INTEGRATING COMPUTATIONAL INTELLIGENCE, MICROSIMULATION AND SEMAPHORIC REGULATION INTO CONVENTIONAL GIS SOFTWARE

Fernando Bernardes de Oliveira, fernando@lsi.cefetmg.br

Paulo Eduardo Maciel de Almeida, pema@lsi.cefetmg.br

Gray Farias Moita, gray@dppg.cefetmg.br

Federal Center for Technological Education of Minas Gerais State (CEFET-MG), Brazil

ABSTRACT

Traffic flow management and control has become a serious problem for large cities around the world, because of the increasing amount of circulating vehicles and the lack of infrastructure and modern technology available to traffic agencies. An efficient semaphore planning strategy can guarantee a safe and regular vehicle flow, avoiding traffic jam and delays. Besides that, it can contribute to safe transportation, to pollution emission minimization and to drivers and pedestrians' satisfaction. However, semaphoric regulation is a difficult task to accomplish, mainly if a specific region has high vehicle flow rates and a lot of semaphores to be simultaneously regulated. Moreover, conventional techniques have become obsolete and are very ineffective to modern demands and actual traffic conditions. This paper focuses the development of an integrated software environment, composed by a GIS module, a computational intelligence based semaphoric regulation tool and a microsimulation engine, all of them encapsulated into an open architecture platform to perform, validate and test semaphoric planning. GIS module allows importing of existing traffic network digital documents into the software environment, selecting of specific regions inside a city to be studied and execution of effective simulation without needing to redraw street map and traffic details. To perform semaphore regulation, an immune system inspired on clonal selection principle, called CLONALG (CLONal selection ALGorithm), is applied and effectively tested. Experiments were undertaken with actual data collected from a region downtown Porto Alegre, Brazil. Obtained results from the simulations are evaluated and compared to similar results from literature. Effectiveness of the integrated tool and the intelligent technique to solve the proposed approach are presented and discussed showing that the overall technique has good potential to solve practical traffic problems.

1 – INTRODUCTION

Nowadays, a major problem currently faced in large cities is the traffic control. This task is not trivial and becomes even more difficult due to the increasing number of vehicles and pedestrians. The city of Sao Paulo, for example, according to DETRAN-SP (2009), reached in May 2009 a fleet of more than 6.5 Million vehicles. This growth in cities, in general, involves expansion and maintenance of traffic routes and setting policies and actions for an effective traffic management.

An important element in traffic management is the traffic control system, which in addition to controlling the conflicting movements at an intersection, can ensure that the vehicle flow is safe and steady, avoiding traffic jams, too many stops and high consumption of fuel, among other problems. In this context, efficient traffic engineering is essential, being able to mitigate the impact of uncontrollable city growth. To this end, practical, fast and efficient solutions are needed for traffic planning and management, in order to ensure proper flow and safety.

Geographic Information Systems (GIS) allow the modeling, processing and analysis of spatial information. Foote and Lynch (1998) cited by Silva (2006) claim that GIS provide powerful tools for dealing with geographical and environmental issues. These systems allow, through the use of a computer, for organizing information about a particular region as a set of maps, each one displaying specific information about a feature of the region under study. An example of such systems is the OpenJUMP (OpenJUMP, 2009), which is a free and open source GIS development platform, written in Java (Sun, 2009). It is an extensible application, which allows the inclusion of new features by connecting modules (plugins) developed by the user. With the support of GIS, Saliba Neto (2009) developed an open source tool for microscopic simulation of urban traffic, where you can reproduce actual traffic situations and the results are used to aid decision making by traffic engineers. As the traffic control system has an important role in the management of traffic, we used the characteristics of OpenJUMP and the simulator developed by Saliba Neto (2009), to build a tool capable of generating traffic semaphoric timing by means of computational intelligence techniques. The use of GIS prevents road networks having to be redrawn in other simulators, reducing the possibility of error and rework, besides using the possibility of applying the techniques of analysis and evaluation of geographic information present on that platform. The timing generation tool here presented is also extensible enough to allow for the development and implementation of various computational intelligence techniques.

Several studies involving the definition of traffic lights timing can be found in the literature. Among these, some can be briefly discussed because of their relevance and distinct manner in which the problem has been addressed. Oliveira et al. (2005) performed the dynamic coordination of traffic lights in a real region of Porto Alegre (RS). Sanchez et al. (2008) applied an evolutionary technique based on genetic algorithms for optimization of traffic lights in an actual test case. Lämmer and Helbing (2008), inspired by the observation of oscillations of self-organization of the flow of pedestrians in traffic jams, proposed a model for self-traffic control system.

This article aims to present the effectiveness of integrating a tool, TL-GISSIM, which is responsible for generating traffic timing, with an algorithm inspired by artificial immune systems, specifically the principle of clonal selection, called CLONAL selection algorithm (CLONALG). TL-GISSIM was incorporated into the GIS platform OpenJUMP and an extension to simulate traffic in microregions, the GISSIM, which retrieves traffic networks directly from the GIS, with no need for redrawn. Practical results are evaluated by means of experiments and comparisons with the literature.

This paper is organized as follows. In section 2, we present criteria and approaches for definition of traffic light timing, both through conventional techniques and computational intelligence, and the algorithm used in this work. Section 3 presents a brief description of GIS, the simulator used and the time generator here developed. In section 4, practical experiments are described and compared with the technical literature. Final discussions are presented in section 5.

