraint: Ideal Offsets

The first constraint is offsets between two signals along coordinated arterials. An offset between two signals is defined as the time difference between green time initiations of an upstream to that of an adjacent downstream signal. The phase of green time for both signals is not necessarily the same. If two signals are coordinated with the objective to dissipate queue at the downstream signal, an offset must be determined based not only on the distance between signals, speed of the released platoon, and platoon dispersion, but also the time required for the queue to dissipate. One cannot provide all phases to be coordinated with the upstream signals. Only an offset between a phase of downstream signals and a phase of the upstream signals that control coordinated movements is set ideal. For a multiphase signal plan, queues at the signal approaches are expected to dissipate for different directions. The direction for which the ideal offset is provided plays an important role in determining the magnitude of the offset.

To clarify the concept of ideal offsets, let us consider a two-signal model as shown in Figure 1. Suppose that one coordinates the signals for the northbound through traffic movement. Thus, as revealed in the figure, the through movement that enters signal i during phase three is coordinated with the through movement that departs signal j during phase one. A time-space diagram for the signal timing is also shown in the figure. A black bar indicates a red interval, and a white bar indicates the interval of effective green time. The offset between these two green phases is set ideal, which means it is set with the objective to dissipate queue of the coordinated movement at signal j approach. A set of queue at the beginning of phase h of cycle

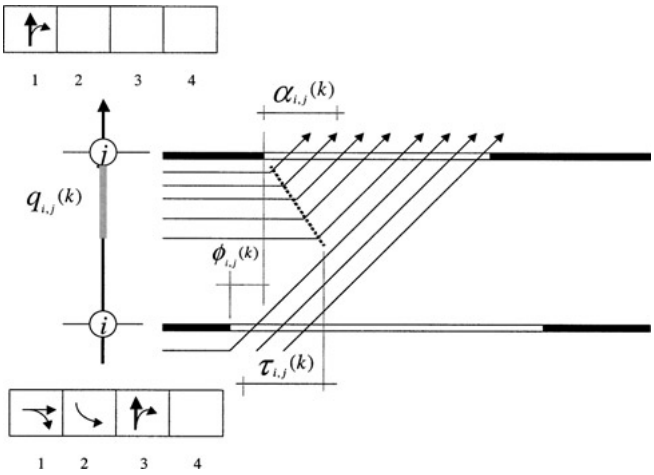

Figure 1 A two-signal system $i - j$ with phase 3 (signal i) and phase 1 (signal j) coordinated.

k is represented by $q_{i,j}^{h}(k) = \{q_{i,j}^{h,L}(k), q_{i,j}^{h,TR}(k)\}$, where $q_{i,j}^{h,L}(k)$ is a queue that is expected to turn left and $q_{i,j}^{h,TR}$(k) is a queue that is expected to turn right and to make a through movement. Thus, if green phase for through movement at signal j with $h^* = 1$, is coordinated with the green phase for that movement at signal i *with* $h^* = 3$, offsets between signal i and j, $\phi_{i,j}^{h*}(k)$, must satisfy Eq. (2).

$$\phi_{i,j}^{h*}(k) = \tau_{i,j}(k) - \alpha_{i,j}(k) = \frac{l_{i,j}(k)}{v_{i,j}} - \frac{q_{i,j}^{h*}(k) l_{veh}}{\lambda}$$

$$= \frac{d_{i,j} - q_{i,j}^{h*}(k) l_{veh}}{v_{i,j}} - \frac{q_{i,j}^{h*}(k) l_{veh}}{\lambda}$$

$$= \frac{d_{i,j}}{v_{i,j}} - \left(\frac{(v_{i,j} + \lambda) l_{veh}}{v_{i,j}\lambda} \right) q_{i,j}^{h*}(k)$$

$$\forall (i, j) \in L_p, \; k = 1, \ldots, K \tag{2}$$

Where $\tau_{i,j}(k)$ 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). v_{ij} is the speed of a released platoon, and $l_{ij}(k)$ is the unoccupied space along street (i, j) at the beginning of cycle k. $\alpha_{i,j}(k)$ is the time required for the tail to start moving and is calculated by dividing the queue length at the approach of signal j, $q_{i,j}^{h*}(k)$ (l_{veh}), by the starting shock-wave speed, λ. l_{veh} is the average length of vehicles. Note that $q_{i,j}^{h*}(k)$ is the queue for coordinated movement at the beginning of phase h^* of cycle k. For the case shown in Figure 1, $q_{i,j}^{h*}(k) = q_{i,j}^{1}(k)$, since at signal j the coordinated phase $h^* = 1$ and serves right-turning plus through movements.

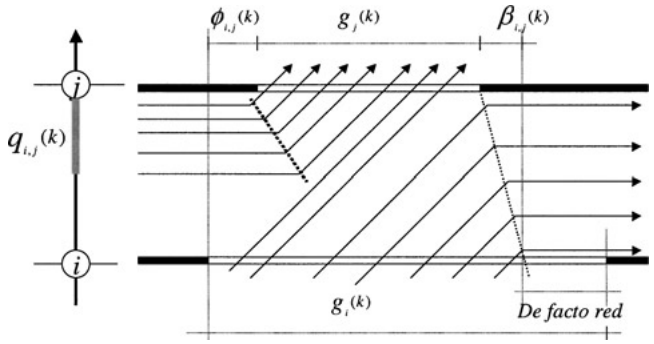

Figure 2 2 De facto red.

Constraint: De Facto Red

De facto red exists when the signal is green but traffic cannot proceed because of backed-up traffic on a receiving street. This situation is shown in Figure 2.

