Optimization of Traffic Signals on Urban Arteries through a Platoon-Based Simulation Model

CHIARA COLOMBARONI, GAETANO FUSCO, ANDREA GEMMA
Department Hydraulics, Transport and Roads
Sapienza University of Rome
via Eudossiana, 18 – Roma 00184
ITALY
gaetano.fusco@uniroma1.it   http://w3.uniroma1/gaetanofusco

Abstract: - The paper describes an optimization procedure to synchronize traffic signals along an urban road artery. The solution procedure applies first a genetic algorithm and then a hill climbing algorithm for local adjustments. The fitness function is evaluated by means of a traffic model that simulates platoon progression along the links, their combination and possible queuing at nodes. The potential benefits of the synchronization procedure have been assessed by simulating a real urban artery through the micro-simulation model Transmodeler.

Key-Words: Optimization, Traffic control, Signal synchronization, Traffic models, Genetic algorithms.

1 Introduction

Traffic signal timing is implemented either by fixed time or traffic-responsive control. In the fixed time control, signal timing plans are designed according to the prevalent traffic conditions observed through historical surveys. Traffic-responsive control makes use of real-times measurements provided by automatic traffic detectors. It can be implemented in two different ways: plan selection or plan generation. The first method selects the most appropriate pre-calculated plan according to the traffic conditions observed in real-time. The plan generation method applies a control logic that adjusts signal settings on-line, according to real-time traffic counts. The last method is theoretically the most effective, since it is flexible enough to carry out quick adjustments of signals to better accommodate the traffic at each junction. It has also some shortcomings, related to the difficulty of obtaining stable solutions for all possible traffic conditions as well as to a higher number of traffic detectors, which implies, on one hand, a greater cost and, on the other hand, a lower robustness of the control system with respect to detector failures. For these reasons, plan selection methods are even often applied and off-line optimization methods for traffic signal synchronization are still widely studied in order to pre-compute the optimal plans.

Usual traffic signal optimization methods seek either to maximize the green bandwidth or to minimize a general objective function that typically includes delays, number of stops, fuel consumptions and some external costs like pollutant emissions. Without loss of generality, we refer to this method in the following as minimum delay problem. The maximal bandwidth method maximizes an opportunity of progression for drivers and does not reduce delays necessarily; nevertheless, it is a quasi-concave problem and efficient solving algorithms exist to find the optimal solution. The latter method is related to physical variables that are to be minimized; anyway, it is a non-convex problem and existing solution methods do not guarantee to achieve the optimal solution.

Traffic signal optimization on road arteries consists of two problems: the solution algorithm and the progression model used to compute the values of the objective function.

In order to improve the algorithm, several authors combined in a different way the two synchronization approaches, that is the minimum delay and the maximal bandwidth. Cohen (1983) used the maximal bandwidth as initial solution of the former problem; Cohen and Liu (1986) constrained the solution of the former problem to fulfill maximum bandwidth; Hadi and Wallace (1993) used the bandwidth as objective function; Malakapalli and Messer (1993) added a simple delay model to the maximal bandwidth algorithm; Gartner and Hou (1994) introduced a flow-dependent bandwidth function; Papola and Fusco (2000) expressed the delay at nodes as a closed form function of the maximal bandwidth solution.

Since the first platoon dispersion model introduced by Robertson (1969), progressively more complex models have been developed.

Park et al. (1998) introduced a genetic algorithm-based traffic signal optimization program for oversaturated intersections consisting of two modules: a genetic algorithm optimizer and mesoscopic simulator.

Dazhi et al. (2006) proposed a bi-level programming formulation and a heuristic solution approach for dynamic traffic signal optimization in networks with time dependent demand and stochastic route choice.

Chang and Sun (2004) proposed a dynamic method to control an oversaturated traffic signal network by utilizing a bang-bang-like model for oversaturated intersections and TRANSYT-7F for the unsaturated intersections.