2 –SEMAPHORE TIME AND COMPUTATIONAL INTELLIGENCE

According to the Manual of Traffic Lights from the National Traffic Department at Brazil (DENATRAN, 1984), the performance of the traffic in terms of flow and safety, is directly related to the regulation of traffic on the existing road system, which basically means defining an optimum cycle time of the intersection, calculating green times for each phase, for the adopted cycle, and calculating the gaps, if necessary.

This book uses, as a method for regulation of isolated traffic lights, the Webster method. As quotes DENATRAN (1984), this method was chosen because it is a complete and comprehensive method that enables the determination of green times and cycle length, causing the least possible delay in the overall intersection. This book, although old, is still in use and is the official reference for calculating semaphore timing in Brazil.

As described above, this approach involves the regulation of traffic lights alone. For the adjustment with more lights and intersections, the Webster method can produce unsatisfactory results, affecting the overall traffic flow and safety. This problem can also be observed when there is variation in the flow of traffic routes throughout the day.

In this respect, the use of modern computer resources to traffic engineering becomes more efficient and essential. One of these resources is a traffic simulator, with characteristics of analysis and assessment of traffic situations, the definition of semaphoric timing based on traffic flow, traffic planning, among others. As finding the exact resolution of problems in this area, such as setting the timing for traffic lights, is a difficult task, the use of computational intelligence techniques has become more and more common, providing practical and reliable solutions. Thus, the following discusses some relevant work that used these techniques to resolving issues in traffic engineering.

Rouphail et al. (2000) researshed a small area of urban traffic, with nine signalized intersections, in the city of Chicago (USA). In this study, the authors carried out timing optimization through TRANSYT-7F software (T7F) and compared the results with a Genetic Algorithm (GA) developed and implemented in only one computer. The criteria for measuring efficiency were the delays in the link and the total queue amount of the network. To simulate traffic, along with the evolutionary algorithm, the authors used CORSIM, which is a microscopic simulation model, applied to traffic lanes. The GA generates the input data for this model, which performs the simulation and returns the result to evaluate the GA fitness. According to the authors, this process slows down the convergence of the algorithm and therefore causes a delay in obtaining good results.

Sanchez et al. (2008) applied an evolutionary technique based on genetic algorithms for optimization of traffic lights in an actual test case. They used a simulation model based on cellular automata. The process was run in parallel on a Beowulf cluster with five nodes (master and slaves), which is thought to make the system scalable. According to these authors, after some tests they decided to employ, as the evaluation function, the absolute number of vehicles leaving the traffic network. They also claim that the timing can be improved with this simulation environment, thereby reducing overall travel time. In this work, the study area had to be discretized for use in the adopted simulation model, which resulted in 1643 cells, 42 lights, 26 entry points and 20 exit points for vehicles. This discretization, of course, required a reasonable time, which makes it impossible or very difficult to apply this technique in other regions.

The approach defined by Lämmer and Helbing (2008) was inspired by the observation of oscillations of self-organization of the flow of pedestrians in traffic jams. From this observation, they derived a model for the self-traffic control system. The problem was treated as multi-agent, with interactions between vehicles and traffic lights. The authors assume the priority control of traffic lights by the flow of vehicles, considering the short term in prediction of this flow and platoons. Similarly, some laws have been borrowed from the area of electrical circuits to model the problem, as the Kirchhoff's Law of Currents, considering the conservation of flow (vehicles) at the nodes (intersections). In their results, they considered not only the reduction of average travel times, as well as their variations. They believe that this approach could also be applied to logistics and production processes. The simulation model is simplistic, considering at intersections that all streams are incompatible by default, and only one direction of movement is allowed at each specific time.

Oliveira et al. (2005) performed the dynamic coordination of traffic lights based on two approaches: Social Insects (Swarm Intelligence) and Cooperative Mediation. In the first scenario, they used part of an actual road network in the city of Porto Alegre (RS), Brazil. Two or more plans for the traffic lights at each intersection were stablished *a priori* and the actors (social insects) dealed with the choice of plans to ensure smooth traffic flow. This scenario is described in detail in Section 4.2, because it was used as a reference for the results of this article. The second approach for dynamic formation of groups of coordinated traffic lights, according to those authors, is based on a function to maximize utility that determines which semaphore plan to be used. Those authors state that mediation is a cooperative method of solving distributed optimization problems whose main advantage is the division of the problem in several sub-problems to be resolved by an arbitrator. The scenario for testing this approach is hypothetical: a 5x5 square grid with a light (agent) at each joint was drawn and simulated. For more details on these experiments, we suggest reading Oliveira (2005) and Oliveira et al. (2005).

Besides these, several other studies address the problem of defining traffic lights timing. What is important to note in the literature work is: up to now the employment of GIS, in which the network could be directly retrieved to simulation, with no rework for network drawing, was not adequately addressed by the majority of authors. This approach, besides ensuring agility and evaluation flexibility, almost completely reduces the possibility of error in this redrawing process.

As discussed above, various computational intelligence techniques have been used to aid in traffic engineering. This paper explores the optimization of timing for traffic lights using an algorithm inspired by artificial immune systems, specifically the principle of clonal selection, called CLONAL selection algorithm or CLONALG. This algorithm has been incorporated into a micro-simulation engine, running inside an open source GIS application.