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 coordinated downstream signal, $g_j^{h*}(k)$, the offset between the two signals, $\phi_{i,j}^{h*}(k)$, and the time it takes for a stopping shock wave to propagate upstream, $\beta_{i,j}(k)$. This mechanism is formulated by Eq. (3).

$$g_i^{h*}(k) \le g_j^{h*}(k) + \phi_{i,j}^{h*}(k) + \beta_{i,j}(k) \quad \forall (i,j) \in L_p \tag{3}$$

Constraint: Coordinated Loops

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). The lock-in constraint is formulated by Eq. (4). $C_j(m)$ is a cycle length in cycle number m, $N(r)$ is a set of nodes of directed loop r, and $F(r)$ is the set of forward links in loop r, where traffic flow moves in the same direction as the loop's direction. $R(r)$ is the set of reverse links in loop r, where traffic flow moves in the opposite direction as loop's course. The green time of signal j at cycle k serving movements in loop r is represented as $g_{j,r}(k)$. n_k is an integer number indicating the cycle number, and R is the number of loops in the network. Δ corresponds to a lost green time.

$$\sum_{(i,j)\in F(r)} \phi_{i,j}(k) - \sum_{(i,j)\in R(r)} \phi_{i,j}(k) + \sum_{j\in N(r)} (g_{j,r}(k) + \Delta)$$

$$= \sum_{m=k,\, j\in N(r)}^{k+n_k} C_j(m) \quad \forall r \in R \tag{4}$$

Constraint: Queue Storage Capacity

Queue along noncoordinated arterials must be less than the storage capacity of approach links. This is formulated in Eq. (5), where L_s is a set of streets carrying noncoordinated movements. The equation dictates that queue length at the end of each time interval must not exceed available queue storage capacity of the approaches, $q_{i,j}^{\max}$, which depends on the length of streets and the length of average vehicles.

$$q_{i,j}^h(k) \le q_{i,j}^{\max} \quad q_{i,j}^h(k) \le \frac{d_{i,j}}{l_{veh}} \quad (i,j) \in L_s \tag{5}$$

Equation (5) dictates that along noncoordinated arterials, a certain number of vehicles stored in queue are tolerated, but the length of the queue has to be short enough such that it does not block the traffic movements of the upstream intersections. For coordinated arterials, the number of queue stored along coordinated arterials depends on queue management strategies used for signal coordination (see the second term of the objective function, Eq. [1]). One extreme case is when one insists on not having queues at all along coordinated arterials and, thus, $q_{\max} = 0$. Another extreme case is when one allows storing a large number of vehicles in queue during a congested period, thus $q_{\max}$ = storage capacity.

Constraint: Control Variables

All control parameters should be within reasonable ranges. Eq. (6) formulates the allowable effective green time, where $g_{\min}$ is the lower bound and $g_{\max}$ is the upper bound, and queues at $k = 0$ or $t = 0$ are initially defined for all signals' approaches.

$$g_{\min} \le g_j^h(k) \le g_{\max} \;\; \forall j \in N, \;\; \forall h \in H, \;\; k = 1, \ldots, K \tag{6}$$

Network Flows

Flows and queues on a signalized intersection network can be measured at certain discrete times. If cycle lengths are approximately the same for all signalized intersections, then the measurement can be made at end of every cycle. The total number of vehicles processed by a network can then be calculated by adding all departure flows leaving intersections measured at the end of cycle. On the other hand, if the cycle lengths vary, the measurement of flows and queues should be made at smaller time intervals than cycle lengths. Figure 3 reveals a two-signal model $i - j$ connecting two intersections. In this article, the signal

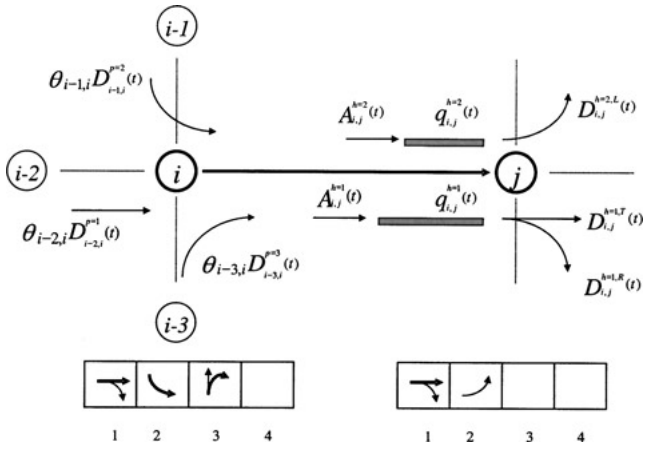

Figure 3 Flows on streets connecting signals i and j.

network is loaded using a discrete time approach with sample time ΔT. We define $I^h_{i,j}(t)$, $A^h_{i,j}(t)$ and $D^h_{i,j}(t)$ to be inflow, arrival and departure flows of phase $h \in H$ over a period of $[t\Delta T, (t+1)\Delta T]$ on an $i-j$ street system.

This system serves traffic flow movement that enters street (i, j) through signal i (upstream) and the departing movement by way of signal j (downstream). Both signals work with a four-phase plan, $H = 4$. The phase number for signals i and j and its associated traffic movements are shown in the figure. The inflow for a given phase h of signal j moving from phase p of signal i during interval t is formulated by Eq. (7), where U_h is set of signal phases at the upstream signal that feeds traffic for phase h of the downstream signal. $\theta_{b,i}$ is the percentage of the departed traffic volume of upstream streets (b, i) that enters road section (i, j), and $b \in B_i$, where B_i is a set of upstream intersections connected to intersection i.

$$I^h_{i,j}(t) = \sum_{p \in U_h, b \in B_i} \theta_{b,i} D^p_{b,i}(t) \tag{7}$$

For the system represented in the figure, $I^h_{i,j}(t) = (\theta_{i-1,j})\, D^{p=2}_{i-1,j}(t) + (\theta_{i-2,j}) D^{p=1}_{i-2,j}(t) + (\theta_{i-3,j}) D^{p=3}_{i-3,j}(t)$. The demand for left-turning movement expected to leave signal j is represented by $A^{h=2}_{i,j}(t) = (\theta^L_{i,j})(I^h_{i,j}(t))$, where $\theta^L_{i,j}$ is a left-turning factor for road section (i, j). Similarly, $A^{h=1}_{i,j}(t) = (1-\theta^L_{i,j})(I^h_{i,j}(t))$ for through and right-turning movements. These demands arrive at the end of downstream queues or stop lines at some distance away. If we assume that the departed platoon disperses, arrival flows can be formulated following Robertson's platoon dispersion model, as shown in Eq. (8). The smoothing factor F equals $1/(1+(\gamma)(\tau_{i,j}))$, where γ is the platoon dispersion factor, empirically derived (0.5), and τ_{ij} is, as defined before, the cruise travel time of a released platoon, factored by 0.8 (Wallace and Kenneth, 1991). Note that the equation assumes deterministic flows with no source and sink nodes between intersections.