This paper presents an optimization method consisting of a mixed genetic-hill climbing algorithm, which applies a new platoon based delay model that generalizes the analytical model developed by Papola and Fusco (2000). In such a way, it is possible to deal with even non stationary traffic demand and non synchronized signal settings. It introduces also more general assumptions on drivers' behavior. The algorithm is rather similar the well-established Transyt solving procedure (Transyt-7F, 2006), which respect to it introduces some additional flexibility aimed at improving the algorithm efficiency.

2 The traffic model

The traffic model provides the queue length and the average delay at signals of platoons traveling along an urban artery.

The delay at nodes is defined here as the excess travel time relative to travel at the synchronization constant speed. So, the transient phases are all included in the effective red time.

2.1 Node delay model

The delay at nodes depends on the arrival time and the length of the platoon as well as on the starting and the time length of the red at the signal. Three different cases may occur.

Case A: Platoon p arrives at node i during the time interval necessary to clear the queue (if any) at the end of red time. In this case, the whole is delayed (*front-delayed platoon*, illustrated in Fig.1):

$$D_{i,p} = q(v_s) l_{i,p} \left(\vartheta_i + \frac{r_i}{2} + \tau_i - t_{i,p} \right) \quad (1)$$

$$\vartheta_i - \frac{r_i}{2} \le t_{i,p} < \vartheta_i + \frac{r_i}{2} + \tau_i \quad (1')$$

where:

- ϑ_i is the offset of node i, defined as the difference between the instants of half red time of node i and node 1;
- r_i is the effective red time of node i;
- τ_i is the time needed to clear the queue at the end of red at node i: it is given by the total number of vehicles delayed at node i, before the platoon p arrives, divided by $q(v_s)$;
- $q(v_s)$ is the traffic flow at the cruise speed along the artery;
- $t_{i,p}$ and $l_{i,p}$ are, respectively, the arrival time and the time length of platoon p at node i;
- $D_{i,p}$ is the total delay of the vehicles of platoon p stopped at node i.

The term between parentheses in the definition of the total delay represents the average delay per vehicle, d.

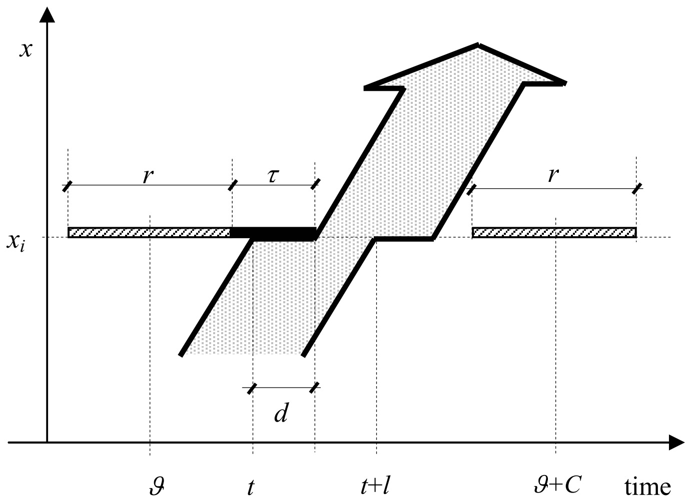

Figure 1. Node delay for a front-delayed platoon (case A).

Case B: Platoon p arrives at node i after the queue (if any) at the end of red time has been cleared and ends after the start of red time, so that the rear of the platoon is delayed (Fig.2).

$$D_{i,p} = q(v_s) \left(t_{i,p} + l_{i,p} - \vartheta_i - C + \frac{r_i}{2} \right) r_i \quad (2)$$

$$\vartheta_i + \frac{r_i}{2} + \tau_i \le t_{i,p} < \vartheta_i + C - \frac{r_i}{2} \quad (2')$$

where C is the cycle length at node i.