2.1 – CLONALG Algorithm

De Castro (2001) developed computational tools for solving complex engineering problems, inspired by artificial immune systems. One such tool is CLONALG, a simplified computer implementation of the principle of clonal selection of B lymphocytes (or B cells) during an adaptive immune response. A specific immune response, as the production of antibodies to a particular infectious agent, is known as an adaptive immune response. Antibodies are produced by B lymphocytes in response to infections, and their presence in an individual reflects the infections to which it has already been exposed. The principle (or theory) of the clonal selection (or expansion), as described by the author, is associated with the basic features of an adaptive immune response to antigenic stimulation. He states that only one cell which can recognize a specific antigenic stimulus will proliferate, and therefore selected over the others. Two versions of the algorithm have been proposed: the first to solve problems of machine learning and pattern recognition and the second to solve optimization problems. In this work, only the second algorithm will be discussed, because it fits the objectives and proposals set forth herein.

2.1.1 - Definitions

The repertoire of available antibodies is represented by Ab. The computational implementation of CLONALG for function optimization can be made using the pseudocode shown in Figure 1, given the input parameters and output:

- Input:
    - the length L of antibodies;

- *Ab*, the population composed of *N* antibodies;
- *Gen*, the quantity of generations to be performed (stop criterion);
- *n*, the number of antibodies to be selected for cloning;
- *β*, the multiplicative factor used to define the number of clones;
- *d*, the quantity of antibodies with low affinity to be replaced.

- Output:
  - the *Ab* matrix, corresponding to the memory antibodies;
  - the fitness *f* of each antibody matrix *Ab*.

```
function [Ab,f] = clonalg(Ab,L,gen,n,β,d)
% At each generation, do:
for t = 1 to gen,
      f        :=  decode(Ab);
      Ab{n}    := select(Ab,f,n);
      C        := clone(Ab{n},β,f);
      C*       := hypermut(C,f);
      f        := decode(C*);
      Ab{n}    := select(C*,f,n);
      Ab       := insert(Ab,Ab{n});           % Ab ← Ab{n}
      Ab{d}    := generate(d,L);              % Randomly generates Ab{d}
      Ab       := replace(Ab,Ab{d},f);
end;
f := decode(Ab);
```

Figure 1 *– Pseudocode of CLONALG algorithm*
Adapted from De Castro (2001)

The functions used in the pseudocode and their descriptions are listed below:

• `decode(`*Ab*`)`: decodes and calculates the *Ab* population fitness *f* (each element of the vector *f* represents the fitness of a single individual of the *Ab* population) for an objective function *g*(•) to be optimized;

• `select(`Ab, f, n`)`: selects n antibody of Ab population in accordance to f fitness, generating thus a subpopulation Ab(n);

• `clone(`*Ab*(*n*), *β*, *f*`)`: clones the elements of the population *Ab*(*n*) in proportion to the fitness *f* and a multiplicative factor *β*, generating a new population. The generated amount of N_c clones can be defined from Equation (1).

$$N_c = \sum_{i=1}^{n} round(\beta \cdot N/i), \quad (1)$$

where *β* is a multiplicative factor, *N* is the total amount of antibodies *Ab* repertoire and `round()` is the operator that rounds the value in parentheses to the nearest integer. In this case, the antibodies are ordered from highest to lowest fitness, so the higher *i* in

Equation (1), the lower the fitness and therefore the number of clones generated for Ab_i (DE CASTRO, 2001).

• `hypermut(`*C*, *f*`)`: mutates elements of the *C* population in proportion to the fitness *f*, generating another population. A hypermutation is inspired by the somatic hypermutation, which is used by the immune system to enter and maintain the diversity of lymphocyte repertoire and also to improve the affinity (ability to recognize) of antibodies in relation to the stimuli applied. This is the primary operator of the CLONALG and therefore must have a high mutation rate.

• `insert(`*Ab*, *Ab*(*n*)`)`: inserts the matrix *Ab*(*d*) into the population *Ab*(*m*) or, put another way, concatenates the *Ab*(*m*) matrix with the *Ab*(*d*) matrix.

• `generate(`*d*, *L*`)`: generates a population containing *N* antibodies *Ab*, each of length L.

• `replace(`*Ab*, *Ab*(*d*), *f*`)`: replaces *d* antibodies *Ab* by *d* elements of the population *Ab*(*d*) in proportion to the fitness *f*.

All these concepts are used together to propose a viable solution to the problem of generation of traffic lights timing optimization, as will be depicted in details on Section 4.

3 – GEOGRAPHIC INFORMATION SYSTEMS AND SIMULATION

The term Geoprocessing denotes the knowledge discipline that uses mathematical and computational techniques for the treatment of spatial information. It has been influencing a lot the areas of cartography, analysis of natural resources, transportation, communications, energy and urban and regional planning. Computational tools for geoprocessing, called Geographic Information Systems (GIS), allow you to perform complex analysis, by integrating data from various sources and creating georeferenced databases (CAMERA, Davis JR., 2001).

According to Oliveira (2009), SIG are software applications with capacity for management, analysis and storage of georeferenced information and/or geoprocessed, as well as alphanumeric data. Thus, its application is fundamental to knowledge and analysis of regions and geographical situations in general.

An example of GIS is the OpenJUMP (OpenJUMP, 2009), which is a free software and open source platform, written in Java (Sun, 2009). Some advantages of the use of this GIS is the facility for viewing and processing spatial data, and its popularity by being developed and maintained by a large group of volunteers working together around the world. Figure 2 shows the main screen of OpenJUMP with a traffic network task loaded into the application. It is suggested reading Oliveira (2009) for further clarification about this software and the extensions implemented on it by means of this research project.

