



Roma Tre University

Ph.D. in Computer Science and Engineering

Optimization Algorithms for Signal Synchronization and Bus Priority on Urban Arteries

Andrea Gemma

Optimization Algorithms for Signal Synchronization
and Bus Priority on Urban Arteries

A thesis presented by
Andrea Gemma
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy
in Computer Science and Engineering
Roma Tre University
Dept. of Informatics and Automation
December 2010

Tutor:
Prof. Dario Pacciarelli

Reviewers:
Prof. Michael Florian
Prof. Gaetano Fusco

To my wife

Abstract

This thesis is aimed at defining a simulation model representing the vehicular flow along an urban road artery with traffic signals. This platoon-based model can simulate the behavior of private and public transport vehicles and gives the opportunity to implement priority strategies for the latter. As a result of the simulation, it is possible to calculate the vehicles delay caused by the presence of traffic lights and this information was used to define a fitness function representing the quality of traffic light synchronization.

In the second part of the work two optimization algorithms have been realized to minimize the fitness function: A genetic algorithm and particle swarm algorithm. The two algorithms, known to have comparable performance, were compared on real instances of some main roads in the urban area of Rome.

Acknowledgements

I am very grateful to all the people who have directly or indirectly contributed to the birth of this Ph.D. thesis. Special thanks to my tutor Prof. Dario Pacciarelli, to Prof. Gaetano Fusco of “La Sapienza” University of Rome, to Prof. Cipriani of the Department of Civil Engineering of Roma Tre University of Rome and to my colleagues and friends. I would also like to thank the members of AuTORI Automation and Industrial Management Laboratory of Roma Tre University. At last, I would like to thank Ing. Valentina Conti for her support throughout the PhD.

Contents

Contents	x
List of Figures	xii
1 Introduction	16
1.1 Definitions	18
2 State of Arts	22
2.1 Steady-state control	25
2.2 Dynamic control	29
2.2.1 Principles of the dynamic traffic flow control	29
2.3 Priority measures for buses	32
2.3.1 Strategies of signal priority for buses	34
2.4 Existing simulation and optimization models	44
2.4.1 Simulation models	44
2.4.2 Synchronization and optimization models	69
3 Delay calculation model	99
3.1 The out flow models	100
3.2 Formulation of the minimum travel time problem	106
3.3 Kinematic wave	115
3.3.1 Stop wave	117
3.3.2 Start wave	119
3.4 The simulation model	121
3.4.1 Law of generation	123
3.4.2 Law of progression	129
3.4.3 Law of classification	137
3.4.4 The pseudo-code	138
3.4.5 The model for public transportation	142
4 The optimization algorithms	152
4.1 The fitness function	152
4.2 The genetic algorithm	156
4.2.1 Genoma	157

4.2.2	Initial population	158
4.2.3	Crossover	160
4.2.4	Mutation	167
4.2.5	Elitism.....	171
4.2.6	Final parameters	172
4.3	Particle swarm optimization	173
4.3.1	The agent	175
4.3.2	The movement function.....	176
4.3.3	Final parameters	182
4.4	Hill-Climbing.....	183
4.5	Comparison.....	186
4.6	Performace	188
5	Evaluation of results on real case	190
6	Conclusion	195
	Bibliography	197

List of Figures

Figure 2-1 - Update schema for signal plan	28
Figure 2-2 - Vehicle trajectory without traffic light priority	36
Figure 2-3 - Bus Trajectory with green extension priority	39
Figure 2-4 - Bus Trajectory with green anticipation priority	40
Figure 2-5 - Bus Trajectory with green insertion priority	40
Figure 2-6 - Resolution algorithm structure of the dynamic traffic assignment model.....	48
Figure 2-7 - Dynamic equilibrium assignation algorithm in dynameq	50
Figure 2-8 - Triangular flow-density diagram.....	51
Figure 2-9 - Simplified model of queued vehicle.....	52
Figure 2-10 - Spatial discretization of the network arcs.....	55
Figure 2-11 - v-k relation (Greenshield's equation changed)	56
Figure 2-12 - Equation of state of the Cell Transmission Model	62
Figure 2-13 - Representations of convergence and divergence.....	64
Figure 2-14 - Node delay for a fore-delayed platoon (case A).....	67
Figure 2-15 - Node delay for a rear-delayed platoon (case B).	68
Figure 2-16 - IT architecture of the TRANSYT resolution model and procedure (Source: Vincent, Mitchell, Robertson. User guide to Transyt Version 8, 1980).....	72
Figure 2-17 - Transyt arc delay model (source: Dion et al., 2004)	78
Figure 2-18 - Spillback queue and impossibility for the upstream vehicles to exploit the green (from synchro 6 manual).....	81
Figure 2-19 - Reduction in capacity effect due to queues (from synchro 6 manual).....	81
Figure 3-1 - Greenberg law	102
Figure 3-2 - Greenshield linear model.....	104
Figure 3-3 - Greenberg logarithmic model.....	105
Figure 3-4 - Delay of a vehicular platoons at a sincronized artery node	108

Figure 3-5 - Representation of the progression of a platoon	109
Figure 3-6 - Example of platoon of type A	112
Figure 3-7 - Example of platoon of type B.....	113
Figure 3-8 - Example of platoon of type C.....	114
Figure 3-9 - Wave front	116
Figure 3-10 - Representation of kinematic wave in the flow-density diagram.....	117
Figure 3-11 - Stop wave	118
Figure 3-12 - Start wave	119
Figure 3-13 - Delay model flow chart	121
Figure 3-14 - Example of demand curve	124
Figure 3-15 - Recombination of platoons.....	125
Figure 3-16 - Number of platoons required for a simulation applying or not applying the compaction property with the range of the simulation time.....	127
Figure 3-17 - Cycles order.....	128
Figure 3-18 - Catch-UP	130
Figure 3-19 - Example of spillback phenomenon	131
Figure 3-20 - Spillback in the space-time diagram.....	131
Figure 3-21 - Example of reclassification in case of spillback.....	133
Figure 3-22 - Spillback condition (a). Push-back method (b). Reclassification method (c).....	134
Figure 3-23 - Spillback condition.....	136
Figure 3-24 - Push-back method	137
Figure 3-26 - Trajectories of vehicles in different platoons in case of oversaturation	138
Figure 3-27 - Extended delay model flow chart	139
Figure 3-28 - Green advance	145
Figure 3-29 - Green extension.....	146
Figure 3-30 - Conflict in the assignment of the priority	148
Figure 3-31 - Public Transport delay model flow-chart	150

Figure 3-32 - Relations between models and data.....	151
Figure 3-33 - Example of the progression of platoons generated by the model (buses are highlighted in black)	151
Figure 4-1 - Scanning of six different dimensions of the objective function	154
Figure 4-2 - Scanning of the objective function between two generic configuration of three road artery. Alpha is the coefficient of convex combination between the configuration.	155
Figure 4-3 – Comparison between trend of algorithm with random initial population or with intelligent construction of initial solution	160
Figure 4-4 - Matrix representation of Genome.....	161
Figure 4-5 - Crossover of type 1.....	163
Figure 4-6 - Crossover of type 2.....	164
Figure 4-7 - Crossover of type 3.....	165
Figure 4-8 - Trend of the algorithm with the three crossover methods	166
Figure 4-9 - Trend of the algorithm with different crossover rate.....	167
Figure 4-10 - Simulation for different mutation rate.....	169
Figure 4-11 - Simulation for different interval. $\Delta=[pmin_{mut}, pmax_{mut}]$	170
Figure 4-12 - Simulation for different interval. $\Delta=[pmin_{mut}, pmax_{mut}]$ in Colombaroni, Gemma and Fusco (2009)	170
Figure 4-13 - Simulation for different elite rate	172
Figure 4-14 - Trend of the objective function for the genetic algorithm with best parameters.....	173
Figure 4-15 - Comparison between trend of algorithm with random initial population or with intelligent construction of initial solution	176
Figure 4-16 - Agent movement relative to two sizes in the space of research	177
Figure 4-17 - Testing of stall parameter	179

Figure 4-18 - Trend of the algorithm with different speed parameters	180
Figure 4-19 - Trend of the algorithm with different best local parameters	180
Figure 4-20 - Trend of the algorithm with different best global parameters	181
Figure 4-21 - Trend of the algorithm with different random parameters	181
Figure 4-22 - Trend of particle swarm algorithm for different values of inertia parameter.....	182
Figure 4-23 - Trend of the objective function for the particle swarm algorithm with best parameters	183
Figure 4-24 - Application of Hill-Climbing after 300 iterations or with premature stop at 50 iterations	185
Figure 4-25 - Comparision between PSO and GA	186
Figure 4-26 - Average Fitness	187
Figure 4-27 - Execution time.....	189
Figure 5-1 - Average unitary delays at nodes on the whole artery for 3 demand levels (high, average, low).....	191
Figure 5-2 - Total vehicles served in simulation for the 3 demand levels (high, average, low).....	192
Figure 5-3 Average unitary delay at each node for the high demand scenario	193
Figure 5-4 - Optimization results.....	194

1 Introduction

In an urban network, a point where more delay accumulates is at the intersections. The cause of this delay is the sharing of the intersections by the different traffic flow: the same road space used for traffic movements must be shared, alternating, between two or more current vehicular conflicts. The average capacity of the intersection is therefore less than half the capacity of the road. Especially with very high traffic flows, the intersection becomes a bottleneck due to the progression of vehicles that are delayed. In order to better manage the sharing of the intersection, it will be necessary to use the traffic lights. When along a road artery there are most signalized intersections, coordination between the different traffic lights becomes crucial: the interruption of progression due to the alternating cycles of the traffic signal makes the flow not constant, therefore, to the generic approach, there will be time intervals characterized by a large number of incoming vehicles and other intervals with a more modest flow or even zero. Hence it is possible to organize the traffic lights in such a way that red time interval overlaps, as far as possible, with the interval of low flow, properly synchronizing the beginning of the green in such a manner that interval of green starts when the intense traffic flow arrives. Ideally, one would expect that no vehicle is stopped and then the delay may be canceled. This condition is essentially impossible to fulfill but in most real cases an appropriate offset of the green of traffic lights can reduce travelling times significantly. Although it is not possible to allow all vehicles to progress along the route without stops at the nodes, in other respects, it is possible to reduce the waiting time or the number of vehicles stopped along the artery. In the case of one-way streets, the

problem is complicated by the presence of incoming / outgoing vehicles that make sure that in the artery there is not a single compact platoon. As for two-way streets, the problem is further complicated by the requirement to take into account each node in both directions. The offset of a generic node, in fact, affects not only the delay at that node, changing the sequence of the vehicles from that node, but also the delay in all downstream nodes, and thus, in the two-way street, the offset of a generic node affects the delay in all nodes of the artery. Within the scope of the synchronization problem, then, the average delay of the artery is a scalar function of vector of offset (or stages) of all nodes. The independence of the effects of the various nodes, which could be positive or negative, easily shows that the synchronization problem of minimum delay is a not convex problem. For this reason it is still frequently used an another approach: 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, henceforth this method well be called minimum delay problem. This 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. The maximal bandwidth method maximizes an opportunity of progression for drivers and does not reduce delays necessarily. Nevertheless, it is an almost concave problem and there are efficient solving algorithms to find the optimal solution. The traffic lights cause not only the delay but also allow to apply priority strategies of transit. As for private transport, the strategies attempt to reduce delay in two ways: by reducing the probability that a transit vehicle encountering a red signal, and, if this does

occur, by reducing the wait time until the green signal. Since the passage of the bus is more sparse, it is possible to implement the signal priority strategies temporarily only in their passage. The crucial importance of traffic lights and the need for tools to optimize the delay by the synchronization was the starting point of the research presented in this work.

1.1 Definitions

Although this work has been done in computer engineering and operations research for its full understanding requires a good knowledge of the transport sector. Where possible, to make the text as complete as possible, many aspects of transport engineering will be explained. It is not of interest in this work to explain all phenomena that are behind the flow of vehicular and so many concepts were implied assumption that the reader's knowledge. For those who need to know more about you suggest the following reference books (Cascetta, 1998) and (Sheffi, 1985)

Below, to give uniformity to the reading of this work will be a short description of the main variables used in the traffic light synchronization and in particular in this work. Each of these parameters will be given a uniform meaning and representation in later chapters:

C_i [sec]: Cycle time for the generic traffic light intersection i . Traffic light cycle is defined as any complete sequence of switch on (and off) of traffic lights at the end of which returns the same configuration of the lights existing at the beginning of the sequence. When will speak of synchronization between multiple traffic lights will refer to this size without linking to any intersection.

In such cases, the cycle is the common cycle at all intersections also called synchronization cycle.

q [vehicles/sec]: vehicular flow. It defines the flow of a current the average number of vehicles passing through a section in unit time.

θ : offset. In the case of synchronization of plans between different traffic light intersection the offset represents the phase shift with respect to the common cycle synchronization. The offset is represented with adimensional values in the range $[0 \div 1)$. Where will be appropriate, in this work, this size will defined in seconds and in the range $[0 \div C)$.

s [vehicles /sec]: saturation flow. The saturation flow is the maximum number of vehicles that can cross a stop line of an intersection per unit time in the presence of continuous queue. The saturation flow depends on the geometric characteristics of the intersection, on the composition of the flow and on the control mode of traffic lights.

y : saturation degree. The saturation degree is the ratio between traffic flow and the flow of saturation. This quantity is an indicator of the level of congestion.

g_{if} : effective green split (in this work simply green split) is the ratio between the effective green and the duration of the cycle of the traffic light on intersection i in phase f . For the analysis of the traffic, sometimes, is convenient to consider, instead of the real length of the green, the duration of effective green for which it is assumed that vehicles may flow to the values of saturation flow, and then with a constant time distance equal to $1/s$. The

introduction of the concept of effective green split can easily determine the maximum flow on the basis of saturation flow and the traffic light cycle.

$$q = s \cdot g$$

r_{if} : effective red split is defined as $1 - g_{if}$ or the part of the cycle for the phase f is not used by the traffic flow for transit across the intersection i .

L [sec]: The traffic light cycle cannot be used to completely by the vehicles stream to the values of saturation flow of each current. In fact there is the time lost in which the intersection is not used completely. The time lost was due mainly to three contributions:

- Transient state of vehicles in the queue at the beginning of the green phase;
- The transitory of exiting of vehicles from intersection at the end of the green phase and during the yellow phase;
- the time between the end of yellow and the beginning of green of the next phase.

The lost times at the beginning and at the end of green are used to determinate the duration of effective green.

F : Phase means that part of the plan during which a particular traffic light signal configuration is constant.

A traffic light plan for a single intersection is defined by the duration of a cycle as well as the phase transition and duration. It can be described in two way:

- on the structure: by the phase, their duration and the transaction between their;
- on the time step: by start times and end times for each light for each phases in the interval $[0 \div C]$

2 State of Art

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 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 often still applied and off-line optimization methods for traffic signal synchronization are still widely studied in order to pre-compute the optimal plans.

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 two synchronization approaches: the minimum delay and the maximal bandwidth.

Cohen (1983) used the maximal bandwidth as initial solution of the former problem; Cohen et al (1986) constrained the solution of the former problem to fulfill maximum bandwidth; Hadi et al. (1993) used the bandwidth as objective function; Malakapalli (1993) added a simple delay model to the maximal bandwidth algorithm; Gartner (1994) introduced a flow-dependent bandwidth function; Papola (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. (1999) 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 et al (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 work 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 et al (2000). In such a way, it is possible to deal even with non stationary traffic demand and non synchronized signal settings. It introduces also more general assumptions on drivers's behavior. The algorithm is rather similar to the well-established Transyt solving procedure Transyt-7F (McTrans-Center, 2006), which respect to it introduces some additional flexibility aimed at improving the algorithm efficiency.

In addition to private transport, in this work, the problem of minimum delay for public transport with priority strategies was tackled. Several methods exist to ensure priority to buses with respect to general traffic in urban areas. Among these, signal priority strategies attempt to reduce delay in two ways: by reducing the probability of a transit vehicle encountering a red signal, and, if this does occur, by reducing the wait time until the green signal. The objective of this study is modeling and simulating a mathematical procedure to provide bus priority along a synchronization artery, through the combination of passive and active bus priority strategies.

Passive priority is defined as the use of static signal settings to reduce delay for transit vehicles. Such strategies can be as simple as increasing the green split for the phase in which the transit vehicle has right of way. Signal coordination is another strategy that can be used to benefit transit vehicles. Arterial progression, for example, can be designed to favor transit vehicles by timing the green band at the average transit vehicle speed instead of the average automobile speed, which is typically faster (Davol, 2001). More effective coordination strategies can combine the maximum green bandwidth and minimum delay problems used in private transport and already mentioned. However, it has been observed that passive strategies have limited value in order to improve the global transport performances (Skabardonis, 2000). Active strategies address these limitations by altering signal settings dynamically and only when necessary, in order to minimize delay to an approaching transit vehicle. Several studies have been performed to apply active priority strategies: Liao et al. (2007) takes advantage of the already equipped Global Positioning System on buses to develop an adaptive signal priority strategy that could consider bus schedule adherence, number of

passengers, location and speed; (Stevanovic, et al., 2008) presents a genetic algorithm formulation that optimizes four basic signal timing parameters and transit priority settings using VISSIM micro simulation as the evaluation environment. A bus priority algorithm could also be integrated into an adaptive network signal control model. For example, SCOOT (McDonald, et al., 1991) system has a number of facilities that can be used to provide priority to buses or other public transport vehicles; the signal timings are optimized to benefit the buses, either by extending a current green signal (an extension) or causing succeeding stages to occur early (a recall). Priority facilities are also available in UTOPIA (Mauro, 1991) system, in which optimal strategies are determined at the higher level on the basis of area traffic prediction, while traffic light control is actuated at the local level according to traffic conditions at individual intersections. The aim of control strategies is to minimize the total time lost by private vehicles, while ensuring that public transport vehicles are not stopped at signalized intersections. The present paper introduces a traffic platoon model and a heuristic algorithm to optimize preset signal synchronization plans that can include and simulate active bus priority strategies at some signals.

2.1 Methods of control

It is used to classify the types of traffic control according to the characteristics and dimensions of the problem which involves:

- Methods of control for synchronized intersections deal directly the space-time interaction of vehicular flow;

- Methods of Traffic Actuated Control (also known in real time) adapting the characteristics of traffic lights to demand variability;
- The fixed time control methods that assume the validity of steady-state;
- Methods of equilibrium traffic control that look for a good network configuration compatible with the resulting change of route of users, instead, the traditional methods of traffic lights control calculate the parameters assuming constant traffic demand and the choice of route as invariants.

The methods of control for synchronized intersections can be distinguished, on the base of parameters of control, between methods of maximum green bandwidth (or "green wave") and maximum performance or can be distinguished on the base of technology used to make synchronization between several intersections, in systems with centralized architecture and distributed systems architecture.

The methods of Traffic Actuated Control can be classified into systems of plan-selection (real-time actuation and off-line calculations) or plan-generation (on-line calculations and dynamic regulation); the latter is then divided into open circuit control systems (if the variables measured are the inputs for the models) or closed circuit (if the variables measured are the product of models of regulation). Finally, an important division concerns the possibility to discern specific vehicle categories, such as public transport and emergency vehicles, with the possibility to advantage them.

The steady-state traffic control can be achieved with a fixed time method, if the traffic flow remains constant for a fixed period of time at least one order of magnitude higher than the traffic light cycle, or with a plan-selection method, where the temporal variations of traffic flow occurs slowly from one period to another, so as that by measuring the value of the flow in some sections, it is possible to recognize the particular configuration of traffic and to actuate a pre-calculated plan to control the measured condition.

2.1.1 Steady-State of control

The steady-state control is used, therefore, to optimize the traffic light parameters for particular time interval, assuming that in each interval flows are known and constant. In the fixed time control the flows are calculated on statistic information. In the plan-selection control, instead, the flows are detected real-time with traffic sensor; on the bases of this measures, the control system sets the most appropriate plan, applying predefined plan.

From the methodological point of view, both the flows models and resolving procedure are similar for the two control methods. In both cases, the flow is considered constant in the time of application of the plan and therefore it is possible to apply models of progression or loading of the network in steady-state flow. It is possible to define, for each intersection, a traffic light cycle, which represents the time period in which, within the range of stationary, the same configurations of traffic light are repeated. In both cases, moreover, it is possible to define a policy control based on knowledge of the average of each flows of traffic and, in the case of synchronized intersections, on the progression of vehicles on the network.

Update

An important aspect of fixed time control or plan-selection is a periodic update of traffic lights plans. The adjustment is needed to adapt the control to the changes in long-term demand and in traffic flows on the network, dependent on seasonal or structural variations. Figure 2-1.

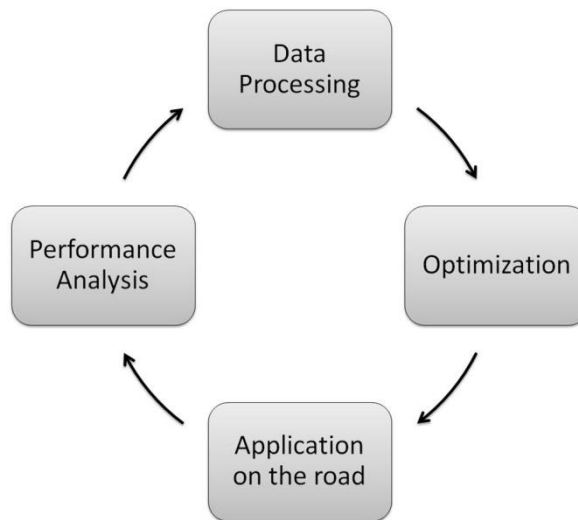


Figure 2-1 - Update schema for signal plan

The plan-selection method requires a monitoring system to classify the traffic pattern and a recognizing system to find the best match between the pattern and the pre-calculated plans.

2.1.2 Dynamic control

Principles of the dynamic traffic flow control

Dynamic control strategies are designed to adapt in the short term the signals configuration to the variability of the traffic flow at intersections. These strategies are potentially more efficient but, in turn, much more expensive from the operational costs point of view, since they require the installation, implementation and maintenance of control systems in real time (measurements, transmission of information, a central controller, and local controllers).

Dynamic control strategies use instantaneous measurements carried out by detectors (generally, inductive), placed in proximity of the intersection at the beginning or at the end of each approach. Through appropriate algorithms they set the green time, the offset and the cycle time for the plan in question on the basis of the traffic flow detected.

The time horizon of each planning is estimated for a medium-long period H (e.g. 60s), but the results are actually applied only for a much shorter period of time (e.g. 4s). Then, new information are collected and a new optimization issue is solved by planning an time horizon H as much long. This way the process is prevented from adopting myopic strategies based only on the optimization of the present situation.

One of the most spread algorithms for this kind of strategies is SCOOT (Hunt, et al., 1981). Recently, new multiple traffic-adaptive methods have been proposed: OPAC (Gartner, 1982), PRODYN (Farges, et al., 1983), CRONOS

(Boillot, et al., 1992), COP (Sen, et al., 1996), RHODES (Mirchandani, et al., 2001), UTOPIA (Mauro, et al., 1989).

Binary variables in the formulation of the problem are the main issue of these strategies, because they require the employment of algorithms with exponential complexity for a global minimization. Practically, some algorithms resolve the problem with the complete enumeration of the solution (brute force), whereas others adopt a dynamic programming. Since the resolution of these algorithms is extremely complex, the control strategies, theoretically adaptable to the whole network, can be applied in real time to an only one intersection. That leads to the realization of the system by employing several excellent decentralized strategies (at each road junction), whose choices are heuristically coordinated by a high-level control. An exception is CRONOS, which uses a global heuristic optimization method with polynomial complexity, enabling to take into consideration simultaneously multiple intersections and obtaining this way a local but not global minimum.

"Store and forward" approach

A particular category of the traffic-adaptive strategies is represented by the approaches based on the "store-and-forward" philosophy. This type of network formation was suggested for the first time by (Gazis, et al., 1963) and by then it has been used in many worthy works.

The main idea of the store-and-forward model is the introduction of a simplification providing the mathematical description of the traffic flow without recurring to binary variables. It is crucially important because it allows to use many highly efficient polynomial-complex methods of control

and optimization (linear, quadratic, nonlinear programming and multivariable regulators). On the other hand, it permits real time coordinated control also on networks on a large scale.

The fundamental simplification is introduced when modelling the outflow q_i of an arc i . The outflow q_i in the discrete time k derives from:

$$q_i(k) = \frac{g_i(k)}{C} s_i$$

Eq. 2-1

where $g_i(k)$ represents the green duration for the arc i and s_i is the corresponding saturation flow. If the discrete time interval is equivalent to the cycle time C , then q_i will be equivalent to the average flow during the corresponding cycle, instead of being formed as equivalent to s_i during the green phase and null during the red one. In other words, each network arc is assumed to get a constant and uninterrupted outflow (until the demand keeps to be sufficient). As a consequence of this simplification:

- time discretization cannot be lower than the cycle time C , so decisions in real time could not be made more than once per cycle;
- the oscillation of the platoon on the arcs, due to the alternation between the green and red phases, are not described by the model;
- the effect of the offset for the consecutive intersections cannot be represented by the model.

In spite of the restrictions, the appropriate use of the "store-and-forward" model can bring to efficient coordinated control strategies for networks on a

large scale, as demonstrated by several studies of simulation. A recent application of this strategy is originated by the TUC algorithm (Diakaki, et al., 2001) based exactly on a formulation of this kind.

2.2 Priority measures for buses

A wide range of methods were developed in order to let the public means get priority on the general traffic in urban areas. Some of them are:

- virtual or physical vehicle flow split systems;
- traffic light priority systems.

The fact of considering buses separately from the general traffic already provides the strongest kind of bus priority. Some of these systems include:

- Bus-only streets: roads where entrance is denied to all vehicle but the ones with priority (not only means of public transportation but also emergency vehicles);
- Busways: fully detached ways for buses travelling on one or both directions, usually placed at the middle of the carriageway. Buses travelling on a Busway can work normally or be driven through physical or electronical tools. The Busway can be associated to a platooning management with platoons of buses and trolleybuses stopping simultaneously and providing so the same features of an underground or LRT (Light Rail Transport);
- Reserved lane: a reserved lane for buses and other vehicles having priority and travelling in the same direction, placed next to the lane for vehicles with no priority. This one can be physically detached or

simply signaled through road surface markings. Some countries are characterized by full-time or part-time traffic hours (rush hour and off-peak hour);

- **Reverse reserved lane:** a lane of the carriageway reserved for public and other means with priority; in this lane though the means with priority always travels in the opposite direction of the means with no priority travelling in the contiguous lane. This kind of reserved lane is quite always physically detached from the rest of the infrastructure and works full-time; it is often used on one-way streets in order to reduce the travelling distances of buses and to provide reserved access to areas of interest for passengers (shops, offices, etc.).

In order that these reserving tools perform completely their priority function of public transportation, in terms of travel time saving and reduction of the delay during its working, they must be integrated with the traffic regulation systems to realize a traffic light priority system.

The methods to grant priority to public means at traffic lights can be grouped into the following categories:

- Passive priority: traffic signal timing is designed to provide priority benefits to bus flows without buses being monitored one by one.
- Active priority: traffic signal timing is changed to give priority at traffic lights to each bus individually approaching intersections, on the basis of the control strategies. Active priority can be reached through:
 - ordinary sensors on the reserved lanes;

- special sensors on the mixed lanes able to identify public means (by their length or weight);
- systems included in the onboard equipment (like tags or transponders) communicating with the street infrastructures at the intersections.
- Traffic light priority using pre-signals: it is a queue management method using traffic lights to manage congestion and queues, in order to put buses in a position where their travel is not delayed by queues of private vehicles. This method involves pre-signals, whose task is to keep general traffic at the upstream of the intersection, in order to assure the buses a reserved access to them.

Road space reservation		Traffic light priority	
Physical	Virtual	With sensors	Without sensors
Bus-only street	Bus lane	Active Pre-Signals	Passive
Busway			
Reserved lane			
Reverse reserved lane			

Table 1 - Diagram of reserving systems of classification

For this work active and passive traffic light priority measures were employed without using pre-signals, which can represent a possible future development.

2.2.1 Strategies of signal priority for buses

The signal priority strategies for public transportation is aimed at reducing the delay of buses at the signalized intersections. The basis for this special consideration is the high capacity of vehicles. Typically, traffic signals are

designed to minimize the total delay of all the vehicles at an intersection. However, minimizing the delay of the vehicles could not be optimal considering the passenger load of the vehicles. For example, a delay of 30 seconds for a crowded bus clearly is not equivalent to a 30-second delay for a vehicle with an only one passenger. So it would be better as a unit of measurement the total delay per person instead of the total delay for vehicle. Granting priority to public means is then more suitable way to minimize the total delay per person and maximize the number of people passing. In Figure 2-2 it is shown how the delay for a means of public transportation can be caused by a traffic signal in absence of traffic light priority methods. The trajectory of a means of public transportation is drawn in a space-time diagram and the horizontal straight line represents a traffic signal with its indication displayed during the time. If in the vehicle trajectory an forbidden-way traffic light comes, the delay increases until the displaying of free-way traffic signal when the vehicle can proceed. Signal priority strategies attempt to reduce the delay in two ways:

- by reducing the probability that a means of public transportation gets an forbidden-way traffic signal;
- by reducing the waiting time, in which the traffic signal gives again free way, when encountering an forbidden-way signal.

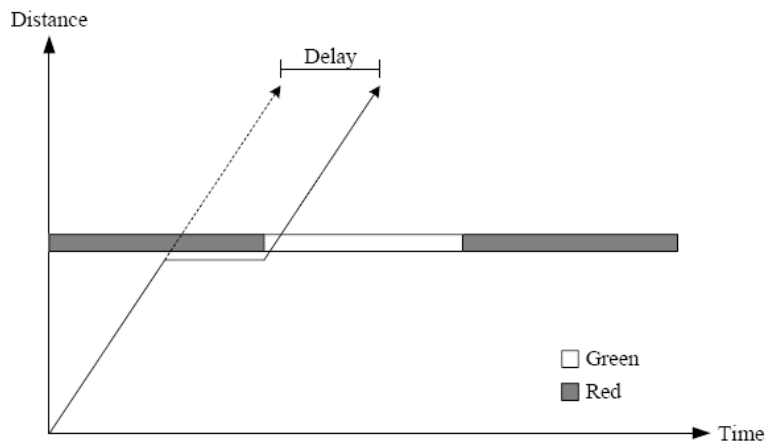


Figure 2-2 - Vehicle trajectory without traffic light priority

Passive signal priority strategies

Passive priority is defined as the use of static traffic regulation methods to reduce the delay of public transportation.

Many are the static signal priority strategies:

- Increase of the green time for the road artery with public transportation lines. It is the simplest kind of passive priority strategies and provides more green time for roads with public transportation lines through increased green split for the phase in which the public transport vehicles have the right to pass. Since it reduces the percentage of cycle time during which the phase of the public transportation receives the forbidden-way signal (reduction of red time), both the probabilities that the bus arrives during the red phase and the average waiting time decrease.

- Use of short traffic light cycles. Another passive strategy is to use a certain duration of the short cycle, that can reduce the delay shortening the waiting time up to the successive green phase. However, this implies a reduced capacity for the intersection, especially due to the increasing of the time lost (the time in each cycle during which there are no vehicle movements). The loss of time is typically caused by full-red time plus the starting delay at the beginning of each phase (time of evacuation at setting in move), then in each cycle it is independent of the cycle duration lost during the loss of time. If an intersection is close to saturation, delays could be currently increased. But if there is an excess of capacity, this strategy can reduce the delays for single vehicles.
- Division of the green phase across the corridors of the public transportation. With this strategy the free-way phase across the corridors of the public transportation implies a double phase within the same cycle. The duration of the cycle can remain unchanged if each of the two green phases is half of the original duration. The benefits means of transport obtained through this strategy include the reduction of the time amount between the green phases reducing the waiting time for the vehicles that received the red time.
- Traffic light synchronization. In this strategy the arterial progression, for example, can be designed to favor the public transportation vehicles through the timing of green bands with the average bus speed instead of the average private vehicles speed. Although this strategy increases the travel time for the drivers, it helps assuring that the public transportation vehicles can be in line with the traffic signal

progression. However, in the urban area the progression for the buses could be difficult to respect because of the stops, which impede this vehicles to move at a constant rate within the network.

A general problem occurring with the passive priority strategies is the fact that they typically lead the intersection to work overall less efficiently, especially if the frequency of the public transportation is not very high. For this reason, the strategies cannot be always realizable in particular in conditions of oversaturation. In some cases, using short cycles or longer green splits for the public transportation, the oversaturation of the intersection occurring would cause long queues and delays. They represent the limits of the passive priority strategies, but in different cases they are the only realizable options in particular when the evaluations of the costs require the use of the existing control system, so the recent research quantity within the passive strategies is minimal (Skabardonis, 2000).

Active signal priority strategies

The active strategies face the limitations of the passive strategies by alternating dynamically the traffic light regulations and, only when necessary, by making adjustments in real time and by timing traffic lights to minimize the delay of the public means to the approach. That is intensive from the infrastructural point of view with respect to the passive strategies, because it requires tools to detect the public vehicles upstream on the intersection and control systems ahead to use the strategies to accept the priority.

There are three basic actions that a control system can perform to react to the detection of the means of public transportation:

- Green extension in the current phase. If the bus approaches to the intersection at the end of the green time interval in its direction, then the current green time interval can be extended until the vehicle overcomes the intersection as shown in the picture below.

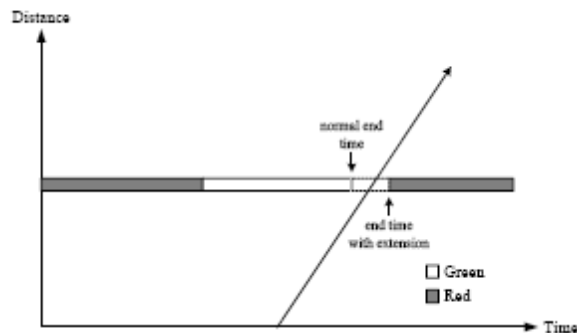


Figure 2-3 - Bus Trajectory with green extension priority

Without extension, the vehicle would have to wait for the green light of the successive cycle with a consequent significant delay.

If the vehicle is approaching to a red light signal, there are two options:

- If the vehicle gets normally the green light in the successive phase, the current phase could end in advance to enable the vehicle to get sooner the green light. That occurs if the vehicle arrives at the intersection almost at the end of the red phase for its direction, as shown in the picture below.