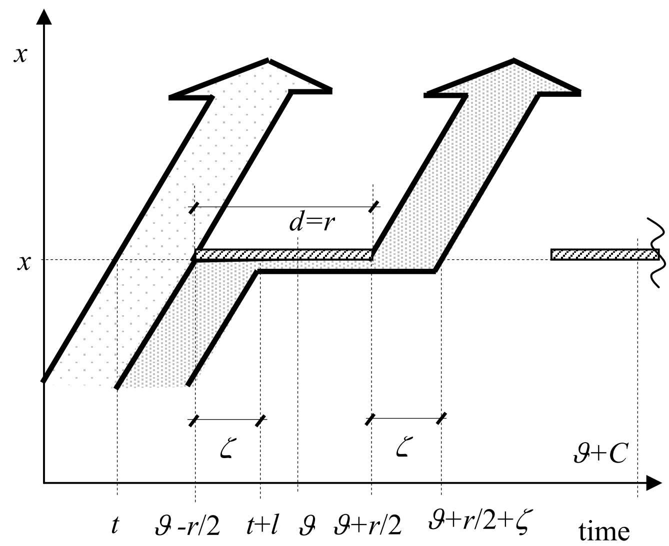

Figure 2. Node delay for a rear-delayed platoon (case B).

Case C: Platoon *p* arrives at node *i* after the queue (if any) at the end of red time has been cleared and ends before the start of red time, so that it is not delayed:

$$\vartheta_i + \frac{r_i}{2} + \tau_i \le t_{i,p} \le \vartheta_i + C - \frac{r_i}{2} \qquad (3)$$

$$t_{i,p} + l_{i,p} \le \vartheta_i + C - \frac{r_i}{2} \qquad (3')$$

Remark: The classification of one platoon depends on the arrival times of other platoons, which determine the queue clearance time, τ_i. Thus, delay at node *i* is a function of signal settings and traffic patterns at nodes 1, 2,…*i*.

2.2 Arterial delay model

The arterial delay model has been developed specifically to assess synchronization strategies along signalized arteries. The algorithm logic is depicted in the flow chart in Figure 3.

The node delay model explained in Section 2.1 is at the core of the arterial model. It performs the platoon classification at nodes, which is needed to determine the average delay and the number of stopped vehicles. It determines also the departure time and the time length of platoons, by taking into account that they can recombine at nodes.

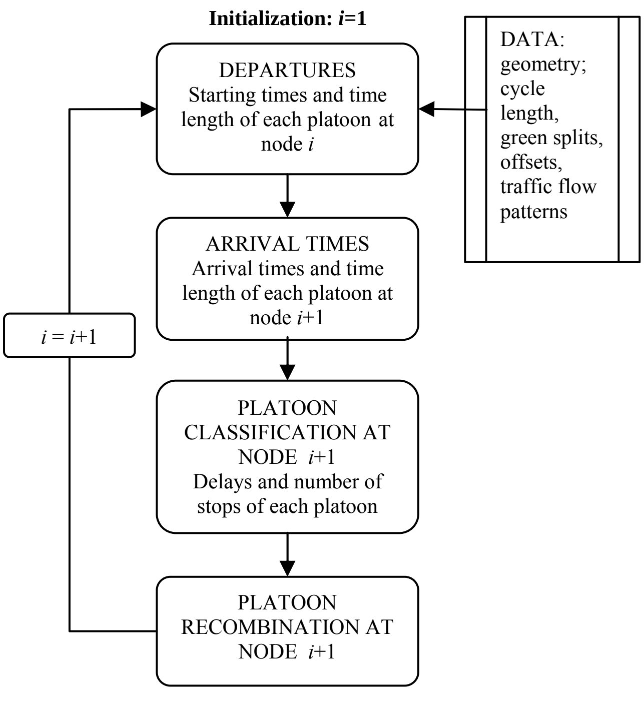

Figure 3. Flow chart of the algorithm applied to compute delay model at nodes of synchronized arteries.

The mechanism of platoon recombination at nodes is exemplified in Figure 4, where two platoons, A-type and B-type, arrive at the node i and a third platoon of vehicles entering the artery from side streets starts at the beginning of the effective red time for the artery (*i.e.*, at the beginning of the effective green time for side streets).

