



**HAL**  
open science

# A multimodal traffic responsive strategy using particle swarm optimization

Sofiane Kachroudi, Neila Bhourri

► **To cite this version:**

Sofiane Kachroudi, Neila Bhourri. A multimodal traffic responsive strategy using particle swarm optimization. 12th IFAC Symposium on Transportation Systems, Sep 2009, LOS ANGELES, CA, United States. pp 531-537. hal-01488125

**HAL Id: hal-01488125**

**<https://hal.science/hal-01488125>**

Submitted on 27 Mar 2017

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

## A multimodal traffic responsive strategy using particle swarm optimization

Sofiane KACHROUDI\* Neila BHOURI

INRETS, 2 rue de la butte verte 93166 Noisy-Le-Grand, France

**Abstract:** The paper describes a model predictive control strategy for multimodal urban traffic regulation for private vehicle (PV) traffic and public transport (PT). By acting on the traffic lights, the strategy aims to achieve two objectives. First to make public transport vehicles respect their timetable and, second, to improve the global traffic conditions. Based on a complex model, the strategy determines the optimal green splits and the offsets of the traffic lights. The optimization problem is multiobjective and resolved by the particle swarm optimization algorithm. Two criteria are proposed for private vehicle traffic and public transport. One of the originalities of this work is that the two criteria are expressed globally and are related since the PT progression model depends on the PV traffic. Two PSO algorithm versions are used: the original algorithm and a modified version expected to be more efficient. The efficiency of the strategy is evaluated on a large virtual urban network.

Keywords: Particle Swarm Optimization(PSO); Model predictive control; traffic model; bus priority; private vehicle (VP); public transport (TC).

### 1. INTRODUCTION

Many decades ago, the regulation of private vehicle urban traffic became a necessity due to the huge increase in the volume of traffic in urban areas associated with the impossibility of infinitely extending the road infrastructure. It had also become urgent to regulate public transport in cities. Today, reducing PV traffic has become insufficient to regulate the progression of PT vehicles. Many urban traffic control strategies exist, including TUC, PRODYN, UTOPIA, SCOOTs, CRONOS. These strategies were first developed to deal just with PV traffic. Some of them were then extended to deal with PT traffic. Urban traffic control strategies giving an active priority to public transport in a large urban network has been developed by (Bhourri and Lotito [2005]). However although they are methodologically different, these strategies have the same objective which is to relieve traffic congestion over the whole network by doing more for the links when and where public transport vehicles are present. This paper proposes a new strategy which, in addition to improving traffic conditions over the whole network, aims in particular to enable public transport vehicles respect their timetable. Based on a multimodal model that predicts the state variables, two optimization criteria are formulated for PV and PT traffic. The nature of the multimodal model and thus of the optimization criteria and the large-scale nature of the problem have led to using a sophisticated optimization method that guarantees fast convergence for a complex problem. The optimization method used is Particle Swarm Optimization (PSO) which is an efficient metaheuristic.

The paper is organized as follows. The first section gives an overview of particle swarm optimization and presents both the original algorithm and a modified algorithm. The extension of the PSO algorithm to the case of multiobjective optimization is introduced. The second section presents the urban traffic control strategy. The multimodal model is presented and the problem is formulated as a multiobjective optimization problem. The third section concerns the simulation results achieved on a virtual urban network with different receding horizons. This is followed by a conclusion.

### 2. PARTICLE SWARM OPTIMIZATION (PSO)

Particle swarm optimization (PSO) is an evolutionary algorithm to find an optimum in a search space. It was proposed first in 1995 by James Kennedy and Russel C. Eberhart. The PSO algorithm is a stochastic, population-based algorithm. Initially a population is spread randomly in the search space. The movement of the population is based on the social physiological behavior of birds. The trajectory of each particle depends not only on its own past trajectory but also on the best positions occupied by a defined neighborhood. Without loss of generality, a mono objective optimization problem can be formulated as follows:

$$\begin{aligned} & \text{Minimize} && f(\vec{x}) \\ & \text{subject to :} && g_i(\vec{x}) = 0; \quad i = 1, \dots, m \\ & && h_j(\vec{x}) \leq 0; \quad j = 1, \dots, p \\ & \text{With} && \vec{x} \in \mathbf{R}^n \end{aligned}$$

where  $f$  is the function to be minimized,  $g$  and  $h$  are the constraints,  $\vec{x}$  is the vector of decision variables.