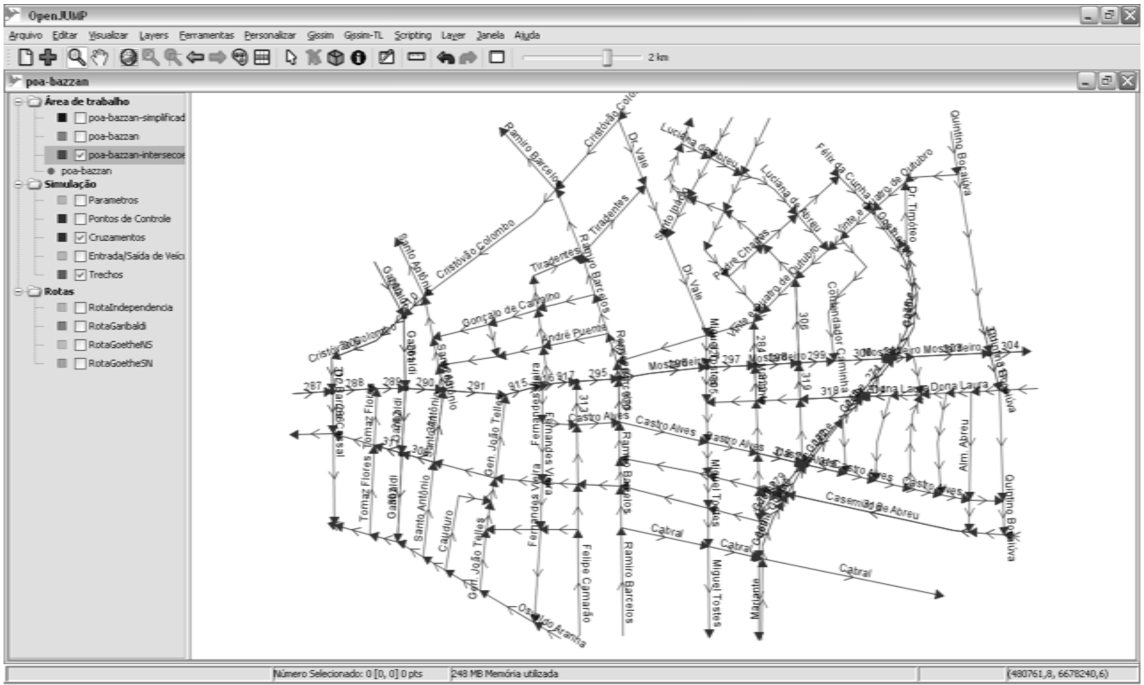


Figure 2 *– Main screen of OpenJUMP with a loaded traffic network task*

4.1 – Micro-simulation

The importance of simulation to help in traffic management is very significant, because it incorporates new features for controlling and planning of traffic in cities (OLIVEIRA, 2009). The traffic simulators, especially the microscopic ones, according to Saliba Neto (2009), are characterized by requiring a large amount of input data, for example, the road network in the region to be studied. Assuming that it is increasingly common the use of GIS for municipalities as a tool for urban planning, it is plausible to use such tools as a data source for simulators, making it possible to reuse spatial information of the road network, such as routes and location of traffic lights, among others, already available for a large number of cities around the world.

The micro-simulator developed by Saliba Neto (2009), called GISSIM, is a microscopic simulator of urban traffic written in Java, an open source plugin incorporated into OpenJUMP. The main sources of information for the simulator are the cartographic city maps, which often have data about the roads and streets network. It is natural that such information covers the entire length of the city. However, due to its microscopic nature, the simulation area should be restricted to a small region of study (SALIBA NETO, 2009).

For this work, the simulator has been extended. The use of multi-lane traffic roads was incorporated and the possibility to include traffic light intersections in the networks under study was implemented. Also, a traffic lights timing generator, called GISSIM-TL, has been developed and incorporated into OpenJUMP. For the timing generation to be realized, a module for managing intersections, allowing for definition of necessary parameters, such as cycle and number of stages, has been designed.

The timing generation module uses, as of now, the CLONALG technique, as discussed in Section 2. Neverthless, this module was modelled and implemented in such a way to allow for the incorporation of other optimization techniques. It is worth to note that the timing generated by this module is for green lights. The timing for red lights for the same stage in other phases is the sum of the time of green and yellow time, both of the other phases. Yellow times are fixed and can be defined in the configuration for each intersection cycle in the network under study.

4 – PRACTICAL EXPERIMENTS AND RESULTS

As a proof of concept for the proposed tool, a practical experiment has been conducted considering an actual region of a big city in Brazil. A small region was represented inside OpenJUMP and recovered by the simulator. An objective comparison with numerical and practical results of the literature was realized. Thus, the experiments were inspired by Oliveira et al. (2005), which despite having a different focus from those of this study, has relevant and consistent results, which met the criteria set out here to choice.

4.1 – CLONALG applied to Semaphoric Timing Optimization

The implementation details of the concepts presented in Section 2, specifically to perform these experiments are discussed below. The antibodies were encoded using binary representation, with a resolution of 10 bits per variable, corresponding to an accuracy of approximately 0.02 seconds in the calculated times. As the only timing that is generated by the techniques is the time to green, because the time of the red stage in question is the sum of the times in other stages, the number of variables depends on the number of green stages in each intersection. The yellow time can be defined for each intersection or from the standard yellow timing defined by the user.