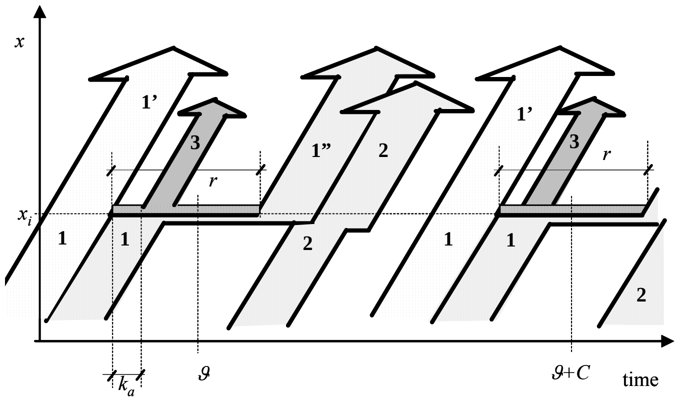

Figure 4. Example of platoon recombination at node *i*.

Since A-type platoon (denoted as 1 in the figure) is split into 2 sub-platoons (1’ and 1” in the figure) and platoon 2 arrives before the queue has been cleared, it joins platoon 1”. Departures at the node are then composed by platoon 1’, whose starting time coincides with its arrival time; by platoon 2, which starts at the end of the effective red time, and by platoon 3, entering the artery from side streets.

The link module computes the arrival times and the time length of platoons at downstream intersection. The arrival time is determined by applying either the synchronization speed or an acceleration rate and verifying if following platoons can catch up the preceding one. The time length is computed by subtracting the vehicles that leave the artery at the upstream node and assuming that all vehicles belonging to the platoon can accelerate, compressing then the platoon. Platoon progression and recombination along the links is illustrated by the example in Figure 5.

The first vehicle of platoon 1 travels at the synchronization speed v_s. Dashed line represents the trajectory of the last vehicle if no vehicle of the platoon had left the artery or if, in any case, it had traveled at the synchronization speed; however, due to exiting vehicles, all vehicles within the platoon can travel at a higher speed, as indicated for the last one, whose speed is indicated as *v*”. All vehicles of the entering platoon (numbered as 2 in the figure) travel at a speed $v' > v_s$ and may catch up the preceding platoon, depending on the link length and the value of *v*”, as it occurs in the case of platoon 2. However, platoon 3 starts at the end of effective red time, travels at speed v’ and the leading vehicle can not catch up the tail of platoon 2. Successive vehicles within the platoon can accelerate in order to fill the empty spaces left by exiting vehicles and such a condition is applied to compute the time length of the platoon at node i+1.

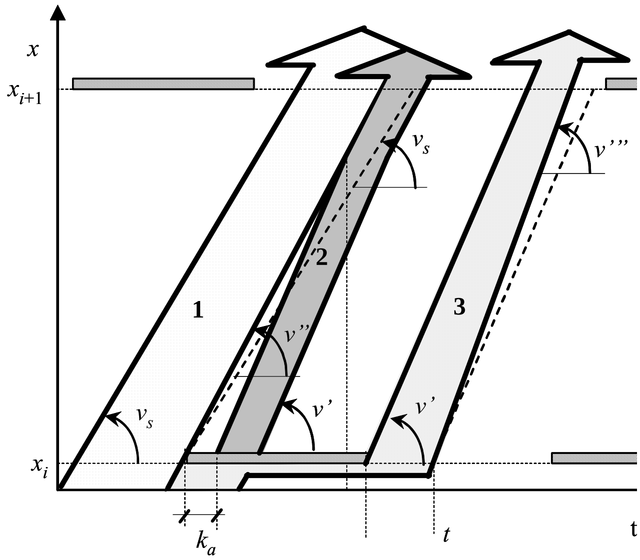

Figure 5. Example of platoon progression and recombination along a link.

The node delay model computes delays at every approach of the artery by checking, for each arriving platoon, which condition occurs among the A), B), C) cases introduced in the previous section. Since the existence and the length of a queue can not be determined before all platoons have been analyzed, the delay computation requires an iterative procedure that classifies the different platoons progressively. It is worth noting that such a procedure involves few iterations, because the platoons can both catch up each other along the links and recompose themselves at nodes, when more platoons arrive during the red phase.