\* Corresponding author. Tél +33 (0)1 45 92 56 32  
E-mail address : sofiane.kachroudi@inrets.fr

The algorithm described by J. Kennedy and R-C. Eberhart in 1995 is the original version of PSO. Since then many versions have been proposed. In this paper, the original version and a modified PSO proposed by Van Der Bergh in 2001 are used.

### 2.1 Original PSO algorithm

The PSO algorithm is based on an efficient move of the population within the search space. Each particle is defined by a position and a velocity. In fact, the velocity is not defined as a real speed but as a displacement vector. At each computational iteration, these two variables are updated. The displacement of a given particle in the current iteration depends on (i) the displacement in the previous iteration, (ii) the difference between the current position and the best position of the particle in the past and (iii) the difference between the current position and the best position of the particles that make up the neighborhood. Each term of this update is weighted by a factor that will be discussed later. The position update is simply given by adding the displacement updated to the past position. The best position is defined as the one that minimizes the objective function  $f$ . As mentioned implicitly above, each particle has a neighborhood. There are many ways to define a neighborhood. Further details can be found in the work of (Clerc and Kennedy [2002] and Reyes-Sierra and Coello. [2006]). Let  $X_i(k)$  and  $V_i(k)$  designate the position and the displacement vector of particle  $i$  at iteration  $k$ , the update equation of particle  $i$  is given by:

$$\begin{aligned} V_i(k+1) &= \omega \cdot V_i(k) + c_1 \cdot r_{1,i}(k) \cdot [Xbest_i - X_i(k)] \\ &\quad + c_2 \cdot r_{2,i}(k) \cdot [Xbest_V - X_i(k)] \\ X_i(k+1) &= X_i(k) + \chi \cdot V_i(k+1) \end{aligned}$$

$Xbest_i$  and  $Xbest_V$  are respectively the best position of particle  $i$  and the best position of the particles making up the neighborhood of particle  $i$ .

$r_{1,i}(k)$  and  $r_{2,i}(k)$  are two random numbers included in set  $[0; 1]$ .

$\omega$ ,  $c_1$ ,  $c_2$  and  $\chi$  are the setting parameters of the swarm movement. These parameters are very important for the convergence and the speed convergence of the swarm.

The convergence of the original PSO algorithm was studied by Clerc and Kennedy in (Clerc and Kennedy [2002]). They established the following inequality that guarantees the convergence of the algorithm  $\omega > \frac{1}{2} \times (c_1 + c_2) - 1$ . However, an empiric combination of these parameters has given a good balance between the quality of the optimum found and the rate of convergence. The empiric combination is given by:  $\omega = \frac{1}{2 \cdot \ln(2)}$  and  $c_1 = c_2 = 0.5 + \ln(2)$ , where  $\chi$  is the constriction factor. In the original version, this factor was set to 1. It was observed that the optimal setting parameters depend on the nature of the problem formulated, i.e. on the objective function. In this paper, we use the empiric combination so that we do not substitute our optimization problem by searching for the optimal setting parameters.

The last point concerns a large class of optimization problems when the decision variables  $\vec{x}$  are bounded in the set  $[x_{min}; x_{max}]$ . In this case, when the position goes beyond the bounds, it is set to the closest bound. Therefore, the

complete update is given by:

$$\begin{aligned} V_i(k+1) &= \omega \cdot V_i(k) + c_1 \cdot r_{1,i}(k) \cdot [Xbest_i - X_i(k)] \\ &\quad + c_2 \cdot r_{2,i}(k) \cdot [Xbest_V - X_i(k)] \\ X_i(k+1) &= X_i(k) + \chi \cdot V_i(k+1) \\ &\quad \text{if } (X_i(k+1) < x_{min}) \text{ then} \\ &\quad \quad X_i(k+1) = x_{min} \\ &\quad \text{if } (X_i(k+1) > x_{max}) \text{ then} \\ &\quad \quad X_i(k+1) = x_{max} \end{aligned}$$