In the implementation of CLONALG algorithm to solve the problem of semaphoric timing optimization, the number of clones per antibody, N_C, is determined from Equation (1). This number is proportional to the fitness of the antibody and a multiplicative factor β. The antibodies are then cloned according to N_C, resulting in subsets C_n, where n = [1..N_C]. Each clone is modified from the mutation operator and then is evaluated. After completion of the evaluation process, the antibody with higher fitness for each subset is selected to compose the new population *Ab*. Among these, *d* antibodies with low fitness are replaced by new ones, randomly generated. The adopted stop criterion is the number of generations.

4.1.1 –Solution Representation

The representation of a solution to the problem is made by a string of binary values (vector), according to the number of variables in question. Figure 3 illustrates a solution to a problem with two variables. Every 10 binary numbers in the vector corresponds to each variable. This vector thus symbolizes the genetic characteristics or the genotype. The set of traffic light timings resulting from the elements of this vector are the actual characteristics or phenotype. In the Figure 3 example, considering the accuracy of 10 bits per variable and the time range of generated times varying between 15 and 40 seconds, the resulting traffic lights times are 18 seconds of green for the variable 01 and 34 seconds of green variable 02.

Variable 01										Variable 02									
1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10
0	1	0	0	0	1	1	0	0	1	1	1	0	0	0	1	1	0	1	1

Figure 3 *– Solution representation for a two variable problem*

4.2 – Adjusted Parameters

The parameters used by CLONALG for the timing optimization were obtained through experimentation and are described as follows:

Table 1 *– CLONALG parameters to timing optimization*

Parameter	Value
Size of repertoire *Ab*	50
Number of generations	61
Clones by antibody (β Factor)	20%
Hypermutation probability	10%
Antibodies replacement (*d* Factor)	20%

4.3 – Experiments Scenarium

The chosen scenario represents an actual region of the city of Porto Alegre, Brazil. The experiments were inspired by those realized by Oliveira et al. (2005), which despite having a different focus from the discussed ones in this work, have relevant and consistent results, which met the desired criteria. Figure 4 represents the region addressed by those authors, highlighting the main intersections and traffic lights considered in the study. In this experiment, 13 traffic light intersections were considered, resulting in a problem with 26 variables to be represented by each genotype in the algorithm.

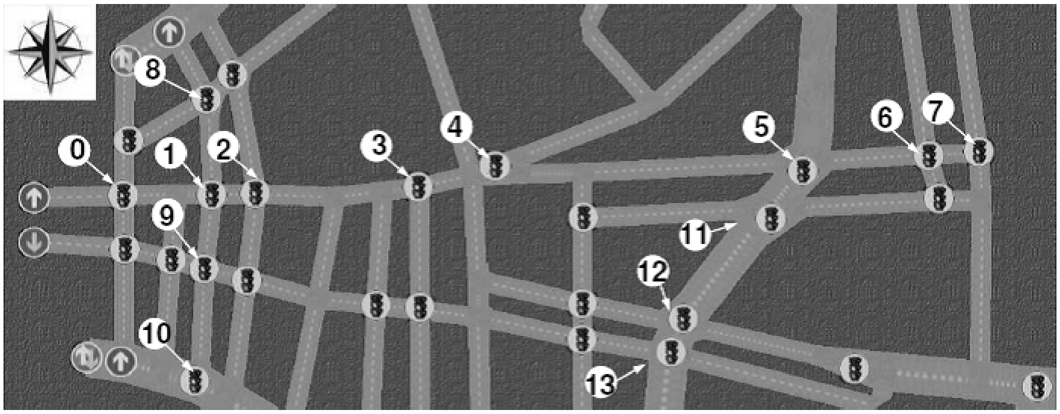
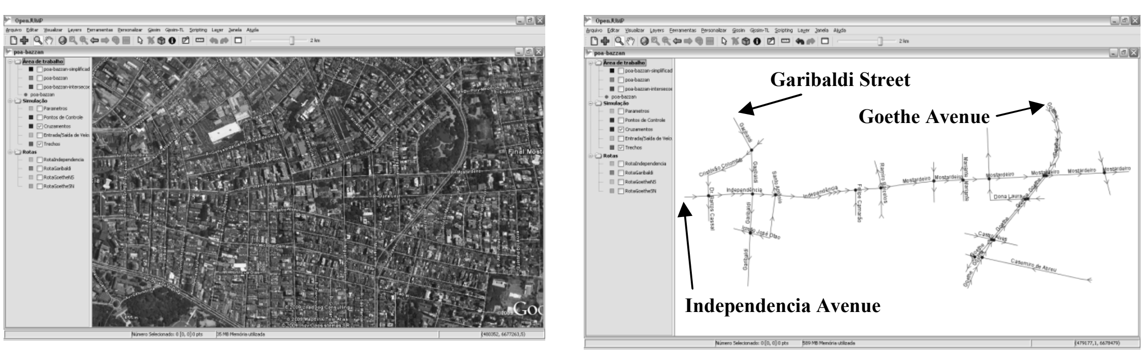


Figure 4 *– Micro-region considered by Oliveira et al. (2005), with 13 traffic lights highlighting*
Source: Oliveira et al. (2005)

According to those authors, the main street (in the map of Figure 1, the west-east corridor represented by Independencia Avenue and followed by Mostardeiro Avenue) has eight traffic lights to be controlled (intersections 0-7 in the Figure 4), and two cross-roads: the Garibaldi Street (comprised of intersections 1, 8, 9, 10) and Goethe Avenue (intersections 5, 11, 12, 13). Vehicle flow entering the network consists of 36 vehicles per minute in Independencia Avenue and of 24 vehicles per minute in Garibaldi Street and Goethe Avenue.