$$A^h_{i,j}(t) = F\theta_{i,j} I^h_{i,j}(t-\tau_{i,j}) + (1-F) A^h_{i,j}(t-1) \tag{8}$$

The departure flows (veh/sec) of phase $h \in H$ at signal j serving flows from signal i over a period of $[t\Delta T, (t+1)\Delta T]$ is denoted by $D^h_{i,j}(t)$ and its magnitude is expressed by Eq. (9).

$$D^h_{i,j}(t) = \min \begin{pmatrix} c^h_j(t)\Delta T \\ A^h_{i,j}(t)\Delta T + q^h_{i,j}(t-1) \end{pmatrix} \quad \forall i, j \in L \tag{9}$$

Where $c^h_j(t)$ indicates the capacity during effective green interval and it equals the saturation flow s^h_j (veh/sec) if phase h has a right of way. Otherwise, it equals zero. Queue cannot be negative, so Eq. (10) also governs. The queue length during phase h at the end of the time interval t, $q^h_{i,j}(t)$, is the sum of any queues transferred from the previous time interval $(t-1)$ and the difference between input and output at the current time interval.

$$q^h_{i,j}(t) = \max \begin{pmatrix} 0 \\ q^h_{i,j}(t-1) + \left(A^h_{i,j}(t) - D^h_{i,j}(t)\right) \end{pmatrix} \tag{10}$$

Problem formulation

The structure of a signal coordination model is formulated as a constrained dynamic optimization problem with the objective to maximize the number of vehicles released by a signal network. The disutility function was introduced to account for queue accumulation along coordinated arterials. If queue accumulates, the objective function is reduced. The effective green time intervals for all signalized intersections and for the duration of an oversaturation are the decision variables. The objective function follows Eq. (1). A set of constraints are defined in Eqs. (2–6), and network loading is evaluated using Eqs. (7–10). The optimization problem is then summarized as follows.

$$\text{Max } Z = \sum_t^T \sum_{(i,j)\in L} \sum_h^H \frac{d_{i,j}}{d_{\max}} D^h_{i,j}(t) - \sum_k^K \sum_{(i,j)\in L_P} \times \delta_{i,j}(k) \max \begin{pmatrix} 0 \\ q^{h*}_{i,j}(k) - q^{\max}_{i,j} \end{pmatrix} \tag{11}$$

Subject to:

$$\phi_{i,j}^{h*}(k) = \frac{d_{i,j}}{v_{i,j}} - \left(\frac{(v_{i,j}+\lambda)l_{veh}}{v_{i,j}\lambda}\right) q_{i,j}^{h*}(k) \quad \forall (i,j) \in L_p, \quad k = 1, \ldots, K$$

$$g_{i,j}^{h*}(k) \le g_j^{h*}(k) + \phi_{i,j}^{h*}(k) + \beta_{i,j}(k) \quad \forall (i,j) \in L_p, \quad k = 1, \ldots, K$$

$$\sum_{(i,j)\in F(r)} \phi_{i,j} h(k) - \sum_{(i,j)\in R(r)} \phi_{i,j}^{h}(k) + \sum_{j\in N(r)} \left(g_{j,r}^{h}(k) + \Delta\right) = \sum_{(m=k,j)\in N(r)}^{k+n_k} C_j(m) \quad \forall h \in H, \quad r \in R, \quad k = 1, \ldots, K$$

$$q_{i,j}^{h}(t) \le \frac{d_{i,j}}{l_{veh}} \quad \forall h \in H, \quad \forall (i,j) \in L_s, \quad t = 1, \ldots, T$$

$$g_{\min} \le g_j^h(k) \le g_{\max} \quad \forall j \in N, \forall h \in H, \quad k = 1, \ldots, K$$

Network loading is made using the following set of equations:

$$I_{i,j}^{h}(t) = \sum_{p\in U_h, b\in B_i} \theta_{b,i} D_{b,i}^{p}(t) \quad h \in H, t = 1, \ldots, T$$

$$A_{i,j}^{h}(t) = F\theta_{i,j} I_{i,j}^{h}(t - \tau_{i,j}) + (1 - F)\, A_{i,j}^{h}(t-1) \quad h \in H, \quad (i,j) \in L, \quad t = 1, \ldots, T$$

$$D_{i,j}^{h}(t)\Delta T = \min \begin{pmatrix} c_j^h(t)\Delta T \\ A_{i,j}^{h}(t)\Delta T + q_{i,j}^{h}(t-1) \end{pmatrix} \quad h \in H, \quad (i,j) \in L, \quad t = 1, \ldots, T$$

GENETIC ALGORITHMS

Because the signal coordination model finds optimal signal timing for the entire period of oversaturation, the problem of signal coordination becomes a large combinatorial optimization problem and cannot be efficiently solved using traditional calculus-based optimization techniques. We use SGA to solve the problem, a search technique based on the mechanics of natural selection and natural genetics (Goldberg, 1989). SGA searches for a set of optimum effective green times by maximizing the objective function subject to a set of constraints and network loading equations as defined in the previous sections. SGA transforms the constrained signal coordination problem into an unconstrained problem by associating penalty with all constraint violations (Goldberg, 1989; Dasgupta and Michalewicz, 1997). SGA's fitness functions equal the objective function and are degraded in relation to the degree of constraint violations.