In this paper, the neighborhood of a particle is made up of all the other particles of the swarm. If we note  $nb.max$  the maximum number of iterations,  $\epsilon$  a fitness objective fixed by users and  $min.p$  the global minimum of the swarm, the pseudo-code of the PSO algorithm is given by:

```

Initialize the population
While (i < nb.max or min.p > ε) do
  For each particle
    determine the best neighborhood position;
    update velocity and position;
    update the best personal position;
  end for;
  i ++;
end while;
report the final global minimum min.p;

```

### 2.2 Modified PSO (GCPSO) algorithm

Clerc and Kennedy (Clerc and Kennedy [2002]) have given the convergence condition of the original PSO algorithm. Shi and Eberhart (Shi and Eberhart [1999]) indicated that choosing a constriction factor of less than 1 improves the convergence rate of the algorithm. However, Van der Bergh has proved in his PhD thesis (Van Den Bergh [2002]) that the original PSO does not guarantee a global optimum. It is important here to recall that the algorithm is stochastic and hence it is impossible to prove rigorously the convergence or not of the algorithm. The fact remains that, as for the studies of Clerc and Kennedy, Van Der Bergh has assumed the hypothesis that a mathematical proof is possible. Finally, if the convergence to a global optimum is not guaranteed, the original algorithm may in some cases find a global a such optimum.

To address this issue, Van Der Bergh proposed a modified version of the original algorithm that is supposed to guarantee a global optimum or at least to find an optimum of a better quality than the original algorithm. Considering  $\tau$  the index of the global best particle of the swarm, the only modification concerns the velocity update of this particle. The new velocity update of particle  $\tau$  is given by:

$$V_\tau(k+1) = -X_\tau(k) + Xbest_\tau + \omega \times V_\tau(k) + \rho(k) \times (1 - 2 \times r_{2,\tau})$$

The other velocity particles are unchanged. The new update equation of the best particle position is  $X_\tau(k+1) = Xbest_\tau(k) + \omega \times V_\tau(k) + \rho(k) \times (1 - 2 \times r_2(k))$ .

The best global particle position is reset to its best position to which is added the current displacement. The random

term  $\rho(k) \times (1 - 2 \times r_2(k))$  is used to explore a hypercube of side  $2 \times \rho(k)$ . The values of  $\rho(k)$  are adapted at each time-step and depend on the nature and search space of the optimization problem. However, this parameter is generally increased when the global best of the swarm is improved at each time-step and decreased in the contrary case. The most simple way to adapt  $\rho$  is to double it if the global best of the swarm is improved successively for a number of time-steps higher than a fixed value and to divide it by 2 if the global best remains the same for a number of time-steps below a fixed value.

### 2.3 Multiobjective PSO

A multiobjective optimization problem can be formulated as follows:

$$\begin{aligned} & \text{Minimize} && [f_1(\vec{x}), f_2(\vec{x}), \dots, f_k(\vec{x})] \\ & \text{subject to:} && g_i(\vec{x}) = 0; \quad i = 1, \dots, m \\ & && h_j(\vec{x}) \leq 0; \quad j = 1, \dots, p \\ & \text{With } && \vec{x} \in \mathbf{R}^n \end{aligned}$$

In monoobjective optimization, the optimizer determines a single solution that minimizes the cost function. In multiobjective optimization, the optimizer determines a set of solutions that form the Pareto front. The difficulty then is the choice of a single solution within the Pareto front. For reasons easily related to the features of the monoobjective PSO algorithm, extension to a multiobjective PSO raises the question of the comparison of two or more solutions during the optimization process. In particular three questions have to be asked:

- (1) How to update the best position of a given particle?
- (2) How to select the best particle of the neighborhood?
- (3) How to determine the best global particle of the swarm?

The most commonly used way to retain nondominated solutions is the use of external archives (Reyes-Sierra and Coello. [2006]). Strictly speaking, three external archives are necessary. One to retain the best position of a particle; the second to retain the best particle of the neighborhood of a given particle; the third to store the global best particle of the swarm. However, for evident reasons of combinatory complexity, no more than one external archive is reported in the literature (Reyes-Sierra and Coello. [2006]). Actually, if we choose a full connected neighborhood topology, the best particle of the neighborhood is the global best of the swarm. For this present work, we choose a full connected neighborhood topology and one external archive for the global best particle of the swarm. The corresponding pseudo-code of the multiobjective optimization algorithm is as follows:

```

Initialize the population
Initialize the swarm archive
While (i < nb_max or min_p > ε) do
  For each particle
    select the global best;
    update velocity and position;
    update the particle best position;
  end for;
  i ++;
  update the swarm archive;
end while;
report the final swarm archive;

