

Methodology for Parameters Optimization for an Hybrid Architecture of Control

Lounis Adouane, Nadine Le Fort-Piat

▶ **To cite this version:**

Lounis Adouane, Nadine Le Fort-Piat. Methodology for Parameters Optimization for an Hybrid Architecture of Control. 16th IFAC World Congress, Jul 2005, Prague, Czech Republic. hal-01714877

HAL Id: hal-01714877
https://hal.science/hal-01714877

Submitted on 23 Feb 2018

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.

METHODOLOGY FOR PARAMETERS OPTIMIZATION OF AN HYBRID ARCHITECTURE OF CONTROL

Lounis Adouane, Nadine Le Fort-Piat

Laboratoire d'Automatique de Besançon
UMR CNRS 6596, 24 Rue Alain Savary,
25000 BESANÇON, France

Abstract: This paper presents a methodology for parameters optimization applied to an hybrid behavioral control architecture. The coordination between behaviors in this architecture is insured using both hierarchical and fusion action mechanisms. This global mechanism of coordination is characterized by a multitude of parameters which must be finely tuned to enhance the efficiency of the execution of cooperative tasks. The proposed parameters optimization is obtained using genetic algorithms. Appropriate genetic operators are used to manipulate real chromosomes with a normalization relationship between its genes. The validation of the results is established using a large number of simulations. *Copyright © 2005 IFAC*

Keywords: Behavioral architecture of control, Cooperative mobile robotics, Parameters optimization, Genetic algorithms.

1. INTRODUCTION

Behavioral architectures of control are based on the concept that a complex behavior of a robot can be obtained from the coordination of several elementary behaviors. This coordination can be either competitive or cooperative.

In competitive architectures, the command is given by an unique behavior, which has been selected temporarily among a set of active behaviors. The principle of competition can be defined by a set of fixed priorities like in subsumption architecture where a hierarchy is defined between behaviors (Brooks, 1986). This principle can be also dynamic like in the actions selection architecture (Maes, 1989) where the behavior with a great level of activation is selected.

In cooperative architectures, the final command is the result of a compromise or a fusion between different behaviors, which are active at the same time. This include the fuzzy control (Saffiotti *et al.*, 1993) via the process of defuzzification, the schema-based architecture (Arkin, 1989) which uses vectorial addition to encode the robot's behavioral response.

Competitive like cooperative process of selection need to set or to adapt specific parameters according to the task to achieve. These parameters are generally tedious to obtain. For this reason, many works deal with their optimization. Parker (94) uses reinforcement learning for parameters adaptation of motivational behaviors blocks used to enhance the pushing of a long rectangular box. In (Ram, *et al.*, 94), authors use a genetic algorithm to learn appropriate parameters corresponding to schemas motors dedicated for local navigation of a mobile robot.

This paper deals with Cooperative Box-Pushing Task "***CBPT***". The proposed architecture of control combines the coordination principles of subsumption and motor schemas to obtain an efficient

control architecture for cooperative robotics. The appropriate parameters of fusion blocks are determined using specific parameters optimization based on *genetic algorithms*.

The rest of this paper is organized as follows. Section 2 gives the specifications of the ***CBPT*** and a short description of the simulated mini-robots. Section 3 presents the details of the proposed hybrid architecture of control. Section 4 is devoted to the description and the analysis of the proposed methodology for parameters optimization. A brief description of ***MiRoCo*** simulator is also done. Conclusions and future directions are presented at the end of the paper.

2. COOPERATIVE BOX-PUSHING TASK

CBPT remains a privileged complex task for the study of reactive and distributed features for cooperative robotics control (Kube, 1997). This task is difficult to control due to the large dynamic of entities interactions in direct surrounding of the object to push. In addition to that, the used robots in this study are very limited in structural, decisional and sensory point of views. Indeed, the mobile mini-robot named ALICE (Fig. 1) has very reduced dimensions of 2cm×2cm×2cm (Caprari *et al.*, 2002) and has, in its original structure, only four infrared sensors *IS1*, *IS2*, *IS3*, *IS4* which measure light intensity and local distance to obstacles.