Recall that there are two different classes of equations defined in the previous section. The first is a set of equations that are needed for traffic simulation, that is, Eqs. (7–10), and SGA directly uses these equations to calculate traffic flows and queues for a given set of candidate solutions, that is, a set of green times. The second is a set of equations that determine the feasibility of solutions, that is, Eqs. (2–6). The range of green times, that is, Eq. (6), can be checked without requiring a traffic simulation. In this article, the green time is represented as a four-bit binary string with all zeros representing the minimum green time $g_{\min}$, and with all ones representing the maximum green time $g_{\max}$. Any other string is decoded to a green time, g, as formulated by Eq. (12), where DV is the decoded value of a string. Thus, for $g_{\min} = 20$ and $g_{\max} = 80$, the decoded value for string 1001 is $(1)(2^3) + (0)(2^2) + (0)(2^1) + (1)(2^0) = 9$, and the corresponding green time for such a string is $20 + [(80–20)/(2^4-1)] \times 9 = 66$ seconds. With this procedure, green time is always within the specified range.

$$g = g_{\min} + \left(\frac{g_{\max} - g_{\min}}{2^{d-1}}\right) DV \tag{12}$$

Equations (2–5) require a simulation to check for a violation and may have limits that are solution dependent. The basic procedure is to define the fitness value of an individual i by extending the domain of the objective function Z_i using Eq. (13), where μ_j is a penalty coefficient for constraint j, m is the number of implicit constraints, and H_j denotes j's constraint function (inequality and equality). $C_{\min}$ is an input coefficient, as the absolute value (or larger) of the worst possible value of the augmented objective function, Z_i-$\sum \mu_j H_j$. This input coefficient is introduced to overcome the negative value of the augmented objective function (Goldberg, 1989), which occurs during early generations of SGA as constraints are violated. Eq. (13) becomes a mapping of the objective function to fitness form.

$$\text{fitness}_i = C_{\min} + \left(Z_i - \sum_{j-1}^{m} \mu_j H_j\right) \tag{13}$$

Equation (4) requires that offsets and green times around any loop within a network equals an integer multiple of the

cycle time. This constraints is not active if a coordinated signal is an open-loop system, that is, when multiple coordinated arterials cross a single coordinated arterial, and it is active when a coordinated signal form a closed-loop system. For an open-loop system, the augmented objective function becomes as formulated below.

$$\begin{aligned}\text{Max} \quad & C_{\min} + \sum_{k}^{K} \sum_{(i,j)\in L} \sum_{h}^{H} \frac{d_{i,j}}{d_{\max}} D_{i,j}^{h}(k) - \sum_{k}^{K} \sum_{(i,j)\in L_P} \\ & \times \delta_{i,j}(k) \max \begin{pmatrix} 0 \\ q_{i,j}^{h*}(k) - q_{\max} \end{pmatrix} - \mu_1 \sum_{k,(i,j)\in L_p}^{K} \\ & \times \left(\phi_{i,j}^{h}(k) - \left(\frac{d_{i,j}}{v_{i,j}} - \frac{(v_{i,j}+\lambda) l_{veh}}{v_{i,j}\lambda} q_{i,j}^{h*}(k) \right) \right)^2 - \mu_2 \\ & \sum_{k,(i,j)\in L_p}^{K} \max \begin{pmatrix} 0 \\ g_i(k) - (g_j(k) + \phi_{i,j}(k) + \beta_{i,j}(k)) \end{pmatrix} \\ & -\mu_3 \sum_{k,(i,j)\in L}^{K} \max \begin{pmatrix} 0 \\ q_{i,j}^{h}(k) - \frac{d_{i,j}}{l_{veh}} \end{pmatrix} \end{aligned} \tag{14}$$

Working with a set of individuals coded in a fixed-length string, SGA selects good individuals according to their fitness values from a mating pool. The intention of the selection procedure is to pick above-average individuals (a set of building blocks) from the current population and to insert their duplicates in the mating pool. There exist a number of selection procedures, such as proportional, (Goldberg, 1989), tournament (Goldberg and Deb, 1991), and truncation selections. Proportional selection, for example, selects an individual with a probability proportional to its fitness value. Tournament selection randomly chooses a number of individuals from the population, picks the best of them, and stores them in the mating pool. Truncation selection picks the portion of the best solutions from the population. The selection is repeated until enough solutions are selected. Crossover is the next genetic operator applied to the individual of the mating pool and used to explore the space of solutions but preserve the available information stored in the parents' individuals. Two individuals are randomly picked from the mating pool, and they are cut at a randomly selected site(s). The portions of both individuals bracketed by this site(s) are swapped among themselves to create two new individuals. In the single-point crossover, the search is not extensive, but the information of the parents is preserved at a maximum level. On the other hand, in a larger number of crossover points, the search is more extensive, but the information is preserved at a lower level. To preserve some of the previously found good individuals, not all individuals are participating in the crossover operation. Crossover with p_c probability is applied. To maintain the diversity of individuals, mutation is applied to them. This second genetic operator serves as a local search and is applied by changing a 1 to a 0 and vice versa, with a small mutation probability, p_m. The complete application of selection and the two genetic operators (crossover and mutation) to the whole population concludes one generation of SGA.

Ideally, with a large population size, SGA converges to good solutions. However, a large population size undesirably affects the convergence time. To overcome this situation, we use multiple executions, that is, we use a small population size, reinitialize the population, and restart the search for optimum solutions after an unsuccessful convergence (Krishnakumar, 1989). Every run of genetic algorithms between two restarts is called an epoch. With the multiple epochs, at least one good individual of the previous epoch and generation is kept as one member of the current candidate solutions. The basic procedure of multiple-epoch SGA is described as follows:

1) Select randomly either a population of size n, or n − 1 and one good individual from any previous search or epoch
2) Evaluate fitness and determine the best individual, and carry it to the next generation (elitist strategy)
3) Determine the remaining individuals using genetic operators, such as selection and crossover ($p_c = 1$)
4) Check for convergence. If the search converges to a solution proceed to step 1) (one epoch), otherwise proceed to step 2). SGA converges when the highest fitness value is sufficiently close to the average fitness of the population.

The process of genetic operation continues until the prescribed limit of generation or when the number of functional evaluation (FE) is reached. FE is a function of the population size and the number of generations for SGA to converge. When SGA evaluate the fitness function for one generation, FE equals exactly the population size. Alternatively, SGA stops when the highest fitness value is sufficiently close to the average fitness of the population. In this article, SGA stops when FE = 20,000. Note that SGA with multiple epoch does not apply a mutation operator ($p_m = 0$), since diversity can be maintained from epoch to epoch (due to population re-initiation).