```

However, two questions remain unanswered. The first concerns the update of the best position of a particle. Given the past best position and the new updated position, how do we update the best position? If one position dominates the other, the new best position is the dominating one. In the case where the two positions are nondominated, M. Reyes-Sierra and C.A. Coello have reported in (Reyes-Sierra and Coello. [2006]) several ways of choosing between the two solutions. In the work presented in this paper, 4 ways were tested. The first is to consider an aggregate function and the solution that minimizes this function is chosen. The second is to choose randomly between the two solutions. The third is to choose the solution that minimizes only one criterion among the  $p$  criteria. In our case, there are two objective functions to be minimized which results in two possibilities: giving priority to the first or the second criterion. The two possibilities were considered in the simulation. The second question concerns the selection of the global best position of the swarm. As far as updating the best particle position is concerned, three cases are studied: random choice, choice based on an aggregate function and choice based on giving priority to one criterion.

An exhaustive overview of multiobjective particle swarm optimization can be found in the paper of M. Reyes-Sierra and C.A. Coello "Multiobjective Particle Swarm Optimizers : A Survey of the State-of-the-Art".

## 3. MULTIMODAL TRAFFIC CONTROL

Reducing congestion in urban road networks and regulating the traffic of public transport vehicles has become an important issue which needs to be resolved. The aim of this work is to propose a closed-loop control strategy that reduces congestions in each link of the urban network and regulates public transport vehicle traffic. From traffic measures collected from the real network, the strategy has to determine the optimal inputs to apply to the system in order to respect the objectives cited above. The strategy uses an internal model or predictor that predicts the state variable of the system.

### 3.1 Traffic model

The global system considered is made up of four elements: the urban road network, the Private Vehicle traffic, the Public Transport vehicles and the traffic lights controlling

the whole of the traffic.

To establish a mathematical or computational model, a mathematical description of these four elements is necessary. The urban road network is considered as a directed graph. The nodes represent the intersections and the links represent the roads connecting two intersections. Mathematically, traffic lights are described by four variables: the cycle time  $C$ , the green splits that are the durations of the green phase, the offsets between two consecutive intersections and the phase sequence. To describe the traffic mathematically, several variables may be used depending on the objective of the model and the features of the problem formulation. The variables may be the mean waiting time at intersections, the travel time for an origin-destination combination, the flow in each link or the number of private vehicles in each link of the urban network. As far as the PV mode is concerned, there are several ways to describe the progression of PT vehicles in the network. It may be the mean waiting time at intersections, the travel time between two points or the positions collected for a regular time interval.

For this present work, the PV mode is described by the number of private vehicles in each link of the network and the PT mode is described by the position of each vehicle of each line present in the network. The control variables are the green splits of all the traffic lights of the network and the offset of just those traffic lights crossed by PT vehicles. cycle time and the phase sequence are assumed to be fixed and known.

*PV traffic model* The dynamics of the PV mode in an urban network is described by the number of private vehicles in each link at the beginning of the time-step. For a reason that will be explained later, the time-step is the cycle time  $C$ . Given a link  $i$  that connects  $j_1$  to  $j_2$ , the number of VP in this link at the beginning of cycle time  $k + 1$  is given by the following conservation equation:

$$X_i(k + 1) = X_i(k) + C \cdot (q_i(k) - u_i(k))$$

where  $q_i(k)$  and  $u_i(k)$  are respectively the inflow and the outflow of link  $i$  at cycle time  $k$ . The inflow of link  $i$  may be expressed as the sum of outflows of upstream links weighted by their respective turning rates. It results that the knowledge of the outflows is sufficient to completely determine the conservation equation. Gazis and Potts in (Gazis and Potts [1963]) have established a simple model called *store – and – forward* model. The model expresses the outflow of a link  $i$ , controlled by a traffic light with  $G_i$  the duration of the green phase, as follows,  $u_i = S_i \times \frac{G_i}{C}$  where  $S_i$  is the saturation flow of link  $i$ . Incorporating this in the last conservation equation, the final dynamics of link  $i$  is given by:

$$X_i(k + 1) = X_i(k) + C \cdot \left[ \sum_{w \in I_{j_1}} \tau_{w,i} \cdot \min\left(S_w \cdot \frac{G_w(k)}{C}, \frac{X_w(k)}{C}\right) - \min\left(S_i \cdot \frac{G_i(k)}{C}, \frac{X_i(k)}{C}\right) \right]$$

where  $I_{j_1}$  is the set of upstream links of link  $i$  and  $\tau_{w,i}$  is the turning rate from link  $w$  toward link  $i$ . The whole dynamics of PV traffic in the urban network is inferred by applying this equation to all links.