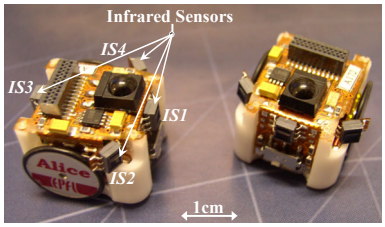


Fig. 1. Mini-robots ALICE

The ***CBPT*** can be summarized as follows. The environment contains a number $\boldsymbol{N}$ of robots that must push a box $\boldsymbol{B}$ towards a circular target $\boldsymbol{T}$. The displacement of $\boldsymbol{B}$ requires in less the cooperation of a critical number $\boldsymbol{Nc}$ of robots.

3. CONTROL ARCHITECTURE

The proposed behavioral architecture of control (Fig. 2) is composed by a set of elementary behaviors (Adouane and LeFort-Piat, 2004*a*) where each of them can send a command $\mathcal{C}$ among a set of *discrete* commands $\mathbb{C}_d$:

- Go Forward $\mathbf{GF} = \mathcal{C}(0°,\ \boldsymbol{Tra}\text{ cm})$,
- Go Backward $\mathbf{GB} = \mathcal{C}(0°,\ \text{-}\boldsymbol{Tra}\text{ cm})$,
- Go Right-Rotate $\mathbf{GRR} = \mathcal{C}(\text{-}\boldsymbol{Rot}°,\ 0\text{ cm})$,
- Go Left-Rotate $\mathbf{GLR} = \mathcal{C}(\boldsymbol{Rot}°,\ 0\text{ cm})$,
- Go Right-Turn $\mathbf{GRT} = \mathcal{C}(\text{-}\boldsymbol{Rot}°,\ \boldsymbol{Tra}\text{ cm})$,
- Go Left-Turn $\mathbf{GLT} = \mathcal{C}(\boldsymbol{Rot}°,\ \boldsymbol{Tra}\text{ cm})$,
- Remain There $\mathbf{RT} = \mathcal{C}(0°,\ 0\text{ cm})$.

Where: ***Rot*** and ***Tra*** constants correspond respectively to rotational and translatory commands which will be executed by the robot in a coupled movement.

Each mini-robot can measure the θ angle that it makes with an active landmark in the environment. θ angle goes from -180° to 180° and it can be measured with an incertitude of ±15°.

3.1 Elementary behaviors

Exploration This behavior sends commands to the motors according to predetermined constant coefficients of the occurrence probability of each command $\mathcal{C} \in \mathbb{C}_d$.

Obstacles avoiding This behavior uses the four infrared sensors only in boolean mode. It consists therefore in avoiding the obstacles according to a pre-established strategy giving the response according to the sixteen possible stimuli of the infrared sensors.

Attraction to the Box This behavior consists in attracting the mini-robot towards the box, which broadcasts light. This is performed using the infrared sensors just in reception mode.

Alignment This behavior consists in insuring that the mini-robot is aligned with the box $\boldsymbol{B}$ before activating the behavior of *Box-pushing* or the one of *Repositioning*. Therefore, this behavior controls the relative angle δ that the mini-robot makes with $\boldsymbol{B}$, so that $\delta \in [\text{-}\eta\ °,\ \eta°]$.

Box-pushing After the detection of θ_1 and θ_2 angles that the mini-robot makes with respectively the box $\boldsymbol{B}$ and with the target $\boldsymbol{T}$ (Fig. 3), this behavior generates a response according to the following rule:

> ***If*** $((|\theta_1| \text{and } |\theta_2|) \leq \theta_{Max})$ ***Then*** apply the **GF**
> ***Else*** apply **RT** command.

With θ_{Max} is a positive constant less than 90°.