3 The optimizing algorithm

3.1 Definition of the objective function

The objective function (or fitness function in genetic algorithm jargon) is defined as a linear combination of the total delay on each direction of the artery and the total delay at the approaches of the cross streets.

$$f = (1 - w_t)\left(w_1 \sum_{i=1}^{n} D_i^{(1)} + (1 - w_1) \sum_{i=1}^{n} D_i^{(2)} \right) + + w_t \sum_{i=1}^{m_i} D_{h_i}^{(t)} \quad (4)$$

where:

- $D^{(1)}_i$ is the total delay at node i in direction 1
- $D^{(2)}_i$ is the total delay at node i in direction 2
- $D^{(t)}_{h,i}$ is the total delay of transversal lane group h at node i
- w_1 is the weight of the delay in direction 1
- w_t is the weight of delay at the cross streets

It is worth noting that the total delay instead of the unitary average delay has been included in the objective function as the simulation procedure has been devised to analyze and simulate the whole demand. Thus, all possible signal settings solutions are homogenous as for the number of vehicles served.

3.2 Solution framework

The solution framework is based on a genetic algorithm that combines the feasible signal settings in order to optimize the objective function, which is evaluated, for each possible solution, through the traffic model described in section 2.

3.3 Genome definition

Each possible signal setting solution for the artery is represented through a genetic coding (genome) whose elements symbolize the cycle length, the green split in each of the 2 directions and the offset of each signal.

3.3 Initial population

The first step of the genetic algorithm is to generate the initial population, formed by a given number of possible solutions, each of them is characterized by a different genome patrimony.

Since the quality of the initial population affects the algorithm convergence significantly, a subset of the initial solutions has been designed by applying simple but reliable criteria that are usual good practice in traffic engineering, while the remaining has been chosen by random.

More specifically, the following special designed solutions have been considered:

- the actual signal settings;
- a maximal green bandwidth solution corresponding to the maximal of the actual cycle lengths of the artery and the actual green splits;
- a good practice solution obtained by applying the following simple rules, that is:

a) cycle length equal either to the minimum cycle or the optimum value for isolated junctions following Webster's assumptions on probabilistic arrivals:

$$C = \max\{C_{Web,i}\} \quad \text{OR} \quad C = \max\{C_{\min,i}\} \quad (5)$$

where:

$$C_{Web,i} = \frac{1{,}5 L_i + 5}{1 - \sum_{h}^{m} (1 - y^*_{h,i})}$$
$$C_{\min,i} = \frac{L_i}{1 - \sum_{h}^{m} (1 - y^*_{h,i})} \quad (6)$$

b) green splits according to the either equisaturation criterion or a priority criterion that assigns all the available green to the artery

$$\lambda_{k,i} = \frac{y^*_{h,i}}{\sum_h^m y^*_{h,i}} \quad \text{OR} \quad \begin{array}{l} \lambda_{h,i} = y^*_{h,i} \\ \lambda_{k,i} = 1 - \sum_h^m \lambda_{h,i} \end{array} \qquad (7)$$

c) offset set according to the maximum bandwidth criterion.

The following notations have been used in the equations above:

- $y^*_{h,i}$ is the saturation degree at node i for the critical lane group of stage h
- $\lambda_{h,i}$ is the green split for the critical lane group of stage h
- L_i is the lost time at node i

In addition, all the solutions must fulfill a set of constraints on the minimum and maximum values for the cycle length, the green splits and green time for pedestrian crossing.

3.4 Cross-over

Cross-over operator combines the genetic patrimony of a pair of individuals (parents) to generate new individuals (children). The cross-over rule applied is to generate a couple of children for each couple of parents, choosing by random (0.5 probability) which child receives each of the chromosomes of the first parent and assigning each chromosome of the second parent consequently.

The probability of each individual to be selected for reproduction is proportional to its fitness value. Better individuals have so a higher probability to transmit their genetic patrimony. Moreover, to improve the algorithm stability, the cross-over operator is applied only to a given quota β of the whole population.

3.5 Mutation

Mutation operator applies, with a probability γ, a random alteration to one or more elements of each individual of the population. Through the mutation, the algorithm can explore different zones of the space of solutions. To prevent the algorithm being trapped in local minima, after a given number of iterations without achieving any improvement of the objective function, the mutation probability γ is changed linearly until a threshold value γ_{max}.