*PT traffic model* The dynamics of the PT vehicles is represented by their positions in the network. These positions are referenced with respect to the starting point of the PT vehicle in the network. Apart the geometric features of the urban network, the free bus speed and the dwell times at stops, strictly speaking the evolution of this position in time depends on the PV traffic conditions and on the configurations of the traffic lights, especially the green splits and the offsets. In (Kachroudi and Bhourri [2008]), a detailed description of a similar model was given. This model only involves the green splits, the offsets were ignored and assumed to be random. A detailed description of the model used here will be proposed in future papers. Given bus  $n$  of line  $m$ , its position at the beginning of cycle time  $k$  is  $P_{m,n}(k)$ . The change in bus position can be given by the following general equation:

$$P_{m,n}(k + 1) = F\_TC(P_{m,n}(k), Td_{m,n}(k), G_{m,n}(k), Nb_{m,n}(k), Ts_{m,n}(k))$$

$Td_{m,n}(k)$  and  $G_{m,n}(k)$  are the vectors that contain respectively the beginning instants and the durations of the green phases of all traffic lights that can be crossed by buses during cycle time  $k$ .  $Nb_{m,n}(k)$  is the vector containing the number of private vehicles ahead of the bus in each link that may be crossed by the bus during cycle time  $k$ . Finally,  $Ts_{m,n}(k)$  is the vector containing the dwell time at the stops where the bus may dwell during cycle time  $k$ . Function  $F\_TC$  is an *if – then – else* type algorithm. We introduce a new variable  $tr$  that is the remaining time before the end of the current cycle time. Initially,  $tr$  is set to the value  $C$  and at the end of the cycle time  $tr = 0$ . The algorithm is given as follows:

```

P_p = P_{m,n}(k);
tr_p = C;
While (tr_p > 0) do
  if (next_stop = station) then
    (P_n, tr_n) = g_st(P_p, tr_p, G, T_d, Nb, T_st);
  else (next_stop = traffic_light)
    (P_n, tr_n) = g_light(P_p, tr_p, G, T_d, Nb);
  end if;
  P_p = P_n;
  tr_p = tr_n;
end while;
P_{m,n}(k + 1) = P_n

```

The algorithms  $g\_st$  and  $g\_light$  update the position and the remaining time of a bus respectively when the bus dwell on a stop and when it crosses over a traffic light. These two algorithms will be detailed in future papers and

thus will not be discussed here.

### 3.2 Problem formulation

As described above, the urban traffic is considered as a discrete-time system. The state variables or outputs are the number of private vehicles in each link of the network and the position of each bus present in the road network. The controllable variables or inputs are the beginning instant of green phases (equivalent to offsets) of traffic lights crossed by bus lines and the green splits of all traffic lights of the network. At the beginning of cycle time  $k$ , the number of private vehicles in each link of the network is spliced into the vector  $X(k)$  and the position of all buses present is spliced into the vector  $P(k)$ . The objective of the control strategy proposed is to minimize the total number of private vehicles present in the network and to minimize the quadratic difference between the real position  $P(k)$  of the buses and a pre-specified position  $P^s(k)$ .

Mathematically, using the following notations:

$$\begin{aligned} P(k) &= [P_1(k), P_2(k), \dots, P_{nl}(k)]^T \\ P^s(k) &= [P_1^s(k), P_2^s(k), \dots, P_{nl}^s(k)]^T \\ X(k) &= [X_1(k), X_2(k), \dots, X_{N_a}(k)]^T \\ G(k) &= [G_1(k), G_2(k), \dots, G_{N_a}(k)]^T \\ Td(k) &= [Td_1(k), Td_2(k), \dots, Td_{N_{ac}}(k)]^T \end{aligned}$$