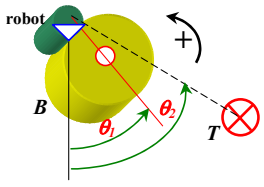


Fig. 3. *Box-pushing* necessary information

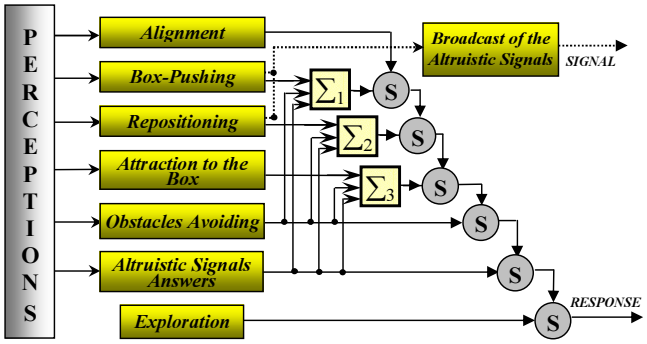


Fig. 2. Control based hybrid architecture to achieve the ***CBPT***

Repositioning The aim of this behavior is to reposition appropriately and quickly the robot around $\boldsymbol{B}$. According to the measured θ angle between the robot and the target $\boldsymbol{T}$ at the instant "t", this behavior generates a sequence of commands as following:

- the robot rotates with an angle $\alpha[^\circ]$ such as

$$\alpha = \boldsymbol{f}(\theta) \tag{1}$$

- after the rotation is ended, the robot moves forward for a distance of $\boldsymbol{d}$[cm] such as

$$\boldsymbol{d} = \boldsymbol{g}(\theta) \tag{2}$$

Where α and $\boldsymbol{d}$ are as big as $|\theta|$ is big.

Altruistic behaviors The altruism is the fact to generate an effect (an action) with the objective to help its neighbors, and this without immediate obvious gain for the entity that generates this effect. The notion of altruism introduced in the proposed architecture of control is represented by the following two behaviors.

a - Broadcast of the altruistic signals: This behavior is activated when behaviors of *Box-pushing* or *Repositioning* are activated (Fig. 2). This purely altruistic behavior consists in giving out attractive or repulsive signals (Fig. 4) as follows.

> ***If*** $(|\theta| < \theta_{\boldsymbol{Max}})$ ***Then*** the signal is attractive
> ***Else*** the signal is repulsive.

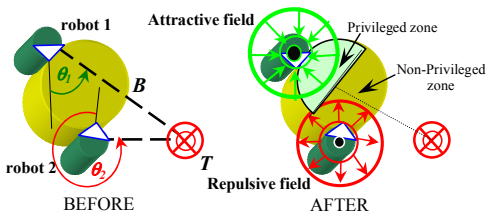


Fig. 4. Attractive and repulsive signals

These signals have the objectives to attract mini-robots around the privileged zone (zone where the mini-robots are susceptible to appropriately push $\boldsymbol{B}$ toward $\boldsymbol{T}$) and at opposite, to repulse mini-robots from non-privileged zone. Figure 4 shows roughly these two zones.

b - Altruistic signals answers: The answers to the altruistic signals are commands that attract the mini-robot towards the most attractive altruistic signal (of bigger intensity) and move away from the most repulsive altruistic signal received by the four infrared sensors.

3.2 Coordination of behaviors

Hierarchical coordination In the proposed architecture of control, the hierarchical coordination between behaviors is managed entirely through their responses. Indeed, among all the possible responses given by each behavior, it exists a specific response called ***Refuge Response*** "***RR***" (Adouane and LeFort-Piat, 2004*a*) which is generated for a particular stimulus of sensors.

The principle of using the "***RR***" to hierarchically coordinate a set of behaviors is the following: starting from the behavior "n" of highest level in the architecture of control (*Alignment* behavior in (Fig. 2)) ***If*** it generates a command $\mathcal{C}_n \neq$ "$\boldsymbol{RR}_n$" ***Then*** $\mathcal{C}_n$ is executed by robot's motors ***Else*** behavior "n" gives the hand to the behavior "n-1" which applies the same rule. Thus, by chain effect, the hierarchical coordination is achieved.