As an improvement of the objective function is obtained, the mutation probability is reset to its initial value γ.

6.2.6 Elitism

Elitism feature consists of keeping a percentage η of the best solutions.

6.3 Local adjustment

The genetic algorithm stops after a given number of iterations. To improve the solution, local adjustments are undertaken by applying a hill climbing algorithm based on a set of predetermined steps, similar to that implemented in Transyt. The algorithm tries to improve the objective function by applying firstly some incremental steps and then the symmetrical decreasing steps to each design variable: cycle length, green splits, offsets.

4 Application to a real urban artery

The procedure described here has been applied to synchronize Via Tiburtina, a 3 km long urban artery in Rome, containing 8 signalized intersections. In the rush hour it is usually heavily congested, with an average speed of about 8 km/h in direction of the town center and about 16 km/h in the opposite direction.

In order to validate the present procedure, both the actual scenario and the optimal solution have been simulated using the microsimulation model Transmodeler (Caliper, 2007).

This model is characterized by many parameters and so a careful calibration of the arterial model has been required to fit observed traffic counts. Three demand scenarios have been considered to verify the robustness of the synchronization solution with respect to possible demand fluctuations. Starting from the actual average demand, two other scenarios, high and low, have been obtained by increasing and reducing the average demand level as +15% and -15%, respectively.

The simulation results highlight that the optimizing procedure improves the average unitary delay at the nodes of the arterial of 40%, 22% and 23% in high, average and low demand level, respectively (see Figure 6).

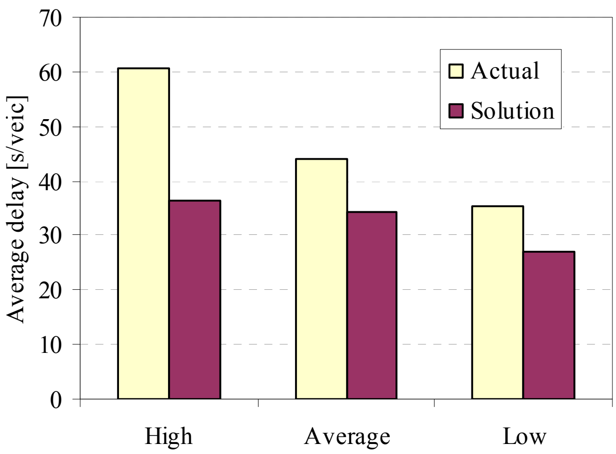

Figure 6. Average unitary delays at nodes on the whole artery for 3 demand levels (high, average, low)

The signal settings, although optimized with respect to the average delay, improve also the total capacity of the artery and allow increasing the total number of vehicles served of 9%, 9% and 5% in the high, average and low demand level, respectively (see Figure 7).

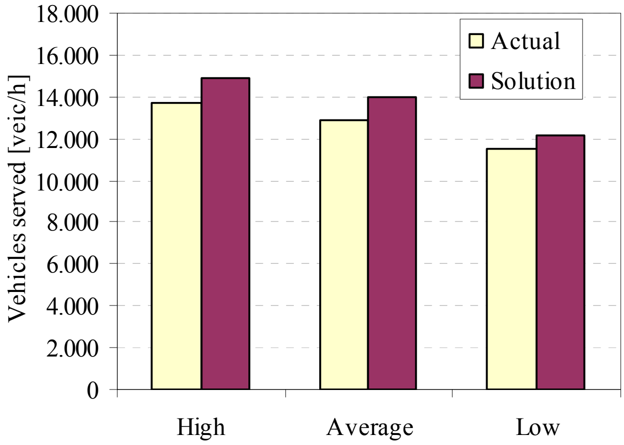

Figure 7. Total vehicles served in simulation for the 3 demand levels (high, average, low)

Figure 8 shows the average unitary intersections delay at signals, computed in microsimulation for the actual scenario and optimal solution, for the highest demand level.