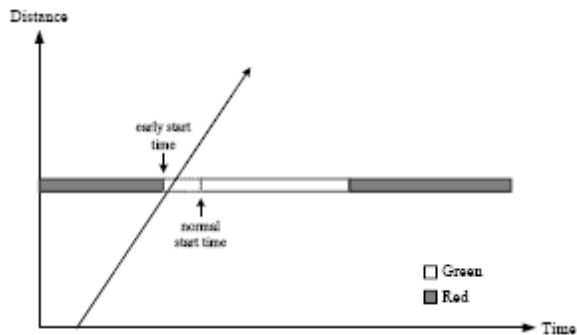


Figure 2-4 - Bus Trajectory with green anticipation priority

- If other phases have to be served before the return of the normal green time, a short phase can be inserted for the approach of the public means with the control system that can function normally again once the vehicle is passed. This case is shown in Figure 2-5, where the control system interrupts its normal traffic signal plan to serve the phase of public transportation before returning to the regular timing.

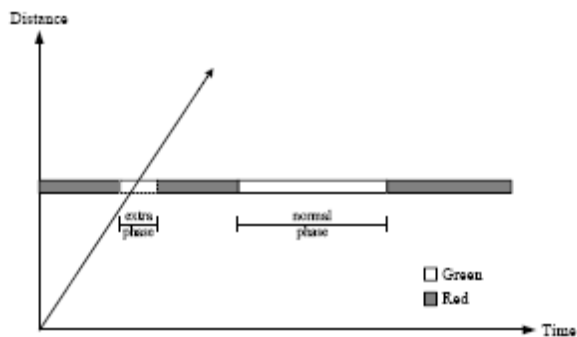


Figure 2-5 - Bus Trajectory with green insertion priority

More interesting is the effect on private traffic that active priority strategies can cause.

Under conditions of light traffic flow, active priorities can have little effects on the general traffic flow, because the exceeding capacity within the cycle can be redistributed to the phase of public transportation.

However, active priority can have greater negative effects during the rush hour, when the intersections operate very close to saturation with short or no time to lose for the movements of the non-public means.

In turn, the active strategies divide into 3 categories:

- Unconditional strategies;
- Conditional strategies;
- Adaptive strategies.

The unconditional strategies provide the state of priority to each public means detected. This means that the control system will try to start one of the priority actions described above, once any means of public transportation is detected. The disadvantage of this strategy is the fact that priority could be granted to any public vehicle, which might not need it, as for example a vehicle in advance with respect to its schedule. However, such priority requires only information about the presence of the vehicle sent to the control system.

The conditional strategies grant the state of priority on the basis of certain criteria, which in most cases are linked to the specific means of transport. The most common criterion for conditional priority is the vehicle delay with

respect to its schedule. However, other criteria, such as the vehicle frequency or the passenger load, are taken into consideration for different applications.

The adaptive signal priority strategies for public means use control patterns based on the optimization, in order to determine if and how to grant priority. In some patterns, the delay of the public means is considered together with the delay of all the other vehicles. The control system then calculates the optimal solution consisting in the way to distribute the green time among the current approaches. As phases and times are not fixed, adaptive strategies do not require the predefinition of specific priority actions, like the extension or the insertion of a phase. The control system changes constantly the distribution of green time according to the demand. Priority strategies for public means can easily be implemented within more adaptive existing systems, giving more importance to public transportation vehicles in the optimization routine. The implementation of the traffic light priority for public means within an already existing adaptive network control system can cause some errors.

This kind of system in fact considers as part of the optimization the effects of the global regulation, while giving signal priority is competence of a local control system. That can lead to conflicts of aims in the optimization giving then sub-optimal results. Another problem is the fact that the most part of the adaptive control system use macroscopic traffic models in their routine of estimation and optimization. These models cannot catch certain details in the movements of the public transportation means. For example, the waiting time at the stops of the public transportation and the interactions among the public vehicles and the others would not be considered, so that the travel time for the public means could be underestimated. Finally, the restrictions on the

optimization could limit the opportunities for the priority of the public means, especially during the rush hour when it is more important. For example, a binding factor can be the maximum length of queue acceptable on each approach; in very congested conditions this restriction could be always active, not enabling to assign the additional green time to the approach to reserve.

The tools to detect the presence of public means, to which the active traffic light priority is granted, can be:

- ordinary sensors on the reserved lanes;
- special sensors on the mixed lanes able to detect public means (by a different Earth's magnetic field alteration);
- systems included in the onboard equipment (like tags or transponders) communicating with the street infrastructures at the intersections.

The most recent systems help managing platoons in real time with information to passengers at the bus stops, assuring the traffic signal reserving on the basis of the service needs, aiming at improving regularity and punctuality. There is a clear tendency to increase the use of GPS (Global Positioning System) technology for bus localization. This technology helps lower the operative costs and represents a flexible solution for the localization of the public means.

2.3 Existing simulation and optimization models

The model proposed in this work is built both on a simulation model to evaluate the performance of a signalized road artery and on an optimization algorithm for the traffic light synchronization. This chapter presents the most known simulation and optimization models of the traffic light synchronization, available in the literature and often applied on real cases or on commercial developments. Many models proposed are in fact software and appear as black-box to the user. So the most of the information presented is taken from the manuals coming with software or available online.

2.3.1 Simulation models

Traffic simulation models are divided into:

- microscopic and discrete models;
- macroscopic and continuous models (discretized to implement a simulation);
- mesoscopic models.

The traffic microsimulation models help viewing lifelike movements of the single vehicles and following the development of traffic on the road network. These models, simulating the behavior of each single vehicle with its own origin and destination, provide all the elements for a detailed quantitative analysis. They are disaggregate models, as they reproduce the movement of

the single vehicle, but often are also used for aggregate analysis (flows, lengths of the queue, travel times, etc.).

In the macrosimulation the traffic is normally described as a flow defined by behavioral rules based mainly on the interaction of vehicles among them and with the infrastructure. The macroscopic models, also known as continuous time and space models, are based on the continuous simulation of the traffic. The mathematical theory underlying such models is set on the time and unidimensional dynamics, respecting the flow conservation laws.

In the last few years some mixed, mesoscopic models were proposed, based on the simulation of the microscopic or quasi-microscopic kind, which presents macroscopic features (as for example the concepts of aggregate speed and density). In general, mesoscopic models have the great advantage of being less onerous from the computational point of view, but present the disadvantage of less detailed representations of the vehicles behavior.

Dynameq

DYNAMEQ, designed by M. Florian (2005), is a dynamic traffic assignment (DTA) model, which uses some variants of the gradient method and the method of successive averages (MSA) to determine the choice of path in a condition of dynamic equilibrium.

The choice of path are shaped as decisional variables ruled by the user optimal principle, according to which each user of the network aims at minimizing their costs and so the travel time on the path used. All the users access the network information and therefore to the travel time on all the paths (used and not used).

The resolution algorithm is an iterative procedure realized to converge towards these conditions and is characterized by two main components:

1. a method to determine a new set of path flows variable in time considering the knowledge on the travel times in the previous iteration;
2. a method to determine the arc flows and the travel times resulting from a given set of incoming path flows. That refers to the network loading phase carried out through an efficient event-traffic simulation model. The model represents explicitly the available regulation systems and catches realistically the congestion phenomena, such as the queue formation and their propagation from arc to arc (spillback).

Two different approaches are normally used to simulate the user choice behavior: a dynamic "en route" assignment and a dynamic equilibrium assignment. In DYNAMIQ the approach used is the search for an approximate solution in equilibrium conditions.

In the equilibrium assignment problem only the pre-trip path choices are taken into consideration and they are the choices made by the user before taking their path. The choice of path are shaped as decisional variables ruled by the user optimal principle, according to which each user of the network aims at minimizing their costs and so the travel time on the path used. All the users access the network information and therefore to the travel time on all the paths (used and not used). The resolution algorithm is an iterative procedure realized to converge towards these conditions.

The resolution algorithm

The resolution algorithm used in DYNAMEQ consists of two main components beyond the phase of calculation of the minimum dynamic paths:

1. a method to determine a new set of path flows variable in time considering the experience developed on the travel times in the previous iteration;
2. a method to determine the arc flows and the travel times resulting from a given set of incoming path flows.

Moreover, the algorithm requires an initial set of path flows. The general structure of the algorithm is shown in Figure 2-6.

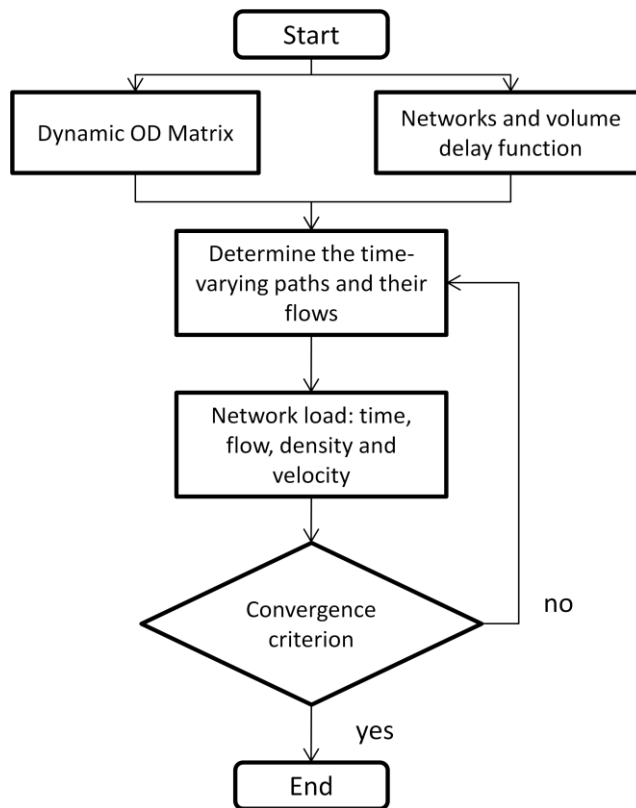


Figure 2-6 - Resolution algorithm structure of the dynamic traffic assignment model

Defining K the set of all paths, the path flows of h_k^τ , $k \in K$ input are determined by a variation of the method of successive averages (MSA), applied to each origin/destination couple and time interval τ .

An initial set of possible solutions is calculated assigning the demand for each time interval on a set of successive minimum paths. From the second iteration and up to a maximum predefined N number of iterations, the arc travel times variable in time after each loading are used to determine a new set of dynamic minimum paths added to the set of current paths.

In the iteration $n, n \leq N$, the volume assigned on each path belonging to an initial set is equivalent to $\frac{g_i^\tau}{n}$, where g_i^τ is demand of O/D pair in the time interval τ . Afterwards, for each iteration $m, m > N$, the minimum paths are identified among those actually used and the path flows are redistributed across the paths known.

If the flow on a particular path goes under a predetermined value, the path is not taken into consideration and its flow is distributed on other paths used. This heuristic approach is similar to the Lawphongpanich, and Hearn's algorithm (1984) for the solution of the stable fixed-demand equilibrium model.

The algorithm is summarized below (Figure 2-7):

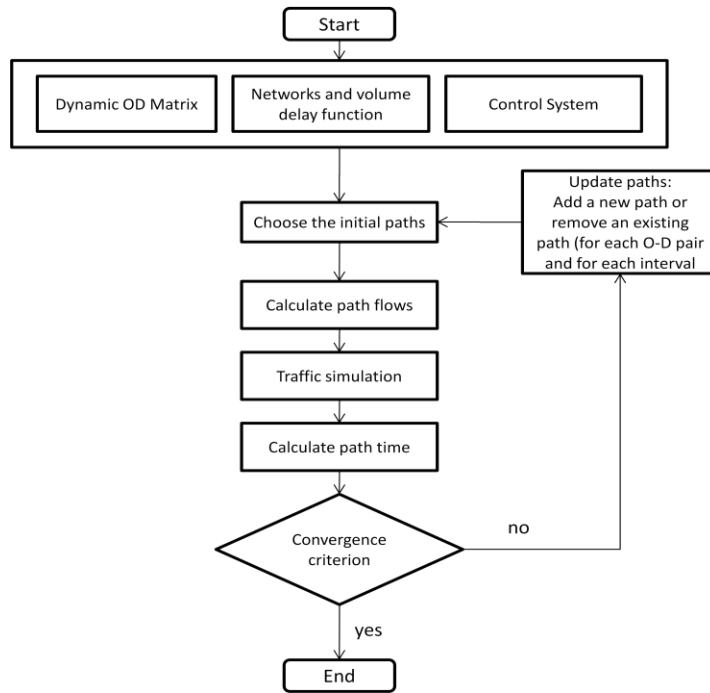


Figure 2-7 - Dynamic equilibrium assignment algorithm in dynameq

It is important to know that although this model is very close to the macroscopic models of outflow, is indeed a discrete model. The network loading procedure, carried out through an event simulation, moves the single vehicles on the network arcs.

The network loading model

The simulation model core is the car following simplified model:

$$x_f(t) = \min[x_f(t - \varepsilon) + \varepsilon V, x_l(t - R) - L]$$

Eq. 2-2

where $x_f(t)$ is the trajectory of a vehicle (function in time), L is the actual length of the vehicle, R is the user reaction time, V is the free speed and ε is an arbitrarily short time interval. The subscripts f and l denote respectively the trajectories of the follower and the leader vehicles. This model take into consideration the only free speed (without acceleration constraints) combined with a simple model of collision avoidance and can be easily represented through the well-known triangular flow-density diagram, also called fundamental diagram of traffic flow (Mahut, 2000).

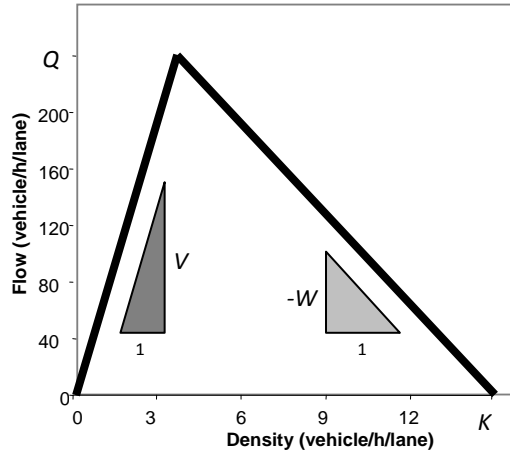


Figure 2-8 - Triangular flow-density diagram

This relation can be solved rigorously, as well as it is possible to calculate the time in which a vehicle enters and exits each arc using this formula:

$$t_n(0) = \max \left[t_n(-X_1) + \frac{X_1}{V_1}, t_{n-1}(0) + \left(R + \frac{L}{V_2} \right), t_{n-X_2/L}(X_2) + R \left(\frac{X_2}{L} \right) \right]$$

Eq. 2-3

where X_1 and X_2 are the lengths of the upstream and downstream arcs with respect to the position $x=0$, V_1 and V_2 are the free speeds respectively on these two arcs and the subscripts represent the vehicles sequentially numbered. This arc formulation provides a very practical and computational-efficient way to shape the traffic flow without calculating analytically the state variations (position, speed, etc.) of each vehicle at each instant. Please note how this expression can be applied only to the case of an arc with just one lane (Figure 2-9).

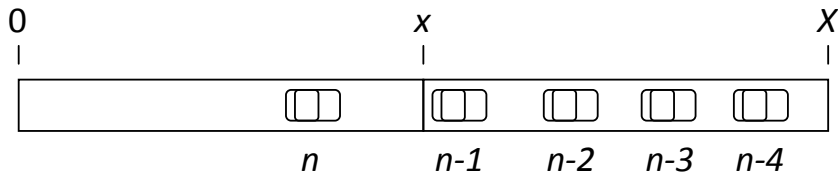


Figure 2-9 - Simplified model of queued vehicle

In case of a multi-lane road, a different formulation needs to be used to calculate the entrance and exit time for each vehicle, but also to catch the interactions among vehicles because of the lane change (Mahut, 2000). The model for multi-lane roads develops a series of heuristic procedures to form the lane choice by the user, who bears in mind their wished path upstream to their current position, the lanes likely to be used for the successive turn and the main traffic flow conditions on each lane interposed between the driver and the end of the arc.

The above said model of arc determines the instant when a vehicle wishes to enter the arc as function of the entrance and exit time history. The information is used for a rigorous application of the previous model of queued vehicle.

In short, the car following model is extended in a such a recursive way that it is applied to a sequence of vehicles. For instance, rather than forming the relation between vehicle 1 and vehicle 2 (the higher number follows the lower one) and the consequent relation between vehicle 2 and vehicle 3, the model enables to express directly the interaction between vehicle 1 and vehicle 3. That can be extended to any number of vehicles, and this is actually the essence of the arc expression previously mentioned. From a conceptual point of view, since the first cause for the delay in a network is due to the intersections, the role of dynamics in this model is to propagate appropriately the delay between upstream and downstream vehicles. That is to say that these delays can affect the entrance and exit times in the arc. Instead of forming explicitly the position of each vehicle to determine when the congestion of the arc starts to affect these entrance times, the exit delays are transmitted directly in entrance. This peculiar feature - the ability to solve rigorously the traffic flow model on the whole arc - has been proved also in the case of the kinematic wave model based on the fundamental triangular diagram (Newell, 1993).

This dynamic traffic flow model, that can be characterized as a model continuous in time and space and with discretized flow, is combined with a node model representing explicitly the predetermined traffic signal systems and also shaping the interactions among vehicles at non-signalized intersections through a *gap-acceptance* logic. The combined system is then solved by using an event algorithm enabling to form the whole network. Event models are basically different from the models for discrete time interval, that is:

- the models for discrete time intervals provide that at each time step (usually 1 second or less) all the data required for each vehicle in each single model (queued vehicle, lane change, *gap-acceptance*, path choice) are updated and the outputs (acceleration, deceleration, etc.) are re-calculated;
- event models provide that the single submodels are updated only if one of the inputs changes considerably. So the single submodels assure that their outputs keep their validity as long as the input data keep constant. An event is generated in a specific point if the information changes. Usually, the performance of an event results in the creation of one or many sequences of events.

The result in this case is a very computational-efficient event model, which though respects also the basic laws of the traffic flow and represents explicitly the congestion mechanism that occurs in real life.

From all that, higher calculation time savings derive with respect to the time interval simulation models (in one or two levels of size).

The computational time are particularly important in the context of dynamic traffic assignment models, which must perform several iterations to solve the equilibrium assignment. The time factor is even more important, because dynamic traffic assignment models must be applied to big networks.

Dynasmart

In Dynasmart (Mahmassani, et al., 1992) the traffic flow simulation derives by the MPSM model¹: the generic vehicle (or the group of vehicles) travels on the network according to the discrete space intervals and its position is updated every 6 seconds (spatial and time discretization).

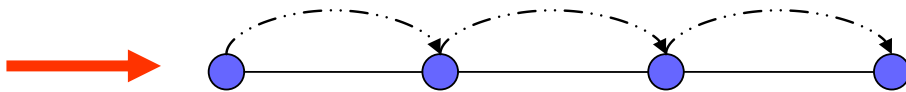


Figure 2-10 - Spatial discretization of the network arcs

The traffic flow model used consists in the speed-density relation deriving by the change of the well-known Greenshield's equation:

$$v = v_0 + (v_f - v_0) \cdot \left(1 - \frac{k}{k_j}\right)^\alpha$$

Eq. 2-4

where:

- v_0 : minimum speed
- v_f : speed with null flow
- k_j : maximum density
- α : calibration parameter

Dynasmart then lets vehicles move on each arc at the speed resulting from Greenshield's equation changed, knowing the density at the end of the previous time step.

¹ Macroparticle Simulation Model (Chang, Mahmassani, Herman 1985)

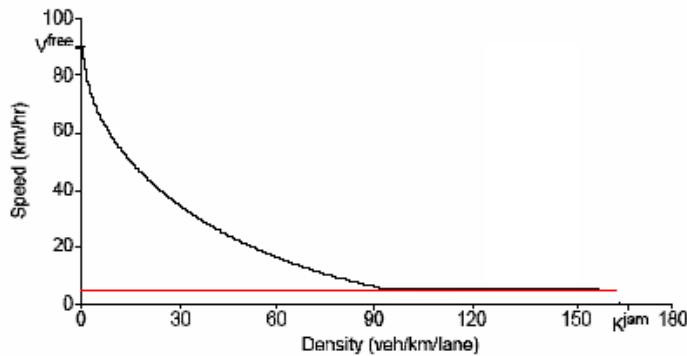


Figure 2-11 - v-k relation (Greenshield's equation changed)

From Figure 2-11, the model always assures the minimum speed also if the prefixed maximum density is overcome.

At the intersections, restrictions are determined on the inflow and the outflow. Especially for inflows, they are required to be equivalent to the minimum value between the sum of flows travelling on the arc and the value of the flow travelling on the arc until they reach the maximum density or capacity. As regards outflows, they are required to be equivalent to the minimum value between the vehicles that must exit the intersection in the interval Δt and the vehicles that can really enter the arc in Δt .

The conflicts occurring at the non-signalized intersections are solved by providing at each approach a fictitious green time calculated in relation to the critical volumes in the generic interval Δt .

Furthermore, predetermined and applied traffic light controls, as well as the ramp metering systems, can be introduced.

Going to the core of the dynamic traffic assignment model, it is important above all to underline the fact that Dynasmart enables to realize both an equilibrium assignment (1st Wardrop's principle) and a system optimum assignment (2nd Wardrop's principle).

In particular, the equilibrium assignment follows an iterative procedure of this kind:

1. it starts from building a set of acceptable paths and assigns the demand on this set;
2. it carries out the traffic flow simulation obtaining the network performance;
3. for each origin/destination couple and for each departure instant τ , it calculates the minimum-cost set of path;
4. it performs the total or null assignment detecting the auxiliary flows on each arc and for each instant τ ;
5. it updates the flows through the method of successive averages;
6. it performs a check of convergence on the path flows between the iteration $i+1$ and the iteration for the o/d couple and for each τ .

As regards the information systems, Dynasmart enables to simulate variable message signs (VMS) and the route guidance systems, and enables to form the user reaction till the definition of different classes and the prevision of a bounded rationality model (Mahmassani and Stephan 1988), whose generic user decides whether to change the current path only if the benefit exceeds a prefixed threshold bound. A software like Dynasmart is clearly mainly addressed to the experimentation and checks the different strategies using ATIS systems.

Simulation and optimization with cell transmission models

In the Cell Transmission Model, proposed by Deganzo (1995), the road is divided into homogeneous sections, called *cells*, numbered consecutively from the extreme initial with $i = 1, \dots, I$. The lengths of the sections are not arbitrarily chosen: they are equal to the distance traveled by a typical vehicle, in a not too intense traffic situation (with free flow speed), during a moment of time. Hence, in such conditions, it is assumed that all vehicles in a cell move ahead in the next one after each interval, without knowing their actual position within the cell. The evolution of the system respects the rule:

$$n_{i+1}(t+1) = n_i(t)$$

where $n_i(t)$ is the number of vehicles in the i cell at t time. This condition occurs for all flows, unless the traffic is slowed by a downstream queue due to a "bottleneck" narrowing. This assumption seems reasonable because, under crowded conditions that may arise during rush hour, most of the delays can be attributed to the jam traffic, where the flow temporarily exceeds capacity, rather than to any dependence between flow and speed. Two constants are introduced to include the queue:

- $N_i(t)$ is the maximum number of vehicles that may be present in the i cell at the t time; is the product of the cell length and its "*critical density of congestion*";

- $Q_i(t)$ is the maximum number of vehicles that can enter the i cell when the time interval proceeds from the t to the $t+1$, corresponding to the minimum "capacity flows" of $i-1$ and i cells.

$Q_i(t)$ will be called "capacity" of the i cell, as it represents the maximum flow that can be transferred from $i-1$ to i . The two constants can vary over time, so as to shape transitory traffic episodes. The number of vehicles flowing from the $i-1$ to the i cell during the time interval from t to $t+1$ (or inflow in i for the time interval after t), indicated with $y_i(t)$, is the smallest of the three values:

- $n_{i-1}(t)$: number of vehicles in the $i-1$ cell at the t time;
- $Q_i(t)$: flow capacity in the i cell for the t time interval,
- $N_i(t) - n_i(t)$: amount of empty space in the i cell at the t time.

The simulation, first proposed by Daganzo (1995) and called "Cell Transmission Model", is based on recursion, where the cell occupation at the $t+1$ time is obtained by balancing the vehicles present at the t time with inflows and outflows.

$$n_i(t+1) = n_i(t) + y_i(t) - y_{i+1}(t)$$

Eq. 2-5

where the flows are related to current conditions at the t time, as indicated below:

$$y_i(t) = \min \{n_{i-1}(t), Q_i(t), N_i(t) - n_i(t)\}$$

Eq. 2-6

The simulation proceeds over time, updating the amount of occupation of the cells at any time.

The simulation result is independent of the order in which cells are considered at every step. This important property of the *Cell Transmission Model* arises because the number of vehicles entering a cell is not related to the number of vehicles that will come out; so only the current conditions will affect the inflow. Moreover, with queue, it is assumed that the propagation speed w of the backward density wave is equal to that of the free flow ($w = v$).

It is demonstrated that the equations Eq. 2-5 and Eq. 2-6 are the discrete approximation of the hydrodynamic model, with the flow-density relation (k - q) in the form of an isosceles trapezoid that can be expressed by the following relation:

$$q = \min\{v \cdot k, q_{\max}, v \cdot (k_j - k)\}, \text{ by } 0 \leq k \leq k_j$$

Eq. 2-7

with the maximum flow $q_{\max} \leq v \cdot k_j / 2$. Changing Eq. 2-7 in the flow conservation equation:

$$\partial q(x, t) / \partial x = -\partial k(x, t) / \partial t$$

Eq. 2-8

the differential equation, that defines the evolution of the system in the hydrodynamic model, is obtained as below::

$$\partial \min\{v \cdot k(x, t), q_{\max}, v \cdot (k_j - k(x, t))\} / \partial x = -\partial k(x, t) / \partial t$$

Eq. 2-9

under suitable assumptions (the characteristics of the cells are independent of i and t , so $N_i(t) = N$ and $Q_i(t) = Q$; moreover dt is equal to one instant of time of the Cell Transmission Model and unitary cell length is equal to $v \cdot dt = 1$), the variable with curly braces is equivalent to:

$$\min\{n_i(t), Q, N - n_i(t)\},$$

which coincides with the definition of $y_{i+1}(t)$ in the Eq. 2-6 equation (provided that the density is differentiable in x). As a result, the left hand side of the equation Eq. 2-9 is $y_{i+1}(t) - y_i(t)$, while the right one corresponds to $-[n_i(t+1) - n_i(t)]$ and the equality of the two relations justifies the recursive simulation Eq. 2-5.

The Cell Transmission Models is equivalent to the hydrodynamic model even with gradual change of density in space. It also deal with the discontinuities of the density tested on sections of transition (consistent with the width of the grid of cells), affecting one or two cells.

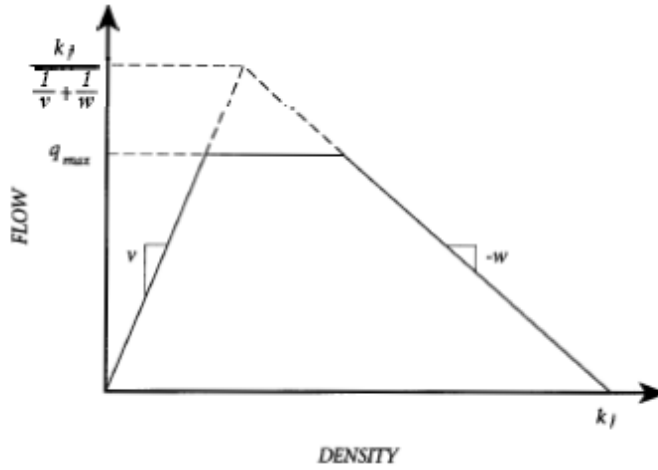


Figure 2-12 - Equation of state of the Cell Transmission Model

The equation of state Eq. 2-7 is extended to the case where the backward propagation waves have a lower speed than that of the free flow $w \leq v$ (as occurs in the real case) with the flow-density relation $(k-q)$ that takes the form of a scalene trapezium and it becomes:

$$q = \min\{v \cdot k, q_{\max}, w \cdot (k_j - k)\}, \text{ by } 0 \leq k \leq k_j$$

Eq. 2-10

where $w \leq v$ and $q_{\max} \leq k_j / (1/v + 1/w)$.

Ultimately the cell transmission model is defined as:

$$n_i(t+1) = n_i(t) + y_i(t) - y_{i+1}(t)$$

Eq. 2-11

$$y_i(t) = \min\{n_{i-1}(t), Q_i(t), d_i^t \cdot (N_i(t) - n_i(t))\}$$

Eq. 2-12

where

$$d_i^t = 1, \text{ if } \cdot n_{i-1}(t) \leq Q_i(t)$$

$$d_i^t = w/v, \text{ if } \cdot n_{i-1}(t) > Q_i(t)$$

and the differential equation that defines the evolution of the system in the hydrodynamic model is:

$$\partial \min\{v \cdot k(x, t), q_{\max}, v \cdot (k_j - k(x, t))\} / \partial x = -\partial k(x, t) / \partial t$$

Eq. 2-13

The cell transmission model has the potential to describe the phenomenon of real instability not included in the hydrodynamic theory.

Daganzo (1995) describes how to replace the representation of a general network with a more detailed graph, where the cells that constitute the nodes of the new structure, are connected by connectors or links (so called to avoid confusion with the arcs of the original network). For every cell, the values of Q_i^t and N_i^t are indicated, while each link is characterized by the moving flow y_{ij}^t from the beginning cell i to the end cell j in the t time interval. It's important to emphasize the distinction between the arcs of the original network, which are elements on which physically you experience movements of vehicles and connectors and which do not cause any traffic, but that indicate only the way in which vehicles move from one cell to another.

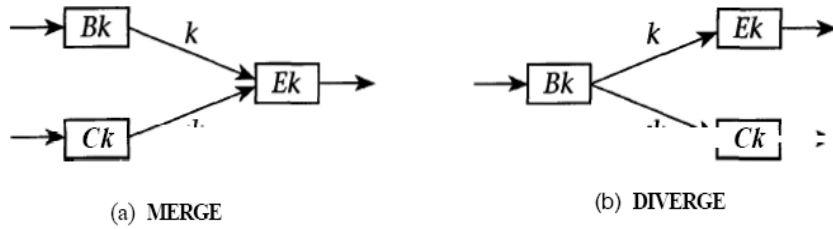


Figure 2-13 - Representations of convergence and divergence

The set cells is divided into five subsets: the ordinary cells (C_O), characterized by one previous section and by a subsequent one, the diverge cells (C_D), with only one upstream section and two subsequent, the merge cells (C_M), with two previous sections and a subsequent, the source cells (C_R), with only one subsequent cell and the target cells (C_S), characterized by only a previous cell. Even the links follow a similar classification; they are divided into: ordinary connector (E_O), combining two ordinary cells; diverge links (E_D), when the beginning cell is diverge; merge connectors (E_M), when the beginning section is converge; source links (E_R), with one initial source cell and destination connectors (E_S), when the end cell is a destination cell. It is suggested that there are no links that connect directly a merge cell with a diverge cell (or vice versa), as it is possible to structure the sections (related to the duration of time intervals with which the planning period is discretized) so that there is always interposed an ordinary cell.

The general procedure of the *Cell Transmission Model* for networks includes two steps for each time interval:

Step 1- Determining the flow on each connector with the equivalent of the equation Eq. 2-12;

Step 2 - Updating the occupations of the cell by transferring the flow of *Step 1* from the beginning cell to the end cell of every link.

The original definition by Deganzo does not explicitly mention the signalized approaches that can easily be implemented either with reduced capacities as a function of free way time but assuming a constant flow, or with a signalized capacity of on / off.

Platoon Progression Algorithm

The model developed by Papola and Fusco (1996) is based on the hypothesis of Robertson (1969) to represent groups of vehicles through entities called platoons. The vehicles within platoons are characterized by uniform density and speed.

With this assumption Papola and Fusco (1996) developed an analytical model of delay, as well as the advantage of linearity with the flow, under some simplifying assumptions, also had the significant property to allow to express the delay as a function of bandwidth. This is a functional link between the two problems highlighted in Papola and Fusco (1998b) in the case of an artery with a constant flow and then extended in Papola and Fusco (2000) to the general case.

The model provides an expression for the delay of a generic platoon dependent moments of arrival and end of each platoon in comparison to the moments of beginning and end of the red light. In particular, there are three different cases, described below.

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 Figure 2-14):

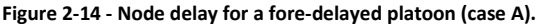
$$\mathcal{G}_i - \frac{r_i}{2} \leq t_{i,p} < \mathcal{G}_i + \frac{r_i}{2} + \tau_i$$

$$D_{i,p} = q(v_s)l_{i,p} \left[\mathcal{G}_i + \frac{r_i}{2} + \tau_i - t_{i,p} \right]$$

where:

- \mathcal{G}_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 .


$$\mathcal{G}_i + \frac{r_i}{2} + \tau_i \leq t_{i,p} < \mathcal{G}_i + C - \frac{r_i}{2}$$

where C is the cycle length at node i .

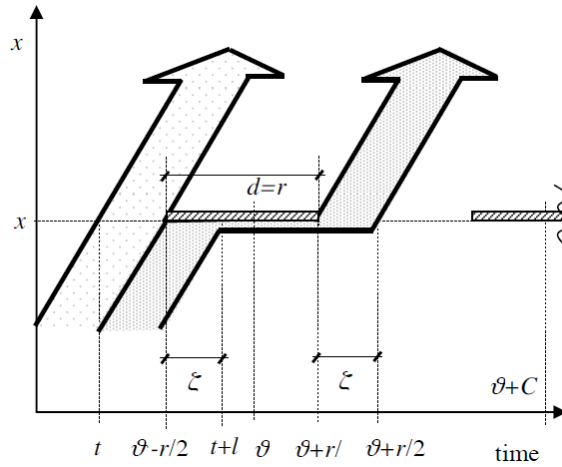


Figure 2-15 - Node delay for a rear-delayed platoon (case B).

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:

$$\mathcal{G}_i + \frac{r_i}{2} + \tau_i \leq t_{i,p} \leq \mathcal{G}_i + C - \frac{r_i}{2}$$

$$t_{i,p} + l_{i,p} \leq \mathcal{G}_i + C - \frac{r_i}{2}$$

A seguito di queste regole di classificazione, è stato sviluppato da Papola e Fusco un algoritmo di simulazione per valutare il ritardo complessivo su un'arteria sincronizzata con ciclo semaforico unico per tutte le intersezioni.