Fusion organization To avoid undesirable effects linked to the pure hierarchical coordination, fusion blocks $\Sigma_1, \Sigma_2, \Sigma_3$ (Fig. 2) are introduced. These blocks compute a command as follows:

For each command "$\mathcal{C}_j(\boldsymbol{Rot}_j,\ \boldsymbol{Tra}_j)|_{j=1..3}$" in input of a fusion block $\Sigma_i|_{i=1..3}$ is associated a gain, respectively $\boldsymbol{g}_{ij}|_{j=1..3}$ such as:

$$\sum_{j=1..3} \boldsymbol{g}_{ij} = 1 \qquad with : \boldsymbol{g}_{ij}\,|_{j=1..3} \in]0\ 1[\tag{3}$$

The output of the fusion block $\Sigma_i|_{i=1..3}$ is a command $\mathcal{C}_i(\boldsymbol{Rot}_i,\ \boldsymbol{Tra}_i)$ given by this method:
If *Box-pushing*, *Repositioning*, or *Attraction to the box* behaviors give their ***RR*** ***Then*** the corresponding fusion block (respectively Σ_1, Σ_2 and

Σ_3) gives in output this ***RR*** which allows to not inhibit behaviors or fusion blocks of lower levels. ***Else*** the $\boldsymbol{Rot}_i$ and $\boldsymbol{Tra}_i$ components of $\mathcal{C}_i$ are computed as follows:

$$\boldsymbol{Rot}_i = \angle \left(\sum_{j=1..n} (\boldsymbol{g}_{ij} \cdot \overrightarrow{v}_{ij}) \right) \quad (4)$$

Where $\overrightarrow{v}_{ij}$ are unit vectors as $\angle \overrightarrow{v}_{ij}$ (angle $\overrightarrow{v}_{ij}$) = "$\boldsymbol{Rot}_j$".

$$\boldsymbol{Tra}_i = \sum_{j=1..n} (\boldsymbol{g}_{ij} \cdot \boldsymbol{Tra}_j) \quad (5)$$

4. PARAMETERS OPTIMIZATION

To improve the achievement of the ***CBPT***, the mini-robots must compute their commands in order to satisfy many criteria at the same time. For the case of Σ_3 which has as global objective to lead the robot toward the box, the different criteria are as follows: to move toward the box while avoiding others mini-robots (and/or obstacles) and if the mini-robot detects altruistic signals, to react consequently.

The choice of $\mathbf{g}_{ij}|_{i=1..3,j=1..3}$ gains of the fusion blocks Σ_1, Σ_2, Σ_3 can be obtained empirically like in (Adouane and LeFort-Piat, 2004*b*) where they are chosen according to the emergence of some visual configurations of the mobile robots. However, this phase can be tedious. Thus to avoid a too long manual trial-error method to obtain the more suitable gains for the fusion blocks, we propose to use *genetic algorithms* "GA" which are very suitable for multicriteria and parameters optimization. In what follows, the proposed methodology of optimization will be explained through its application on the gains $\mathbf{g}_{ij}|_{i=3,j=1..3}$ of Σ_3 fusion block.

The *population* used for the GA is constituted of a large number of *chromosomes*. Each of them contains three genes corresponding the the gains attributed to Σ_3 and which must satisfy (3).

These gains coupled to the commands generated by corresponding elementary behaviors (i.e., *Attraction to the box*, *Obstacle avoiding* and *Altruistic signals answers*) characterize completely the motion of the mini-robot when the fusion block Σ_3 is activated. Therefore, commands generated by Σ_3 should give the best compromise between keys objectives that lead the robot toward the box.

Figure 5 shows the method applied to the evaluation of each chromosome of the population.

The *temporary fitness evaluation* $\boldsymbol{Fitness}_{kij}$ of one chromosome, is obtained according to (6):

$$\boldsymbol{Fitness}_{kij} = \alpha_1 \times \boldsymbol{CRB}_{kij} + \alpha_2 \times \boldsymbol{DRB}_{kij} - \alpha_3 \times \boldsymbol{DRO}_{kij} + \alpha_4 \times \boldsymbol{DRSA}_{kij} - \alpha_5 \times \boldsymbol{DRSR}_{kij} \quad (6)$$