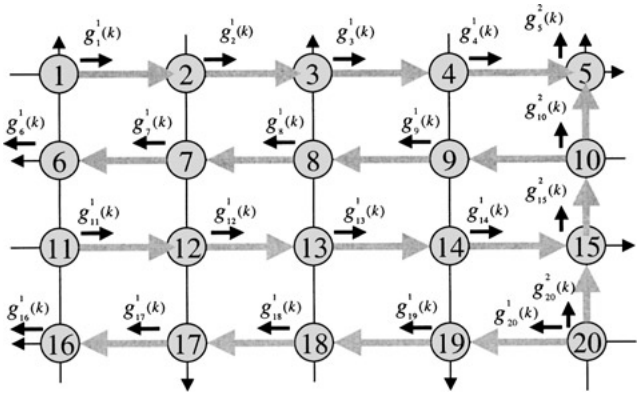

Figure 4 A one-way arterial network with 20 signals.

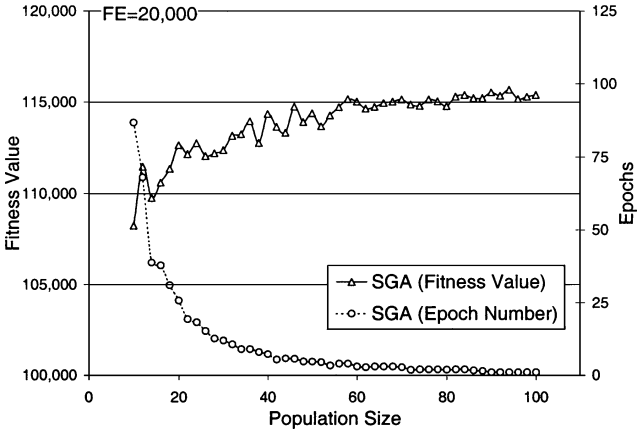

Figure 5 Fitness values for SGA executed with multiple epochs (FE = 20,000).

SOLVING SIGNAL COORDINATION PROBLEMS

The signal coordination model is used to coordinate signals on a one-way two-lane street system with $N = 20$ intersections as shown in Figure 4. Each signal works with a two-phase plan and turning movements are not allowed. Signal coordination is made for signals along a northbound arterial from signal 20 to 5 and for those along arterials running E-W. The thick lines in the figure show the coordinated movements. Two arterials are assumed to carry very heavy traffic, that is, 2,000 vehicles per hour per lane (vphpl). These are the northbound arterial from signal 20 to signal 5 and the westbound arterial from signal 10 to signal 6.

Furthermore, the eastbound traffic enters the network at signal 1 and signal 11 with a flow rate of 1,800 vphpl, whereas the westbound traffic enters the network at signal 20 with a flow rate of 1,800 vphpl. These are defined as major movements. In addition, all minor traffic enters the network with a flow rate of 1,500 vphpl at signals 2 and 4 for the southbound traffic, and at signals 16 and 18 for the northbound. These entry flows are assumed constant during an oversaturated period. For coordinated movements, $g_{\max} = 90$ and $g_{\min} = 30$ seconds; for noncoordinated movements, $g_{\max} = 60$ and $g_{\min} = 20$ seconds. The lost green time = 5 seconds. The number of arterial lanes is 2. Speed limit, or desired speed, equals 40 ft/sec, while vehicle acceleration/deceleration is 4 ft/sec^2. Saturation flow = 1,800 vphpl. Initial queues at all coordinated approaches are 20 vehicles per lane. Effective vehicle length is 25 ft. Starting and stopping shock wave speeds are 16 ft/sec and 14 ft/sec, respectively. Flows and queues are evaluated at sample time $\Delta T = 10$ seconds, and the duration of oversaturated period $T = 15$ minutes, or K is about 15 cycles.

Every possible value of effective green times defined in the genetic solutions is first coded in a fixed-length of binary substring. The length of a substring is decided by the precision needed in a variable. In this article, 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 is $(80-20)/5 = 12$. This number can be set equal to 2^d and d can be computed as $\log_2$ (12), and d should equal at least four bits of binary string. The string length is $(d)(m)(K) = (4)(40)(15) = 2400$. The average of fitness value, the number of functional evaluation, as well as the execution time are recorded.

Figure 5 displays the fitness values of SGA executed with multiple epochs for FE = 20,000. For this functional evaluation threshold, SGA with a small population size can be executed with a larger number of epochs. As shown in the figure, for population size $n = 10$, SGA is executed with $e = 87$ epochs. As a population size increases, the number of epoch decreases. Thus, for example, for SGA when $n = 20$ or larger, the number of epochs $e = 20$ or fewer, and when n is close to 80, only one epoch is executed. Note that the decrease in the epoch size is associated with the increase in fitness values, that is, quality solutions. For $n = 10$ or $e = 87$, for instance, the fitness value for SGA is 108,200, while for $n = 100$ or $e = 1$, it is 115,400.

As shown in the figure, starting from $n = 100$, decreasing the population size (a larger number of epochs) does not considerably worsen the fitness value until $n = 60$. With $n = 60$, the fitness value is about 115,000. Decreasing further the population size, however, greatly affect quality solutions. For $n = 10$, for example, SGA provides fitness value of approximately 108,250, which is about 6.0 % less than that with $n = 100$. This worst quality solution is due to a finite rework occurring for SGA with large epochs. A smaller population size causes considerable penalties to the run duration due to large rework for

resolving those parts of the solutions that have already been decided in the previous generations. On the other hand, a larger population size with marginal improvement in quality solutions is not a better option. Thus, for efficiency, suffice it to use $n = 60$ or $e = 3$.

TRAFFIC PROGRESSION AND QUEUE DISSIPATION

Figure 6(a) illustrates how queues and offsets are interrelated and how the load of traffic is distributed among signals along the eastbound arterial, $p_{11,15}$. Each graph corresponds to a street section between two intersections. A graph for signal 14–15, for example, shows the queue of vehicles, released by signal 14, on the approach to signal 15 and shows the offset between signals 14 and 15. The heavy load of the queue-dissipation process at signal 15 (critical signal) postpones the green initiation of signal 14 (upstream signal) by setting a negative offset, thus causing a postponement of local queue dissipation along the approach to signal 14. The graph shows an initial queue of 40 vehicles and it cannot be cleared until 12 cycles. The mechanism of load shifting among signals propagates upstream. Notice that the positive offsets (forward traffic progression) are set at an earlier stage of cycles in the upstream direction. For example, an offset of larger than 15 seconds is set after cycle 9 at signal 15, but it is after cycle one at signal 12. However, this does not mean that the queue-dissipation process propagates to downstream signals. In fact, it is the other way around. The reason why early positive offsets are set at upstream signals is that the algorithm restrains entering traffic by allocating close to the minimum green times at the entry signals, which in turn releases existing local queues downstream. It is not shown in the figure, the queue of vehicles on the approach to the entry signals, that is, signal 11, increases.