Advantages and disadvantages

The advantages of this model are mainly the possibility of evaluating the delay of an entire platoon in analytical form and then in constant time and the

ability to assess the delay of a particular traffic light synchronization quickly. On the other hand the simulation model developed by Fusco and Papola can evaluate the delay only in case of under saturation.

2.3.2 Synchronization and optimization models

In order to determine the best strategy for the network management, the algorithms need a special function, called "performance function", to assess precisely the network performance for every possible solution, thus indicating its quality.

Depending on whether the function is a favorable or unfavorable indicator, the purpose of the optimizer will be respectively maximize or minimize it. The variables to respect to perform this optimization are the control variables of the problem: typically cycle times, green times and offset.

The functions that can be assigned to evaluate the performance of a network are numerous and every function refers to a different logic approach to the problem.

A widely used target, initially in the case of individual intersections, was the minimization of cycle time that determined the minimum cycle time which can withstand the demand so that the cycle time can be increased up to the necessary to keep the network synchronized.

Another kind of adopted maximization is defined as "network capacity". Writing the demand matrix as μQ , where Q is the "ordinary" demand matrix,

the criterion tends to maximize μ meeting the eligibility restrictions of the flow carried by the network. The idea is that the higher the value of μ obtained, the lower the saturation of the network when it had to withstand a demand of Q (to be precise for every arc the saturation rate is equal to z_i/μ , with z_i equal to the saturation rate obtained in the optimization with demand of μQ).

A very widespread performance function nowadays is the function of total delay on the network, defined as the sum of time spent in queue by all vehicles on the network.

A similar assessment of the performance of the network is given by the appropriately weighted sum of the delay of vehicles on the network and the number of stops they are forced to perform along their path.

In their algorithm based on sensitivity analysis (Ying et al., 2004), as evaluation function of the network traffic Ying et al. use the total time spent by vehicles while crossing the network to make their origin-destination travel.

De Schutter presents a problem formulation as a problem of complementarity (De Schutter, 2001) and provides many possible objective functions:

- Weighted average of the length of all queues on the network in the time period;
- Weighted average of the longest queue length for every instant of time within the time period;
- Longest queue length obtained within the time period;

- Weighted average of the waiting times of all queues on the network in the time period;
- Weighted average of the longest waiting times in the queue for every instant of time within the time period;

The first model / software to be described is Transyt, given the high degree of similarity with the model proposed and developed in this thesis.

Transyt

TRANSYT is an off-line application that, assuming stationarity of flows, simulates the progression of platoons and determines the traffic signal timing coordinated for a road network of which the average flows are known.

It was developed by the British Transport and Road Research Laboratory in 1969 based on the platoon progression model by Robertson (Robertson, 1969) and, although the basic approach has remained essentially the same, over the years there have been many following versions to implement the features to make it a more flexible tool to the service of transportation engineering. The program is now at version 14, one of the first to be developed as a fully graphical version. In the U.S. in parallel it has been developed a version with the right directions, called TRANSYT-7F, which adopts the same resolution procedure. Both versions of TRANSYT are structured in two modules working in parallel: the traffic model and the optimization algorithm.

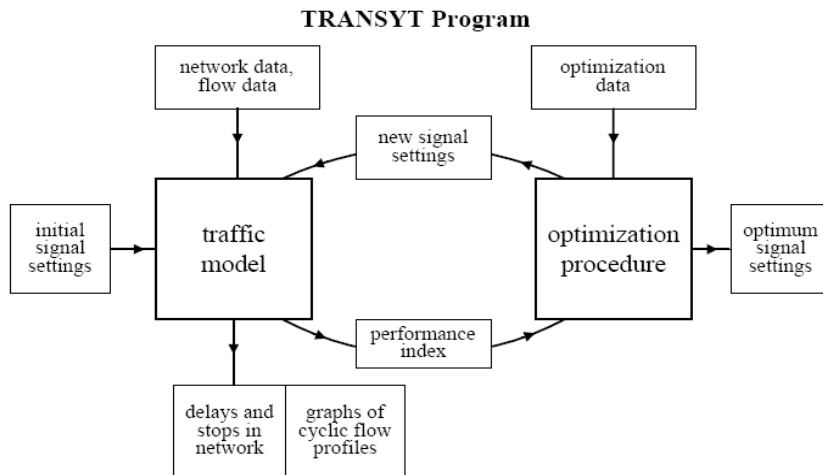


Figure 2-16 - IT architecture of the TRANSYT resolution model and procedure (Source: Vincent, Mitchell, Robertson. User guide to Transyt Version 8, 1980)

The model

The traffic model of the network calculates a *Performance Index* (PI) given by a weighted average of stops and delays within the system. The PI will then be used by the optimization procedure that will return a new solution, developed in turn again from the traffic model to obtain the new IP value, in an iterative sequence that will stop when it will reach the optimum.

As external inputs, it requires the features of the network (nodes, arcs, lanes, green phases, maneuvers that are permitted, not permitted or in contrast with each other or not), the average flow and the saturation flow of every arc and a given initial assignment.

Then the optimization algorithm determines the timing which reduces the PI to a minimum (local) value, satisfying constraints such as minimum green

time, or other user-specified constraints. So the inputs received are the constraints specified by the user and the IP returned by the traffic model.

Every stretch of road between two intersections forms one or more arcs, and each of them represents a possible maneuver. It is assumed that the inflow is regular and that its division into platoons is caused solely by the traffic lights that disrupt its passing. The magnitude of flows is obtained from historical data and the model does not redirect flows according to the network state.

The model assumes a vertical queue of vehicles; however the *spillback* phenomenon can be represented by introducing a strong disadvantage to achieve a predetermined queue level.

Driving times or traveling speeds must be determined to calculate the platoon dispersion. This may be calculated differently in the case of public transport, to consider also the time required to stop.

Groups of adjacent intersections can be specified to require that their offsets are changed at the same time; this choice is made when a network is very large and it is necessary to treat it as a set of sub-areas.

There is also the possibility to generate controls that give priority to certain vehicles or roads. The program can then be used to establish timings that give priority to public transport or emergency vehicles, without the need to specifically detect this type of vehicles within mixed traffic flows, as occurs in the case of real time systems that require the application of whether fixed or on board detection systems and the interfacing of these with the central system.

The optimization algorithm

The TRANSYT optimization algorithm aims at minimizing the PI. If it were possible to reduce this to 0, the PI would have a planning that allows to each user to find always the green traffic light during his path and to have no queue delays and no stops and then it would mean to get the minimum travel time. Following this philosophy, TRANSYT aims at obtaining "green waves" along several possible routes within the network.

Up to version 8, the optimization was performed by varying only the offsets and the green times of each traffic light. It was later provided with algorithms that can also change the cycle time. The optimization of the sequence of phases has been the last to be introduced using a genetic algorithm.

The cycle time

A common cycle time value is a prerequisite for the coordination of traffic lights on a network. Initially (up to version 8) TRANSYT did not alter this cycle time: it was assumed as given and common for all the traffic lights of the analyzed network. Where it was more convenient within the scope of optimization, the software was still able to recognize that a traffic light performs two cycles within the allotted time.

As the change in cycle time has significant effects on the optimal value of the network, it was recommended to the user to perform an analysis to determine the most appropriate value to impose for the subsequent offsets and green times optimization. Then this heuristic exploration was implemented in the program and managed internally.

Offset optimization

The earlier releases of TRANSYT software calculates the optimum synchronization setting using a hill climbing process. First, the algorithm calculates the PI for a given initial assignment. Subsequently, the program varies the offset of every traffic light of a predetermined amount and corresponding to it, the program calculates the network IP. If this is reduced, the offset is further changed along the same research direction and of the same amount, until it is obtained a minimum value. If the initial move had produced an increase of the IP, the offset is changed in the opposite direction, again until it reaches a minimum value.

Obviously, the optimum setting obtained won't be a global but only a local optimum. To reduce the possibility of obtaining optimum but not very favorable local settings, TRANSYT alternates in subsequent iterations the use of more or less wide research steps. The size of these steps can be specified by the user, but the program provides recommended values by default and in any case they cannot be automatically modified during the progression of the algorithm depending on the results achieved, as occurs in the most popular optimum search algorithms like the gradient method.

Optimization of green times

During the offset optimization, all green times are retained. TRANSYT can optimize the green duration for the different phases, but only changing them individually and trying to reduce the network PI. However at this stage, green times cannot be reduced below the minimum possible value of green.

Phases optimization

The traffic light phases have been one of the last elements introduced within the TRANSYT optimization. In later releases a genetic algorithm is adopted to explore the solution space while avoiding a comprehensive solution. The procedure comes from studies (Hadi, et al., 1993) relating to the interaction between a genetic algorithm and the parallel version of TRANSYT-7F.

Every solution consists of a single cycle time and offsets, green times and sequence of phases for every signalized intersection. In the population of the genetic algorithm, the solution is then encoded by a concatenation of binary substrings corresponding to each one of these parameters. The usual crossover and mutation operations are performed on these substrings for pairs of solutions.

Wallace explores two possible implementations: one in which the genetic algorithm works in conjunction with normal TRANSYT optimizations and one in which the genetic algorithm precedes a final optimization performed by the program.

In the first method, the solutions are composed only of the substrings related to the cycle time and to the sequence of phases. The genetic algorithm then modifies only these two figures, and based on these, the offsets and the green times are automatically determined by TRANSYT for every solution and the associated PI calculated by the traffic model.

In the second method, the string of each solution contains information such as cycle time, sequence of phases and offsets. So the TRANSYT calculates only green times in the optimization and the consequent PI. At the end of the

population evolution performed by the algorithm, when the limit number of generations required is reached, it is carried out a comprehensive optimization (this time including also cycle time and offset) by the TRANSYT on the best four individuals identified. The basic idea is that the genetic algorithm identifies the peaks of the function while the hill climbing is adopted to achieve them exactly.

Among the two methods, only in some cases the second one generates results comparable to the first one, but it has a much lower complexity. A calibration of the genetic algorithm parameters (probability of mutation, crossover, density, etc.) can result in significant improvements in the obtained solutions.

The limits

A weakness noted by several researchers is the way in which TRANSYT manages the queues. It assumes that with queues, vehicles will place along a single front rather than along a linear queue. This ensures that the green propagation is not considered along the queue because with the green traffic light all vehicles would leave simultaneously accumulating at the end of every arc, without taking into account the *spillback* phenomenon. In non-congested traffic conditions, this approximation is acceptable, but when the flows reach the saturation levels, the simulation is not more representative. So with oversaturated flows, TRANSYT is unable to perform significant optimizations.

In addition, the hypothesis that the flow is constant and equal to the average one is not negligible, although the limit of all the controls is fixed time.

The figure below shows the deterministic oversaturation delay model used by Transyt.

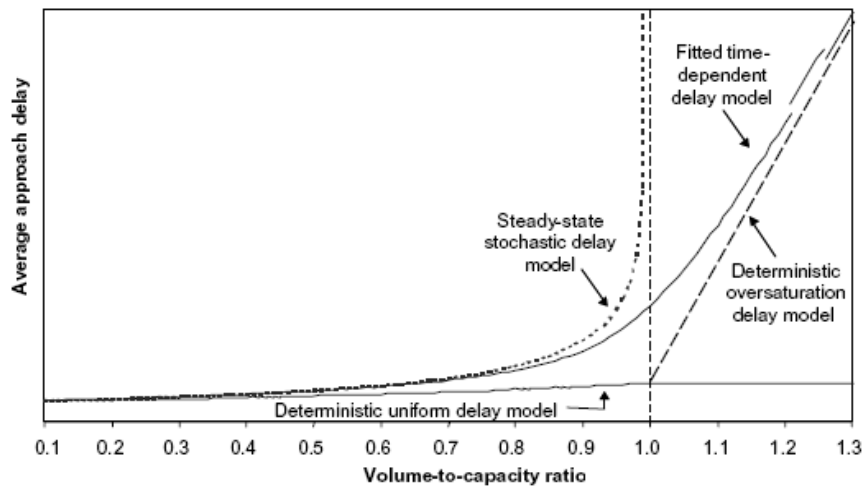


Figure 2-17 - Transyt arc delay model (source: Dion et al., 2004)

Passer IV

PASSER IV was developed by the Texas Transportation Institute in the early 90s and it is based on the MAXBAND program. It is a DOS-based program used to optimize a traffic light network and it is aimed at maximizing the green band.

Obtaining the maximum green band means maximizing the period of time in which a vehicle is potentially able to traverse the given network with the minimum possible number of stops at signalized intersections. To do this, the program determines the optimum cycle times, offsets and phases length. The green duration is instead established in a previous step using the Webster method that formulates the green split in function of saturation grade.

Alternatively, the software can try to minimize the cycle time or provide the optimum planning for different alternative objectives.

It is able to optimize traffic light timings for arteries as well as for networks with closed loops, as the center or industrial districts. Unlike other software can also manage multiple arteries intersect as well as one-way and streets two-way.

The user can make many specifications such as:

- assign a priority order to the arteries;
- set the speed limits, minimum or maximum and seek to use the same optimum speed between consecutive arcs
- set the reference traffic light and direction for choosing all the offsets;
- use the different units for input and output data;
- take a number of different optimum plans.

The model does not simulate the evolution of network traffic and this is one of the major limitations of this software since the only maximization of the band does not guarantee the minimization of total delay on the network however giving good result.

Synchro

Synchro, developed by Trafficware Inc., is a software package capable of modeling and optimizing traffic light timing on a road network.

The goal can be specified to maximize the capacity of intersections or the minimization of total delay. The algorithm is able to determine both the optimum offsets for pre-assigned timing and to establish the optimum cycle and phases.

Synchro is also able to model actuated traffic lights : to do so the software needs additional information from the user to analyze the actuated network. So it can be interfaced with the instrumentation of real-time control to get this data provided by the detectors.

Synchro 3.2 implements in order to calculate the delay, the analytical method HCM (Highway Capacity Manual)-94, Chapter 9 on signalized intersections. Synchro 4 implements the new methods of HCM 1997, which include actuated traffic lights and congested intersections. Version 6 was also introduced the ability to identify the spillback and the capacity of the generated code as shown in the figures below.

The limitations of this software is to use an analytical formula for calculating the delay. This is not able to represent the phenomena which can be calculated with a simulation model

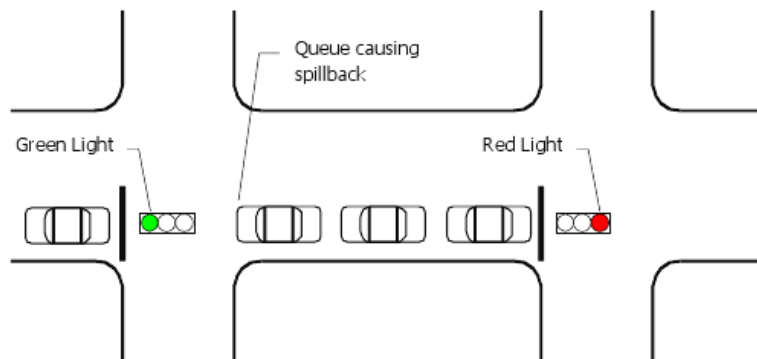


Figure 2-18 - Spillback queue and impossibility for the upstream vehicles to exploit the green (from synchro 6 manual)

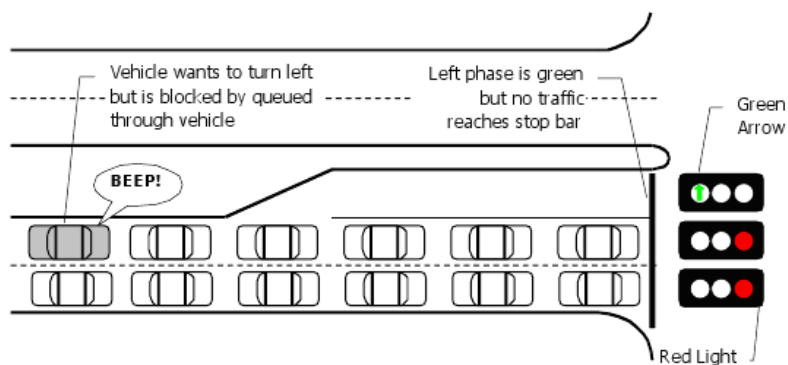


Figure 2-19 - Reduction in capacity effect due to queues (from synchro 6 manual)

SCOOT

The acronym S.C.O.O.T. stands for "Split, Cycle and Offset Optimization Technique" (McDonald, et al., 1991). It has been successfully installed in over 70 cities in Britain and in many cities around the world, including Beijing, Cape Town, Madrid, San Diego and Orlando.

The first version was developed for research purposes in 1975 in Glasgow; in 1979 it was performed the first working installation in the city of Coventry; in 1991, with version 2.4, it is adopted for the first time in the United States. Today it is active in over 170 cities worldwide.

It is a plan generation system based on traffic model updated in real time. In fact its implementation requires a central planning unit connected to several traffic detectors located within the network.

The idea is to continually update the chosen optimum cycle time, green time and offset according to current surveys, generating almost at all times a new plan calculated as the most effective at that time. The time intervals are small enough to ensure a rapid answer and a reliable forecast. It uses an objective function given by the linear combination of delays and number of stops within the network, similar to the TRANSYT function, and SCOOT for this can be seen as a traffic-adaptive version of the latter.

The inputs that are required to the user are for each approach, travel time from the sensor to stop, saturation flow, capacity of arc accumulation, beginning and evacuation time loss.

The occupation values detected by the sensors placed upstream of the strings are used by the central unit to predict the profile of the incoming platoons to the downstream approach, and integrating it numerically, the incoming and outcoming flow, in order to estimate the queue and the intersection delay.

The method

The measures being used are delay, queue length, number of stop and other measures of the degree of network congestion.

The network is divided into regions. Within each region, the flow has homogeneous characteristics. Each region is divided into arcs and nodes and it is measured the degree of saturation of each node by measuring:

$$x=q/s$$

where

- q : flow detected;
- s : saturation flow;

For each region, the critical node will be the one with the highest degree of saturation: the cycle time assigned to all nodes of the same region will be determined by the algorithm optimizer applied to this node.

Optimization of the cycle

Every plan has a duration of 150-300 seconds, after which it is applied the new plan. the transition from one plan to another is always gradual and occurs in small increments: the cycle time per time unit may be altered by a value between 1 and 4 seconds. The algorithm optimizer also has the ability to schedule double cycles, thus increasing its effectiveness.

Optimization of green times

Shortly before activating the new plan, the algorithm evaluates for each intersection whether to increase, decrease or leave unchanged the percentage

of green. In this case, the change is always an amount between 1 and 9 seconds. The goal of each phase is to minimize the maximum degree of saturation in the approaches to the critical node. This module works in coordination with the offset optimization algorithm while both are dependent on optimizer of the cycle time.

Offset optimization

It is targeted at improving the traffic flow, based on estimates of the impact of any possible intervention. The adopted approach is to minimize the performance indices of the nearby intersections. For the evaluation of the traffic flow of every node it also explores the neighboring nodes: the approach is justified by the common cycle time of the nodes. On this base, the planner can impose "green wave" phases.

Control parameters

From the model it is possible to obtain information about the monitoring and the evaluation of performance, such as the degree of saturation, the degree of network congestion, traffic density, maximum flow and more.

Advantages and disadvantages

The key benefits of SCOOT are the ability to optimize remarkable flows, close to the saturation level of the network, heterogeneous flows and to respond to unexpected changes of the flows. It also tends to favor the main arteries, encouraging the flow of traffic along them. The fact that it does not require a priori knowledge of the network on which it must be applied, makes it extremely versatile. Finally, it is flexible and controllable thanks to the large

amount of information returned and to the possibility of introducing specific weights in the definition of performance indices.

A possible problem is rather the extreme dependence on traffic detection systems: the system can make up for these small errors, but if the defects are not repaired, quickly the performance degenerate to those of a plan generation setting. Moreover, the distribution of network in regions means that there are similar non optimum situations along the border among two different regions. The experimentation was performed always and only on urban areas, therefore the management capacity of highways has never been treated.

Public transport

Compared to the original version, some amendments were introduced to model special traffic components such as emergency vehicles and public transport with reserved lane.

The system is able to provide traffic light priority to the above particular traffic components. In SCOOT there are three levels of public transport traffic light priority:

- Equal priority to all buses on each approach;
- Differentiated priority according to compliance with the time schedule, priorities for some and not for others.
 - Buses traveling in time are not entitled to priority;
 - Buses with delay have a moderate priority;
 - High priority for public transport with significant delay.
- "Extra" priority through the jumps of bus phases.

Detection and identification

The logic of the traffic light priority does not depend on the method for identifying the buses. SCOOT allows to use different public transport detection systems, such as:

- AVL (Automatic Vehicle Location) used by different public transport management systems;
- SVD (Selective Vehicle Detectors), or other systems that can reliably identify the buses and their position.

If you require differentiated priority, then the bus detection system should provide the level of priority requested as well as the detection of the vehicle. For a typical system that adjusts the priority according to the delay, or the adherence to the time program, the management system will set the delay and send it to the bus. When the AVL system of a bus determines that the bus arrives to the detection point of traffic signal priority, it will transmit a detection of the bus, the priority level and possibly an identification code of the vehicle. SCOOT then provide the appropriate priority as defined by the traffic engineer in the SCOOT data.

Bus stops

Buses normally need to be detected after each stop on the arc, as in its model SCOOT does not consider the time spent at the bus stop. An accurate localization to detect when the bus leaves the bus stop can be critical in this area.

Where the time spent at the bus stop is predictable, for example, with buses with many doors, pre-paid tickets and all the buses waiting at the bus stop, it

is possible to detect it before the stop and take into account the time spent at the bus stop.

Modeling

The buses are modeled according to the street pattern. Where there are no lanes dedicated to them, the bus are modeled as queuing with other vehicles, allowing them to be preferencing although they may be delayed by other vehicles. Where there is a reserved lane, buses are modeled separately from other vehicles, but they are modeled with the general traffic at the end of the reserved lane if the end is before the stop line.

Recovery

Once the bus has passed through the signals, it is necessary a recovery period to return the existing timing with the normal SCOOT optimization. Four recovery methods are provided for timing recovery, after having extended the green phase, the green phase recall and the phase jump to achieve as soon as possible the desired phase.

Restrictions on the priority

The amount of priority given to buses can be restricted by the degree of saturation of the intersection as modeled by SCOOT. This is controlled by a target of degrees of saturation for the extension and the recalls. Normally the degree of saturation should be defined so that the intersection is not possible to become over-saturated. This means that traffic signal priority will be more effective at intersections that have spare capacity.

Rhodes

The main philosophy of Rhodes (Mirchandani, et al., 2001) is to respond to natural fluctuations in the traffic flow in a "proactive" manner. This means making use of models to predict the evolution of the system by updating them at intervals small enough to ensure an effective response, which is impossible in case of detection at intervals longer of a few seconds.

To handle the traffic (flow forecast), Rhodes uses the PREDICT algorithm. This is structured according to a hierarchy of three levels:

1. Control of intersections;
2. Control of the network flow;
3. Network load.

The system also provides the ability to manage traffic, including rails, emergency and public service vehicles.

Intersection control subsystem

Real-time outflow measurements are acquired, crossing times, turns and forecasts on arrivals at the intersections are made and the consequent formation of queues. Based on this, it is calculated the solution on the duration of the intersection phases. The algorithm determines for each intersection the duration for every phase from a series originally assigned.

The optimization variables are the duration of each phase and its offset within a given time interval and different objectives are allowed, such as stop, delay and queues.

Crossing times, queues and turns are processed in real time and in particular dealt with as follows:

1. Driving times are estimated through measures of beginning and ending times of vehicles represented by compact platoons.
2. The queue throughput is estimated by the arrivals and departures generated by the successive phases of green (QUEUE algorithm).
3. Turns are calculated using a dynamic OD matrix and an approximation to the least squares (TURN algorithm).

Network flow control subsystem

This subsystem works with a lower frequency than the previous one, at intervals of approximately 200-300 seconds. It calculates the platoons propagation within a sub-network and it optimizes them using the REALBAND algorithm that is responsible for resolving any conflicts that are generated between platoons. Conflict means the contemporary arrival of two or more platoons at an intersection that would require the simultaneous activation of phases in contrast. To solve the conflicts, it generates a decision binary tree, it determines the optimal solution based on the performance values relating to every possible choice and it sends the coordination constraints to the intersections optimization subsystem.

Network load subsystem

It Estimates the transportation demand (dynamic OD matrix) in time intervals of approximately one hour. This estimate is used to determine the likely size of the platoons within the network in the next period.

Utopia

UTOPIA (Mauro, et al., 1989) is the name for Urban Traffic Optimization by Integrated Automation. It is an adaptive-traffic control system that minimizes the total travel time of vehicles on the network giving priority to public transport.

UTOPIA is composed of two levels, the central level and local level, which operate independently but coordinated with each other.

The central level monitors and controls the network as a whole. It makes forecasts and controls in the medium and long term and determines the coordination times for the network. The local level instead assumes the control of individual intersections by coordinating the decisions both with those of adjacent intersections and with those of the central level. It consists of a dense network of local controllers (SPOT) interconnected and involved in the management of an intersection. Each spot is also responsible for the supervision of sensors, devices and components connected to it and shall report any failure to the upper layer.

Main components

- Traffic control system (central level): it determines the optimal area strategy, it manages the coordination between the planning elements and can display a graphical representation of network and of the traffic on it.
- Local controllers (local level): they develop the optimal intersection strategy on the basis of the strategy of the area established by central

level, local surveys and information exchanged with neighboring controllers.

- Communication network: it manages the information flow between local controllers and between them and the local level. Using a routing mechanism, it is capable of making up for any network failures.

Approach

The functional scheme used is that of closed loop control systems: a functional unit is responsible for estimating the system state on the basis of surveys, while another unit determines the optimal control strategy based on the estimate performed by the forecasting unit. The decomposition of the complex optimization problem of a wide network in simpler sub-problems related to each other leads to obtaining theoretically almost optimum strategies.

The minimized cost function is a function that has as main components delay and the number of stop of private and public transport. The data differentiation between the two categories allows to help the last one giving them priority during the traffic light planning. The presence of a public transport is detected by special sensors and turns into a request to the satisfaction of which is favored by the algorithm. The system update time is about few minutes for the central level while the local level updates every 3 seconds and the horizon forecasted by the solution is 120 seconds (a technique known as "rolling horizon"). This determines the optimum sequence of traffic light phases on the basis of the criteria imposed by the top level, the traffic measures detected locally and adjacent controllers information.

Traffic Urban Control (TUC)

The acronym TUC stands for Traffic Responsive Urban Control (Diakaki, et al., 2001). The TUC was initially developed as part of a traffic control system within the TABASCO European project, and then it was extrapolated and used individually.

It is based on a modeling approach proposed in 1963 by Gazis and Potts defined as "store-and forward", that thanks to the linear quadratic formulation of the problem allows a highly efficient optimization and solution methods with polynomial complexity that are also suitable for networks of significant size.

The model

The representation is one of the most classical dynamic models of discrete time flow, with time intervals equal to the cycle calculated at each iteration.

The arc continuity equation is given by:

$$x_z(k+1) = x_z(k) + T[q_z(k) - s_z(k) + d_z(k) - u_z(k)] = 0$$

where:

- T is the dimension of the time interval;
- K is the generic time interval;
- x_z is the number of vehicles on the z arc;
- q_z is the inflow in the z arc, given by the linear combination of outgoing flows from the arcs belonging to the backward star in z for the relating turn percentage (in z);

- u_z is the outflow from the z arc;
- d_z is the flow is generated inside the z arc;
- s_z Is the flow out along the z arc, percentage of q_z .

Assuming that there is a combination of control values that ensure stationarity of the number of vehicles on the arcs, it can be determined by minimizing a suitable objective function derived from the wording. The solution leads to the optimum control law.

The algorithm

The first version of the TUC worked only on the green phases; then it was improved allowing to monitor in real time also cycle times and offsets. This more comprehensive version has been implemented in the field and evaluated in three European cities: Southampton, in England, Munich, in Germany, and Chania, in Greece.

The developed modules are:

- Control of the green phases: a recursive formula of negative feedback control systems in discrete time aimed at minimizing the number of vehicles on the signalized approaches, cause of the main network delay.
- Cycle time control: an algorithm based on feedback that changes the cycle time in order to adapt it to the current saturation level found in the network.

- Offset control: a feedback control law changes the offset given for the intersections along the main streets in order to obtain the "green waves", but taking also into account the possible presence of queues.
- Priority of public transport: local plan modifications to facilitate the flow of collective transport vehicles.

All modules can be used individually or in combination using as constants the values normally set by other modules.

Cycle time

The cycle time is increased to decrease the importance of time loss until the green.

The algorithm is divided into three points:

1. It calculates the average maximum load on the network;
2. The cycle time is calculated from a negative feedback control law:

$$C(k) = C_N + K_c (\sigma(k) - \sigma_N)$$

where C_N is the generic nominal cycle time of the network, σ_N is the nominal average load and K_c is a control parameter that regulates the intensity of the feedback control;

3. At sufficiently low saturated intersections, it is assigned a cycle time that is twice in comparison with the one calculated.

The cycle time obtained is then used by the algorithm to calculate the green phases.

Green phases

The green times are updated using the following formula:

$$g(k) = g_N + L \cdot x(k), k = 0, 1, \dots$$

where g_N is the vector of green nominal values, that can be obtained for example by optimizing with TRANSYT, L is the control matrix resulting from the stationary Riccati equation and x is the number of vehicles coming to the intersection. In practice, the greater the number of vehicles over a z arc, the lower the green attributed to arcs leading to z .

To make simpler but as effective the algorithm, once the elements values of the L matrix are determined as those with inferior effect on the control (typically corresponding to arcs away from each other), they are placed equal to 0.

The calculation of L is one of the most expensive aspects of the method, but it is performed only a priori before the final application, and then it remains unchanged for the time when the network is not changed.

The offsets

The idea is to encourage the traffic outflow along the main arteries through "green wave" phenomenons.

In the case of one-way arteries, the offset are given (eg by a maximum bandwidth algorithm).

In the case of two-way arteries, each direction generates an offset that would support the green band, and then it is taken an offset given from a weighted

average of the two possible offsets. The weights can be preset or may be determined in real time based on direction at the moment with more load.

In the case of arteries that pass through, TUC prioritizes according to an order preset by the user.

To consider also the presence of queue at the intersections along the arteries, TUC studies the flow behavior through two waves moving in the opposite direction:

1. The wave caused by the green traffic light from which the vehicle come, that moves in the opposite direction to the direction;
2. The wave caused by the green traffic light towards which the vehicles are directed, which moves in the concordant direction to their direction.

Both waves have different ideal offset, that appropriately weighed give the final choice of the algorithm. To implement it, appropriate changes can be made to the cycle time of the target traffic light cycle.

The priority of public transport

TUC can manage the priority of collective transport in two different ways:

1. Through an appropriate decision-maker that would alter locally the decisions otherwise taken by the model. It is rarely relevant in the case of networks where the passages of public transport are particularly frequent, because it could affect the original effectiveness of the model.

2. By giving appropriate weights to the green phases provided by the model for the arcs, where such means are present. This method actually encourages *the paths* taken by public transport lines, not being able to detect the direct presence of a method.

TUC trials in different and even very extreme scenarios, have shown improvements in considerable network performance indices: waiting time at the source nodes reduced by 100% (the network is able to absorb all the demand), driving time reduced between 20% and 80%, total time spent reduced between 30% and 90% and fuel consumption reduced between 25% and 85%, depending on the considered scenario.

It is necessary to clarify the following points:

- The control intervals cannot be lower than the traffic lights cycle time, then the decisions cannot be made more frequently than the higher cycle period on the network.
- Offset effects for adjacent intersections are not considered.
- It is impossible to take into account either of the effects of the turn percentages on the traffic or of the flows approach at the saturation levels.
- Due to the nature of the L matrix, the process does not lend itself to changes or expansions of the network under consideration.

Nevertheless, the control system is particularly effective for large networks thanks to its robustness against possible inaccuracies in the measurements, to its generality, which makes unnecessary any form of initial calibration, to the

scarcity of detection elements required (one for arc) and to the very low operational complexity.

3 Delay calculation model

The aim of this thesis was to define a delay calculation model and to develop some resolution procedures, that are efficient from the calculation point of view, and effective in terms of the quality of the solution.

Because of the non convexity of the problem and of the difficulty to represent it through an exact formulation, the approach to the solution procedure was to define a model for the calculation of delay and to use it as a fitness function to some heuristic algorithms.

One of the main features that a fitness function must have to be used within heuristic algorithms is the ability to be evaluated quickly. In fact, these algorithms have the particularity of having to process a large number of possible solutions with the advantage, however, to adapt successfully to various problems also difficult to handle.

After the study of the state of the art presented in previous chapters, research project developed together with Prof. Gaetano Fusco and Eng. Chiara Colombaroni gave rise to this work. The algorithm proposed by Fusco and Papola and presented in paragraph 2.3.1 was used as a starting point for developing the delay.

From this model, it was tried to remove some unsuitable assumptions while maintaining the delay classification and calculation rules which gave it a linear computational complexity compared to the number of platoons analyzed. In the definition of the new model, the objectives were:

1. Extending Papola and Fusco model to simulate the platoons progression within an artery with any traffic light configuration.
2. Allowing the model to work with dynamic demand and in condition of oversaturation.
3. Analyzing the progression of platoons in both directions of an artery.
4. Simulating the queuing of vehicles on the transversal approaches.
5. Using the model to calculate the delay of public transport.
6. Allowing the simulation of preferencing systems of public transport.
7. Simulating the spillback condition and its impact in the delay calculation.
8. Using the extended model as fitness function for the optimization algorithms.

3.1 The out flow models

In the representation in platoons of vehicles and of their progression, it is the representation of the vehicular outflow that is simplified. In order to better understand some applied choices, it is necessary to introduce the outflow from the formal point of view. To analyze this quantity, it is convenient to use a functional relation between the average flow value and other traffic characteristics.

The relation between the three variables, speed v , density k and flow q , is called *vehicular outflow model* and referring to these quantities the following values are defined:

- q_m : maximum value of vehicular flow;
- v_f : free or null flow speed;
- v_m : speed at which the flow is maximum;
- k_j : critical density or saturation flow density;
- k_m : density at which the flow is maximum.

From a series of empirical observations, it is highlighted how with the increasing of vehicular density on a lane or, more generally, on a road, drivers naturally tend to reduce their speed. The two quantities (v, k) are thus not independent, but closely related by some functional expression. Once that speed and density are known, from the equation of state it is possible to derive $q = k \cdot v$ the vehicular flow.

Here are the main models that link together the speed-density variables:

- Linear model (Greenshield);
- Logarithmic model (Greenberg).