For this experiment, the micro-region in question was drawn on the GIS OpenJUMP, based on a satellite image obtained through Google Earth software (Google, 2009). This only happened because Porto Alegre County Management Office could not be contacted on time to supply a digital cartographic map of the city traffic network. The image was inserted into the GIS using the Universal Transverse Mercator (UTM) coordinate system. This system, as pointed out by Saliba Neto (2009), represents a projection of the Earth's cylindrical surface, whose basic unit is the Meter. GISSIM simulator uses these coordinates to calculate the distance traveled by vehicles. The design carried out to the GIS is in actual scale. Figure 5 shows a satellite image inserted in OpenJUMP and a simplified drawing of the characteristics of the corresponding region represented in the software.



(a) *(b)*

Figure 5 *– Satellite image inserted in OpenJUMP (a) and a simplified region drawn in OpenJUMP (b), with main streets indicated by arrows*

The vehicle entry points to the network, called sources by GISSIM simulator, were considered at Independencia Avenue with 36 cars per minute, and the Garibaldi Street with 24 vehicles per minute. The Goethe Avenue has bi-directional flow, here treated as north-south and south-north, with an entry rate of 24 cars per minute in each of these directions. According to described by Oliveira et al. (2005), the probability of a vehicle to converge in a given intersection was considered to be about 1/10. This probability was configured in the simulator, at each crossing, giving a resulting scenario very close to actual characteristics of the traffic in the region under study.

GISSIM simulator allows for the definition of special vehicles, for statistical purposes, called Instrumented Vehicles. For each of these vehicles, the simulation step to entering the network and an unique identification (name and/or code) are defined. Their behaviour is cyclical, that is, if one of them leaves the simulation at any given time, it must be re-entered in following step.

At the traffic light intersections, cycle parameters, such as phases, stages, focus groups, among others, are determined by the user. The only timing information that can be informed for these intersections are the yellow times and inter-green times. All the others are automatically generated and/or aggregated by the semaphoric timing optimization algorithm.

For the optimization algorithm to be properly run, the parameters described by Table 1, as the size of the *Ab* repertoire, number of generations, cloning by antibody factor *β*, hypermutation probability and the replacement antibody factor *d*, are all assigned in the user interface. Moreover, the time limit for green lights and the number of steps to be performed for evaluation of each individual are also chosen by the user before the algorithm starts.

Based on all these parameters, the semaphoric timing optimization algorithm performs its process of generation and evaluation of populations, until the defined stop criterion is reached. Each antibody repertoire has its traffic timing set evaluated by a special simulation performed with this data timing.

4.5 – Evaluation Function

Each set of configuration timing generated by the optimization algorithm must be simulated to evaluate the results. For this experiment, a period of 3,600 steps (corresponding to one hour) of simulation was used to this evaluation step. Equation (2) presents the evaluation function used to measure the optimality of each timing set generated by the algorithm. Weights were defined to account for traffic priorization in each of the main roads of the system. Maximizing this function defines the ability of the individual or the fitness of the antibody.

$$f(\cdot) = (W_{Ind} \times N_{Ind}) + (W_{Gar} \times N_{Gar}) + (W_{Gns} \times N_{Gns}) + (W_{Gsn} \times W_{Gsn}), \qquad (2)$$

where:

W_{Ind} Weight for Independencia Avenue;
N_{Ind} Number of vehicles which left the simulation by Independencia Avenue;
W_{Gar} Weight for Garibaldi Street;
N_{Gar} Number of vehicles which left the simulation by Garibaldi Street;
W_{Gns} Weight for Goethe Avenue (N/S);
N_{Gns} Number of vehicles which left the simulation by Goethe Avenue (N/S);
W_{Gsn} Weight for Goethe Avenue (S/N);
W_{Gsn} Number of vehicles which left the simulation by Goethe Avenue (S/N).

4.6 – Practical Results and Analysis

As the chosen timing optimization algorithm is a stochastic process, the best result at the end of the optimization was simulated 10 times with 10,800 steps (corresponding to 3 hours) for each instance of simulation. In their experiments, Oliveira et al. (2005) evaluated the average time on the way down to the main roads as pointed out in the last section. For this type of information to be evaluated, cyclic instrumented vehicles were configured in each entry point of the network in steps 1, 10, 100, 1000 and 5000, and the mean path of vehicles on roads was compared with the results in question. Again, it is important to note that Oliveira et al. (2005) did not take into account the timing optimization in their study, but the coordination of traffic lights from pre-defined traffic light timing planes. So they were not working to optimize any traffic criterion. However, the results of those authors were essential for comparison and analysis of the results of this work.

Oliveira et al. (2005) conducted four types of experiments, as follows: the first type (Type I) the setting is ideal and abstract in which there is no semaphores and therefore there is no slowdown caused by queues. In this scenario, which represents an unreal situation, the intersection is dimensionless, so vehicles can cross it without collisions. The second (Type II) is the scenario in which all the main traffic route (Independencia Avenue + Mostardeiro Avenue) are synchronized in fixed and predetermined plans that prioritize running this corridor. The third (Type III) is the scenario in which all the traffic lights carry out plans that do not prioritize any route. In the fourth (Type IV), the lights are controlled by agents with behaviour similar to social insects, in accordance with the proposed model. In this case, the proposed approach was simulated in two different ways: updating the threshold of response using the linear function (Type IV - Linear) or the exponential function (Type IV - Exponential) of success (Oliveira et al., 2005). As the Type I experiments addresses an unreal situation, in this work it was not considered and the results were compared with the other scenarios. The results presented by those authors to Goethe Avenue were used in this study as a reference in both directions (north-south and south-north).