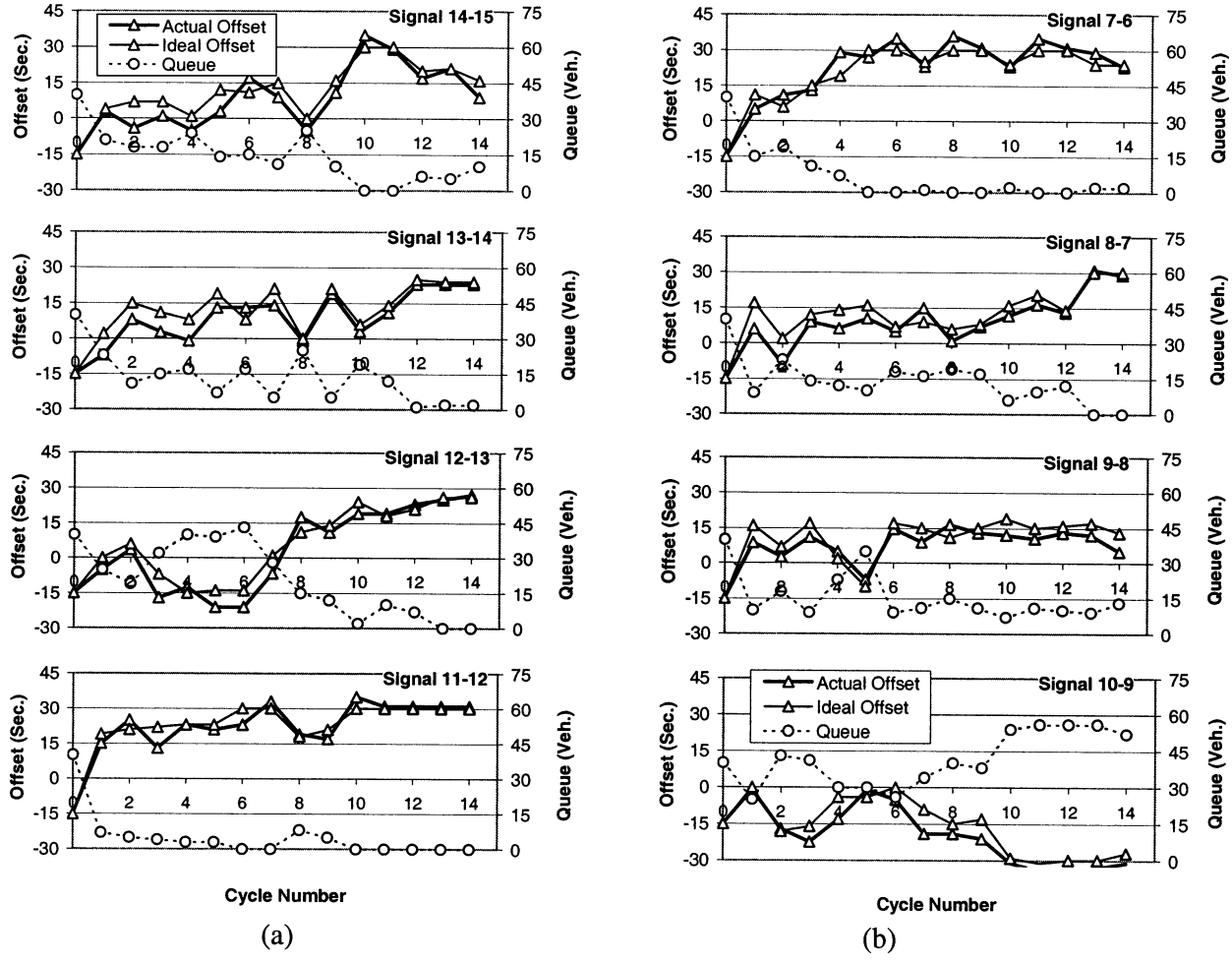

Figure 6 Dynamic of offsets and queues. (a) The eastbound coordinated arterial, $p_{11,15}$ with signal 15 as the critical signal. (b) The westbound coordinated arterial, $p_{10,6}$ with signal 10 as the critical signal.

In contrast, the queue-offset relation along the westbound arterial, $p_{10,6}$, shows that earlier positive offsets and the queue-dissipation process occur at the downstream signals, as shown in Figure 6(b). This is expected since the exit point (signal 6) of this coordinated arterial is not the critical one and is not as restricted as the exit signal on eastbound arterials. Therefore, the algorithm can release more vehicles at the entry signal (signal 10) causing an increase queue size at the immediate downstream section, that is, on the approach of signal 9. It is shown that queues at the further downstream signals can be released faster than the time it takes for vehicles (discharged at the entry signals) to travel to the exit signals. Consequently, the positive offsets are achieved earlier at downstream signals.

In general, forward traffic progression is achieved after several cycles of a signal-coordination process (see Figure 7). Depending on the magnitude of the local queues, the algorithm sets the appropriate offsets to dissipate local queues during the early cycles, and sets less green time to control arrivals. The magnitude of local queues and the position of critical signals determine the beginning of forward progression. When the critical signal is located at the exit point, the progression is expected to occur at a later stage, and when it is located at the entry point, the progression tends to occur at an earlier stage. This mechanism of traffic progression is expected because a system with a critical signal at the exit point operates in a more restrictive environment. The green bandwidth along coordinated arterials also varies over the cycles depending on the existence of local queues and on the magnitude of arrival volumes. This variation of green bandwidth reflects the ability of the algorithm to effectively allocate green times. An inadequate or wasted green time at any signal would jeopardize the performance of the whole signal network during an oversaturated period. Green time must be used productively and therefore is reflected in the variation of the green bandwidth.