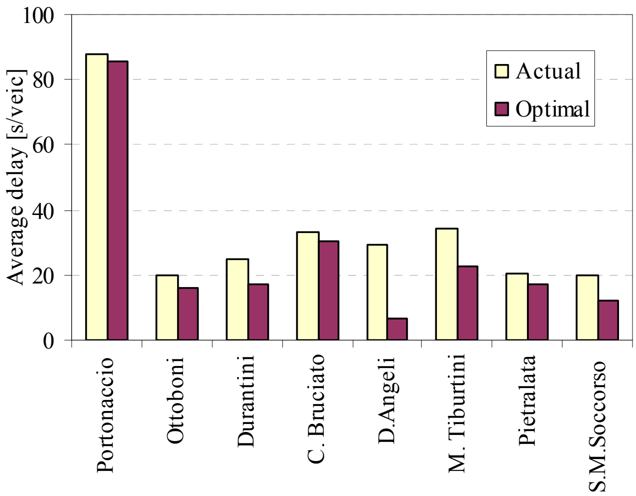

Figure 8. Average unitary delay at each node for the high demand scenario

It is possible to observe that in the most critical signal intersection (Portonaccio, which is the most close to the center), the delay reduction is about 35%, 21% and 3% for the three demand levels. It is worth noting that an even slight improvement has been achieved at each node.

4 Conclusion

An optimizing procedure for signal settings suitable for plan selection traffic-responsive control has been introduced. The procedure has been validated simulating a 8-node urban road artery in Rome, Italy, by using Transmodeler software. Significant benefits have been obtained in terms of both the unitary delay (from 22% to 40% for different demand levels) and the number of vehicles served (from 5% to 9%, respectively). Further developments include the signal bus priority in the procedure, in order to select the most suitable priority rule to apply. The basic idea is to individuate opportunity progressions for buses by minimizing delays for car traffic.

Acknowledgment

This study was carried out within the framework contract between DITS-Department of Hydraulics, Transport and Roads of Sapienza University of Rome- and ATAC, Mobility Agency of the Municipality of Rome.

References:

[1] Cohen S.L. (1983), Concurrent Use of MAXBAND and TRANSYT Signal Timing Programs for Arterial Signal Optimization, *Transpn. Res. Rec*., 906, 81-84.

[2] Cohen S.L. and Liu C.C. (1986), The Bandwidth-Constrained TRANSYT Signal-Optimization Program, *Transpn. Res. Rec*., 1057, 1-7.

[3] Hadi M.A., Wallace C.E. (1993), Hybrid Genetic Algorithm to Optimize Signal Phasing and Timing, *Transpn. Res. Rec*., 1421.

[4] Malakapalli M.P. e Messer C.J. (1993), Enhancements to the PASSER II-90 Delay Estimation Procedures, *Transpn. Res. Rec*., 1421, 94-103.

[5] Gartner N.H., Hou D.L. (1994), Performance Evaluation of Multi-Band Progression Method, 7th IFAC/IFORS *Symp. "Transportation Systems: Theory and Application of Advanced Technology"*, Tianjin, China, 24-26 August, 981-986.

[6] Papola N., G.Fusco (2000). A new analytical model for traffic signal synchronization, *Proc. of the 2nd ICTTS Conference*, Beijing, China, 31 July-2 August, 2000.

[7] Robertson D.I. (1969), *Transyt: a Traffic Network Study Tool*, Ministry of Transport, RRL Report LR253, Crowthorne, Berkshire, United Kingdom.

[8] Park B., Messer C.J., Urbanik II T. (1999), "Traffic Signal Optimization for Oversaturated Conditions: Genetic Algorithm Approach". *Transpn. Res. Rec.*, 1683, pp.133-142.

[9] Dazhi S., Benekohal R.F., Waller S.T. (2006), "Bi-level Programming Formulation and Heuristic Solution Approach for Dynamic Traffic Signal Optimization", *Computer-Aided Civil and Infrastructure Engineering*, 21, pp. 321-333.

[10] Chang T., Sung G. (2004), "Modeling and Optimization of an Oversaturated Signalized Network", *Transpn Res - Part B*, 38, pp.687-707.

[11] McTrans Center (2006). *Traffic Network Study Tool–Transyt-7F*. United States Version. University of Florida.

[12] Caliper (2007). Transmodeler Traffic simulation Software User's Guide. Version 1.5. Caliper Co.