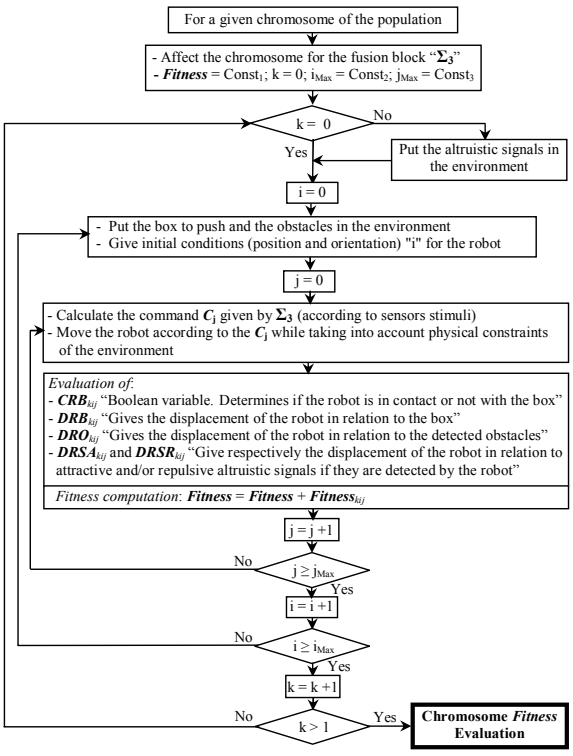


Fig. 5. Adopted methodology for evaluate each chromosome affected to Σ_3

Where:

- $\boldsymbol{CRB}_{kij}$ is a boolean variable which represents the collision or not of the robot with the box, ($\boldsymbol{CRB}_{kij}$ = '1' if the robot collides with the box, and '0' else),
- $\boldsymbol{DRB}_{kij}$, $\boldsymbol{DRO}_{kij}$, $\boldsymbol{DRSA}_{kij}$, $\boldsymbol{DRSR}_{kij}$ represent the relative displacement of the robot in relation to the box, obstacles and altruistic signals (attractive and repulsive) respectively, and this when they are detected by the robot's sensors. These relative displacements give the means to estimate if the robot goes closer (positive value) or moves away (negative value) from the detected agents,
- $\alpha_i|_{i=1..5}$ are the priority (balance) attributed to each corresponding criteria to achieve the *attraction to the box* behavior. We chose to give more priority respectively to the collision with the box $\alpha_1 = 2$ to give more favor to chromosome which leads to reach the box quickly; $\alpha_2 = 1.4$ to go toward the box; $\alpha_3 = 1$ to avoid obstacles; α_4=α_5=0.4 to respond to altruistic signals.

The global fitness ***Fitness*** affected to each chromosome is calculated as follows:

$$\boldsymbol{Fitness} = \sum_{k=0}^{1} \sum_{i=0}^{i_{Max}} \sum_{j=0}^{j_{Max}} \boldsymbol{Fitness}_{kij} \quad (7)$$

Where: "k" corresponds to the iteration made with and without altruistic signals surrounding the box; "i" represents different initial positions

and orientations taken by the robot at each start of simulation; and "j" corresponds to the number of steps before stopping the simulation number "i".

4.1 Simulation environment

To apply the proposed methodology of parameters optimization, we have used the ***MiRoCo*** (*Mini-Robotics Collective*) simulator (Fig. 6) which permits to have a good approximation of: ALICE mini-robot structure, the displacement and the accurate interaction between the mini-robot and its environment, etc. This simulator allows also to make a very big number of simulations to evaluate each chromosome of the population.

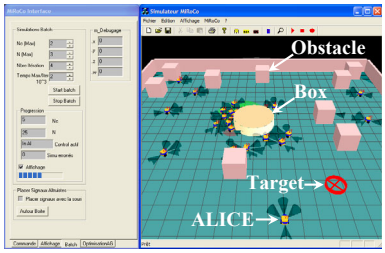


Fig. 6. ***MiRoCo***, a simulator for collective mobile robotics

Figure 7 shows the setup used to perform batch simulations. It shows also an initial position and orientation taken by ALICE to evaluate chromosomes.

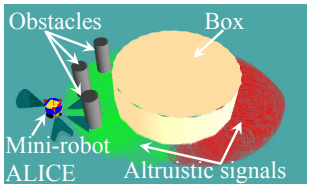


Fig. 7. Batch simulation setup

4.2 Genetic cycles

To achieve genetic cycles which allow to obtain optimal chromosome (gains) for Σ_3 which maximize (7), we use appropriate selection, crossover and mutation operators. These operators are adapted to manipulate real genes (Guvenir and Erel, 1998), (Muhlenbein and Schlierkamp-Voosen, 1993), while taking into account (3).