In a survey on traffic characteristics, Greenshield proposed a linear relation between speed and density, that can be expressed as:

$$v = v_f \cdot \left(1 - \frac{k}{k_j} \right)$$

The model is simple to use and it was found a good correspondence between the actual and theoretical values calculated for a wide range of variables,

although, unfortunately, not always the actual events are linked by linear relations in the whole existing domain for a variable.

Greenberg has instead developed a speed-density model based on what follows:

$$v = v_m \cdot \ln\left(\frac{k_j}{k}\right)$$

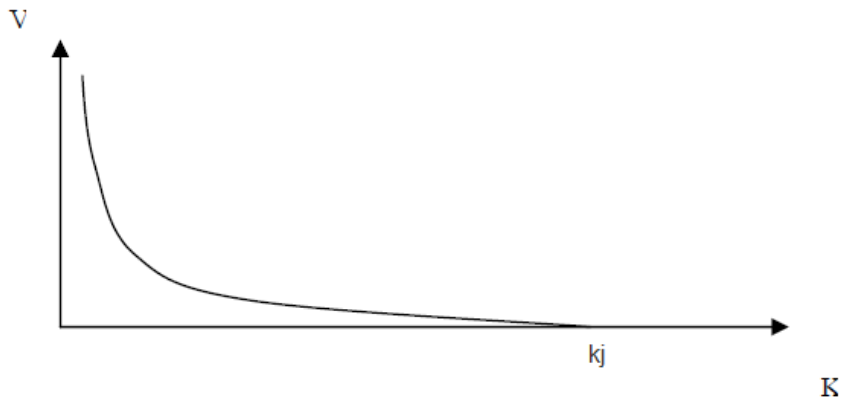


Figure 3-1 - Greenberg law

This model shows a good correspondence to the real values for flows close to congestion situation, but it results less accurate for low values of vehicle density, as can be verified for $k \rightarrow 0$.

To overcome this problem, Underwood proposed a different formalization of the model, as shown below:

$$v = v_f \cdot e^{-\left(\frac{k}{k_m}\right)}$$

The convenience of this formalization is evident if it is considered that for very high densities it does not provide zero speed values, and it does not give too high speed values in the case of very low density.

In general, there is not a single model that is effective in toto, for example, that the Greenshield model is easier to use and it adapts to constant flow conditions providing consistent results to the real values in a quite large existence field.

But if it is necessary to have a more precise evaluation of speeds at high vehicular density values, it may be convenient to use the Greenberg model that, at the expense of a greater formal complexity, is more accurate in these cases.

The relation between flow and density (q-k curve) is often called *fundamental diagram of traffic*. This diagram assumes different patterns depending on the functional form that links the speed and density variables. In fact, with the linear v-k model (Greenshield), it is obtained a relation of parabolic type between flow and density.

$$q = v \cdot k = v_f \cdot \left(k - \frac{k^2}{k_j} \right)$$

This can be projected on different planes as in Figure 3-2.

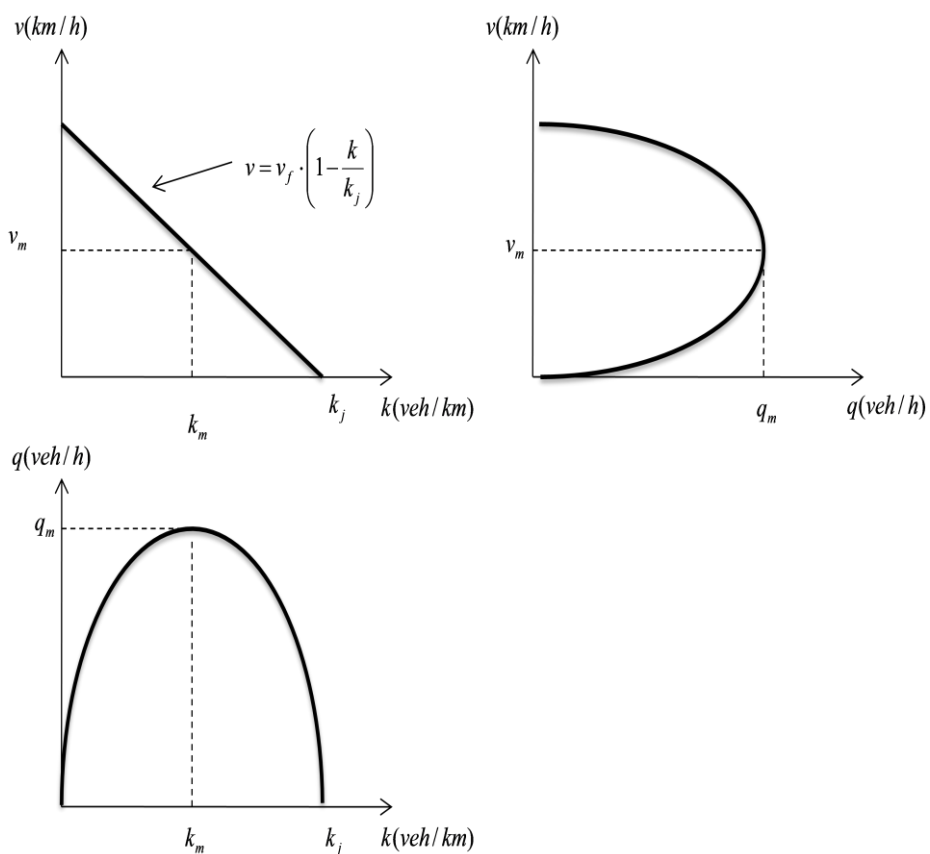


Figure 3-2 - Greenshield linear model

On the contrary, the flow-density Greenberg's function is a logarithmic function multiplied by a linear function.

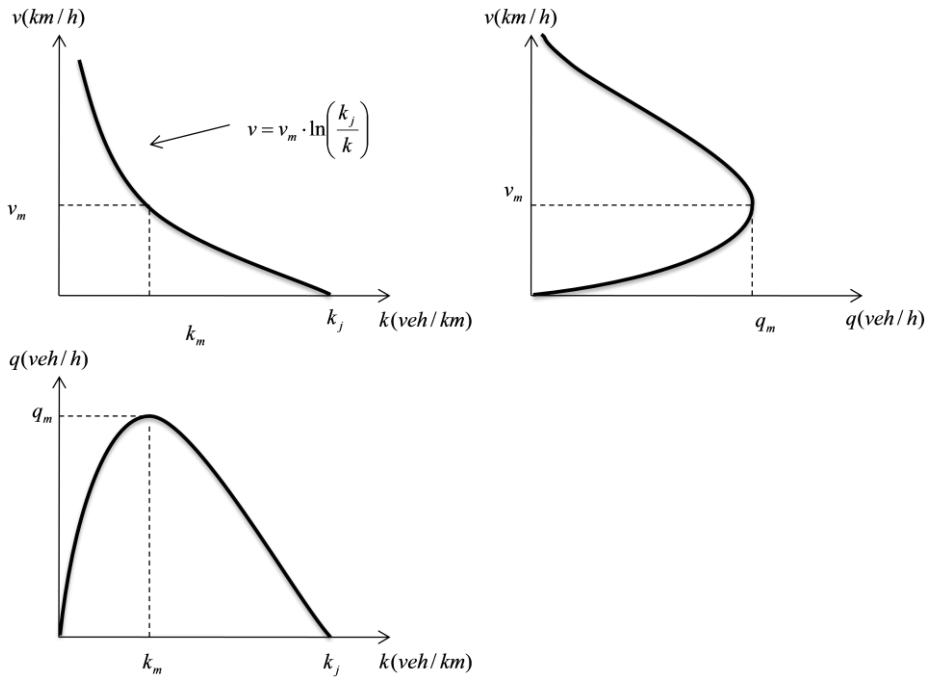


Figure 3-3 - Greenberg logarithmic model

For the platoons model, the vehicles can be standing or can travel at free speed v_f . The flow during their progression is equal to the road saturation flow and the density is equal to the critical density. In the platoons model, fundamental diagrams are then represented in discrete form by only two state points for each representation space:

- $(0,0), (s, k_m)$ in the flow-density diagram;
- $(0,0), (v_f, k_m)$ in the speed-density diagram;
- $(0,0), (v_f, s)$ in the speed-flow diagram;

3.2 Formulation of the minimum travel time problem

Analyzing a road artery, the problem of minimum travel time can be formally expressed as follows:

$$\min t(\bar{C}, \bar{g}, \bar{\mathcal{G}}, \bar{L}, \bar{s}, X, Q)$$

subject to:

$$0 \leq \mathcal{G}_i < 1$$

$$\max \{C_{\min}\} \leq C_i \leq C_{\max}$$

$$C_{\min,i} = L_i / (1 - \max \{y_{i,h}\} - \max \{y_{i,k}\})$$

$$\max \{y_{i,h}\} \leq g_i \leq 1 - L_i - \max \{y_{i,k}\}$$

where:

- \bar{C} : traffic light cycles set for each i intersection;
- $\bar{g}, \bar{\mathcal{G}}$: green splits and offsets set for each phase of traffic light plans of each intersection;
- \bar{L} : time loss set for each i intersection;
- \bar{s} : saturation flows set for each arc;
- X : urban artery geometry;
- Q : demand level;

- $y_{i,h}, y_{i,k}$: saturation degree of the h approach along the artery and of the k transversal approach of the i node;

That is, as expected, the travel time of a road section is closely linked to the geometry of the road itself and to the configuration of traffic light plans.

Among the variables that have been used to express the system, only three are basically configured by the designer: cycles, splits and offsets. Saturation flow and geometry are studied and designed in earlier phases to those of the configuration of traffic light plans and often they cannot be changed a posteriori. In addition, to maximize the intersection performance, time loss are already designed to be both minimum and ensure the safety levels required.

Although not formally expressed, the desired configuration, in an optimization procedure, must obviously be periodic and non-rigid, that is it has to be implemented for periods of variable duration and it has to ensure good service levels also if, within certain ranges, the vehicular flow changes.

The delay caused by a signalized intersection is strictly defined as the difference in the road section travel time in the presence of traffic lights compared to the travel time of the same section if a vehicle could travel along a trajectory at constant speed v .

Assuming that on each arc vehicles can reach speed of synchronization v , even if they are stopped at the upstream node, it follows that upstream and downstream of each node there is at least a section with stationary traffic conditions, in which all vehicles have speed v and are alternated with the amount $1/q(v)$ where $q(v)$ represents the flow at speed v . Therefore, all

vehicles that are stopped experience a similar delay, denoted with \overline{AB} in Figure 3-4.

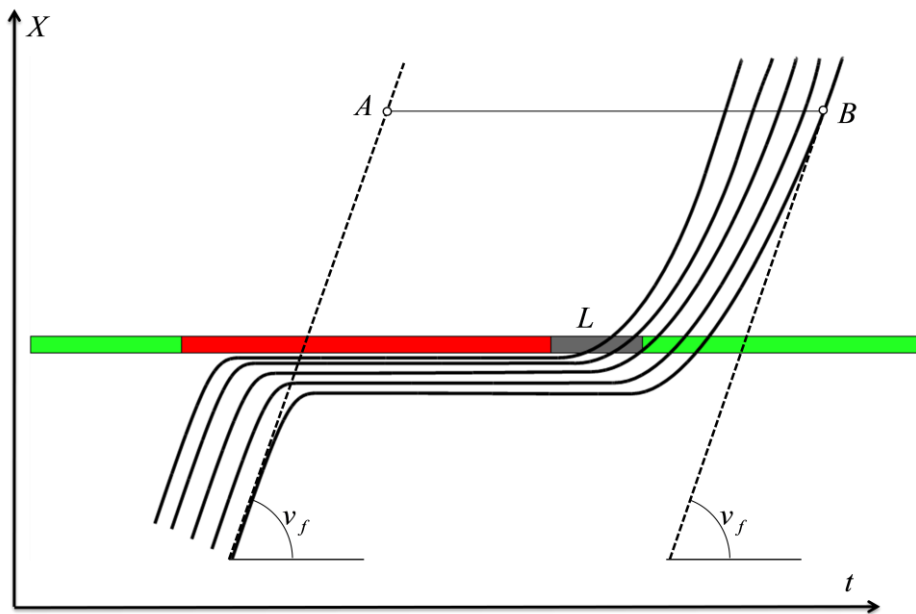


Figure 3-4 - Delay of a vehicular platoon at a synchronized artery node

Compared to traditional delay formulations which consider only the upstream node delay, as occurs in Transyt, this definition is rigorous (Newell, 1989), because it includes both the upstream and the downstream node delay and, as it will be discussed later in this paragraph, it has the additional advantage of providing linear delay functions with the flows, as the average delay is equal for all stopped vehicles.

To take advantage of the possibility to calculate the delay in analytical form, vehicles must be represented through the platoons. Under this hypothesis, vehicles belonging to the same platoon have uniform speeds and in addition the change of the state of motion occurs by assuming an infinite acceleration/ deceleration that allows vehicles to switch from state of stop to a uniform motion with speed v . Similarly, the density of vehicles within the platoon undergoes instantaneous changes switching from the queuing density or maximum density to k_j at critical density $k_c = q(v_f)$ (that is, the density corresponding the maximum flow) during the progression. schematic representation of the progression trend of a generic platoon is shown in Figure 3-5.

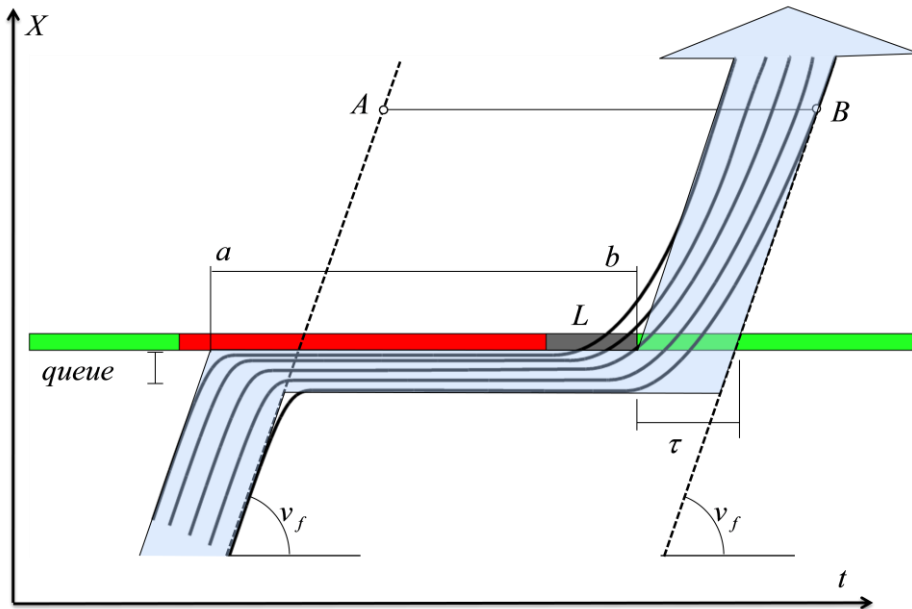


Figure 3-5 - Representation of the progression of a platoon

As it is possible to see in this representation, the delay remains unchanged compared to the Newell (1989) formulation, and the problem can be simplified to a geometrical problem. In fact, projecting \overline{AB} on the abscissa of the intersection, the average delay per vehicle, it can be calculated as the difference in time between the stop instant of the first vehicle at the intersection and the departure instant, that is the distance \overline{ab} .

With this representation, it is easy to calculate the number nq of delayed vehicles, the queue generated by vehicles and the time interval τ necessary to clear the queue.

In particular, their formulation changes slightly depending on the instant of arrival of the first and of the last vehicle at the intersection and on the state of traffic signals lights. As shown in the Papola and Fusco algorithm, platoons are then classified into three types described below.

Type A: totally delayed or fore-delayed platoons

These types of platoons are generated when the first vehicle is stopped in the progression along the artery by the red traffic light or by the queue generated by another stopped platoon. In this case, all vehicles in the platoon are forced to join the queue of the first one according to the laws of the kinematic wave. In this condition, all vehicles are stopped and uniformly delayed of a constant amount. The different quantities can be calculated as follows:

$$delay = \overline{ab} = t_{er} + L + \tau_{-1} - t_1$$

$$nq = n$$

$$queue = nq \cdot k_j$$

$$\tau = \frac{nq}{k_j}$$

where in addition to the nomenclature already established, these variables are defined:

- t_{er} : the instant in which red ends;
- τ_{-1} : the time required to clear the queue already in the intersection;
- t_1 : the instant in which the first vehicle arrives;
- nq : the number of vehicles delayed;
- n : the number of vehicles that form the platoon;

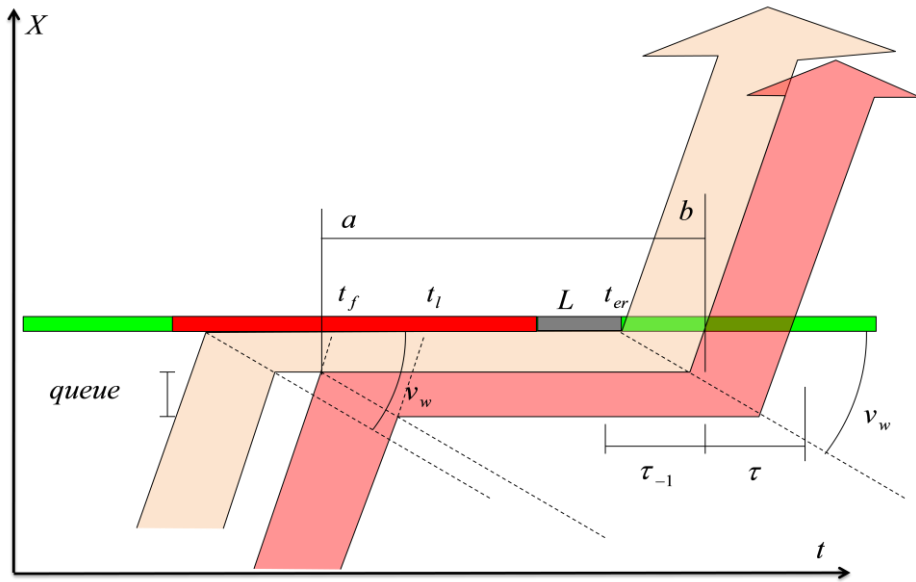


Figure 3-6 - Example of platoon of type A

Type B: partially delayed or hind-delayed platoons

In the case of platoons of type B, only a part of vehicles is delayed because the traffic light switches to red during the platoon transit. In this case, all the delayed vehicles must wait for the completion of the whole red phase.

Within a so classified platoon, the quantities are:

$$delay = \overline{ab} = r \cdot C$$

$$nq = (t_l - a) \cdot s$$

$$queue = nq \cdot k_j$$

$$\tau = \frac{nq}{k_j}$$

where in addition to the nomenclature already established, these variables are defined:

- r : the effective red split. The L time loss is already included in the effective red;
- t_l : the instant in which the last vehicle arrives at the intersection if there is not queue;
- C : the traffic light cycle time.

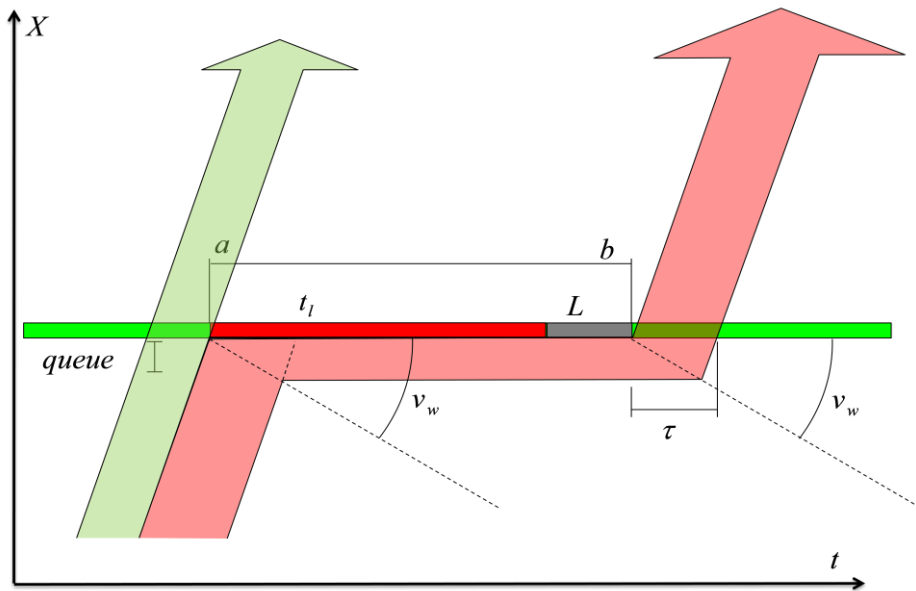


Figure 3-7 - Example of platoon of type B

Type C: no-delayed platoons

The platoons of type C are those who can cross the intersection without being slowed down. This type of platoon does not accumulate delay and does not create queue. Then it is possible to calculate the values as:

$$delay = 0$$

$$nq = 0$$

$$queue = 0$$

$$\tau = 0$$

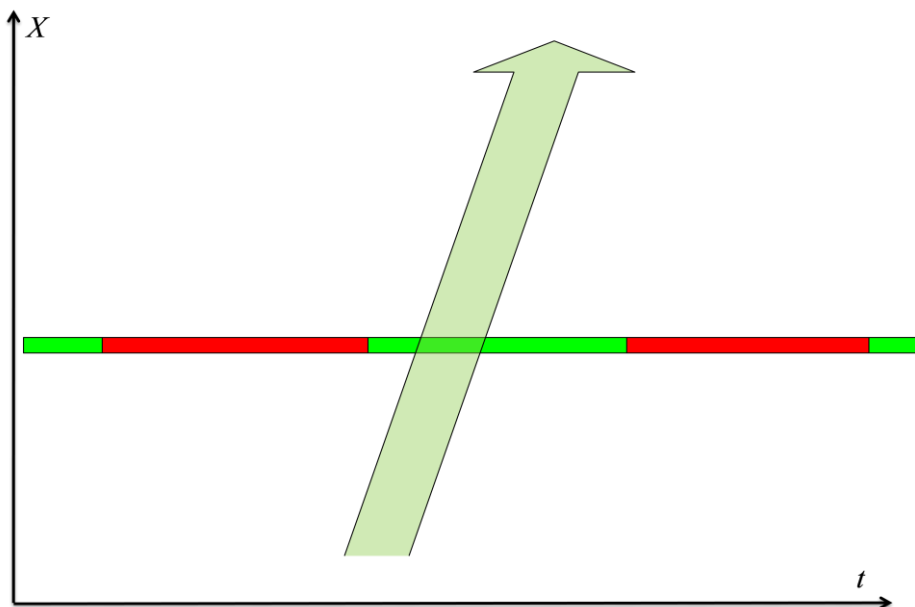


Figure 3-8 - Example of platoon of type C

3.3 Kinematic wave

Of particular interest in the study of vehicular outflow is the behavior of the kinematic wave in the traffic flow, that has a strong impact on the phenomenon of the vehicle queuing and then on the classification methods of platoons.

Given any interruption in the progress of a vehicle along a road, this would result in a queue that will propagate with a certain speed and that can be clearly observed during daily movements allowing to note the progressive lighting of the vehicle stop lights.

Obviously, once the vehicle restarts, the queue will gradually be absorbed over time. These propagations of changes in density and flow within the vehicular flow are called kinematic waves.

Gerlough and Huber presented the following analysis on the kinematic wave phenomenon: two different vehicle density k_1 and k_2 move along an artery road and are separated by a fictitious section S, henceforth called wave front, which travels at a speed v_w . The speed v_w is considered positive if the section moves in the x direction (Figure 3-9).

Defining:

- v_A : average speed of vehicles in the region A;
- v_B : average speed of vehicles in the region B;

- $v_{rA} = (v_A - v_w)$: average speed of vehicles in the region A compared to the S mobile front;
- $v_{rB} = (v_B - v_w)$: average speed of vehicles in the region B compared to the S mobile front.

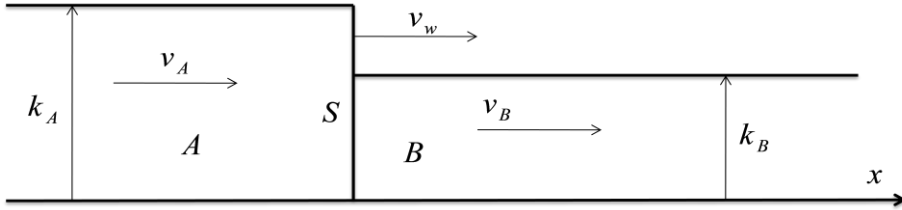


Figure 3-9 - Wave front

In a t time, both v_A and v_B gain compared to S a distance equal to $v_{rA} \cdot t$ and $v_{rB} \cdot t$ and the number of vehicles N crossing the dividing line S , from region A toward region B, will be equal to:

$$N = v_{rA} \cdot t \cdot k_A = v_{rB} \cdot t \cdot k_B$$

The above expression is nothing but a continuity equation which ensures that vehicles leaving the region A are as many of them enter the region B. Expliciting the expressions of the relative speeds:

$$(v_A - v_w)k_A = (v_B - v_w)k_B$$

The last expression can be written as:

$$v_B \cdot k_B - v_A \cdot k_A = v_w(k_B - k_A)$$

and using the equation of state of the vehicular outflow $q = k \cdot v$:

$$v_w = \frac{q_B - q_A}{k_B - k_A}$$

The alternation between red and green traffic light phases leads to the formation of multiple kinematic wave within the transit vehicular flow. Kinematic waves that are created with a traffic light are generally classified in two categories: stop and start waves.

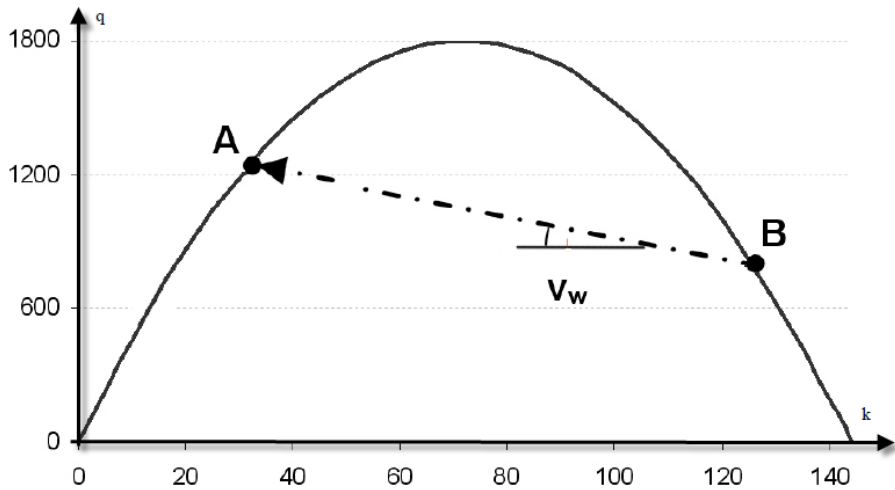


Figure 3-10 - Representation of kinematic wave in the flow-density diagram

3.3.1 Stop wave

A vehicular flow q_1 traveling at a v_1 speed and characterized by a k_1 density can be interrupted by a red traffic light. Corresponding to it, the traffic flow assumes a density equal to the critical density k_j and a flow and a speed

equal to zero, as the vehicles begin to queue up. In this way, it is possible to note the creation of the S wave front between the undisturbed flow and the vehicles that queue lined up at the traffic light. This front, which is representative of the kinematic wave, moves upstream compared the x direction with speed:

$$v_w = \frac{0 - q_1}{k_j - k_1}$$

If the red lasts a t time, the length of the queue formed at the traffic light can also be calculated as $v_w \cdot t$. In the case of platoons representation used in this thesis, the formula can be expressed as:

$$v_w = -\frac{s}{k_j - k_1}$$

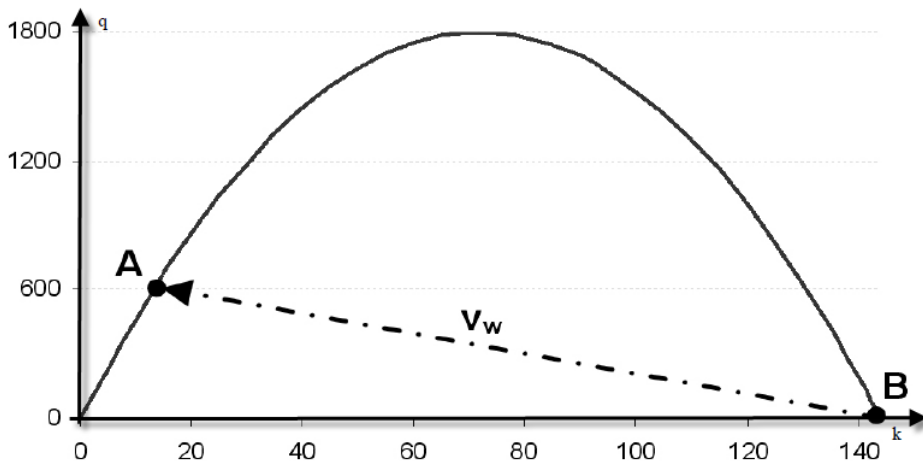


Figure 3-11 - Stop wave

3.3.2 Start wave

The vehicles stopped at traffic light accumulate during the red phase with a density equal to the critical density, and during the red time they form a queue of L length. If at the t_0 instant the traffic light immediately gives the green signal: vehicles accumulated near the traffic light will be faced with a potential maximum flow situation (because the downstream infrastructure is empty), which corresponds to the C point in the flow-density diagram in Figure 3-12.

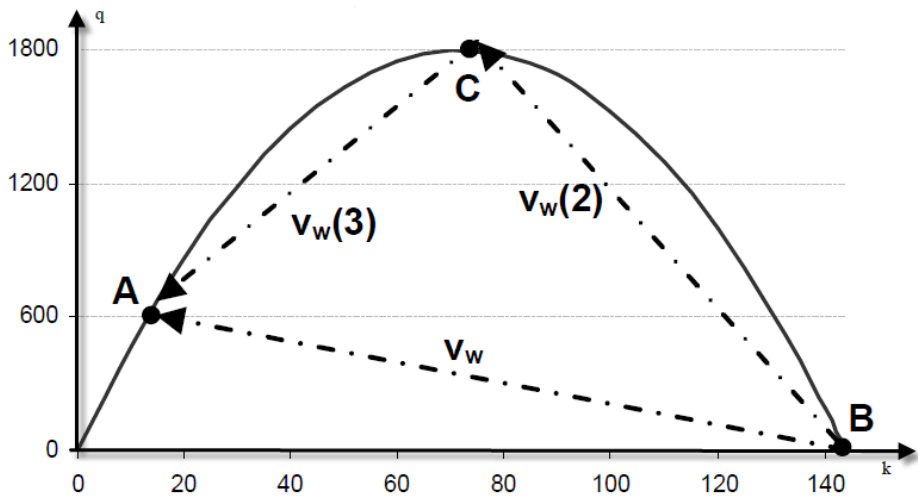


Figure 3-12 - Start wave

Among the B and C states it is generated a second wave that travels upstream with a speed:

$$v_w(2) = \frac{(q_C - q_B)}{(k_C - k_B)}$$

Meanwhile, the kinematic wave formed as a result of the presence of the two states (A and B) continues to move back with its speed v_w . The second wave will reach the first one in the t instant (calculated from t_0 = beginning of green) derived from the following relation:

$$L + v_w \cdot t = v_w(2) \cdot t$$

Then in "t" the queue is completely reabsorbed. When the second wave reaches the first one, a third wave forms between the C and the A state that moves with direct $v_w(3)$ speed towards downstream and that allows the reforming of the initial undisturbed state (A). During the departure of vehicles, close to the wave that is generated between the B and the C state which moves upstream, a second wave moves downstream at the same time. This highlights the change within the vehicular flow between the C state and the condition of zero flow and density and it will cause the formation of a wave with speed:

$$v_{oc} = \frac{(0 - q_C)}{(0 - k_C)}$$

The phenomenon of the restart wave is greatly simplified in the platoon model in which it is assumed an instantaneous acceleration in the restart phase and thus the phenomenon of forward wave propagation between the B and the C states is ignored. Overall, it is assumed the instantaneous transition between zero speed and critical density to free speed and relative density, and the formula to calculate the kinematic propagation speed can be written as:

$$v_w = \frac{q_1 - 0}{k_1 - k_j} = \frac{0 - q_1}{k_j - k_1} = -\frac{s}{k_j - k_1}$$

and then it is identical to the speed of the stop wave.

3.4 The simulation model

The model has been developed specifically to assess synchronization strategies along signalized arteries. The simulation logic is depicted in the flow chart in Figure 3-13.

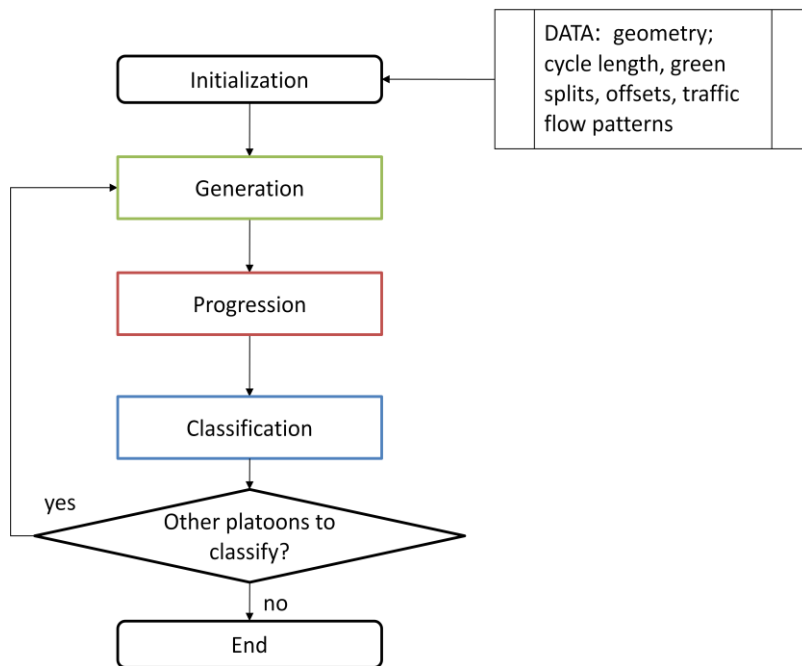


Figure 3-13 - Delay model flow chart

Simulation models can be classified as follows:

- *Deterministic simulation.* The evolution over time of the built model is uniquely determined by its characteristics and initial conditions.
- *Stochastic simulation.* In the model there are aleatory variables.
- *Continuous simulation.* The value of the involved variables varies continuously over time.
- *Discrete simulation.* The state of the studied system, and therefore the value of the relative variables, changes in well-defined time intervals.