where  $nl$  is the number of buses present at cycle time  $k$  and  $N_a, N_{ac}$  are respectively the number of links of the network and the number of links crossed by the  $nl$  buses present in the network ( $N_{ac} \leq N_a$ ), the problem can be formulated as follows :

|   |
|---|
| <p><b>Minimize</b></p> $O\_VP(G) = \sum_{j=k+1}^{k+N_h} \sum_{i=1}^{N_a} X_i(j)^2$ $O\_TC(Td, G) = \sum_{j=k+1}^{k+N_h} \sum_{i=1}^{nl} [P_i(j) - P_i^s(j)]^2$ <p><b>Subject to</b></p> <p><b>For every intersection :</b></p> $G_i = G_j = C - G_k$ $Td_i = Td_j = (G_i + Td_k) \text{ modulo } C$ <p style="padding-left: 20px;">links <math>i</math> and <math>j</math> have the right of way simultaneously contrary to link <math>k</math></p> $G_i^{min} \leq G_i \leq G_i^{max}; i = 1..N_a$ $0 \leq Td_j \leq C; j = 1..N_{ac}$ |
|---|

## 4. SIMULATION RESULTS AND DISCUSSION

The simulation tests are performed on a virtual urban road network made up of 16 intersections and 51 links. It is crossed by 4 bus lines and each bus line has to dwell on two stops for a time assumed to be known and fixed. Each intersection and thus each link is controlled by a traffic light with the same cycle time  $C = 80s$ . The road network has 11 entrances. The nominal inflows considered at these entrances correspond approximately to the maximum of inflows that lead to smooth traffic conditions with maximum link capacity. In other terms, with higher

inflows the traffic conditions will become saturated with vehicle queue lengths increasing indefinitely. However, we will introduce some disturbances. The simulation time is for 40 cycle times. In the first 10-cycle time, the inflows are considered nominal and for the 3 remaining lots of 10 cycles, the inflows are disturbed by increasing the inflows at some entrances and decreasing then at others. Finally, for all links, the saturation flow value is 0.5 *veh/s*.

Two series of simulations were done. Firstly, the optimization horizon  $N_h$  is set to 1, which means that the objective is to regulate the traffic for only the next cycle and that the consequences of these actions on future cycles are ignored. Secondly, we consider an optimization horizon of 6 cycles. Given a current cycle, the objective is to determine the optimal durations and beginning instant of green phases of the 6 future cycles. It is important to point out that with an optimization horizon  $N_h > 1$ , the number of decision variables is multiplied by  $N_h$ . This last remark will have serious consequences on the quality of the optimums found by the PSO algorithm.

The criteria to be compared are given by the following equations:

$$OVP(k) = \sum_{i=1}^{51} X_i(k)$$

$$OTC(j) = \frac{1}{28} \times \sum_{n=1}^{28} |P_n(j) - 350 \times j|$$

where  $j \in \{1, 2, 3, 4\}$

where  $k$  is the index of cycle time so  $k \in [1, 40]$ ,  $i$  is the index of links,  $j \in \{1, 2, 3, 4\}$  is a cycle time index (buses are expected to spend no more than 4 cycle times in the network),  $350 \times j$  is the objective position to be attained at the beginning of cycle time  $j$ , and  $n$  is the index of buses that enter the network over the simulation time.

Figure 1 illustrates the evolution of the criteria  $OVP$  and  $OTC$  for three control strategies: without optimization, with optimization and for  $N_h = 1$ , and with optimization and for  $N_h = 6$ .

It emerges from the two figures that with the optimization algorithm and for an optimization horizon set to 1, PV traffic and the progression of PT are improved by about 22% for the PV traffic at the end of the simulation time and by about 230% for PT progression. It was observed that these improvement were much higher when the maximum number of iterations allowed for the PSO algorithm was higher. However, the two figures also show a surprising result: with the optimization algorithm and for  $N_h = 6$ , the performance of the optimization process is worse than for  $N_h = 1$ . The explanation is that for such a networked system (the outflow of one link is the inflows of others), and for such criteria that include all links, optimal actions for just one future cycle time do not have such bad consequences on the other future cycle times that they can not be overcome for the same optimization process in the respective future cycle times. However, this explanation justifies the absence of difference between cases  $N_h = 1$  and  $N_h = 6$  and does not justify the