Selection operator This is performed using the roulette wheel selection (Goldberg, 1991).

Crossover operator This operator allows to cross two chromosomes while preserving the sum of any sequence of genes. Given two normalized chromosomes $x = \langle x_1,\ x_2,\ x_3\rangle$ and $y = \langle y_1,\ y_2,\ y_3\rangle$, the obtention of two normalized offsprings $x' = \langle x'_1,\ x'_2,\ x'_3\rangle$ and $y' = \langle y'_1,\ y'_2,\ y'_3\rangle$ is such as:

$$\begin{cases} x'_i = sx_i + (1-s)y_i \\ y'_i = (1-s)x_i + sy_i \end{cases} \tag{8}$$

where "s" is a stride and changes randomly between $[0,\ 0.5]$ for each crossover operation.

Mutation operator This operator transforms the value of a random gene to either 0 or 1 with equal probability. Therefore, if a chromosome undergone a mutation, it must be normalized according to (9).

$$x_{i=1..3} = \frac{x_i}{\sum_{j=1}^{3} x_j}. \tag{9}$$

4.3 Simulations Results

The genetic algorithm is applied on a population of 1000 chromosomes, with the probability of crossover and mutation equal respectively to 0.8 and 0.02.

Figure 8 shows the evolution of the maximum and average fitness of consecutive populations. Optimal chromosome is obtained after 120 genetic cycles and it corresponds to:

- Σ_3^*: $\boldsymbol{g}_{31}^*$= 0.181, $\boldsymbol{g}_{32}^*$= 0.400, $\boldsymbol{g}_{33}^*$= 0.419

For the fusion blocks Σ_1 and Σ_2 (Fig. 2), we consider the same parameters as in (Adouane and LeFort-Piat, 2004*b*):

- Σ_1: $\boldsymbol{g}_{11}$= 0.500, $\boldsymbol{g}_{12}$= 0.400, $\boldsymbol{g}_{13}$= 0.100
- Σ_2: $\boldsymbol{g}_{21}$= 0.500, $\boldsymbol{g}_{22}$= 0.450, $\boldsymbol{g}_{23}$= 0.050

With: $\boldsymbol{g}_{11}$, $\boldsymbol{g}_{21}$, $\boldsymbol{g}_{31}^*$ are gains of commands corresponding respectively to *Box-pushing*, *Repositioning* and *Attraction to the box* behaviors; $\boldsymbol{g}_{12}$, $\boldsymbol{g}_{22}$, $\boldsymbol{g}_{32}^*$ correspond to *Obstacles avoiding* behavior and $\boldsymbol{g}_{13}$, $\boldsymbol{g}_{23}$, $\boldsymbol{g}_{33}^*$ correspond to *Altruistic signals answers* behavior.

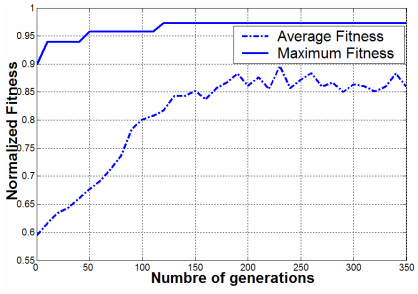


Fig. 8. Evolution of the GA optimization

To validate the relevance of the proposed evaluation methodology, which assigns a fitness to chro-

mosomes according to their adaptation to achieve the *attraction to the box* sub-task, we take the optimal chromosome $x^* = \langle 0.181,\ 0.400,\ 0.419 \rangle$ and the chromosome $y = \langle 0.596,\ 0.397,\ 0.006 \rangle$ which has a fitness value on the middle part of the final population.
The test consists in putting sequentially each of these two chromosomes on the fusion block Σ_3 and to perform a large number of ***CBPT*** with the global architecture of control (Fig. 2). The minimum number of robots to push the box ***Nc*** was chosen equal to 2. In the simulations, the number of robots ***N*** is increased from 2 to 11, and for each ***N***, 50 simulations were made. Initial conditions of the simulations are the same for both chromosomes.