Within this classification, the developed model can be described as deterministic and discrete. In fact, although many of the variables used in the model are continuous, they are strictly dependent on discrete state variables.

The simulation was carried out by defining three basic laws:

- law of generation;
- law of progression;
- law of classification.

Each of these laws, suitably iterated, provides a simulation in which the final result is the calculation of total delay.

In contrast to other traffic simulation models, in this model does not arise the possibility of changing the path selection by the user, as for Dynameq, Dynasmart and Cell Transmission Model. In fact the scope is the evaluation of traffic light synchronization along an artery that does not have therefore alternative paths.

3.4.1 Law of generation

The simulation model implements internally a list of platoons that have to be classified (*list_platoons*). The law of generation has the task to populate this list by entering the platoons as the algorithm proceeds. In particular, the phases in which the platoons are entered in the list are two and distinct:

- Initial generation phase
- Recombination phase

The model developed the opportunity to work with a dynamic demand with a trend that is represented by piecewise linear curve and through the comprehensive demand value. In this way it is possible to easily model various demand distribution assumptions.

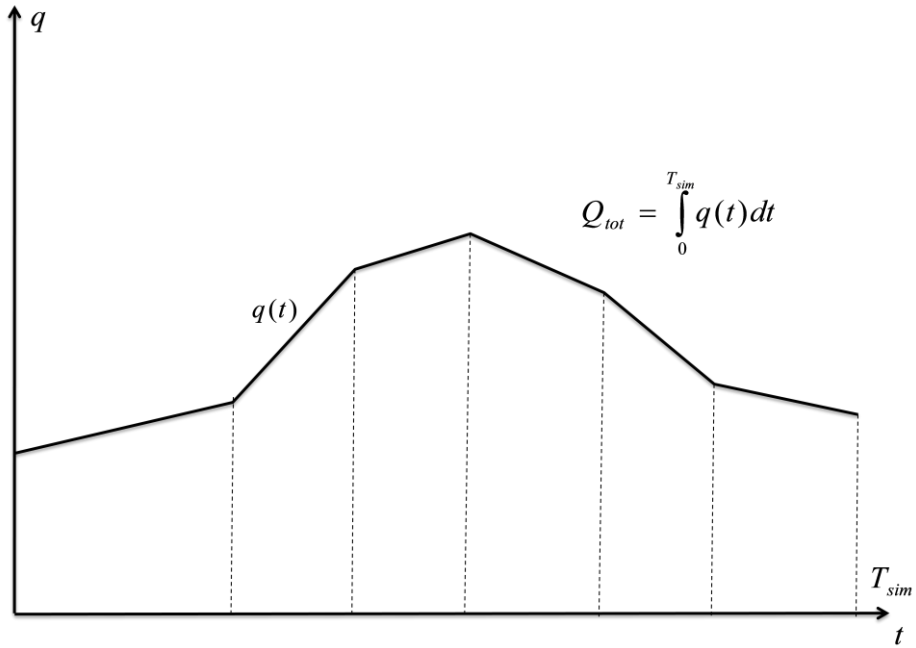


Figure 3-14 - Example of demand curve

During the initial generation phase, procedure analyzes the demand and creates all the platoons that will reach the first node in the whole simulation period. These platoons are enter in the *list_platoons* list waiting to be classified. Size and instants of arrival of platoons are chosen deterministically. One of the input parameters of the procedure is the number of platoons that will be generate during a traffic light cycle of the first node. Based on this quantity, it is possible to define the instants of arrival of each platoon in which the number of vehicles is defined by the timing of the demand curve. A similar argument is made for both access nodes and beginning and end artery and for all transversal accesses. After the first generation phase, in the model were created all the platoons that will enter the artery during the whole

simulations period. For each platoon, during the creation it is calculated the instant of arrival at the node in the absence of congestion. The difference between this instant and the instant of the actual restart from the node will represent the delay accumulated by the platoon in this intersection according to the model presented in section 3.2.

The law of generation also the task of creating new platoons recombining vehicles of different platoons that are temporally contiguous.

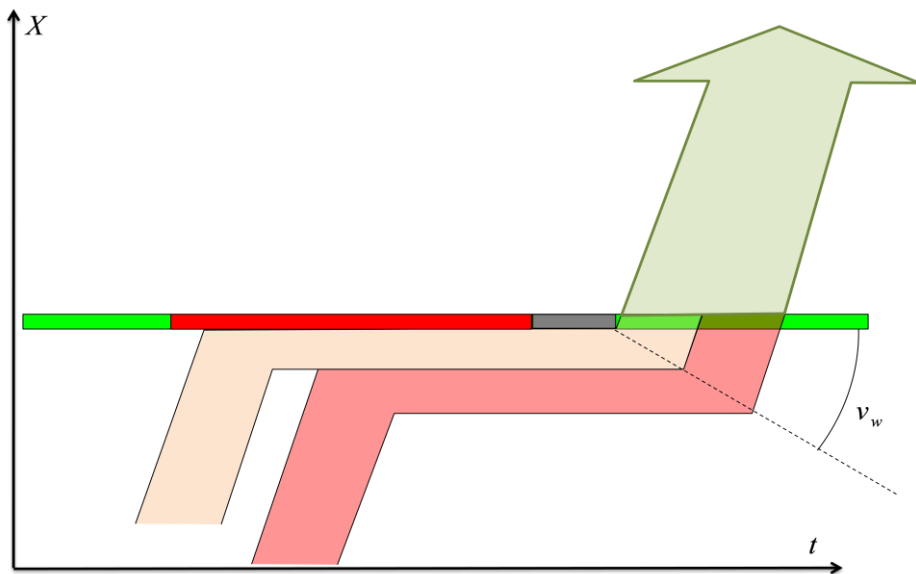


Figure 3-15 - Recombination of platoons

In fact when multiple platoons are delayed at the same intersection and on the same cycle, all vehicles of the platoons will form a single queue. When the traffic light changes to green, with intervals that follow the kinematics wave law, all vehicles in the queue restart from the intersection and behave as if they belonged to the same platoon. Based on this vehicle behavior, the restart

law defined for this model generates new platoons unifying vehicles belonging to more platoons as shown in Figure 3-15.

During the initialization phase, the model stores for each manoeuvre the percentage of vehicles that perform the manoeuvre and through this information and it is able to estimate how many vehicles proceed in the artery and therefore how many vehicles will join the platoon just generated.

The compaction properties of platoons is fundamental as it has a strong impact on performance. The realized model presents a linear complexity compared to the number of platoons to calculate and in Figure 3-16 is possible to see that the number of platoons simulated by applying the compaction property is significantly lower than the number of platoons simulated by not applying it.

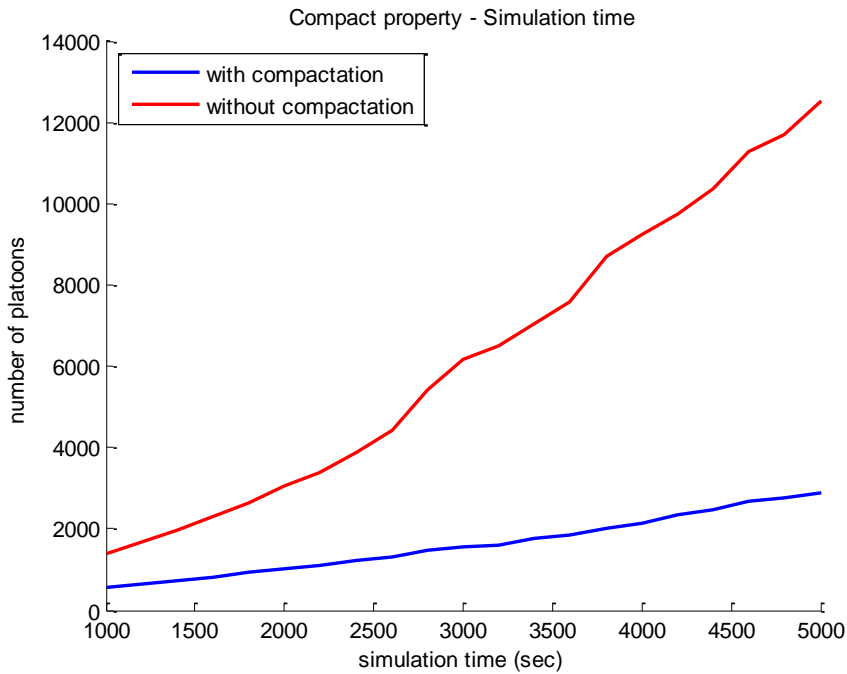


Figure 3-16 - Number of platoons required for a simulation applying or not applying the compaction property with the range of the simulation time

To benefit of the advantage given by the compaction, before generating the new platoons leaving from a particular cycle, the model must classify all the platoon that come to it.

The generated platoons, then, are recalled from the *list_platoons* list in small groups formed by all those who come to the intersection in the analyzed cycle. Therefore the variable along which the simulation proceeds is not explicitly the time, but the traffic light cycles. The order followed for the cycles may be different, but it must ensure the classification of a platoon only when it was possible to classify all platoons that can influence its behavior. This condition

is not satisfied by the model in the case of spillback and the temporal "paradox" is solved as described in 3.4.2.

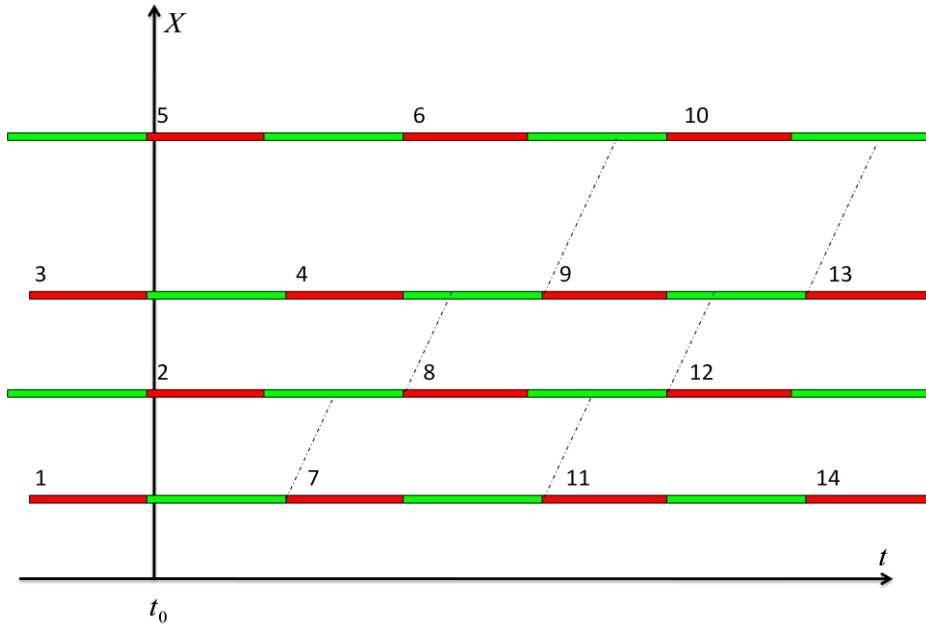


Figure 3-17 - Cycles order

In the realized model, the numbering of the cycles proceeds according to the following rules:

1. It is assigned 1 to the first cycle of the first intersection across the initial instant t_0 of the simulation;
2. Assuming a vehicle that starts at the last instant of the last numbered cycle, the c ending cycle is calculated in the downstream intersection;

3. All cycles are numbered, in the downstream intersection, from the last one not numbered to the c cycle included;
4. The procedure is repeated returning to step 2 up to the end of cycles.

In Figure 3-17 are shown the fictitious trajectories of vehicles assumed in step 2 and the numbering resulting from the application of the procedure.

3.4.2 Law of progression

The law of progression simulates the progress of the platoons along the artery arcs, with vehicles, in the work hypothesis, that travels in compact formation with uniform speed equal to the free speed v_f . This rule does not correctly approximates the behavior of the vehicles belonging to two little spaced platoons. In fact, typically drivers that follow have the tendency to accelerate in order to reduce the distance from preceding vehicles. This behavior only occurs when the distance is small enough to be recovered in the road section between two traffic lights. The difference between the free speed and the speed that the driver is willing to have to close the gap is called *catch-up* speed (v_c), that is the same name used within this text to identify the phenomenon. To model this behavior properly, the law of progression monitors the temporal distance between two or more platoons and it verifies if there are *catch-up* conditions. If this occurs, to the following platoon is assigned a speed $v = v_f + v_c$ up to the instant in which it reaches the platoon that precedes after which the two platoons are compacted and conformed to the speed v_f . The law of progression also aims at analyzing the trajectories of

platoons and at calculating the end point of the first and last vehicle at the next intersection. In the trajectories analysis, the model must take into account whether the downstream arc has sufficient capacity to bring all the vehicles in the platoon, and if not, it must register the *spillback* event that is generated.

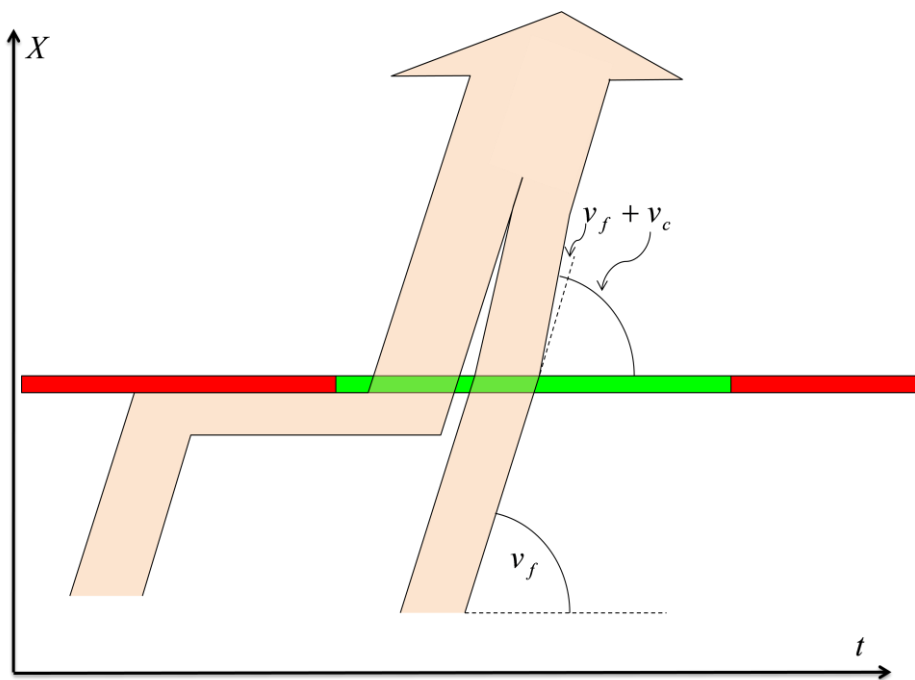


Figure 3-18 - Catch-UP

Spillback

With the term "spillback" it is indicated the case in which the congestion that develops over a downstream arc, compared to direction of the progression of a vehicle, reduces the capacity of an upstream intersection. In the case

described above, the green traffic light is ignored and the capacity is reduced by an amount equal to $q_s = t_s \cdot s$ where t_s is the time for which the state of spillback remains.

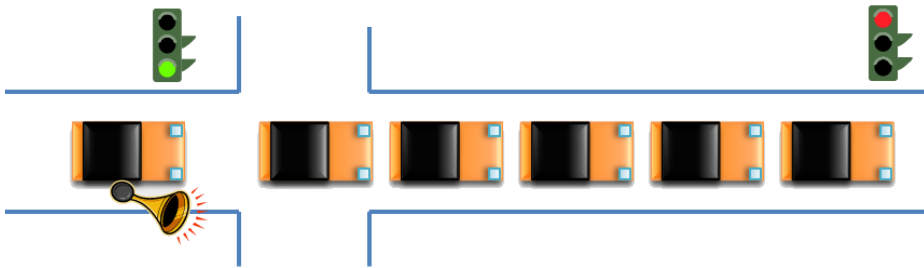


Figure 3-19 - Example of spillback phenomenon

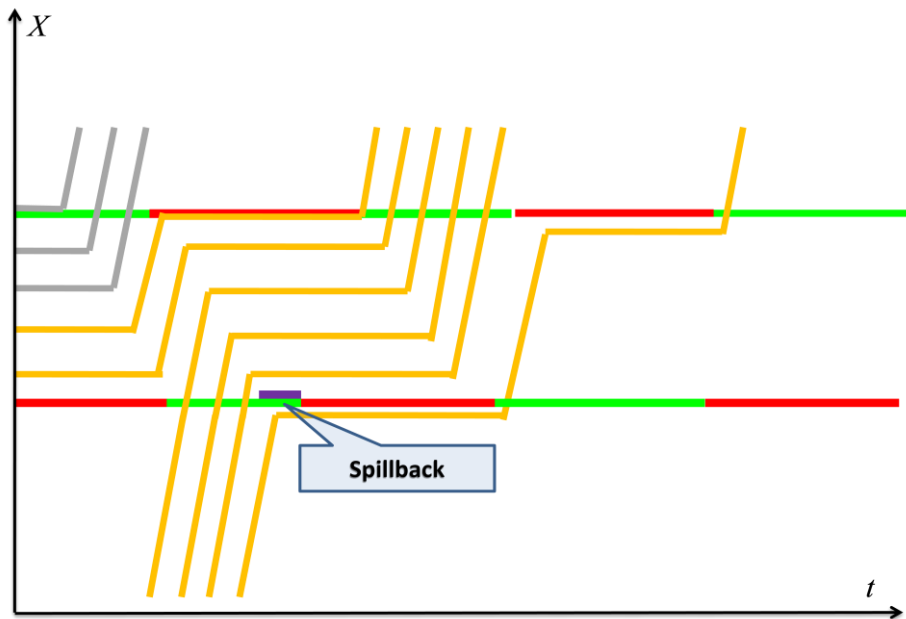


Figure 3-20 - Spillback in the space-time diagram

In the platoons model, the spillback analysis creates some problems. In fact, to benefit of the compaction property, that increases significantly the performance, the model must be able to analyze everything that happens in a traffic light cycle before it is possible to proceed in the simulation. As described in paragraph 3.4.1, this requires a minimum time-step in the simulation that is equal to the cycle duration. This is an interval too wide to be able to well manage the spillback. In fact after classifying the events that characterize a traffic light cycle, it is possible to classify several other platoons before the model can realize that the platoons already classified could be in a spillback condition.

As it is possible to see in Figure 3-21 only in the classification of the platoon 5 the model realizes that part of platoon 3 and part of platoon 4 are in spillback situation. At this point this simulation should reprocess all classified platoons from the first which came into spillback. In the example it should be reclassified platoons 3, 4 and 5 that become 3b, 4b and 5b. This model, here called *reclassification model*, was implemented during this research work, but it has proven to be very inefficient. In most cases, under conditions of oversaturation, the number of reclassified platoons represented more than 50%-70% of the total number of platoons, by doubling or tripling the processing times.

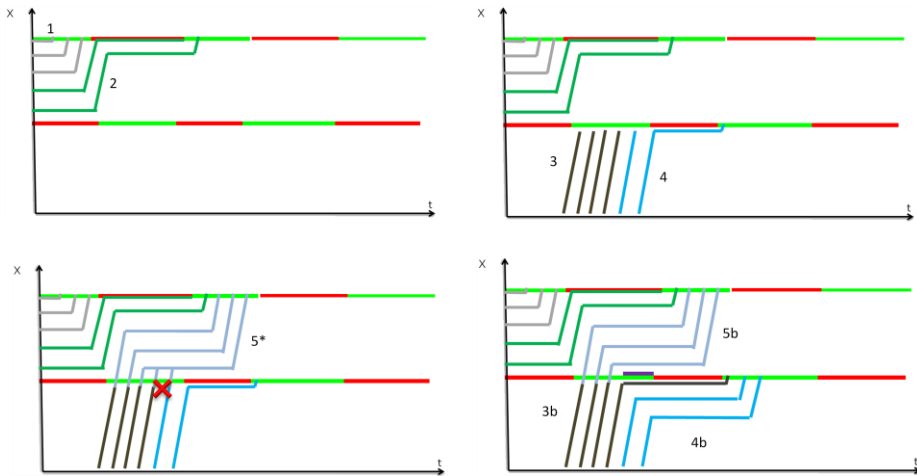


Figure 3-21 - Example of reclassification in case of spillback

The research was therefore aimed at establishing a more efficient model that takes into account the spillback phenomenon. The problem was solved by adopting a method that, by alternating the simulation with space-time inconsistencies, maintains unchanged the calculation of total delay. In particular, this method, here called *push-back*, creates a new platoon formed only by the vehicles that are stopped by spillback and it puts it after all the classified platoons. An example of the described technique is shown in Figure 3-22 (b) which is compared to the method of reclassification submitted at the beginning of the paragraph. As it is possible to see, except for the point at which the spillback occurs, the two simulations are identical.

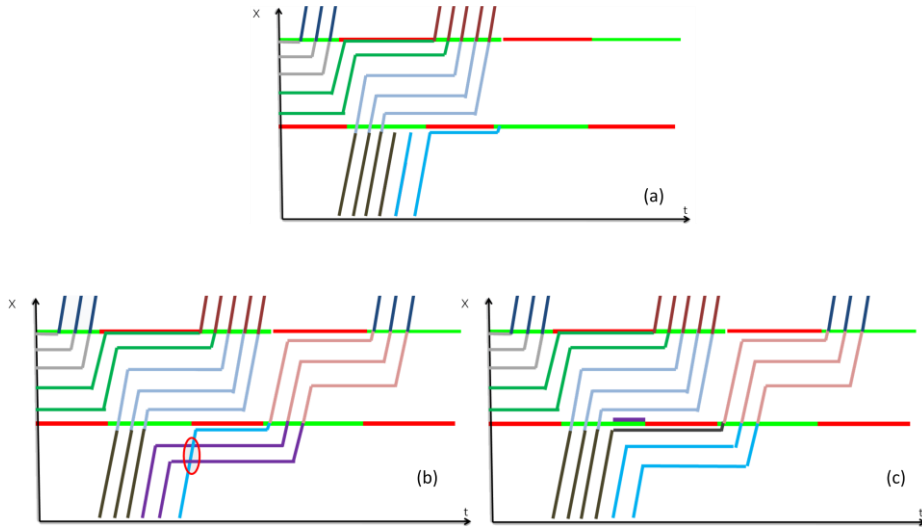


Figure 3-22 - Spillback condition (a). Push-back method (b). Reclassification method (c).

In this thesis, the calculation of total delay at the intersection through the push-back method was demonstrated numerically correct. In fact, as it is possible to see in Figure 3-23, normally analyzing the spillback condition, the total delay accumulated by the three platoons is:

$$d_s = n_1 \cdot 0 + n_2 \cdot (t_s + r_e) + n_3 \cdot \left(r_e - t_3 + \frac{n_2}{s} \right)$$

Eq. 3-1

where:

- n_1, n_2 and n_3 represent the number of vehicles of the three platoons;
- t_s represents the time interval for which the state of spillback remains;

- t_3 represents the temporal distance between the red light and the arrival expected for the first vehicle of platoon 3 in the absence of queues at the intersection;
- r_e represents the effective time of red;
- s is the saturation flow.

If it is adopted the push-back method, the total delay can be calculated as:

$$d_p = n_1 \cdot 0 + n_2 \cdot \left(t_s + r_e + \frac{n_3}{s} \right) + n_3 \cdot (r_e - t_3)$$

Eq. 3-2

equalizing the two expressions:

$$n_1 \cdot 0 + n_2 \cdot (t_s + r_e) + n_3 \cdot \left(r_e - t_3 + \frac{n_2}{s} \right) = n_1 \cdot 0 + n_2 \cdot \left(t_s + r_e + \frac{n_3}{s} \right) + n_3 \cdot (r_e - t_3)$$

$$n_2 \cdot t_s + n_2 \cdot r_e + n_3 \cdot (r_e - t_3) + \frac{n_3 \cdot n_2}{s} = n_2 \cdot t_s + n_2 \cdot r_e + \frac{n_2 \cdot n_3}{s} + n_3 \cdot (r_e - t_3)$$

$$0 = 0$$

Eq. 3-3

The equality is always satisfied by showing that it is possible to use the push-back method for modeling the spillback.

The demonstration assumes that a queue at a signal spills back up to the upstream intersection and blocks it during the remaining green time, like in the example.

However, when the spill-back queue moves forward before the end of green, at least a fraction of the stopped platoon can leave the intersection. The remaining fraction experiences the spill-back delay. Only this fraction can be pushed back in the delay computation (hypothesis in which n_1 and n_2 form a single platoon as in Figure 3-22).

This model has been used without affecting the performances.

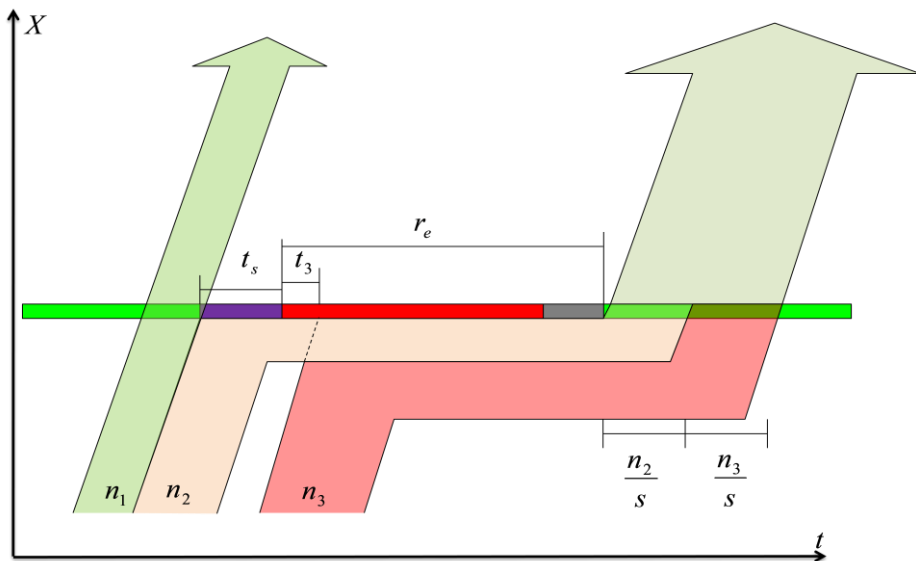


Figure 3-23 - Spillback condition

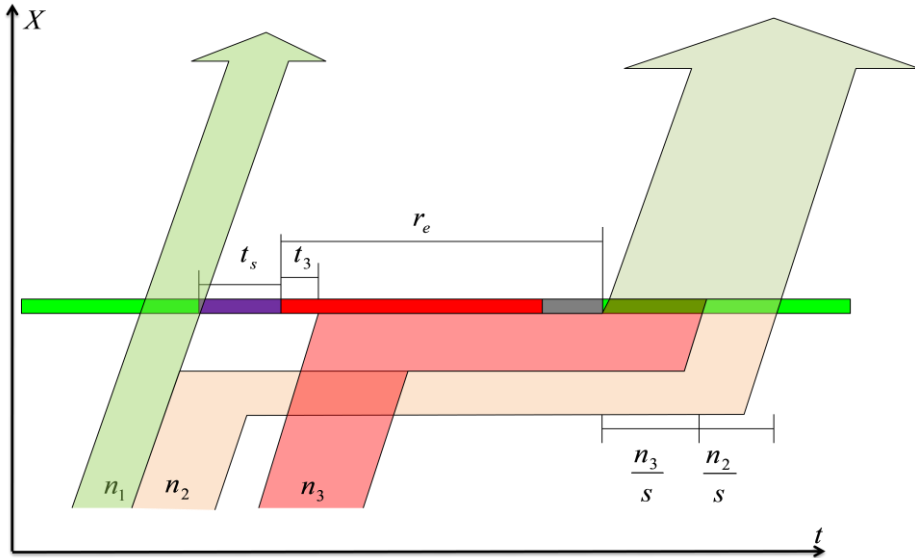


Figure 3-24 - Push-back method

3.4.3 Law of classification

The law of classification is the same used in the model developed by Fusco and Papola and presented earlier in this chapter. The only difference with the Progression Platoons is the generalization for the platoons classification in condition of congestion. In fact, working exclusively in sub-saturation, the original model classifies platoons taking into account only the red time and the time for the queue throughput at intersection for the analyzed traffic light cycle. In addition, Papola and Fusco model calculates in advance the throughput of the whole platoon in one cycle for the platoons of type A and C and in two cycles for those of type B.

In reality, in case of oversaturation, the platoons may have to wait several cycles before they can reach the traffic light and before the green light and,

the vehicles in the same platoon can reach the intersection in different cycles. In fact, in Figure 3-25, it is possible to see how platoon 1 can be disposed in a single cycle, but with the increasing of the congestion, the vehicles of the following platoons must wait more traffic light cycles to be able to flow. In particular, vehicles of platoon 2 are disposed in three distinct cycles.

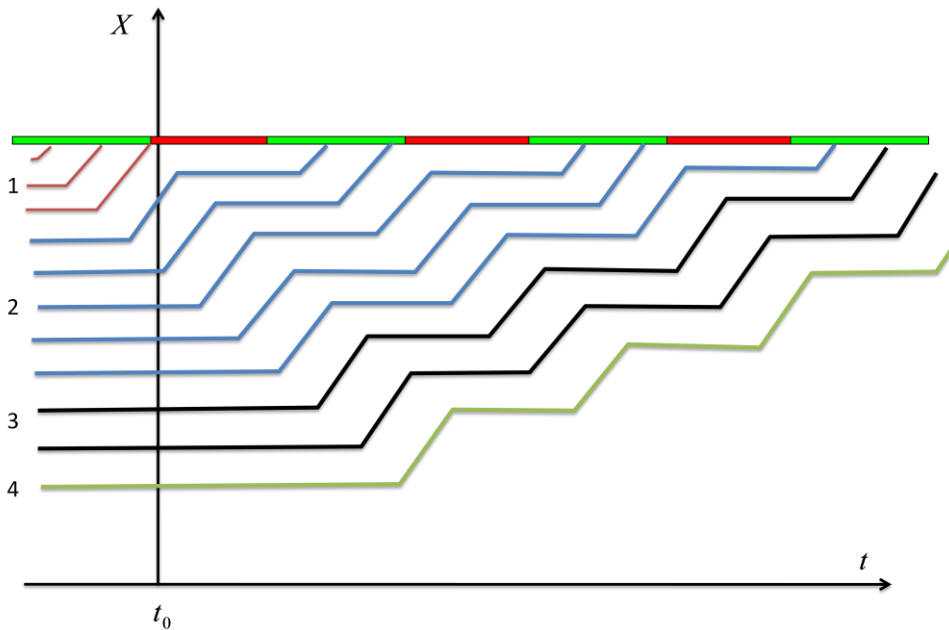


Figure 3-25 - Trajectories of vehicles in different platoons in case of oversaturation

This generalization allowed to apply the Fusco and Papola law of classification in a dynamic model such as the model developed in this thesis.

3.4.4 The pseudo-code

After the deeper description of the rules applied in the simulation model, the flow chart in Figure 3-13 can be extended as in Figure 3-26.

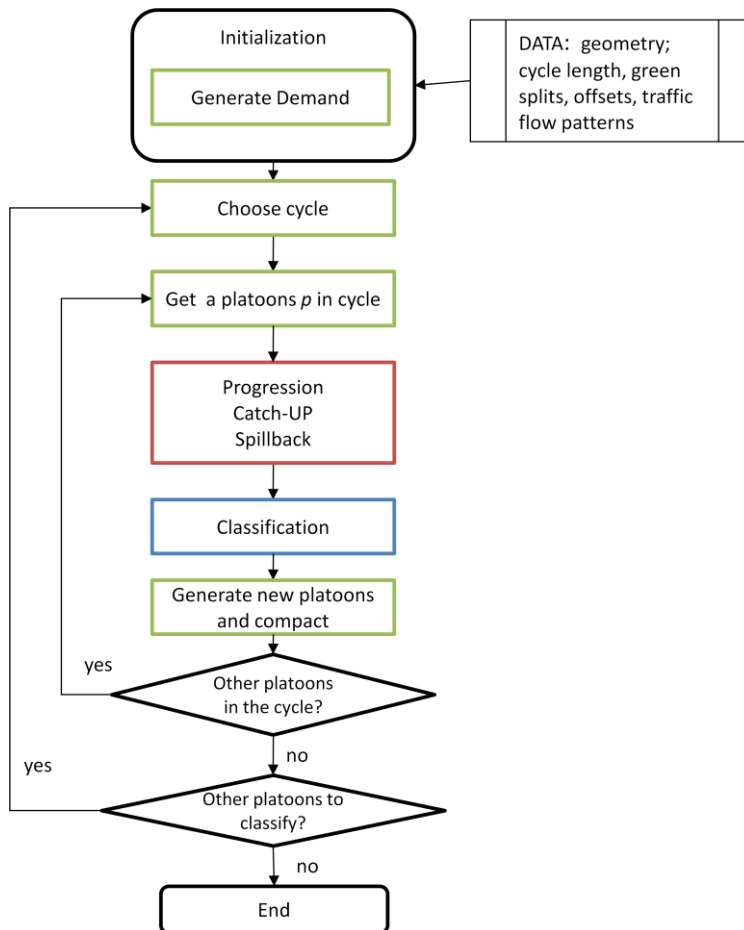


Figure 3-26 - Extended delay model flow chart

The following pseudo-code of the road artery delay model describes the algorithm to simulate the progression of platoons.

```

structure PLATOON
  j1 // intersection of start
  j2 // intersection of end
  n // number of veichles in the platoon
  nq // number of delayed vehicles
  delay // delay for each vehicle
  type // type of classification (0: not classified, a: fore-delayed, ...