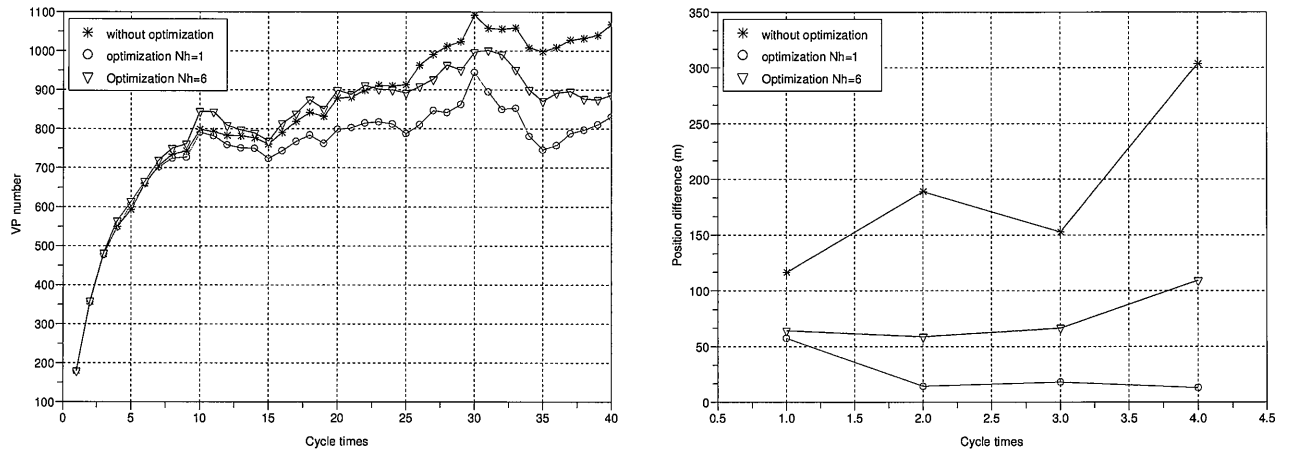


Fig. 1. Comparative results for the original PSO algorithm a) private vehicle criterion b) public transport criterion

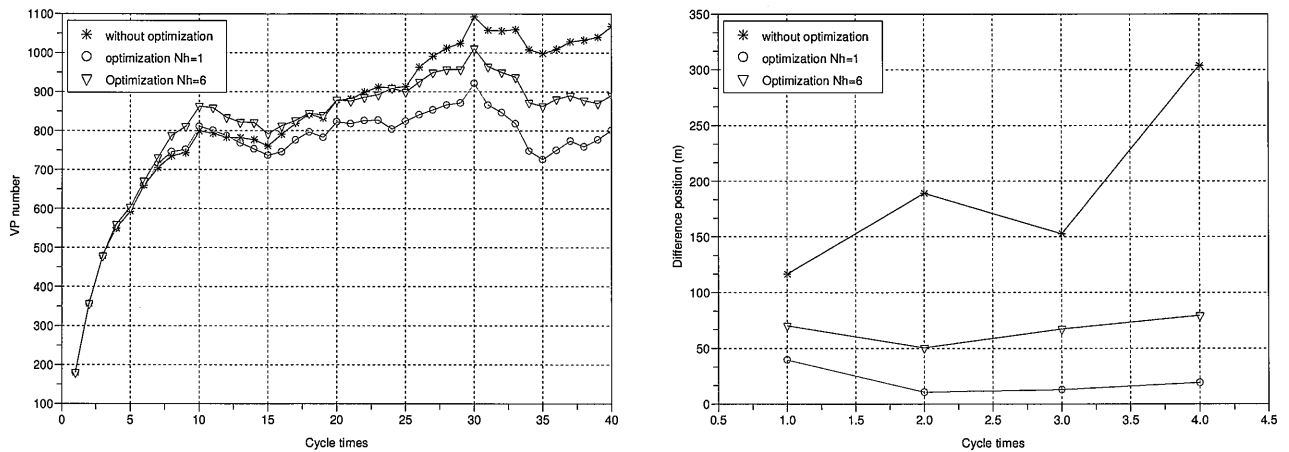


Fig. 2. Comparative results for the modified GCP SO algorithm a) private vehicle criterion b) public transport criterion

