

Adaptive-based Distributed Cooperative Multi-Robot Coverage

Alessandro Renzaglia, Lefteris Doitsidis, Agostino Martinelli, Elias Kosmatopoulos

► **To cite this version:**

Alessandro Renzaglia, Lefteris Doitsidis, Agostino Martinelli, Elias Kosmatopoulos. Adaptive-based Distributed Cooperative Multi-Robot Coverage. American Control Conference 2011, Jul 2011, San Francisco, United States. 2011. <inria-00607085>

HAL Id: inria-00607085

<https://hal.inria.fr/inria-00607085>

Submitted on 7 Jul 2011

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.

Adaptive-based Distributed Cooperative Multi-Robot Coverage

Alessandro Renzaglia, Lefteris Doitsidis, Agostino Martinelli and Elias B. Kosmatopoulos

Abstract—In this paper we present a solution to the problem of positioning a team of mobile robots for a surveillance task in a non-convex environment with unknown obstacles. The problem is addressed taking into account several physical and environmental constraints like limited sensor capabilities, obstacle-avoidance, etc. The goal is to maximize the area monitored by the team, by identifying the best configuration of the team members. Due to the non-convex nature of the problem, an analytical solution cannot be obtained. The proposed method is based on a new cognitive-based, adaptive optimization algorithm (CAO). This method allows getting coordinated and scalable controls to accomplish the task. Furthermore, we propose a different formulation of the problem in order to obtain a distributed solution which allows us to consider also limited communication capabilities. Extensive simulations are presented to evaluate the efficiency of the proposed algorithm and to compare centralized and distributed approach.

I. INTRODUCTION

The use of multi-robot teams has gained a lot of attention in recent years. This is due to the extended capabilities that the teams have to offer comparing to the use of a single robot for the same task. Robot teams can be used in a variety of mission including: surveillance in hostile environments (i.e. areas contaminated with biological, chemical or even nuclear wastes), environmental monitoring (i.e. air quality monitoring, forest monitoring) and law enforcement missions (i.e. border patrol), etc. In all the aforementioned tasks the positioning of limited resources (robots) to maximize the area monitored is the key issue. This can be achieved by optimizing the way that the robots are deployed so that the area monitored by each team member is maximized and at the same time the overlap of these areas is minimized.

Furthermore, in every multi-robot systems, a distributed approach is desirable for several fundamental reasons. The most important are failure of the central station and limited communication capabilities. In a very common scenario each robot has no global knowledge about the surrounding environment or about the group as whole. So, the global behavior of the team can be seen as the sum of the local actions taken by its members, which sense their immediate environment, communicate with their neighbors, process the

information gathered and move according to it. As far as it concerns the optimal coverage using a team of robots, two problems have been identified and formally approached up to now, both by introducing a suitable optimization function. The first problem deals with the optimal arrangement of the team members, so that for every point in the area to be covered, the closest robot is as close as possible to that point. This corresponds to the exigency of having the possibility to intervene as fast as possible, in all the points of the area with at least one robot. In this case, the corresponding cost function which will be minimized depends on the distance of the robots from the points of the area to be covered. The second problem deals with the maximization of the area monitored by a team of robots using vision sensors. In this case, a point is considered monitored as long as it is in the field of view of at least one vision sensor.

Both problems can be solved in very special situations, i.e. when the environment is convex and without obstacles, which is not the case in the majority of real scenarios. Regarding the second problem, the visibility of a given point is often assumed to be independent of the distance of the point from the robot, which is unrealistic since the performance of the optical sensors decreases with the distance. There are cases that when a point is in the field of view of a vision sensor, it can not be monitored because its distance is larger than a given threshold (the value of this threshold depends on many factors, e.g. the resolution of the adopted vision sensor).

Several approaches have been proposed in the literature considering the first problem previously described. In [1], the authors present a controller for the coverage with a team of mobile robots in a convex environment, i.e. without obstacles, based on the Voronoi partition. A similar approach, for a convex environment, is proposed in [2], where additionally the robots estimate a function indicating the relative importance of different areas in the environment, using information from the sensors. An approach for non-convex regions is proposed in [3]. In this work the Voronoi partition is obtained by using the geodesic distance instead of the Euclidean one taking into account the particular topology of the problem. A possible solution for environments with obstacles is proposed also in [4]: the idea is to combine the classical Voronoi coverage and the local path planning algorithm TangentBug.