```

```

        // b: hind-delayed, c: no-delayed)
    cycle // number of cycle, from the start of simulation, ...
        // in which the platoon restarts from the intersection j2
end structure

function ARTERY_DELAY_MODEL
    J=set of intersections;
    n=number of intersections;
    list_platoons = {demand at first node}; // aggregate demand at the first
        // node in platoons
    list_platoons = { list_platoons, demand at lateral approach} // add laterl demand
    for each p in list_platoons
        p.type=0 // no-classified
    end for
    outflow(1..n) = rate of road artery outflows for the intersections in J;
    c0=0; // starting cycle at the first node
    while not all platoons in list_platoons are classified
        c0=c0+1; // cycle to analyze at the first node
        j=0; // intersection analyzed
        while j<n
            j=j+1;
            if j=1 then c=c0
                pl = set the platoons starting at the node j up to the cycle c;
                if pl is not empty
                    c=0; // reset the cycle to analyzefor the next step
                else
                    continue; // return at the start of the while command
                end for
            CATCH_UP(pl); // apply the catch-up rule to the set of platoons
            for each p in pl // analyze and classify the platoons
                if p doesn't arrive from a lateral approach then then
                    // remove the vehicles of outflows from platoon
                    p.n=p.n*(1-outflow(p.j2));
                end if
                // classify the type of platoon and calculate the delay
                CLASSIFICATION(p);
                // to the next step at the next node analyze the platoons untill
                // the last cycle reached from the new platoons
                c=max(p.cycle - 1,c)
                // generate new platoons from the classification of p
                np = GENERATE(p);
                // apply the compactation-rule to the new platoons
                COMPACT(np);
                list_platoons = { list_platoons,np}
            end for
        end while
    end while
    delay=0; // total delay
    // This model can calculate others delays e.g. for each node, mean delay, // etc...
    for each p in platoons
        delay=delay + p.nq * p.delay;
    end for
end function

```

```

    end
    return delay;
end function

function CLASSIFICATION(p)
    spill=CHECKSPILBACK(p) // check if the platoon p causes the spillback
    if spill then
        vs = vehicles unable to enter in the downstream link
        p.n= p.n - vs;
        // create a new platoon of the vehicles not entering in the downstream link
        q= new platoon;
        q.n=vs;
        q.j1=p.j1; q.j2=p.j2;
        q.type = 0;
        list_platoons ={ list_platoons,q}; // the new platoon has to be classified
    end if
    // Test for a-type platoon (FORE-DELAYED PLATOONS)
    // Test for b-type platoon (HIND-DELAYED PLATOONS)
    // Test for c-type platoon (NO-DELAYED PLATOONS)
    if all the vehicles are stopped then
        p.type=a;
        p.nq=p.n;
        p.delay = time of stop;
    end if
    if the firsts vehicles pass and the last ones are stopped then
        p.type=b;
        p.nq= stopped vehicles;
        p.delay = time of stop;
    end if
    if all the vehicles of p are not stopped then
        p.type=c;
        p.nq=0;
        p.delay=0;
    end if
end function

function CATCHUP(pl)
    for each p in pl
        q=next(p)
        if p is adjacent to q
            merge(p,q)
        end
    end for
end function

function CHECKSPILBACK(p)
    // The procedure checks if the residual capacity in the downstream link is
    // sufficient to allow the entrance of platoon p
end function

```

3.4.5 The model for public transportation

The model was extended to simulate the public transportation traffic flow and calculate its delay. In particular, only the cases in which transit vehicles travel on reserved lanes were simulated, representing so the most common case on huge urban road arteries. The model was extended also to take into consideration potential priority strategies.

To represent the public transportation, in line with the private traffic flow model, the means have been modeled as platoons where an only vehicle does not follow the rule of compacting. In the public transportation model, apart from the traffic lights, also the stop nodes were implemented to simulate the stop.

For each vehicle the following parameters are set:

- length
- maximum capacity
- residual capacity
- speed
- stop table
- timetable
- preload
- average time to get on (per passenger)
- average time to get off (per passenger)
- fixed time required to complete a stop

In the same way, for each stop it is defined:

- demand generation
- demand attraction
- level of demand
- preload
- possibility to overtake
- additional fixed time required for the stop request

The effects of the priority strategies are the great changes of the accumulated time of delay for both the public transportation and the private means, though depending on a small number of vehicles. Stochastic elements have been introduced both in the demand model and the supply model, in order to minimize the possibilities to find the solutions that are extremely rigid and not much sensitive to the variation of the traffic flow in the optimization phase.

Demand model

For each line, at the stop the model requires the definition of the number of passengers who are already waiting. During the simulation, an increase hourly of the demand is added to the preload, according to a Gaussian distribution with mean and variance defined by the user. When a vehicle reaches the stop, the model calculates the demand as a sum of the initial demand plus its increase, and if the residual capacity of the vehicle is sufficient, then the model simulates the getting on of all passengers. Otherwise, the users who cannot get on board remain at the stop waiting for the next vehicle of the same line. Also the number of passengers getting off at each hour is formed as a definite Gaussian distribution and the algorithm calculates the number of

passengers getting off on the basis of this the distribution, the time elapsed from the last pass and the number of passengers on board.

Supply model

As previously mentioned, the supply model is represented by the definition of the public transportation lines and stops along the road artery.

The data can be divided into those that are useful to represent the progression of the platoon along the road artery and those that are useful to simulate the interaction with the demand model .

Among the former category there are the vehicle length, the speed and tables of the stops. The model uses these measurements to simulate the progression of the platoon along the road artery, the classification at the intersection and the queuing.

The timetable is also used both to estimate the accumulated delay and to calculate the instant when the vehicle enters the road artery. This instant is measured through a normal distribution.

Among the information useful to simulate the interaction with the demand model, the algorithm uses the maximum capacity, the average time to get on and off, and the fixed waiting time at the stop. It enables furthermore to define the possibility to overtake at each stop. This information is used to calculate whether the vehicles that do not stop at the bus shelter or have completed their passenger load and unload can overtake the other leader vehicles still engaged at the stop.

Priority strategies

The priority strategies implemented, beyond the passive strategy of synchronization, are *the green extension* and *the green advance*. For each of these two strategies, a maximum advance interval and a maximum extension interval are defined. They are applied only if necessary and only if the security requirements for traffic signal phases are satisfied.

Through the advance of the green light the model estimates the instant in which the vehicle arrives at the intersection. If this instant precedes the switching on of the green light, then the system provides the green advance in the necessary interval adding another advance to avoid stopping the vehicle on the part of the driver.

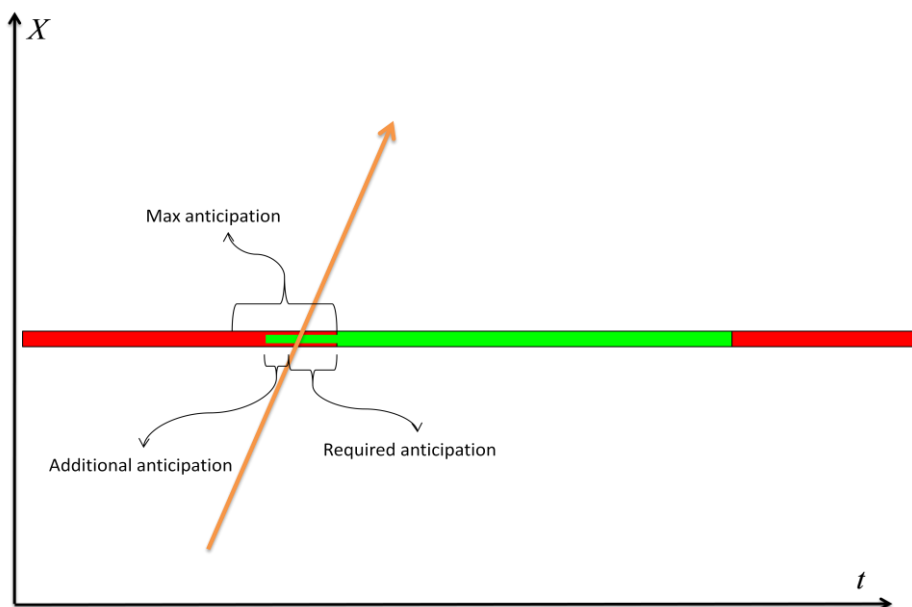


Figure 3-27 - Green advance

In the strategy of green extension, the algorithm checks the instant of the expected arrival of the vehicle and calculates whether this strategy is applicable. If it is possible, it delays the switching on of the red light to enable the vehicle to go on, adds another extension to perform the procedure of switching on of the yellow light (and then of the red one) and grant the margins of security for the vehicles that are taking advantage of the green extension with the bus.

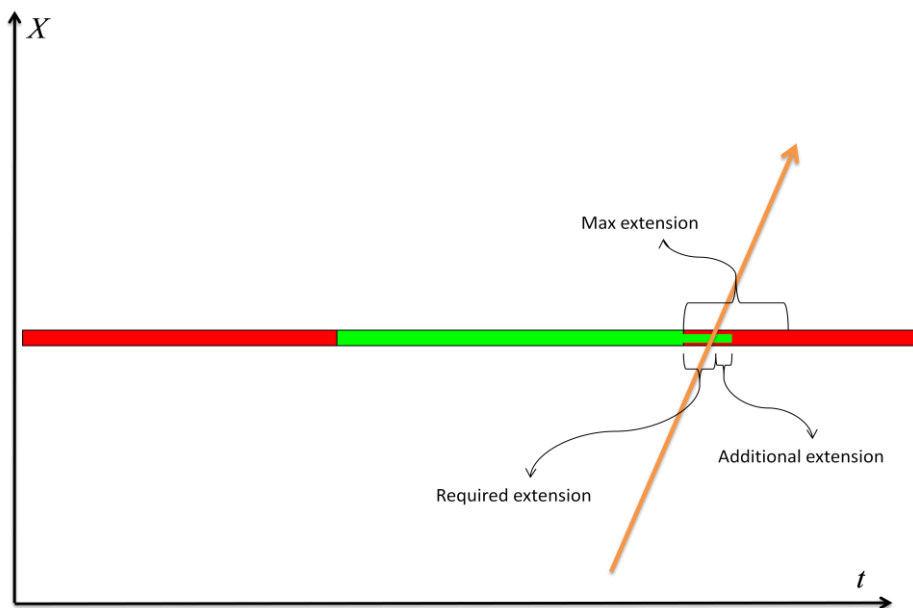


Figure 3-28 - Green extension

In the assignment of priority, the model could have to face conflictual situations. Especially when multiple buses travelling in different directions request the priority, which cannot be applied simultaneously, the algorithm must decide which is vehicle the priority is to be assigned.

The method of choice is based on an assignment of a score to each bus on the basis of the stop request instant (F_{time}), the accumulated delay of the vehicle (F_{late}), the number of passengers ($F_{passenger}$) and the possible priority assigned ($F_{priority}$) to the line:

$$F_{time}(A, B) = \begin{cases} A = W_t, & B = 1 \quad t_A < t_B \\ A = 1, & B = W_t \quad t_A > t_B \end{cases}$$

$$F_{late} = W_L \times T_{late}$$

$$F_{passenger} = W_P \times N_{passenger}$$

$$F_{priority} = W_{priority} \times P$$

$$F_{tot} = F_{time} \times (F_{late} + F_{passenger} + F_{priority})$$

The bus getting the higher score will gain priority.

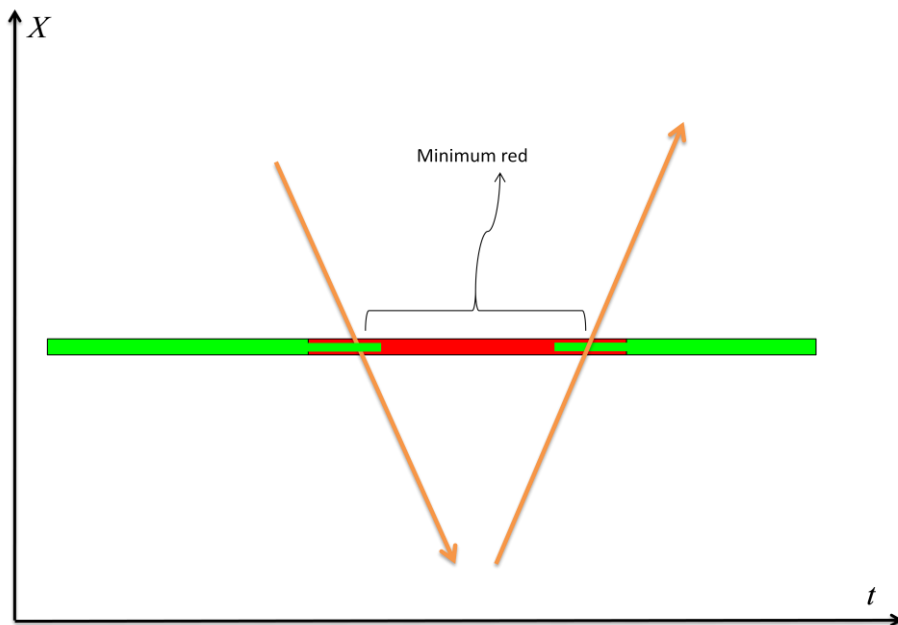


Figure 3-29 - Conflict in the assignment of the priority

Once the priority is assigned, the algorithm grants a certain number of cycles, so that the transversal flows can recover the green light and during these cycles no priority strategy can be applied.

The simulation model for public transportation

The vehicle progression model is very close to the algorithm described for the private transportation in paragraph 3.4.4 with the addition of a simulation of the stop and the overtake, prohibited for the private transportation.

As previously said, the means of public transportation are modeled as platoons of an only vehicle, so the rule of compacting loses its efficacy for them. Moreover, bus drivers behave differently from drivers of private means

and the tendency towards the catch-up phenomenon is almost inexistent. Apart from these two rules, the need to classify contiguous group platoons disappears (Par. 3.4.1) and so a method of single-platoon classification has been proposed in order to reduce the minimum simulation interval and analyze the evolution of the vehicles trajectories in deeper details. This feature has been in fact used to recognize and solve the conflicts in the assignment of the priorities.

The list *list_platoons* has been created by implementing a priority queue that is temporarily ordered on the basis of the instant of expected arrival of the platoon at the end of the road artery without intersections. This system assures that the platoon, which is at the head of the priority queue, cannot be influenced by the other platoons not classified. At each iteration the platoon selected for the classification or the simulation of the stop is therefore the one placed at the head of the priority queue.

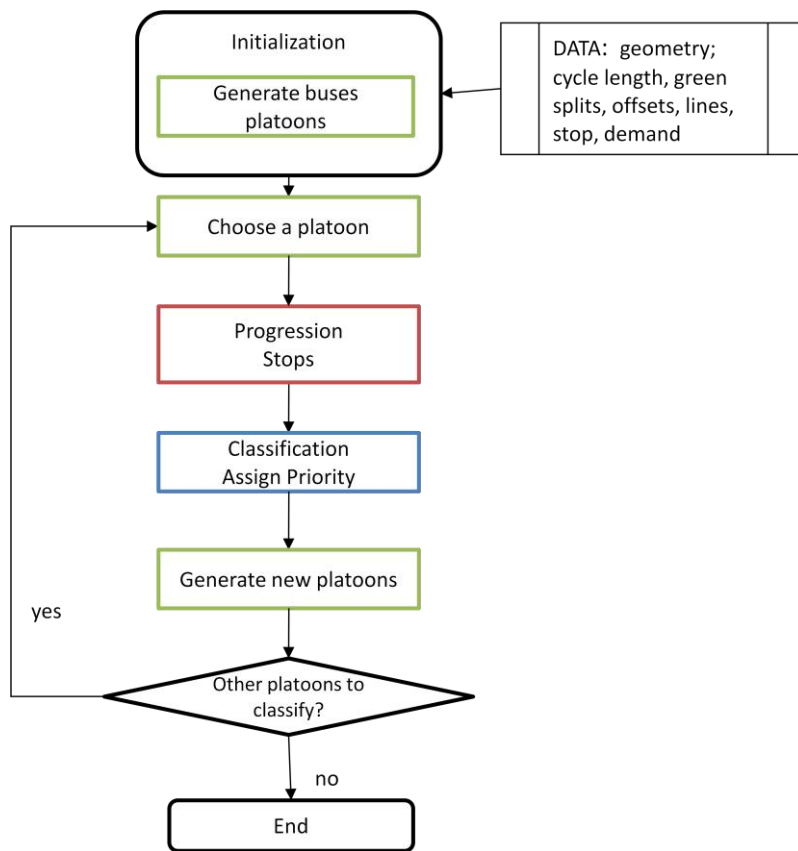


Figure 3-30 - Public Transport delay model flow-chart

Integration in private simulation model

The public transportation simulation model is performed separately from the private transportation simulation model. Travelling on reserved lanes, in fact, public means do not interact with private vehicles, except for the effects on the synchronization resulting from the application of the priority strategies. Therefore, a public transportation simulation a priori helps evaluating in

advance the variations of the traffic signal phases deriving from the priority strategies, and only later the private transportation will be simulated on the basis of such variations.

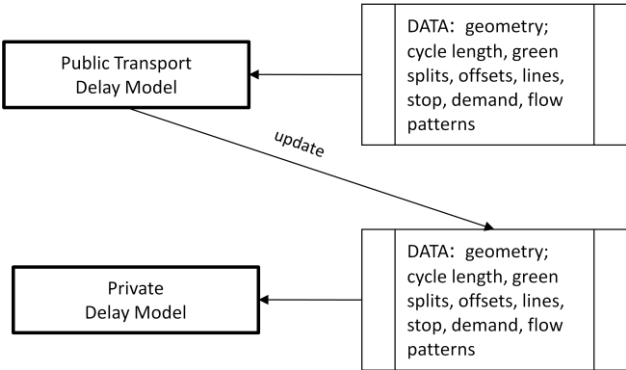


Figure 3-31 - Relations between models and data

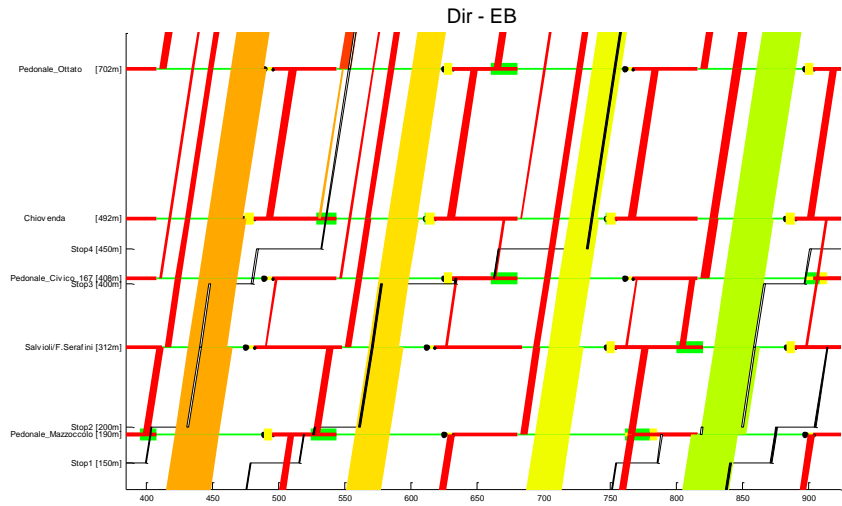


Figure 3-32 - Example of the progression of platoons generated by the model (buses are highlighted in black)

4 The optimization algorithms

The model described in the previous chapter has been used as a method of evaluation of a traffic light synchronization for a urban road artery. On the basis of the delays calculated it was possible to define the fitness function used by two heuristic optimization algorithms: a genetic algorithm and a particle swarm algorithm.

The two procedures have been used as methods of global research and at the end of each procedure an algorithm of local research has been performed: hill-climbing.

Each of the procedures performed has been gauged to search for the best compromise between the performance times and the quality of the solution. In fact, an optimal solution generated by any simulation model would represent only a point of departure or a way of measurement for the work a civil engineer must carry out to implement a traffic light synchronization system to really apply.

4.1 The fitness function

The delays calculated by the model have been used to define a fitness function representing the delay weighted differently on its parts. The function performed is:

$$f = (1 - w_p) \left((1 - w_t) \left(w_1 \sum_{i=1}^n \omega_i D_i^{(1)} + (1 - w_1) \sum_{i=1}^n \omega_i D_i^{(2)} \right) + w_t \sum_{i=1}^n \omega_i D_{h_i}^{(t)} \right) + w_p \left(w_1 \sum_{b=1}^B \sum_{i=1}^n \omega_i D_{b_i}^{(p_1)} + (1 - w_1) \sum_{b=1}^B \sum_{i=1}^n \omega_i D_{b_i}^{(p_2)} \right)$$

where:

- $D_i^{(1)}$: the total delay at node i in direction 1
- $D_i^{(2)}$: the total delay at node i in direction 2
- $D_{h_i}^{(t)}$: the total delay at node i of queue h in lateral approach t
- $D_b^{(p_1)}$: the total delay of passengers in bus b at node i in direction 1
- $D_b^{(p_2)}$: the total delay of passengers in bus b at node I in direction 2
- w_1 : the weight of delay in direction 1
- w_t : the weight of the delay at lateral approaches
- w_p : the weight of delay for transit passengers

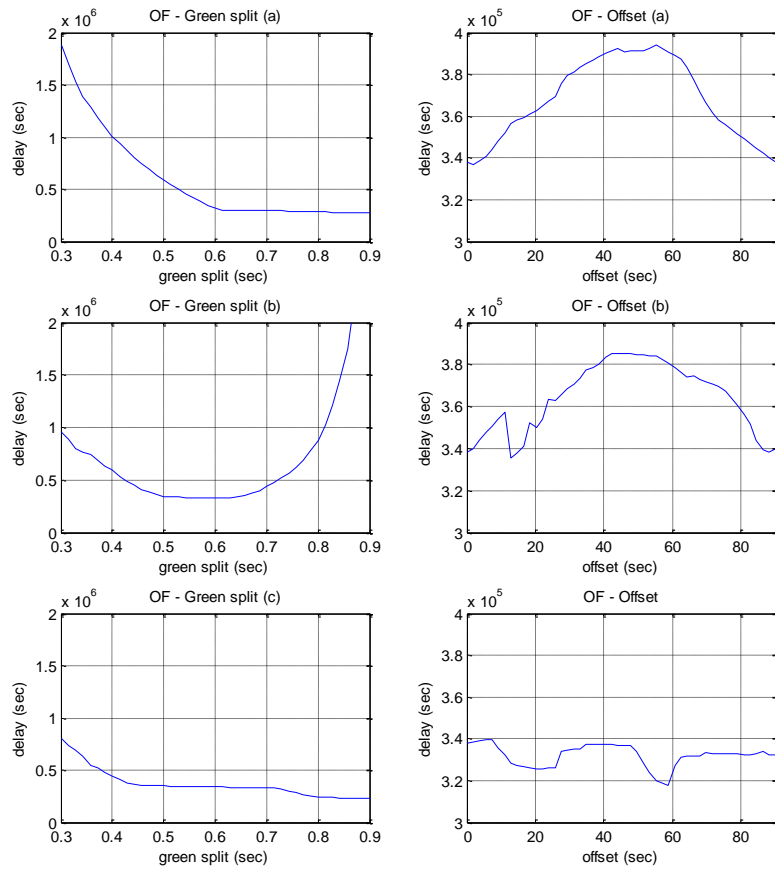


Figure 4-1 - Scanning of six different dimensions of the objective function

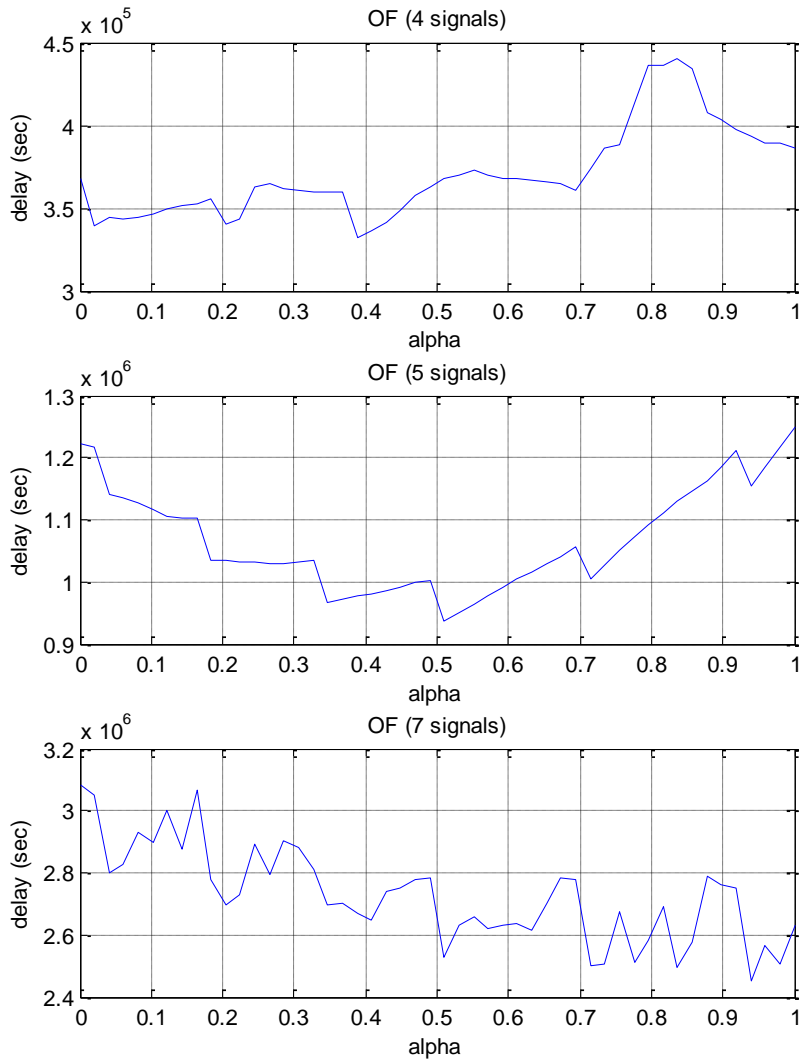


Figure 4-2 - Scanning of the objective function between two generic configuration (c1,c2) of three road artery. α is the coefficient of convex combination between the configuration (OF=delay(α *c1+(1- α)*c2).

4.2 The genetic algorithm

The genetic algorithms are procedures of heuristic optimization taking Darwin's theory "*On the Origin of Species by Means of Natural Selection*" as a starting point. According to this theory, the best specimen of each species have higher chances to reproduce than the others and so to spread their own genetic inheritance. This feature together with the casualty resulting by the genetic mutations leads a species to evolving in better and better individuals. Bringing this theory in the ambit of the optimization issues, it is possible identify in the individuals the solutions to our problem and in DNA its variables. In general, the glossary below for the specific terms can be defined between Darwin's theory and the genetic algorithm:

Darwin theory	Genetic Algorithm
Genome	Variables
Specimen	Solution
Reproduction	Crossover
Genetic Mutation	Mutation
Population	Population
The best specimen	Elite

Defining each of these elements helps adapting the genetic algorithm to any issue for which a fitness function can be calculated.

A generic genetic algorithm can be represented by the pseudo-code below:

```
function GA()  
    population = GENERATE() // It generates the initial population  
    FITNESS(population) // It calculates the fitness for each individual  
    elite=the  $n^{\text{th}}$  best of population  
    while not CONVERGE()  
        next_generation={elite}
```

```

while next_generation.length < population.length
    p=ROULETTEWHEEL(population) // extraction of the father
    m= ROULETTEWHEEL (population) // extraction of the mother
    (c1,c2)= CROSSOVER(m,p) // reproduction
    MUTATION(c1)
    MUTATION(c2)
    next_generation={ next_generation; c1,c2}
end while
population= next_generation
elite=the nth best of population
end while
end function

```

In general, crossover and mutation operators are not always applied, but a probability to be performed is associated to each of them. The choice of the individuals participating to the creation of the next generation is linked to the ROULETTEWHEEL function associated to a higher probability to be chosen for the solutions showing better fitness. Generation by generation, the algorithm preserves its better solution calculated until that moment. In the implementation carried out, this rule has been extended to keep the best N calculated until that moment called elite.

4.2.1 Genoma

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

$$\{C_1, \dots, C_n, g_1^{(1)}, \dots, g_n^{(1)}, g_1^{(2)}, \dots, g_n^{(2)}, \vartheta_1, \dots, \vartheta_n\}$$

where:

- C_i is the cycle of node i

- $g_i^{(d)}$ is the green split of node i in direction d
- \mathcal{G}_i is the offset of node i

A compact representation of this vector was used. Since the offset of is a relative dimension, was set $\mathcal{G}_1 = 0$ and the others offsets were made explicit as differences. In addition, where there is a traffic light at two stages (it needs only one split green for both directions), the green split has been expressed with a single variable. In an instance with n traffic lights and m three stage traffic lights the vector is:

$$\{C_1, \dots, C_n, g_1^{(1)}, \dots, g_n^{(1)}, g_1^{(2)}, \dots, g_m^{(2)}, \mathcal{G}_2 - \mathcal{G}_1, \dots, \mathcal{G}_n - \mathcal{G}_1\}$$

4.2.2 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 are:

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

$$C = \max\{C_{Web,i}\} \text{ OR } C = \max\{C_{min,i}\}$$

where:

$$C_{Web,i} = \frac{1.5L_i + 5}{1 - \sum_h^m 1 - y_{h,i}^*}$$

$$C_{min,i} = \frac{L_i}{1 - \sum_h^m 1 - y_{h,i}^*}$$

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

$$g_{k,i} = \frac{y_{h,i}^*}{\sum_h^m y_{h,i}^*} \text{ OR } \begin{matrix} g_{h,i} = y_{h,i}^* \\ g_{k,i} = 1 - \sum_h^m \lambda_{h,i} \end{matrix}$$

- 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
- $g_{h,i}$ 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.

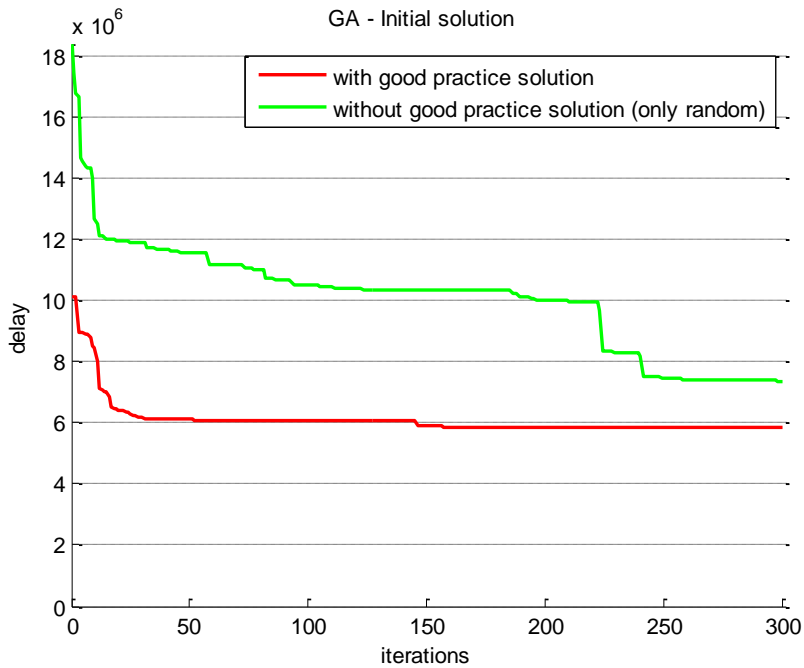


Figure 4-3 – Comparison between trend of algorithm with random initial population or with intelligent construction of initial solution

4.2.3 Crossover

The crossover operator combines the genetic inheritance of a pair of individuals (parents) to generate new individuals (children). The aim of crossover is to generate a couple of children for each couple of parents,

choosing randomly (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 the reproduction is proportional to their fitness value. Better individuals have so a higher probability to transmit their genetic inheritance. Moreover, in order to improve the algorithm flexibility, the crossover operator is applied only to a given rate of the whole population called *crossover rate*.

For example if are choose these parents:

$$\text{father} = \{C_1^f, C_2^f, C_3^f, C_4^f, g_1^{(1)f}, g_2^{(1)f}, g_3^{(1)f}, g_4^{(1)f}, g_2^{(2)f}, g_3^{(2)f}, \Delta g_2^f, \Delta g_3^f, \Delta g_4^f\}$$

$$\text{mother} = \{C_1^s, C_2^s, C_3^s, C_4^s, g_1^{(1)s}, g_2^{(1)s}, g_3^{(1)s}, g_4^{(1)s}, g_2^{(2)s}, g_3^{(2)s}, \Delta g_2^s, \Delta g_3^s, \Delta g_4^s\}$$

the potential children are:

$$\text{1st child} = \{C_1^f, C_2^s, C_3^f, C_4^f, g_1^{(1)s}, g_2^{(1)f}, g_3^{(1)s}, g_4^{(1)f}, g_2^{(2)s}, g_3^{(2)f}, \Delta g_2^f, \Delta g_3^s, \Delta g_4^s\}$$

$$\text{2ndchild} = \{C_1^s, C_2^f, C_3^s, C_4^s, g_1^{(1)f}, g_2^{(1)s}, g_3^{(1)f}, g_4^{(1)s}, g_2^{(2)f}, g_3^{(2)s}, \Delta g_2^s, \Delta g_3^f, \Delta g_4^f\}$$

This implementation can be represented through a matrix, which helps explaining better the types of crossover performed. A matrix representation is shown in Figure 4-4.

	nc	$g^{(1)}$	$g^{(2)}$	$\Delta\theta$
	1	0.5		
	2	0.6	0.65	3

C_{max}	1	0.4		15
180	2	0.7	0.54	75

Figure 4-4 - Matrix representation of Genome

Where C_{\max} represents the synchronization cycle of the road artery and the field nc represents the dividend of C_{\max} used to calculate the cycle C_i for the intersection.

$$C_i = C_{\max} / nc$$

The other variables assume the meaning described previously.

Three were the types of crossover implemented. The first was presented by Colombaroni, Fusco and Gemma (2009). For each parameter it chooses the use of value extracted from the first or the second parent with equal probability to have a child. The second child is complementary to the first one.