The algorithm presented in this article is extended for signal coordination on a two-way oversaturated arterial network (Girianna and Benekohal, 2002c). When oversaturated conditions occur on a two-way street system, signal coordination should be designed to initially clear queue in the primary direction (large negative offsets). When the queue is dissipated and, thus offsets can be positive and can potentially provide forward traffic progression in the primary direction, the processes of queue dissipation can then be made for the opposing traffic. Thus, one can start releasing queue in the opposing direction provided that the process of queue dissipation in the primary direction has been made.

SPEEDING-UP SGA COMPUTATION TIME

In order for the SGA-based signal coordination to be applicable to real-time systems, reducing the duration of SGA computation time becomes a critical issue. To reduce the computation time, genetic algorithms need to be executed on parallel machines. SGA 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 parallelism provides a lower bound of speed-up, but it is useful as a benchmark for measuring the performance of other types of parallel genetic algorithms. The parallel speed-up of the master-slave SGA is defined as T_s/T_p. Where T_s is the elapsed time for one generation of serial SGA and T_p is that for one generation of the master-slave SGA. To evaluate the speed-up of master-slave SGA, one processor is first used, and with an increment of one processor, SGA is subsequently executed until the number of processors equals the population size. Thus, for example, for population size $n = 30$, the master-slave SGA 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

Figure 7 A space-time diagram for the eastbound coordinated arterial from signal 11 to 15, $p_{11,15}$ (Numbers in parentheses indicate signal numbers, "==" for red interval and "—" for effective green).

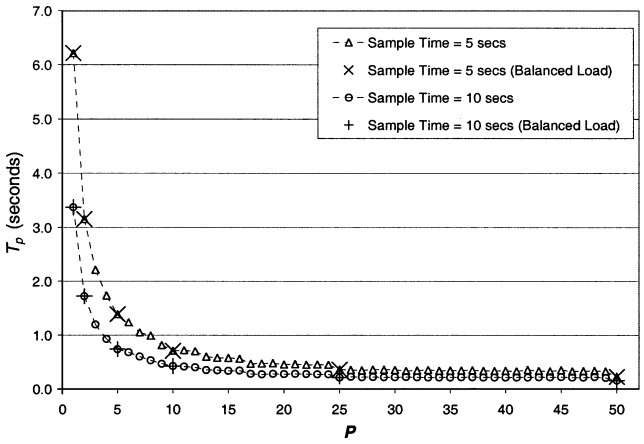

Figure 8(a) SGA Elapsed time per generation.

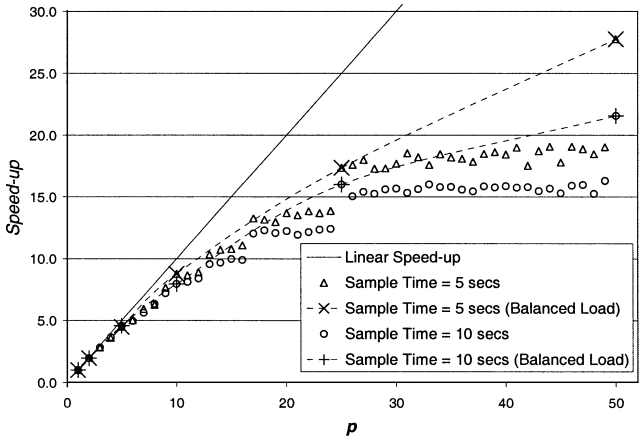

Figure 8(b) SGA speed-up per generation.

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 evaluate three individuals, say processors one to six, and the remaining evaluate only two.

Figure 8 shows the elapsed time and speed-up per generation for different sample time intervals ($\Delta T = 5$ and 10 seconds) and different number of processors. When $\Delta T = 5$ seconds, flows and queues on street networks are evaluated for every 5 seconds. The two measures are represented as a function of the number of processors used. As expected, for a given number of processors, the case with $\Delta T = 10$-second sample times requires less elapsed time for GA to evaluate. When two processors are used, the elapsed time for 10-second interval decreases from 3.369 seconds (one processor) to 1.722 seconds, that is, the speed-up is approximately two. Adding more processors provides less elapsed time and increased speed-up. A linear speed-up is maintained approximately until $p = 5$. When the number of processors equals the population size, that is, $p = n$, the maximum speed-up is attained. Note that the master-slave GA is more effective when the balanced-load conditions occur (shown as dashed curves in Figure 8[b]). For example, with one processor evaluating two individuals, that is, $p = 25$ (balanced-load conditions), the speed-up does not increase by adding more processors until $p = 50$. This is because for $25 < p < 50$, at least one processor has to evaluate two individuals, and this bottleneck determines the duration of the entire computation time. For the case with a five-second sample time interval, a linear speed-up is maintained approximately until $p = 10$. Similarly, the master-slave GA is effective when balanced-load conditions occur, and the maximum speed-up is attained when $p = n$. The maximum speed-up for a case with $\Delta T = 5$ seconds, that is, 28, is larger than that for a case with $\Delta T = 10$ seconds, that is, 22. The optimal number of processors depends on the ratio of the time for SGA to evaluate one candidate solution (evaluation time) to the time required for processors to communicate (communication time). Mathematical formulation to obtain the number of processors and the empirical results of the formulation are described elsewhere (Girianna and Benekohal, 2002d).

Efficiency measures the deviation of master-slave GA's performance from the ideal conditions (linear speed-up) and is formulated as the ratio of the speed-up to the number of processors. Communication time, C_t, and the degree of utilization of processors determine the master-slave GA efficiency. When the number of processors used is small, the communication time, C_t, is considerably small compared to evaluation time, E_t, and, thus, the elapsed time is greatly determined by E_t. However, as a larger number of processors is used, the gap between C_t and E_t becomes closer and both C_t and E_t determine the elapsed time. When $p = 3$ for a case with 10-second sample times, for example, $E_t = 1{,}135.0$ milliseconds and $C_t = 9.8$ milliseconds, or $E_t/C_t = 116$, that is, the evaluation time is 116 times larger than the communication time. When $p = 30$, $E_t = 139.0$ milliseconds and $C_t = 25.5$ milliseconds, or $E_t/C_t = 5.4$. With more processors, the gap between $E_t$and C_t becomes smaller. Clearly, the effect of C_t on the elapsed time becomes significant as a larger number of processors used. In other words, as p increases, a larger fraction of time is devoted to exchange the data among processors, decreasing the efficiency.