As far as it concerns the second problem previously described, different solutions have been proposed in the literature. In [5] the authors propose a gradient-based algorithm for the case of a single robot case. In [6], an approach for the multi-robot problem based on the assumption that the

A. Renzaglia and A. Martinelli are with INRIA Rhône-Alpes, Grenoble, France email: `firstname.lastname@inrialpes.fr`

L. Doitsidis is with the Department of Electronics, Technological Educational Institute of Crete, Chania, Greece & Informatics & Telematics Institute, CERTH, Thessaloniki, Greece `ldoitsidis@chania.teicrete.gr`

E.B. Kosmatopoulos is with the Department of ECE, Democritus University of Thrace, Xanthi, Greece & Informatics & Telematics Institute, CERTH, Thessaloniki, Greece `kosmatop@dssl.tuc.gr`

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement n. 231855 (sFly).

environment is simply connected is presented. The visibility problem is also related with the Art Gallery Problem, where the goal is to find the optimum number of guards so that each point of the environment is visible by at least one guard [15]. All the aforementioned solutions are based on the hypothesis that a given point can be monitored regardless of its distance from the vision sensor.

In this paper we propose a distributed and scalable algorithm to maximize the monitored areas in a 2D non-convex environment, even if it is unknown¹, by using a team of mobile robots. In particular, we consider the second problem previously mentioned by also accounting a maximum distance in the region visible from a vision sensor. The centralized version of the proposed method has been considered in [7], [8]. To the best of our knowledge this problem has never been approached by following a distributed and coordinated control strategy. On the other hand, an incremental deployment algorithm can be found in [14]. Our goal is to approach this problem by introducing a learning strategy able to provide a coordinated control algorithm for all the team members. In particular, the proposed approach is based on the Cognitive-based Adaptive Optimization (CAO) methodology. The CAO methodology, which was recently introduced in [10], [11], possesses the capability of being able to efficiently handle optimization problems for which an analytical form of the function to be optimized is unknown, but the function is available for measurement at each iteration of the algorithm employed to optimize it. As a result, it perfectly suits for multi-robot optimal coverage in non-convex environments, where the analytical form of the function to be optimized is unknown but the function is available for measurement (through the robots' sensors) for each multi-robot configuration. The CAO approach extends the popular Simultaneous Perturbation Stochastic Approximation (SPSA) algorithm [12]. The difference between the SPSA and the CAO approach is that SPSA employs an approximation of the gradient of an appropriate cost function using only the most recent experiments, while the CAO approach employs linear-in-the-parameters approximators that incorporate information of a user specified time window of the past experiments together with the concept of candidate perturbations for efficiently optimizing the unknown function.

It is finally mentioned that the CAO or the SPSA do not create an approximation or estimation of the obstacles location and geometry; instead, they on-line produce a local approximation of the unknown cost function to optimize. For this reason, they require simple, and thus scalable, approximation schemes to be employed.

This paper is organized as follows. In the next section we describe the stochastic optimization approach used in this work, the CAO algorithm, and how it is applicable to a generic coverage-like problem. Then, in section III, we show explicitly the proposed solution based on the CAO

¹In this case each robot has to be equipped also with range sensors in order to get metric information from the environment (as it has been done in the past when the environment is unknown [6], [14]).

algorithm, before for the standard centralized problem and then for the proposed distributed version. Finally, in section IV, we present the results of the numerical simulations.

II. THE COGNITIVE-BASED ADAPTIVE OPTIMIZATION APPROACH

The Cognitive-based Adaptive Optimization (CAO) approach [9]-[11] was originally developed and analyzed for the optimization of functions for which an explicit form is unknown but their measurements are available as well as for the adaptive fine-tuning of large-scale nonlinear control systems. In this section, we will describe how the CAO approach can be appropriately adapted and extended so that it is applicable to the problem of multi-robot coverage. More explicitly, let us consider the problem where M robots are involved in a coverage task, attempting to maximize a given *coverage criterion*. Apparently, the coverage criterion is a function of the robots' positions or poses (positions and orientations), i.e.

$$J_k = \mathcal{J} \left(x_k^{(1)}, \dots, x_k^{(M)} \right) \quad (1)$$

where $k = 0, 1, 2, \dots$ denotes the time-index, J_k denotes the value of the coverage criterion at the k -th time-step, $x_k^{(1)}, \dots, x_k^{(M)}$ denote the position vectors of robots $1, \dots, M$, respectively, and \mathcal{J} is a nonlinear function which depends – apart from the robots' positions – on the particular environment where the robots live; for instance, in the 2D case the function \mathcal{J} depends on the location of the various obstacles that are present, while in the 3D case with flying robots monitoring a terrain, the function \mathcal{J} depends on the particular terrain morphology.

Due to the dependence of the function \mathcal{J} on the particular environment characteristics, the *explicit form of the function* \mathcal{J} is *not known* in most practical situations; as a result, standard optimization algorithms (e.g. steepest descent) are not applicable to the problem in hand. However, in most practical cases – like the one treated in this paper – the current value of the coverage criterion can be estimated from the robots' sensor measurements. In other words, at each time-step k , an estimate of J_k is available through robots' sensor measurements,

$$J_k^n = \mathcal{J} \left(x_k^{(1)}, \dots, x_k^{(M)} \right) + \xi_k \quad (2)$$

where J_k^n denotes the estimate of J_k and ξ_k denotes the noise introduced in the estimation of J_k due to the presence of noise in the robots' sensors. Please note that, although it is natural to assume that the noise sequence ξ_k is a stochastic *zero-mean* signal, it is not realistic to assume that it satisfies the typical Additive White Noise Gaussian (AWNG) property even if the robots' sensor noise is AWNG: as \mathcal{J} is a nonlinear function of the robots' positions (and thus of the robots' sensor measurements), the AWNG property is typically lost.