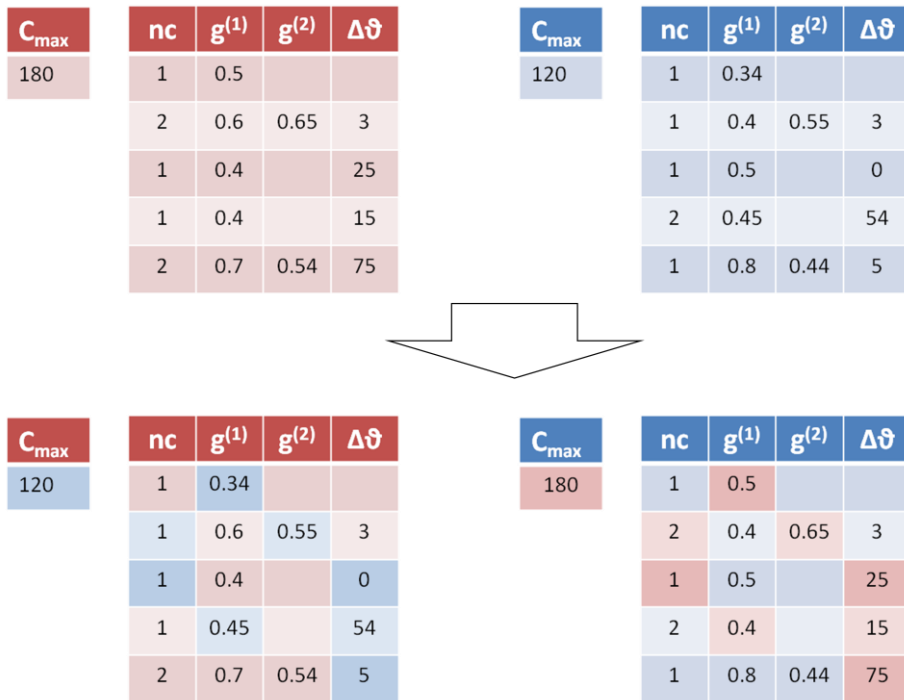


Figure 4-5 - Crossover of type 1

In the second method a child is generated choosing with equal probability the parent from whom the three parameters are to be extracted for each intersection as shown in Figure 4-6:

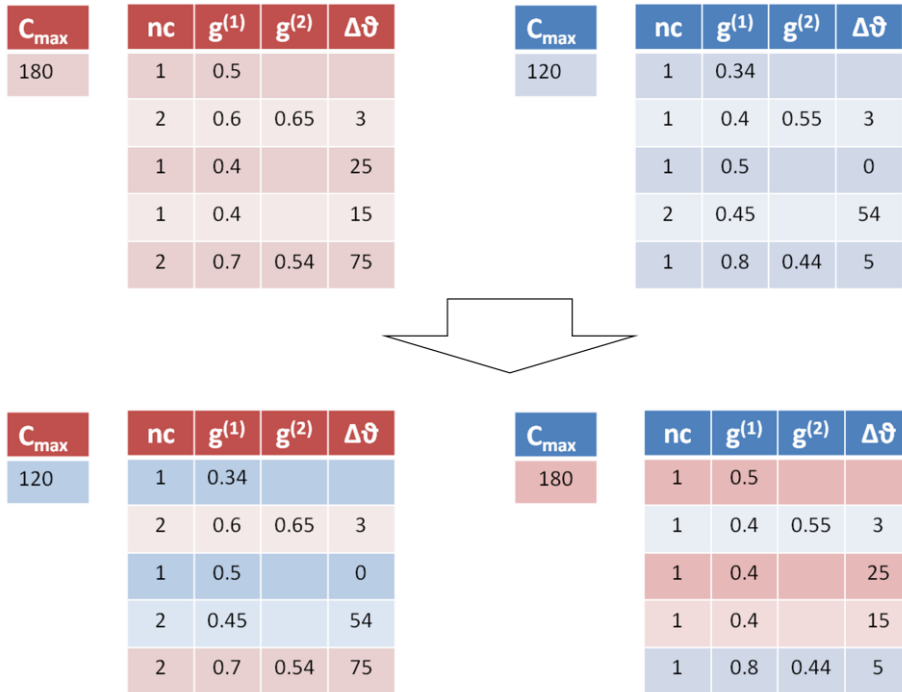


Figure 4-6 - Crossover of type 2

The third crossover method requires the choice of a synchronization cycle between one of the two parents to generate a child and the break of the artery at an intermediate point. The first child will be assigned the parameters of configuration of one parent for a segment of the road artery and those of the second parent for the other segment. The second child is created to be complementary to the first one.

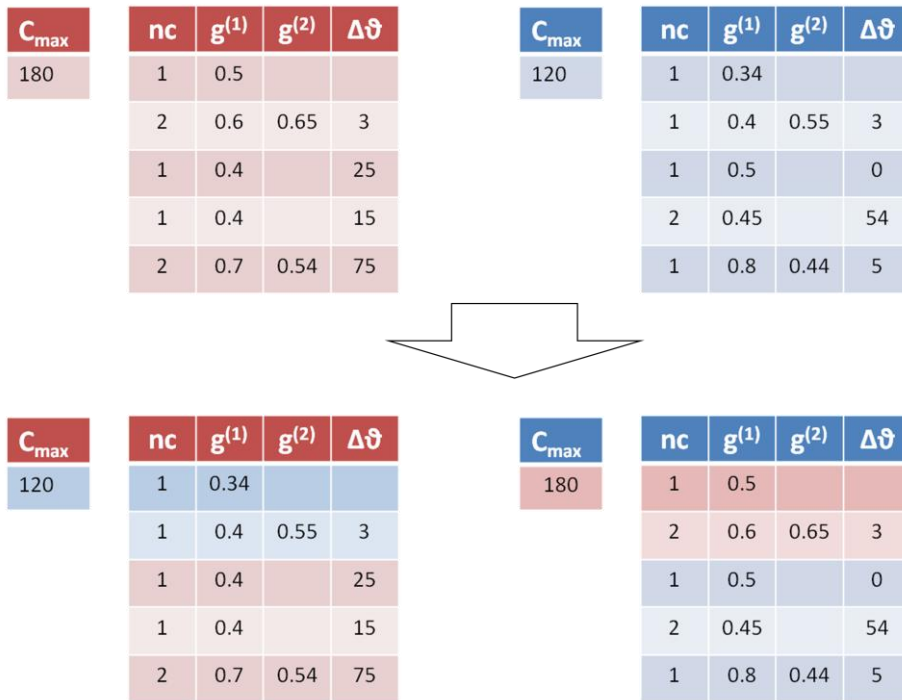


Figure 4-7 - Crossover of type 3

In this research work different tests have been carried out to choose the best method to use and in particular it was decided to aggregate the trend of the algorithm by applying it at 24 different instances and on the basis of the result it was chosen the method to use.

Numerical experiments have been conducted on four test networks considering different dimensions and characteristics as shown in table 2. All instances are major urban roads and were simulated during the peak hour.

Signals	Type	Capacity	Flow	Cross Flow	Way
4	Ring-road	High	High	High	Two

4+1 pedestrian	Radial-road	High	High	Low	Two
5	Ring-Road	High	High	High	One
8	Radial-Road	Very High	Very High	Low	Two

Table 4-1 - Characteristics of instances

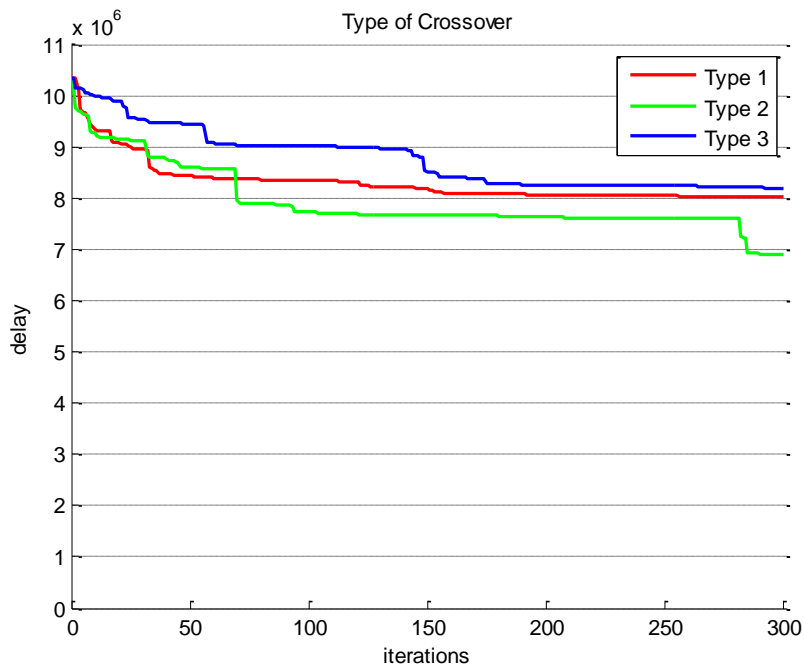


Figure 4-8 - Trend of the algorithm with the three crossover methods

From the simulations carried out it was decided to use the crossover of type 2, which enables the algorithm to find good solutions already after few iterations, to keep a better medium profile than the other two optimizations and to get the best solutions in the 24 instances analyzed.

Furthermore, an analysis has been carried out to decide the value that the *crossover rate* must assume in the interval [0.6,1]. Lower values have not been analyzed, since they would reduce the algorithm capacity to evolve.

The simulations show that the better results can be obtain with value 1 then the whole population have to be subject to crossover.

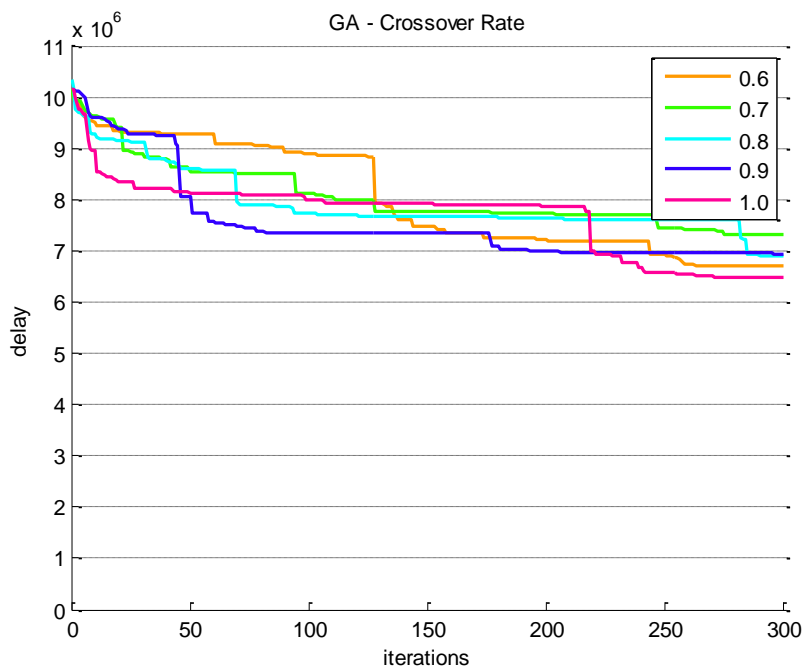


Figure 4-9 - Trend of the algorithm with different crossover rate

4.2.4 Mutation

The mutation operator changes casually one or more parameters of a solution and represents so the part of a casual research of the genetic algorithm, which enables to explore the space of research also far from the points in which the

population concentrates. The mutation operator is applied only with a certain probability, called *mutation rate*. A very high value can turn the algorithm into a simple casual research, cancelling the heuristic effects of the roulette wheel and the crossover. A very low mutation probability can indeed lead to a situation of premature convergence of the genetic algorithm in the neighborhood of local minimum for which there is no solution.

In order to avoid the situations of premature convergence and the development of an algorithm of casual research, several potential solutions have been examined in the implementation of the mutation. It was analyzed both the trend of the algorithm with the variation of the probability of mutation and the trend after the implementation of a model with a variable mutation rate.

The analysis of the possible values that the probability of mutation can assume has been carried out in the interval $[0.1, 0.5]$, whose range appears to be reasonable when taking into consideration what said previously.

In the case of variable mutation, it was calculated the number of iterations during which the best-fitness is not updated, called *stall iterations*. The probability of variable mutation adopted is comprised between a minimum value ($p_{min_{mut}}$) and a maximum value ($p_{max_{mut}}$), and is proportional to the number of *stall iterations*. The probability of mutation assumes the minimum value when the *stall iteration* is equivalent to 0, and the maximum value when it is equivalent to the *maxstall*. This approach enables the algorithm to benefit from the effects of the crossover and the mutation according to the situation of convergence. After an improvement of the best-fitness, the probability of mutation is brought back to the minimum value. This approach it was used in

the first release of this work (Colombaroni, Gemma and Fusco 2009) with good benefits especially with few iterations (Figure 4-12). After the improvement of the simulation model with the introduction of spillback and with the new crossover function, the results of variable mutation were not good as shown in Figure 4-11.

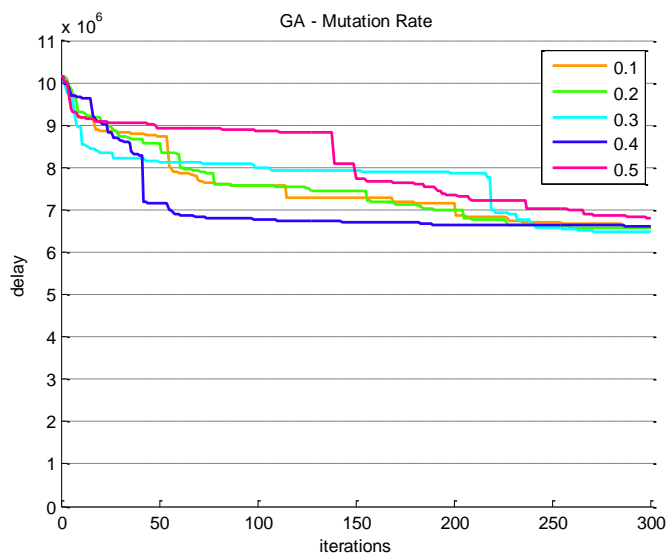


Figure 4-10 - Simulation for different mutation rate

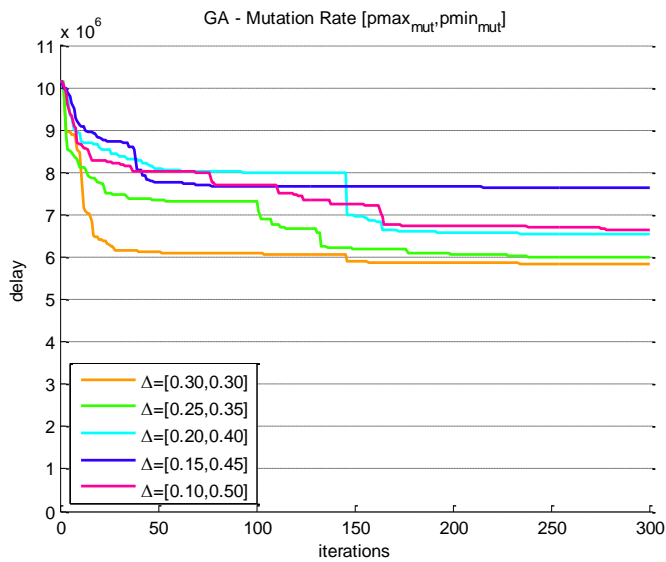


Figure 4-11 - Simulation for different interval $\Delta=[p_{\min_mut}, p_{\max_mut}]$

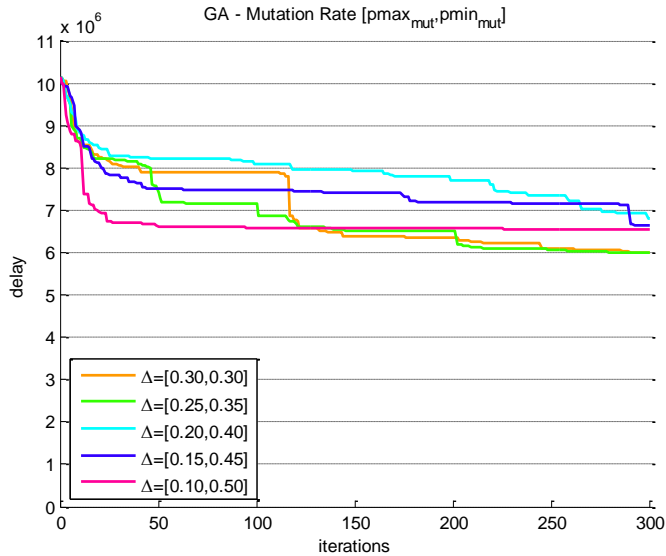


Figure 4-12 - Simulation for different interval $\Delta=[p_{\min_mut}, p_{\max_mut}]$ in Colombaroni, Gemma and Fusco (2009)

4.2.5 Elitism

During the phase of creation of a new population, the genetic algorithm in its standard formulation bequeaths fully to the next generation the genome of the best solution found until that moment (cloning). This concept has been extended in this work by introducing the possibility to clone the whole set of best solutions found until that moment. This set is called *elite*. The algorithm expresses the number of individuals belonging to the elite through a percentage of the total population. If the *elite rate* assumes too low values, no effect can be registered and the genetic inheritance of the best solutions can be easily lost. If on the contrary the values used are very high, the population risks not evolving. In this work it was decided to analyze only the effects of the elitism using the range $[0, 0.2]$ as the interval of definition of the elite percentage. The simulations show that the best value is 0.2. In Figure 4-13 it is possible to see how the elite rate gives a stabilizing effect because the good genome of the elite contributes to create better solution. This condition is rapidly decreasing the objective function.

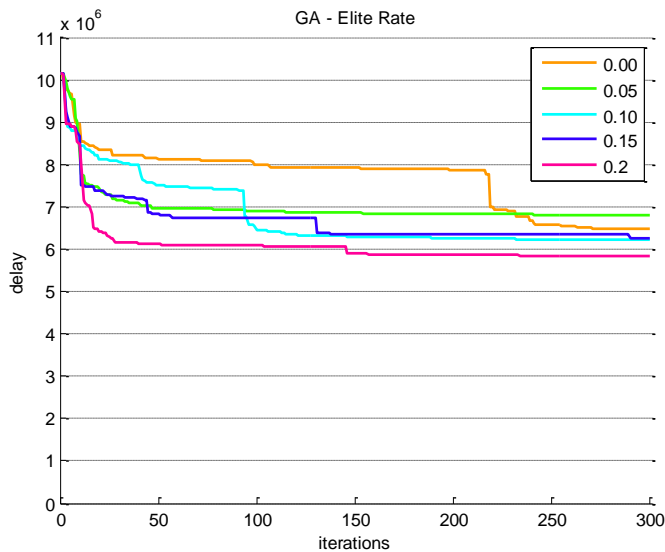


Figure 4-13 - Simulation for different elite rate

4.2.6 Final parameters

On the basis of the analyses described in the previous paragraphs, the configuration below was adopted for the genetic algorithm:

Number of iterations	300
Population Size	25
Crossovere Rate	1.0
Type of Crossover	2 (interchange of intersection)
Minimum Mutation Rate	0.3
Maximum Mutation Rate	0.3
Max Stall	20
Elite Rate	0.2

From these parameters it results a trend like the one below (Figure 4-14):

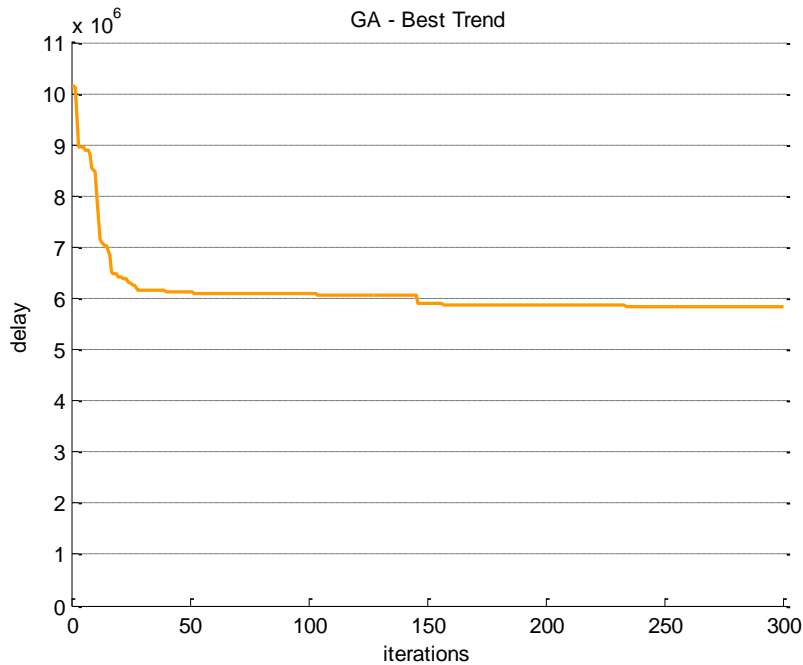


Figure 4-14 - Trend of the objective function for the genetic algorithm with best parameters

A further analysis has been carried out for the number of iterations as shown in paragraph 4.5.

4.3 Particle swarm optimization

The PSO is a heuristic optimization method employed to solve several optimization problems. The simplicity of the underlying concept, together with the capacity to adapt successfully to the various problems to which it has been applied, led to choose this algorithm as an alternative to the genetic algorithm described in the previous paragraphs. The origins of the PSO are rooted in the study of the social behaviors ruling the movements and

dynamics of groups of animals (flights of birds, shoals of fishes, swarms of bugs, etc.).

The most interesting aspect of these studies is the fact that the whole of the individuals moves and organizes in absence of a centralized control. In other words, the collective behavior is the result of simple local interactions and the single individual or agent does not have a global perception. This way, even with a low number of agents, wide spaces of research can be successfully examined, in order to focus then on a specific area, where one or more agents found a potential target according to the particular specifics of the fitness function.

The PS algorithm works as described below:

```
function PS()  
  for each agent  
    agent.pBest=max  
    agent.pBestSolution=null  
  end  
  do  
    for each agent  
      fit = FITNESS(agent)  
      if fit < agent.pBest  
        agent.pBest=fit  
        agent.pBestSolution=solution  
      end if  
    end for  
  
    gBest=the agent with best fitness  
  
    for each agent  
      CALCULATEVELOCITY(agent)  
      UPDATEPOSITION(agent)  
    end for  
  while not converge  
end function
```

From the simple description of how the algorithm works, you can easily understand how it can be applied to any issue for which a fitness function can

be defined. Moreover it is necessary to represent the solutions through a position vector and estimate a speed vector to move the solution within the space of research.

4.3.1 The agent

As well as for the genetic algorithm, for PSO the method of representation of a solution must be defined. And in particular, the solution must possibly have the features of a vector that can be compared to the position of the agent in the space of the solutions. The same structure used for the representation of the solution in the genetic algorithm demonstrated to be suitable also for the representation of an agent in the PSO.

$$\{C_1, \dots, C_n, g_1^{(1)}, \dots, g_n^{(1)}, g_1^{(2)}, \dots, g_m^{(2)}, \vartheta_2 - \vartheta_1, \dots, \vartheta_n - \vartheta_1\}$$

where:

- C_i is the cycle of node i
- $g_i^{(d)}$ is the green split of node i in direction d
- ϑ_i is the offset of node i

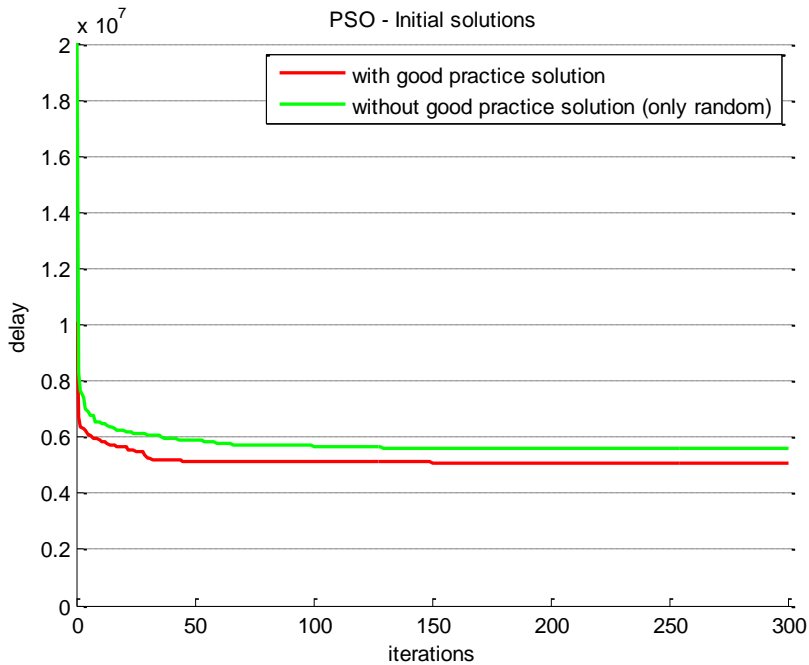


Figure 4-15 - Comparison between trend of algorithm with random initial population or with intelligent construction of initial solution

As well as in the genetic algorithm, also the choice of the initial population is essential and allows the algorithm to find better solutions as shown in (Figure 4-15).

4.3.2 The movement function

In order to adapt a problem to be solved with the PSO, apart from defining an appropriate data structure, a rule of agent movement must be formulated to explore the space of research. Please note the agent movement relative to two different variables in Figure 4-16.

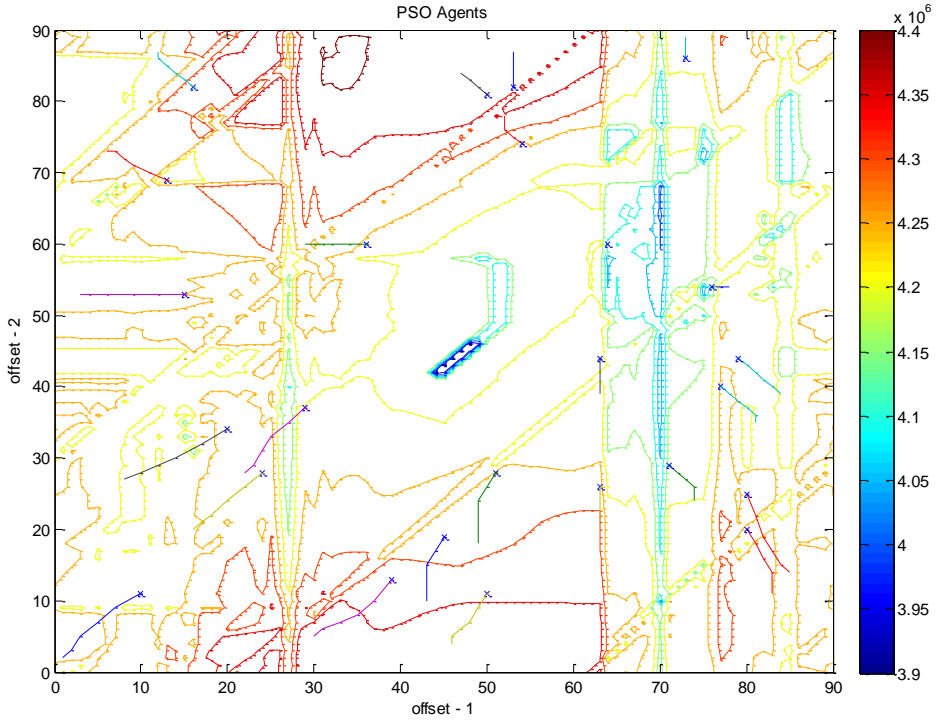


Figure 4-16 - Agent movement relative to two sizes in the space of research. Contour lines depict the values of the total delay functions.

The movement function for the PSO has been defined both with respect to the local minimum found by the agent and the global minimum found by the whole swarm:

$$\begin{aligned}
 R &= rnd() \cdot s_{\min} \\
 V_t^i &= \alpha \cdot V_{t-1}^i + \beta_L \cdot (b_L^i - X_{t-1}^i) + \beta_G \cdot (b_G - X_{t-1}^i) + \sigma \cdot R \\
 X_t^i &= X_{t-1}^i + \gamma \cdot V_{t-1}^i
 \end{aligned}$$

where:

- s_{\min} is the minimum movement vector;
- V_t^i is the speed of the agent i at the simulation instant t ;
- X_t^i is the position of the agent i at the simulation instant t ;
- b_L^i is the position of the best solution found by the agent i ;
- b_G is the position of the best solution found by all the agents;
- R is the random movement vector
- $\alpha, \beta_L, \beta_G, \sigma, \gamma$ are the weights of the different parts of function

Each weight has been gauged through a series of simulations in the neighborhood of some values found in the literature.

The parameters are calibrated in sequence as the following order:

1. Speed parameter α
2. Best local parameter β_L
3. Best global parameter β_G
4. Random parameter σ
5. Inertia parameter γ

After calibration of each parameter, this has been kept fixed for subsequent calibrations.

A variable random parameter has been tested to avoid premature convergence but the results was not good as shown in Figure 4-17. In particular, a multiplier for the random parameter is defined. This multiplier was applied to

each pre-defined number of iterations that do not improve the solution (Figure 4-17).

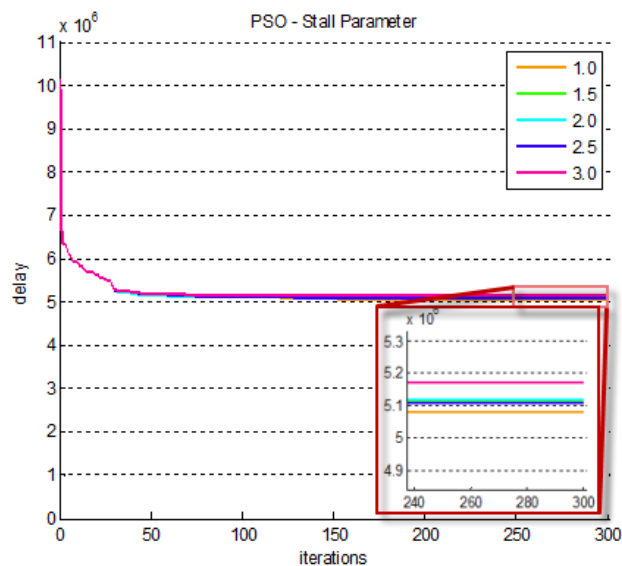


Figure 4-17 – Trend of PsO with different stall parameter

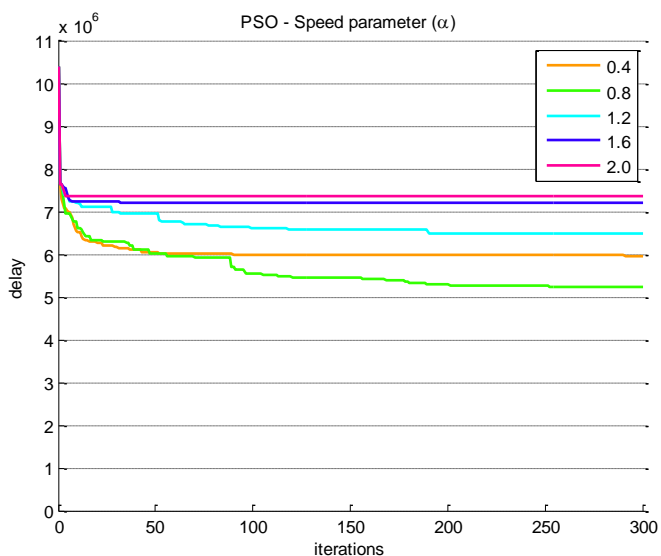


Figure 4-18 - Trend of the algorithm with different speed parameters

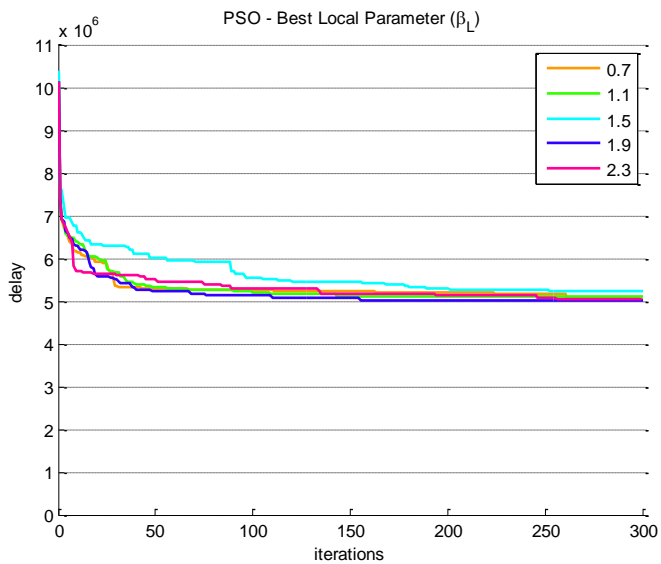


Figure 4-19 - Trend of the algorithm with different best local parameters

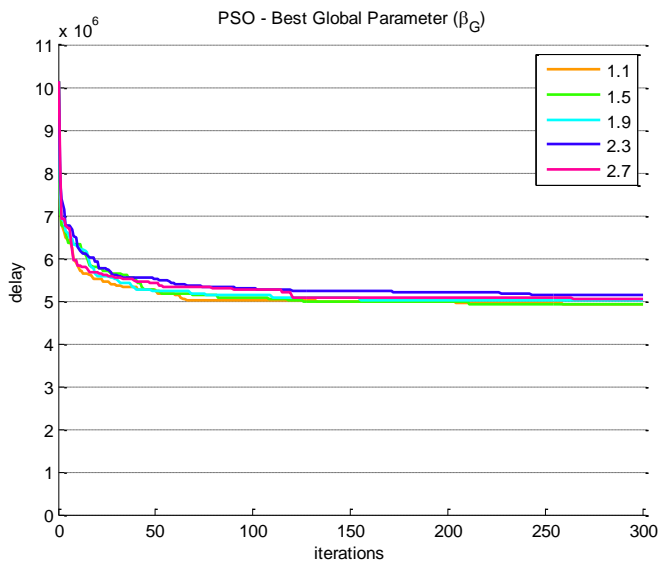


Figure 4-20 - Trend of the algorithm with different best global parameters

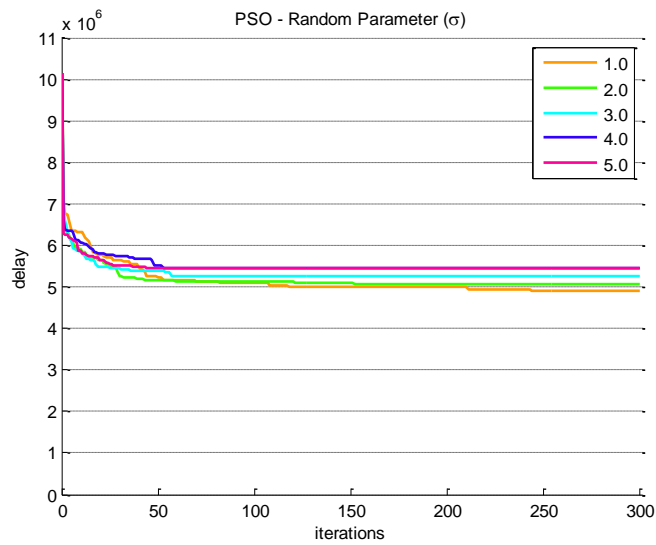


Figure 4-21 - Trend of the algorithm with different random parameters

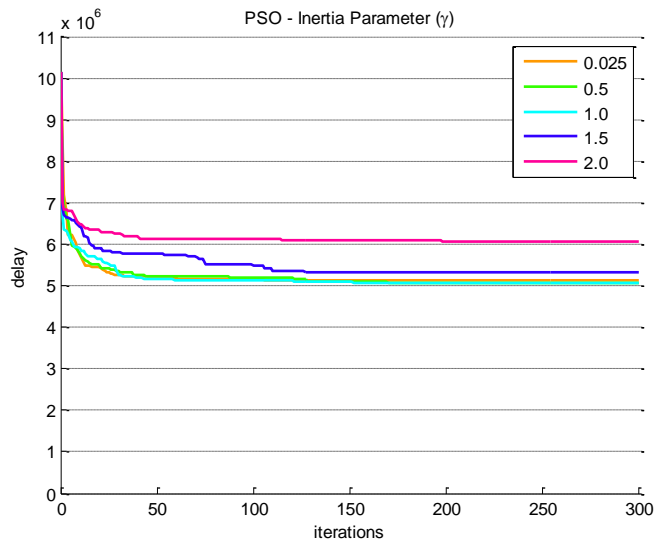


Figure 4-22 - Trend of particle swarm algorithm for different values of inertia parameter

4.3.3 Final parameters

On the basis of the analyses described in the previous paragraphs, the configuration below was adopted for the PSO algorithm:

Number of iterations	300
Number of agents	25
Speed parameter	0.8
Best local parameter	1.9
Best global parameter	1.5
Random parameter	1.0
Inertia parameter	1.0

From these parameters it results a trend like the one below (Figure 4-14):

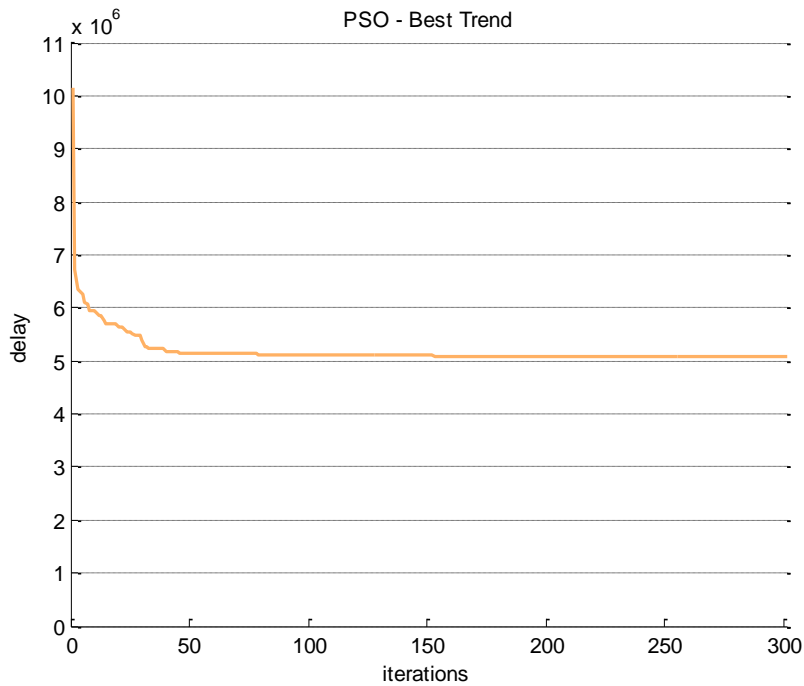


Figure 4-23 - Trend of the objective function for the particle swarm algorithm with best parameters

4.4 Hill-Climbing

On the basis of the analysis carried out for the two global optimization algorithms, a procedure of local adjustment of the solution has been applied in the same way as the model of Transyt. The algorithm developed is a *hill-climbing* algorithm, which carries out sequences of attempts to increase and then decrease the variables of the project.