Moreover, when a larger portion of processors are idle, efficiency drops very quickly. This occurs when the master-slave GA is executed with unbalanced-load conditions. When $p = 25$ (balanced-load conditions), the

speed-up for a case with five-second sample times is 17.4, see Figure 8(b), and the associated efficiency $E = (T_s/T_p)/(p) = (17.4)/(25) = 69\%$. When $p = 36$ (unbalanced-load conditions), the speed-up only increases to 18, but E drops to 50% because a larger fraction of idle time occurs. In other words, $p = 25$ is more efficient than $p = 36$. When p further increases to 50 (balanced-load conditions), the speed-up is 28 and E increases to 56%. Accordingly, the merit of master-slave GA is more noticeable for balanced-load conditions.

CONCLUSIONS AND RECOMMENDATIONS

This article presented an algorithm to design signal coordination for networks with oversaturated intersections. The algorithm successfully provided signal timing that is responsive to the temporal variation of arrival volumes and queues at the intersection approaches. It generated an online load-balancing mechanism in which critical intersections are protected from becoming oversaturated. When critical intersections were located at exit points of the network, during the first few cycles all upstream signals managed volumes of entering traffic by setting lower green times. This was combined with setting negative offsets of longer duration at the exit signals. Later, the positive offsets were gradually set as the algorithm promoted forward green bands. When critical signals were located at entry points, the negative offsets were maintained for longer duration at the entry signals. This was done to ensure that all local queues were cleared before more vehicles arrived at downstream signals.

Applied using two or more processors to solve signal control problems, SGA can be executed faster. The SGA with a master-slave parallelism provided a linear speed-up, reduced the elapsed time per generation considerably, and performed better when a smaller fraction of running time is devoted to communication costs. It is well suited to signal coordination problems when a higher accuracy in evaluating flows and queues is demanded. The master-slave SGA, however, only provided a lower bound speed-up, indicated by low efficiency, and the illustration in this article demonstrated the potential benefits expected from parallel SGA.

The research on signal coordination presented in this article can be further extended into several areas. One aspect is to extend the development of signal coordination that covers major turning movements, stochastic effects of traffic flows, and a demand-responsive signal plan. Decomposing a signal network into a smaller network becomes necessary when a signal network is large, or when the fluctuation of traffic at a certain corridor considerably varies. Signal coordination with network partition and a procedure that can control queue management on one network partition and the adjacent subnetworks should be developed. In addition, to gain a considerable computing efficiency of using more processors, further investigation is needed on using more robust parallel SGA that allow fewer communications than the master-slave SGA uses. Hybrid-genetic algorithms are potential candidates for use to solve signal coordination problems. The search can start with SGA to obtain a basin of optimal solutions and within this basin, greedy or calculus-based optimization techniques are used to find the best solutions. For a larger signal network, the merit of exploring distributed signal coordination should be made. With the existence of parallel SGA strategies, the exploration seems to be more useful. Furthermore, analysis is needed on the scalability of the algorithms for larger networks.

REFERENCES

Abu-Lebdeh, G. and Benekohal, F. (1997). Development of traffic control and queue management procedures for oversaturated arterial, *Transportation Research Record*, **1603**, 119–127.

Cantu-Paz, E. (1999). *Designing Efficient and Accurate Parallel Genetic Algorithms*, IlliGal Report No. 99017.

Dasgupta, D. and Michalewich, Z. Eds. (1997). *Evolutionary Algorithms in Engineering Applications*, Springer-Verlag, Berlin, Germany.

Gartner, N.H. (1972). Constraining relations among offsets in synchronized signal networks, *Transportation Science*, **6**, 88–93.

Girianna, M. and Benekohal, R.F. (2002a). Dynamic signal coordination for networks with oversaturated intersections, *Transportation Research Record*, **1811**, pp. 122–130.

Girianna, M. and Benekohal, R.F. (2002b). Intelligent signal coordination on congested networks using parallel micro genetic algorithms, In *Proceedings of the Application of Advanced Technologies to Transportation (AATT 2002) 7th International Congress*, Ed. K.C., Wang. Cambridge, MA.

Girianna, M. and Benekohal, R.F. (2002c). Signal coordination for a two-way street network with oversaturated intersections, presented and submitted for publication in the *Proc. 82nd Transportation Research Board Annual Meeting*, Washington, DC.

Girianna, M. and Benekohal, R.F. (2002d). Solving signal coordination problems using master-slave genetic algorithms, presented and submitted for publication in the *Proc. 82nd Transportation Research Board Annual Meeting*, Washington, DC.

Goldberg, D.E. (1989). *Genetic Algorithm in Search, Optimization, and Machine Learning*, Addison Wesley Longman, Inc., Boston, MA.

Goldberg, D.E. and Deb, K. (1991). A comparative analysis of selection schemes used in genetic algorithms, in *Foundations of Genetic Algorithms*, Ed. G.J.E. Rowalins, pp. 69–93.

Krishnakumar, K. (1989). Micro-genetic algorithms for stationary and non-stationary function optimization, *Proceedings of SPIE, Vol. 1196: Intelligent Control and Adaptive Systems*, pp. 289–96.

Lieberman, E.B., Chang, J., and Prassas, E.S. (2000). *Formulation of a Real-time Control Policy for Oversaturated Arterial*, Paper presented for TRB 79th presentation, Washington, DC.

Roess, R.P., McShane, W.R., and Prassas, E.S. (1998). *Traffic Engineering*, Prentice Hall, Upper Saddle River, NJ.

Wallace C.E. and Kenneth G. (1991). *TRANSYT-7F User Guide*, Transportation Research Center, University of Florida, Gainsville, FL.

Yagar, S. and Dion, F. (1996). Distributed approach to real time control of complex signalized networks, *Transportation Research Record*, **1554**, 1–8.