difference. Actually, as said above, when we consider an optimization horizon  $N_h > 1$ , the number of decision variables is multiplied by  $N_h$ . Knowing now that the optimums found by metaheuristics in general and the PSO algorithm in particular are very sensitive to the number of decision variables, it can be concluded that the optimums found in the case of  $N_h = 6$  are of worse quality, which leads to the performances above. It is important to add that to have better optimum quality with  $N_h > 1$ , the maximum iteration allowed for the PSO algorithm has to be increased considerably. For the same conditions of the simulation experiment, the modified GCP SO algorithm improves PT progression compared to the original one. And increasing the optimization horizon seems to be of better effect on PT progression. In fact, it was expected that an optimization horizon  $N_h > 1$  improves or at least gives comparable results for the PT progression but as explained above the original PSO algorithm led to non high-quality optimums. When we use the modified GCP SO algorithm, the optimums found are of better quality than

for the original PSO algorithm. Figure 2 illustrates the same variables in the same conditions as in Figure 1.

## 5. CONCLUSION

The paper proposed an urban traffic control strategy that uses traffic light to regulate private vehicle traffic and the progression of public transport vehicles. One of the originalities of this work is the global expression of the two criteria corresponding to the two transport modes. The global multiobjective optimization problem is solved using the Particle Swarm Optimization algorithm (PSO). We considered the original algorithm and a modified algorithm (GCP SO) that guarantees better quality optimums. The simulation results show that the strategy proposed improves the private vehicle traffic and the progression of the public transport vehicles in the network simultaneously. They also show that the optimization horizon (control horizon) in the case of the original PSO algorithm does not have significant consequences on traffic conditions. On the contrary, the modified GCP SO algorithm seems to be

more efficient for the progression of the public transport vehicles for a longer optimization horizon.

#### REFERENCES

- Barrière, J.F., J.L. Farges and J.J. Henry. Decentralization vs Hierarchy in Optimal Traffic control. In *5th IFAC, IFIP, IFORS Conference on Control in Transportation Systems*, pp 321-326, 1986.
- Bhouri, N. and Lotito, P. An intermodal traffic control strategy for private vehicles and public transport. In *10th Euro Working Group on Transportation*, pp 423-428.
- Bhouri, N. Constrained Optimal Control strategy for multimodal urban traffic network. In *IFAC Workshop on Control Applications of Optimization (CAO'09)*.
- Clerc, M. and Kennedy, J. The particle swarm-explosion, stability, and convergence in a multidimensional complex space. In *IEEE Transactions on Evolutionary Computation*, pp 58-73, 2002.
- Diakaki, C., M. Papageorgiou and K. Aboudolas. A multivariable regulator approach to traffic-responsive network-wide signal control. In *Control Engineering Practice*, Pergamon, pp 183-195, 2002.
- Gazis, D.C. Optimum control of a system of oversaturated intersections. In *Operation Research*, 12, pp 815-831, 1964.
- Gazis, D.C. and R.B. Potts. The oversaturated intersection. In *Proceedings of the second international symposium on traffic theory*, pp 221-237, 1963.
- Kachroudi, S. and Bhouri, N. A multimodal model for an urban traffic control policy. In *17th IFAC World Congress*, 2008.
- Kachroudi, S. and Bhouri, N. Modèle de prdiction semi-macroscopique pour les véhicules de transport en commun en milieu urbain. In *Conférence Internationale Francophone d'automatique (CIFA)*, 2008.
- Papageorgiou, M. Traffic control. In *Handbook of transportation engineering*, R.H Hall, Editor, Kluwer Academic Publishers, in press, 1999.
- Reyes-Sierra, M. and Coello Coello, C.A. Multi-Objective Particle Swarm Optimizers: A survey of the state-of-the-art. In *the International Journal of Computational Intelligence Research*, pp 287-308, 2006.
- Shi, Y. and Eberhart, R. Parameter selection in particle swarm optimization. In *Evolutionary Programming VII: Proceedings of the Seventh Annual Conference on Evolutionary Programming*, pp 591-600, 1998.
- Shi, Y. and Eberhart, R. Empirical study of particle swarm optimization. In *Congress on evolutionary computation*, IEEE Press, pp 1945-1950, 1999.
- Van Den Bergh, F. An Analysis of Particle Swarm Optimizers. Ph.D Thesis, Department of Computer Science, University of Pretoria, 2002.