For each variables of the model, the algorithm performs the same procedure:

Step 1a: increase of the variable of step ζ_1

$$x = x + \zeta_1$$

Step 1b: decrease of the variable of step ζ_1

$$x = x - \zeta_1$$

if the objective function improves with one of the two steps, then the change is kept, otherwise the previous value is restored

Step 2a: increase of the variable of the step $\zeta_2 < \zeta_1$

$$x = x + \zeta_2$$

Step 2b: decrease of the variable of the step ζ_2

$$x = x - \zeta_2$$

if the objective function improves, then the change is kept, otherwise the previous value is restored.

Step 3a: increase of the variable of the step $\zeta_3 < \zeta_2$

$$x = x + \zeta_3$$

Step 3b: decrease of the variable of the step ζ_3

$$x = x - \zeta_3$$

if the objective function improves, then the change is kept, otherwise the previous value is restored.

This procedure is applied to all the variables of the project and, if improvements are recorded for at least one of them, it is performed again for all the sizes, otherwise the algorithm ends.

The use of this procedure produced important results both if applied after a great number of iterations on the part of the global optimization algorithms and if applied forcing a premature interruption of the genetic algorithm or of the swarm algorithm. To reduce the calculation times, the results then were so good that they suggested the application of the global optimization algorithms for few iterations and then of the hill-climbing algorithm as shown in Figure 4-24.

The results are being compared with the pure "hill-climbing" algorithm that it shown more efficient but less effective.

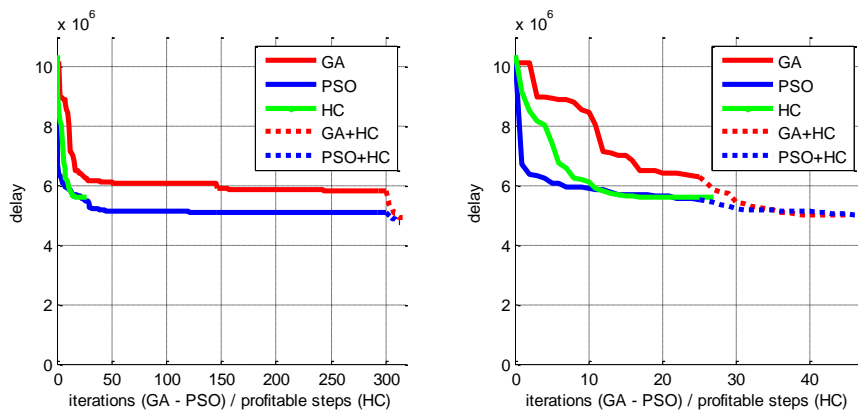


Figure 4-24 - Application of Hill-Climbing after 300 iterations or with premature stop at 50 iterations. The results are compared with pure Hill-Climbing algorithm starting from good practice solution.

4.5 Comparison

In comparing the two algorithms, the best for both speed of convergence that result was, without doubt, the POS as is shown in Figure 4-25. Although at the beginning of this work, the expected result was a substantial similarity between the two algorithms, the best performance of the PSO are due, presumably, to a strong effect of the movements function in the vectorial exploration of the solution space compared to the crossover and mutation operators that move with more discontinuity.

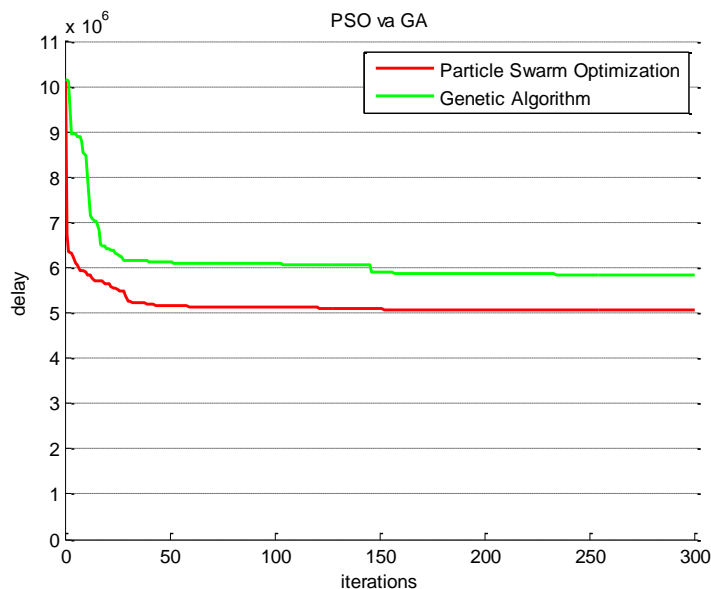


Figure 4-25 - Comparison between PSO and GA

This difference is very abided by the use of Hill-Climbing that is shown a good strategy to reduce the computation time and get a good solution anyway (Figure 4-24). Although the hill-climbing reduces the difference between the

two algorithms, the PSO algorithm is still better in terms of execution time. In fact, the average solutions analyzed the genetic algorithm are worse than those analyzed by the particle swarm optimization. Worst traffic light synchronization require a greater number of platoons to be simulated by reducing the overall performance. As shown in Figure 4-26, the average fitness is significantly different between the two algorithms.

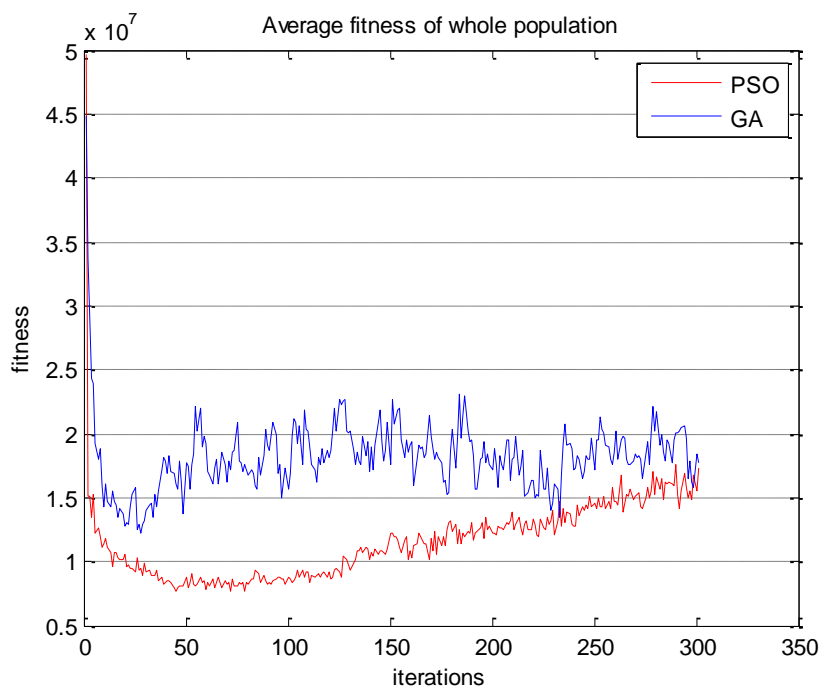


Figure 4-26 - Average Fitness

4.6 Performace

The model and optimization algorithms have been developed entirely in Matlab. The reason for this choice was the flexibility of this language in the development of mathematical models and simulation.

This approach has reduced considerably the development and testing time of the algorithm at the expense of longer simulation time. Although this option the processing time has been contained in the order of a few seconds for instances more complex (Figure 4-27).

The overall performance was about 500 platoons per second on notebook i7 M620 with 4GB Ram on Windows 7 64bit. A partial implementation of the algorithm was done in C#. NET 2.0 to quickly assess the performance on a compiled language. One result has been obtained indicative of about 80,000 platoons per second.

To compare this model with other simulation models consider that on average, in the simulations carried out, a platoon consists of 8 vehicles. For example, an instance of an application 5 intersections with average traffic flow of 2000 vehicles per hour is simulated with 540 platoons in little more than a second..

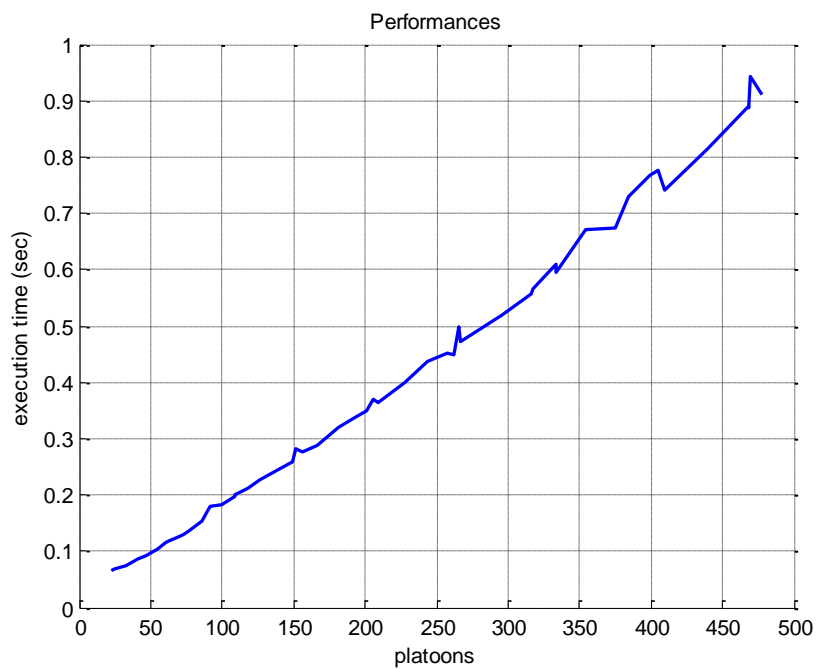


Figure 4-27 - Execution time

5 Evaluation of results on real case

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 40%, 22% and 23% in high, average and low demand level, respectively (see Figure 5-1). Bus priority was not considered in this test.

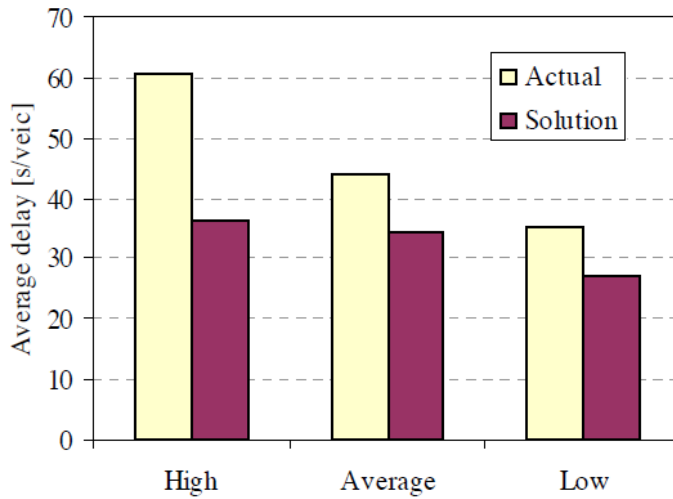


Figure 5-1 - Average unitary delays at nodes on the whole artery for 3 demand levels (high, average, low)

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 5-2).

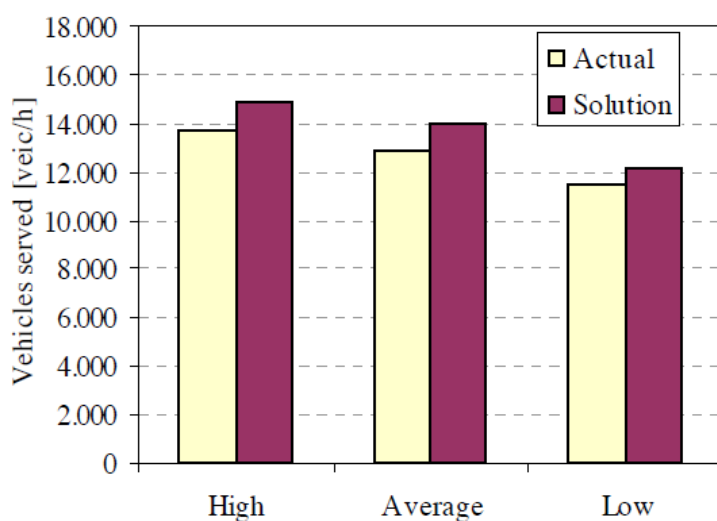


Figure 5-2 - Total vehicles served in simulation for the 3 demand levels (high, average, low)

Figure 5-3 shows the average unitary intersections delay at signals, computed in microsimulation for the actual scenario and optimal solution, for the highest demand level.

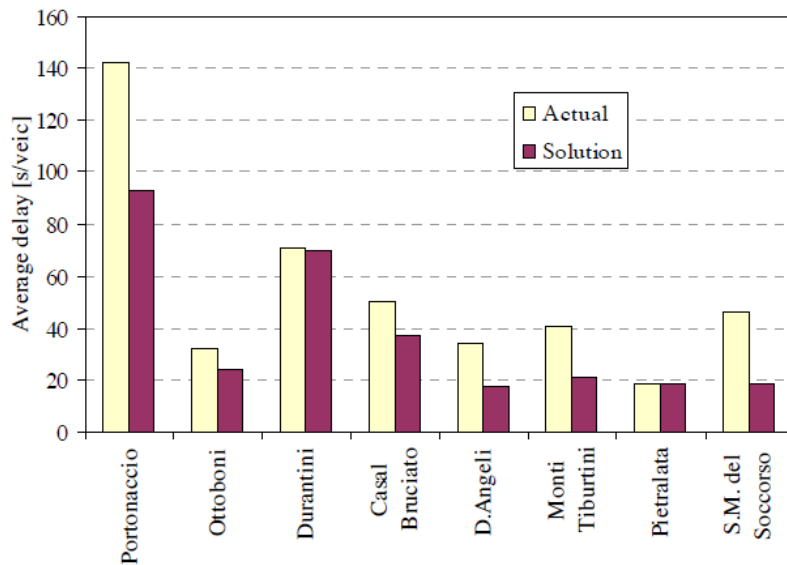


Figure 5-3 Average unitary delay at each node for the high demand scenario

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.

The results obtained, about the public transport, are summarized in Figure 5-4, which shows total delay and transit delay under the following scenarios:

- no project;
- traffic signal synchronization without any transit priority;
- traffic signal synchronization with passive transit priority;
- traffic signal synchronization with active transit priority;
- optimization of the total weighted delay with active transit priority.

The optimization procedure provides noticeable reduction of delay: 55% for car traffic and about 80% for transit with respect to the no project scenario. It's interesting to note that the traffic synchronization with transit priority (without optimization of the total delay) provides the lowest delay for buses and degrades the solution optimized for the whole traffic as 10%.

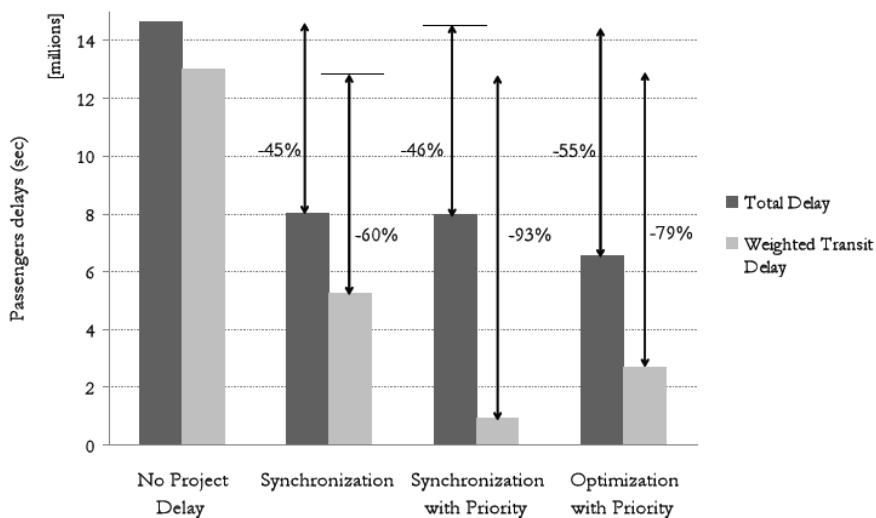


Figure 5-4 - Optimization results

6 Conclusion

At the end of this research a new model has been developed for calculating the delay along an urban road artery particularly. The model has a linear complexity respect to the number of simulated vehicles. The models analyzed and reported in the literature are, or too complex analyzing in detail the phenomena of vehicular flow, or very simplified since it ignores certain aspects of the traffic that can be analyzed with a simulation. However, microsimulation models are not suitable for signal setting design. On the other hand, PASSER is much simpler than the model presented here, which is about as complex as TRANSYT. Anyway, TRANSYT is less detailed in representing oversaturation. Finally, SYNCHRO traffic model is not documented in the literature.

The model created was inspired by the very simplified algorithm proposed by Papola and Fusco, but at the same time improves it introducing the dynamics of a simulation model and the capability to simulate conditions of oversaturation while maintaining the high performance of initial model. This result was obtained mainly with creating the push-back model for calculating the spillback, technique that is not present in the literature that allows at the algorithm of progression platoons to operate using the compaction rules that make it particularly efficient.

The model was used as a fitness function in different optimization procedures that were calibrated to obtain best results with respect of performance and goodness of solutions.

The solution of the optimization was implemented to signal settings of via Palmiro Togliatti, a 18-junction in Rome. Solutions of via Tiburtina and via Nomentana were only tested in microsimulation and not yet applied. The results have obtained the approval of ATAC (Mobility agency of Rome) that thank to have financed the initial part of this research.

The contribution to research of this work is based on improving of the accuracy of the progression platoons algorithm that made it mature enough to be applied to real cases. Further improvements can be made to this model to be applied to networks or to increase the detail of simulation with maneuvers and interaction between lanes. This certainly would require a continuation of research done in this work that the author of this text proposes to do.

Bibliography

Abu-Lebdeh G. and Benekohal. R.F. Design and Evaluation of Dynamic Traffic Management Strategies for Congested Conditions [Journal] // Transportation Research 37A. - 2003. - pp. 109–127.

Boillot F. [et al.] Optimal signal control of urban traffic networks [Conference] // 6th International Conference on Road Traffic Monitoring and Control,. - 1992. - Vol. 355. - pp. 75-79.

Boillot F., Midenet S. and Pierrele J.C. The real-time urban traffic control system CRONOS: Algorithm and experiments [Journal] // Transportation Research Part C 14. - 2006. - Vol. 14. - pp. 18-38.

Caliper Transmodeler Traffic simulation Software User's Guide. Version 1.5. Caliper Co. [Journal] // Transpn. Res. Rec.. - 2007.

Cascetta E, Gallo M. and B Montella Models and algorithms for the optimization of signal settings on urban networks with stochastic assignment models [Journal] // Ann Oper Res. - 2006. - Vol. 144. - pp. 301–328.

Cascetta E. Teoria e metodi dell'ingegneria dei sistemi di trasporto [Book]. - [s.l.] : UTET, 1998.

Ceylan H. A genetic algorithm approach to the equilibrium network design problem // Ph.D. Thesis, University of Newcastle upon Tyne. - 2002.

Chang E.C. P. [et al.] MAXBAND-86: program for optimizing left turn phase sequence in multiarterial closed networks [Journal] // Transportation Research Record. - 1988. - Vol. 1181. - pp. 61-67.

Chang T. and Sung G. Modeling and Optimization of an Oversaturated Signalized Network [Journal] // Transportation Research - Part B. - 2004. - Vol. 38. - pp. 687-707.

Chaudhary N. A. and Messer C. J. PASSER-IV: A program for optimizing signal timing in grid networks [Conference] // 72nd annual meeting of the Transportation Research Board. - Washington, DC : [s.n.], 1993.

Chen Juan and Xu Lihong Road-Junction Traffic Signal Timing Optimization by an adaptive Particle Swarm Algorithm [Conference] // International Conference on Control Automation Robotics and Vision 2006 ICARCV 06 9th. - [s.l.] : Ieee, 2006. - pp. 1-7.

Chiou S. W. Optimization of Area Traffic Control for Equilibrium Network Flows [Journal] // Transportation Science 33. - 1999. - pp. 279-289.

Cohen S.L. and Liu C.C. The Bandwidth Constrained TRANSYT Signal-Optimization Program [Journal] // Transpn. Res. Rec.. - 1986. - Vol. 1057. - pp. 1-7.

Cohen S.L. Concurrent Use of MAXBAND and TRANSYT Signal Timing Programs for Arterial Signal Optimization [Journal] // Transpn. Res. Rec.. - 1983. - Vol. 906. - pp. 81-84.

Colombaroni C., Gemma A. and Fusco G. A Model and an Algorithm for Signal Synchronization and Bus Priority on Urban Arteries [Conference] // Models and Technologies for Intelligent Transportation Systems. - Rome, Italy : [s.n.], 2009.

Colombaroni C., Gemma A. and Fusco G. Joint Problem of Traffic Signal Synchronization and Bus Priority [Conference] // Seventh Triennial Symposium on Transportation Analysis. Tristan VII. - Tromso, Norway : [s.n.], 2010.

Colombaroni C., Gemma A. and Fusco G. Optimization of Traffic Signals on Urban Arteries through a Platoon-Based Simulation Model [Conference] // 11th Int. Conf. "Automatic Control, Modelling and Simulation". - Istanbul, Turkey : [s.n.], 2009.

Davol A.P. Modeling of Traffic Signal Control and Transit Signal Priority Strategies in a Microscopic Simulation Laboratory. - [s.l.] : PhD Thesis, Massachusetts Institute of Technology, 2001.

Dazhi S., Benekohal R.F. and Waller S.T. Bilevel Programming Formulation and Heuristic Solution Approach for Dynamic Traffic Signal Optimization [Journal] // Computer-Aided Civil and Infrastructure Engineering. - 2006. - Vol. 21. - pp. 321-333.

De Schutter Bart Optimizing acyclic traffic signal switching sequences through an Extended Linear Complementarity Problem formulation [Journal] // European Journal of Operational Research. - 2001. - 139. - pp. 400–415.

Diakaki C., Papageorgiou M. and Aboudolas K. A multivariable regulator approach to traffic-responsive networkwide signal control [Journal] // Control Engineering Practice. - 2001. - Vol. 10. - pp. 183-195.

Doherty R.A. A Comprehensive Junction Delay Formula [Report]. - [s.l.] : Department of Transport, UK, 1977.

Dong Chaojun, Huang Shiqing and Liu Xiankun Urban Area Traffic Signal Timing Optimization Based on Sa-PSO [Conference] // International Conference on Artificial Intelligence and Computational Intelligence (AICI), 2010. - Sanya : [s.n.], 2010. - pp. 80 - 84 .

Farges Jean Loup, Hanry J.J. and Tuffal J. The PRODYN real-time traffic algorithm [Conference] // Proceedings of the fourth IFAC symposium on transportation systems. - Baden-Baden, Germany : [s.n.], 1983. - pp. 307-312.

Florian M., Mahut and Tremblay N. A simulation-based dynamic traffic assignment model: Dynameq [Conference] // Proceedings of the First International Symposium on Dynamic Traffic Assignment DTA2006. - [s.l.] : Institute for Transport Studies. University of Leeds, 2006.

Gartner H.N. OPAC: A Demand-Responsive Strategy for Traffic Signal Control [Journal] // Transportation Research Record. - 1983. - Vol. 906. - pp. 75–81.

Gartner N. and Al-Malik M., Combined model for signal control and route choice in urban traffic networks [Journal] // Transportation Research Record. - 1996. - Vol. 1554. - pp. 27–35.

Gartner N.H and Hou G.L. Performance Evaluation of Multi-Band Progression Method [Conference] // 7th IFAC/IFORS Symposium on "Transportation Systems: Theory and Application of Advanced Technology". - Tianjin, China : [s.n.], 1994. - pp. 981-986.

Gartner N.H. Area traffic control and network equilibrium methods. - 1974.

Gartner N.H. Development and testing of a demand responsive strategy for traffic signal control [Conference] // Proceedings of American Control Conference. - 1982. - pp. 578–583.

Gartner N.H., Messer C.J. and Rath A.K. Traffic Flow Theory - A State of the Art [Report]. - Washington, DC : Transportation Research Board, 1999.

Gartner N.H., Tarnoff P.J. and Andrews C.M. Evaluation of optimized policies for adaptive control strategy [Journal] // Transportation Research Record. - 1990. - 1324. - pp. 105-114.

Gazis D.C. and Potts R.B. The oversaturated intersection [Conference] // Proceedings of the second international symposium on traffic theory. . - London, UK : [s.n.], 1963. - Vols. 221–237.

Gerlough D. L. and Huber M. J. Traffic Flow Theory - A Monograph [Journal] // Transportation Research Board. - 1975.

Hadi M.A. and Wallace C.E. Algorithm to Optimize Signal Phasing and Timing [Journal] // Transpn. Res. Rec.. - 1993. - Vol. 1421.

Huang Huixian and Yue Wen Long Real-time traffic signal control in urban intersection with PSO algorithm [Conference] // 10th International conference on application of advanced technologies in transportation AATT 2008. - Athens : AATT 2008, 2008. - pp. 1-11.

Hunt P.B. [et al.] SCOOT, A Traffic Responsive Method of Co-Ordinating Signals [Report] : Report 1014 / TRRL Laboratory. - 1981.

Lawphongpanich S. and Hearn D.W. Simplicial Decomposition of the Asymmetric Traffic Assignment Problem [Journal] // Transportation Research B. - 1984. - Vol. 17. - pp. 123-133.

Lee C. and Machemehl R.B. Genetic algorithm, local and iterative searches for combining traffic assignment and signal control. Traffic and transportation studies. [Conference] // Proceedings of ICTTS 98. - 1998. - pp. 489-497.

Liao C.F. and Davis G.A. Simulation Study of Bus Signal Priority Strategy, Taking Advantage of Global Positioning System, Automated Vehicle Location System, and Wireless Communications [Journal] // Transportation Research Board. - 2007. - Vol. 2034. - pp. 82-91.

Little J. D. C., Kelson M. D. and Gartner N. H. MAXBAND: a program for setting signals on arterials and triangular networks [Journal] // Transportation Research Record. - 1981. - Vol. 795. - pp. 40-46.

Lo K.H., Chang E. and Chan. Y.C. Dynamic Network Traffic Control [Journal] // Transportation Research 35A. - pp. 721-744.

Mahmassani H.S., Peeta S., Hu and T.Y. Ziliaskopoulos, A.K. Dynamic traffic assignment and simulation procedures for ATIS/ATMS applications [Report] / Center for Transportation Research ; The University of Texas. - Austin : [s.n.], 1992.

- Mahut M.** Discrete flow model for dynamic network loading. - [s.l.] :
Département d'informatique et de recherche opérationnelle, Université de
Montréal. Published by the Center for Research on Transportation (CRT),
Université de Montréal, 2000.
- Malakapalli M.P. and Messer C.J.** Enhancements to the PASSER II-90 Delay
Estimation Procedures [Journal] // Transpn. Res. Rec.. - 1993. - Vol. 1421. -
pp. 94-103.
- Mauro V. and Di Taranto C.** UTOPIA [Conference] // Proceedings of the 6th
IFAC/IFIP/IFORS Symposium on Control, Computers, Communications on
Transportation, Paris. - 1989. - pp. 245-252.
- Mauro V.** Road Network Control [Book Section] // Concise Encyclopedia of
Traffic and Transportation Systems. - 1991.
- McDonald M. and N.B. N.B. Hounsell** Road Traffic Control: Transyt and
SCOOT [Book Section] // Concise Encyclopedia of Traffic and Transportation
Systems. - 1991.
- McTrans-Center** Traffic Network Study Tool - Transyt7F. United States
Version [Book]. - 2006.
- Michalopoulos P.G. and Pisharody V.B.** Derivation of delays based on
improved macroscopic traffic models [Journal] // Transportation Research
Part B. - 1998. - Vol. 15. - pp. 299–317.
- Michalopoulos P.G., Stephanopoulos G. and Pisharody V.B.** Modeling of
traffic flow at signalized links [Journal] // Transportation Science. - 1980. - pp.
9–41.
- Mirchandani P. and Head L.** A Real-Time Traffic Signal Control System:
Architecture, Algorithms and Analysis [Journal] // Transportation Research
C. - 2001. - Vol. 9. - pp. 415–432.

Newell G.F. A simplified theory of kinematic waves in highway traffic [Journal] // Transportation Research B. - 1993. - Vol. 27B(4). - pp. 281-313.

Optimal Signal Timing for an Oversaturated Intersection [Journal] // Transportation Research, 34B. - 2000. - pp. 471-491.

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

Papola N. and Fusco G. Relationship between Travel Time Minimization and Bandwidth Maximization Problems [Conference] // Proc. of TRISTAN III Conference, San Juan, Puerto Rico, June 17-23. - 1998.

Park B., Messer C.J. and Urbanik-II T. Traffic Signal Optimization for Oversaturated Conditions: Genetic Algorithm Approach [Journal] // Transpn. Res. Rec.. - 1999. - Vol. 1683. - pp. 133-142.

Pillai R. S., Rathi A. K. and L. Cohen S. A restricted branch-and-bound approach for generating maximum bandwidth signal timing plans for traffic networks [Journal] // Transportation Research B. - 1998. - 8 : Vol. 32. - pp. 517-529.

Qin Zhong [et al.] Approach of intersection signal control based on PSO [Conference] // CCDC'09 Proceedings of the 21st annual international conference on Chinese control and decision conference. - Guilin, China : IEEE Press, 2009. - pp. 4457-4461.

Rakha H.A. A Simulation Approach for Modeling Real-Time Traffic Signal Controls // Queen's University of Kingston, Canada. - 1993.

Robertson D.I. Transyt: a Traffic Network Study Tool [Report] : RRL Report LR253 / Ministry of Transport. - Crowthorne, Berkshire, United Kingdom : [s.n.], 1969.

- Sen S. and Head K.L.** Controlled optimization of phases (COP) at an intersection [Journal] // Transportation Science. - 1996.
- Sheffi Y. and Powell W.** Optimal signal settings over transportation networks [Journal] // Journal of Transportation Engineering. - 1983. - Vol. 106. - pp. 824–839.
- Sheffi Y.** Urban Transportation Networks: Equilibrium analysis with mathematical programming methods [Book]. - [s.l.] : Prentice-Hall, Inc., 1985.
- Siljak D. D.** Decentralized control of complex systems. // York: Academic Press. - 1991.
- Skabardonis A.** Control Strategies for Transit Priority [Conference] // 79th Annual Meeting of the Transportation Research Board. - 2000.
- Stamatiadis C. and Gartner N. H.** MULTIBAND-96: a program for variable-bandwidth progression optimization of multiarterial traffic networks [Journal] // Transportation Research Records. - 1996.
- Stevanovic J. [et al.]** Stochastic Optimization of Traffic Control and Transit Priority Settings in VISSIM [Journal] // Transportation Research Part C 16. - 2008. - pp. 332-349.
- Tsoulos Ioannis G. and Stavrakoudis Athanassio** Enhancing PSO methods for global optimization [Journal] // Applied Mathematics and Computation. - [s.l.] : Elsevier Science, 2010. - Vol. 216. - pp. 2988-3001.
- Wei Yun [et al.]** Intersection Signal Control Approach Based on PSO and Simulation [Journal]. - [s.l.] : IEEE Computer Society.
- Yin Y.** Genetic-algorithm-based approach for bilevel programming models [Journal] // Journal of Transportation. - 2000. - Vol. 126. - pp. 115–120.