Figure 9 shows the number of activation ***NA*** of the fusion block Σ_3 for all the robots which participate to achieve the ***CBPT***. Figure 10 shows the average time needed by the robots to lead the box from its initial position toward the target.

We observe that the ***NA*** corresponding to the best chromosome are smaller than those obtained for the less fitness chromosome. ***NA*** smaller indicates that robots found quickly the box and contribute thus actively to push it. This is confirmed in (Fig. 9) where the time necessary to achieve the ***CBPT*** becomes smaller when the best chromosome is used by the architecture of control.

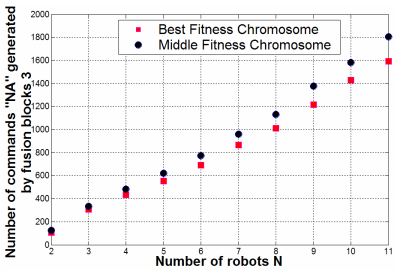


Fig. 9. ***NA*** of Σ_3 according to ***N***

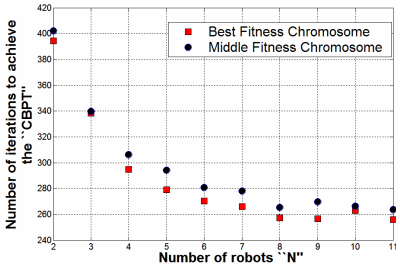


Fig. 10. Evolution of the number of iterations according to ***N***

5. CONCLUSIONS AND FUTURE WORKS

A methodology of parameters optimization for an hybrid architecture of control was proposed in this paper. The optimization has been done using a genetic algorithm via appropriate genetic operators, which deals with real value chromosomes. The adopted genetic operators prove to be very efficient to obtain quick convergence of the genetic algorithm toward appropriate parameters. The validity of the optimization is confirmed through a batch of cooperative box-pushing task simulations. Future works are going to use multiobjective optimization like Pareto-optimal method in order to attribute appropriate parameters of control according to robot's ecological niche. Experimentations with the physical mini-robots ALICE endowed with optimal parameters of control are going also to be made.

REFERENCES

Adouane, L. and N. LeFort-Piat (2004*a*). Emergence of group intelligence from minimalist control of mobile mini-robots. In: *35th International Symposium On Robotics*. France.

Adouane, L. and N. LeFort-Piat (2004*b*). Hybrid behavioral control architecture for the cooperation of minimalist mobile robots. In: *ICRA'04*. USA. pp. 3735–3740.

Arkin, R.C. (1989). Motor schema-based mobile robot navigation. *International Journal of Robotics Research,* 8(4), pp.92–112.

Brooks, R.A. (1986). A robust layered control system for a mobile robot. *Journal of Robotics and Automation, RA – 2*, pp.14–23.

Caprari, G., T. Estier and R. Siegwart (2002). Fascination of down scaling - Alice the sugar cube robot. *Journal of Micro-Mechatronics,* pp. 177–189.

Goldberg, David E. (1991). *Gentic Algorithms*. Addison-Wesley USA.

Guvenir, H.A. and E. Erel (1998). Multicreteria inventory classification using a genetic algorithm. *European Journal Of Operational Research,* pp. 29–37.

Kube, C.R. (1997). Collective Robotics: From Local Perception to Global Action. PhD thesis. Univeristy of Alberta Canada.

Maes, P. (1989). The dynamics of action selection. In: *IJCAI'89*. Detroit. pp. 991–997.

Muhlenbein, H. and D. Schlierkamp-Voosen (1993). Predictive models for the breeder genetic algorithm. I: Continuous parameter optimization. *Evol. Comput.* 1(1), pp.25–49.

Parker, L.E. (1994). Heterogeneous Multi-Robot Cooperation. PhD thesis. MIT.

Ram, A. and R. Arkin and G. Boone and M. Pearce (1994). Using genetic algorithms to learn reactive control parameters for autonomous robotic navigation. *Adaptive Behavior,* 2(3), pp.277–305.

Saffiotti, A., E. Ruspini and K. Konolige (1993). Robust execution of robot plans using fuzzy logic. In: *Fuzzy Logic in Artificial Intelligence: IJCAI'93*. France. pp. 24–37.