The following tables show the results obtained after the timing optimization with the use of CLONALG and the comparison with results obtained by Oliveira et al. (2005). Experiments are reported with and without weighting for Function Evaluation (see Equation 2). For each road, the average travelling time in seconds is presented, according to the experiment, and the percentage of difference from the other used techniques. This difference was calculated taking into account the average travelling time of the reference work and the outcome of those times obtained from simulations, after the time optimization by CLONALG algorithm, according to Equation (3). In the tables, the negative results in bold indicate improvements obtained by the corresponding technique. All experiments were performed on a computer with an Intel Core 2 Duo E6550 2.33 GHz processor with 2GB of RAM, under Microsoft Windows Vista platform.

$$D = \left(\frac{RO - RL}{RL}\right) \times 100, \quad (3)$$

where:

D Percentage of difference (%);
RO Result obtained by CLONALG;
RL Result from reference work.

4.1.1 – Practical Results

Initially, the experiments were carried out taking into account the evaluation function without weights (FASP), in which the values for W_{via} were all set to 1. In these experiments, the roads traffic were equally treated, without prior assessment of the vehicle inflow in each corridor.

Subsequently, the evaluation function with weights (FACP) was utilized. The vehicle inflow on the network considered to Independencia Avenue, as determined by (Oliveira et al., 2005), is higher compared to other routes. So in actual traffic conditions, this avenue needs more time in green to ensure smooth flow of traffic. To consider this road according to actual traffic situations, experiments were carried out considering the relative number of vehicles that left the network during simulations, and the value for W_{Ind} was set to 5. For the other routes, W_{via} weights were set to 1.

Table 2 presents the results for the Independencia Avenue. The results in relation to the literature showed a relative improvement. The weight used was effective, contributing to the literature to be overcome in the experiment of Type IV - Linear difference and reducing the percentual difference for others.

Table 2 *– Results – Independencia Avenue*

Optimization Algorithm	Independencia Avenue + Mostardeiro Avenue			
	Type II	Type III	Type IV – Linear	Type IV – Exp.
CLONALG-FASP	137,85%	28,98%	5,00%	12,93%
CLONALG-FACP	116,86%	17,60%	**-4,27%**	2,96%

The results for Garibaldi Street are shown in Table 3. Comparint to the literature, all results were overcome by both techniques, with values at least close to 36% and higher than 70%.

When the weight was applied to Independencia Avenue, it is observed that the results for this route are increased, especially for the experiments of type III and IV. This may be justified by the fact that this route is close to the point of greatest vehicle inflow at the intersection with Independencia Avenue. Yet the results were very consistent, with improvement over 60%.

Table 3 *– Results – Garibaldi Street*

Optimization Algorithm	Garibaldi Street			
	Type II	Type III	Type IV – Linear	Type IV – Exp.
CLONALG-FASP	**-71,94%**	**-44,93%**	**-36,84%**	**-33,70%**
CLONALG-FACP	**-63,27%**	**-27,91%**	**-17,32%**	**-13,22%**

For Goethe Avenue, north to south direction, the results are presented in Table 4. In both experiments, it can be observed that the results were very close to those of Garibaldi Street, performing better than the literature in almost 70%. The weighting for the main avenue did not appreciably affect the results of that route. This situation may suggest a negative balance for this route or increase the weighting for Independencia Avenue, in order to ensure greater flow in that avenue.

Table 4 *– Results – Goethe Avenue (N/S)*

Optimization Algorithm	Goethe Avenue (N/S)			
	Type II	Type III	Type IV – Linear	Type IV – Exp.
CLONALG-FASP	**-56,68%**	**-67,36%**	**-38,40%**	**-29,14%**
CLONALG-FACP	**-57,65%**	**-68,08%**	**-39,77%**	**-30,72%**

Table 5 presents the results for the Goethe Avenue, now from south to north. For the experiment without weighting, the literature was overcome with values close to 60%. With the use of weighting, it is observed that the results were poorer but yet suffered increase when compared to reference work. This event can be justified by the weight of the main road and/or the need to balance the track in the opposite direction, as discussed earlier. There was also improvement in regard to literature with values above 55%.

Table 5 *– Results – Goethe Avenue (S/N)*

Optimization Algorithm	Goethe Avenue (S/N)			
	Type II	Type III	Type IV – Linear	Type IV – Exp.
CLONALG-FASP	**-15,87%**	**-54,38%**	**-62,86%**	**-60,06%**
CLONALG-FACP	**-1,57%**	**-46,62%**	**-56,55%**	**-53,26%**

4.1.2 – General Analysis

In the experiments, a traffic condition very close to actual situations was represented when the avenue of greatest flow was differently weighted in the optimization algorithm. This situation was able to reduce the average travelling time for the main street, while for other roads this time was increased.

The weighting of the evaluation function implied the increase and reduction of average travelling times around the network under study. This suggests that the evaluation should be made from two or more functions, through multi-objective optimization techniques. Another important point to be noted is the way that evaluations are carried out. The timing optimization algorithm depends on the simulations for the evaluation of results through the evaluation function. As the simulation is stochastic, using discrete event simulation models, therefore, the evaluation function is also stochastic in time. This random characteristic increases the complexity of the problem and indicates the use of other techniques such as robust optimization with uncertainties (noise optimization) to get best results.

The presented results, in general, are satisfactory and show the effectiveness of the computational intelligence approach. The comparison of the results with the literature represented significant improvement, over 60% in some experiments. In almost all experiments in the reference literature has been overcome. Although the approach of reference is different, it was essential for the evaluation and validation of the CLONALG technique here employed.

5 – FINAL DISCUSSION

This paper presented a traffic lights timing optimization technique, using an algorithm based on the principle of clonal selection, called CLONAL selection algorithm, incorporated into the GIS OpenJUMP and a micro-region simulator. The incorporation of these technologies is presented as an important tool for traffic engineering, because it uses a georeferenced base through GIS application for traffic simulation and adds computational intelligence techniques to optimize traffic light timing. With the use of this tool, simulation and optimal semaphore timing is simplified, provided that it is easy to get georeferenced maps of the region of interest and statistical data of traffic flow in that region.

Performing experiments to optimize traffic light timings and comparing simulation results with the technical literature show the effectiveness of the proposed technique. The literature in question was essential for the validation of this technique, because it used an actual region for the experiments, and presented numerical practical results about the taken experiments.

REFERENCES

Câmara, G.; Davis Jr., C. (2001) *Introdução à ciência da geoinformação*. In: Câmara, Gilberto; Davis Jr., Clodoveu Augusto; Monteiro, Antônio Miguel Vieira (ed). Updated on 04/06/2001. Livro on-line Cap. 1. Accessed on jul/2009. Available in: <http://www.dpi.inpe.br/gilberto/livro/introd/cap1-introducao.pdf>.

Carvalho, A. C. P. L. F.; Braga, A. P. & Ludermir, T. B. (2005) *Computação Evolutiva. In: Rezende, S. O. (ed.) Sistemas Inteligentes – Fundamentos e Aplicações*. Manole, Barueri, SP.

De Castro, L. N. (2001) *Engenharia Imunológica: Desenvolvimento e Aplicação de Ferramentas Computacionais Inspiradas em Sistemas Imunológicos Artificiais*. PhD Thesis. Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.

DENATRAN (1984) *Manual de Semáforos*. 2ª Ed, Ministério da Justiça, Brasília, DF.

DETRAN-SP (2009) *Frota de Veículos – 2009*. Departamento Estadual de Trânsito de São Paulo, São Paulo, SP. Available in: http://www.detran.sp.gov.br/frota/frota.asp. Accessed on: jul/2009.

Fogel, D. B (2006) *Evolutionary Computation: toward a New Philosophy of Machine Intelligence*. (3ª ed.), New Jersey: Wiley, John & Sons. IEEE Press Series on Computational Intelligence.

Goldberg, D. E. (1989) *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Publishing Company, Reading, Massachusetts.

Google (2009) *Google Earth*. Available in: http://earth.google.com/intl/pt/. Accessed on: jul/2009.

Lämmer, S. e Helbing, D. (2008) *Self-control of traffic lights and vehicle flows in urban road networks*. Journal of Statistical Mechanics: Theory and Experiment. v. 2008, p. P04019, (34 pág).

Michalewicz, Z. (1996) *Genetic Algorithms + Data Structures = Evolution Programs* . 3ª ed. Springer-Verlag.

Oliveira, D. (2005) *Um Estudo de coordenação dinâmica de agentes aplicado ao gerenciamento de tráfego veicular urbano*. Masters Thesis, Instituto de Informática, Universidade Federal do Rio Grande do Sul.

Oliveira, D.; Bazzan, A. L. C.; da Silva, B. C. e Júnior, P. R. F. (2005) *Coordenação dinâmica de semáforos: dois casos de estudo*. XXV Congresso da Sociedade Brasileira de Computação : V Encontro Nacinal de Inteligência Artificia ENIA'2005, UNISINOS - São Leopoldo/RS, p. 761-770.

Oliveira, F. B. (2009), *Geração de Tempos Semafóricos utilizando Inteligência Computacional em SIG Convencionais*, Masters Thesis, Centro Federal de Educação Tecnológica de Minas Gerais – CEFET-MG.

OpenJUMP (2009) *OpenJUMP Project*. Available in: http://openjump.org. Accessed on: jul/2009.

Rezende, S. O. (2005) (Ed) *Sistemas Inteligentes - Fundamentos e Aplicações*. Manole, Barueri, SP, Brasil.

Rouphail, N. M.; Park, B. B. e Sacks, J. (2000) Direct Signal Timing Optimization: Strategy Development and Results. Relatório técnico, XI Pan American Conference in Traffic and Transportation Engineering in Gramado, Brasil.

Saliba Neto, T. S. (2009) *Construção de uma ferramenta de código aberto para simulação microscópica de tráfego urbano*. Masters Thesis, CEFET-MG, Belo Horizonte, MG.

Sanchez, J.; Galan, M. e Rubio, E. (2008) *Applying a Traffic Lights Evolutionary Optimization Technique to a Real Case: "Las Ramblas" Area in Santa Cruz de Tenerife*. IEEE Transactions on Evolutionary Computation, v. 12, p. 25-40.

Silva, G. (2006) *Modelagem e Implementação de uma Ferramenta Inteligente e de Código Aberto para Inserção Automática de Inferência Fuzzy em SIG Convencionais*. Masters Thesis, CEFET-MG, Belo Horizonte, MG.

Sun (2009) *Developer Resources for Java Technology*. Sun Microsystems, Inc. Available in: http://java.sun.com. Accessed on: jul/2009.