Apart from the problem of dealing with a criterion for which an explicit form is not known, efficient robot coverage algorithms have additionally to deal with the problem of *restricting the robots' positions* so that obstacle avoidance

as well as robot formation constraints are met. In other words, at each time-instant k , the vectors $x_k^{(i)}, i = 1, \dots, M$ should satisfy a set of constraints which, in general, can be represented as follows:

$$\mathcal{C} \left(x_k^{(1)}, \dots, x_k^{(M)} \right) \leq 0 \quad (3)$$

where \mathcal{C} is a set of nonlinear functions of the robots' positions. As in the case of \mathcal{J} , the function \mathcal{C} depends on the particular environment characteristics (e.g. location of obstacles, terrain morphology) and an explicit form may be not known in many practical situations; however, it is natural to assume that the coverage algorithm is provided with information whether a particular selection of robots' positions satisfies or violates the set of constraints (3).

Given the mathematical description presented above, the multi-robot coverage problem can be mathematically described as the problem of moving $x_k^{(1)}, \dots, x_k^{(M)}$ to a set of positions that solves the following constrained optimization problem:

$$\begin{aligned} & \text{maximize} && (1) \\ & \text{subject to} && (3). \end{aligned} \quad (4)$$

As already noticed, the difficulty in solving – in real-time and in real-life situations – the constrained optimization problem (4) lies in the fact that explicit forms for the functions \mathcal{J} and \mathcal{C} are not available. To circumvent this difficulty, the CAO approach is adopted – appropriately modified to be applicable to the problem in hand – which is capable of efficiently dealing with optimization problems for which the explicit forms of the objective function and constraints are not known, but noisy measurements/estimates of these functions are available at each time-step. Next we describe the CAO approach as applied to the multi-robot coverage problem described above.

As a first step, the CAO approach makes use of function approximators for the estimation of the unknown objective function \mathcal{J} at each time-instant k according to

$$\hat{J}_k \left(x_k^{(1)}, \dots, x_k^{(M)} \right) = \vartheta_k^T \phi \left(x_k^{(1)}, \dots, x_k^{(M)} \right). \quad (5)$$

Here $\hat{J}_k \left(x_k^{(1)}, \dots, x_k^{(M)} \right)$ denotes the approximation/estimation of \mathcal{J} generated at the k -th time-step, ϕ denotes the nonlinear vector of L regressor terms, ϑ_k denotes the vector of parameter estimates calculated at the k -th time-instant and L is a positive user-defined integer denoting the size of the function approximator (5). The parameter estimation vector ϑ_k is calculated according to

$$\vartheta_k = \underset{\vartheta}{\operatorname{argmin}} \frac{1}{2} \sum_{\ell=\ell_k}^{k-1} \left(J_\ell^n - \vartheta^T \phi \left(x_\ell^{(1)}, \dots, x_\ell^{(M)} \right) \right)^2 \quad (6)$$

where $\ell_k = \max\{0, k-L-T_h\}$ with T_h being a user-defined nonnegative integer. Standard least-squares optimization algorithms can be used for the solution of (6).

Remark 1: In order for the proposed methodology to guarantee with efficient performance, special attention has to be paid in the selection of the regressor vector ϕ . Polynomial or polynomial-like regressor vectors as well as sigmoidal

regressor vectors can be employed for the construction of ϕ . The particular choice adopted in this paper is described in section III. See [9]-[11] for more details on the design considerations for the regressor vector. \diamond

As soon as the estimator \hat{J}_k is constructed according to (5), (6), the set of new robots' positions is selected as follows: firstly, a set of N candidate robots' positions is constructed according to²

$$x_k^{i,j} = x_k^{(i)} + \alpha_k \zeta_k^{i,j}, i \in \{1, \dots, M\}, j \in \{1, \dots, N\}, \quad (7)$$

where $\zeta_k^{i,j}$ is a zero-mean, unity-variance random vector with dimension equal to the dimension of $x_k^{(i)}$ and α_k is a positive real sequence which satisfies the conditions:

$$\lim_{k \rightarrow \infty} \alpha_k = 0, \quad \sum_{k=1}^{\infty} \alpha_k = \infty, \quad \sum_{k=1}^{\infty} \alpha_k^2 < \infty. \quad (8)$$

Among all N candidate new positions $x_k^{1,j}, \dots, x_k^{M,j}$, the ones that correspond to *non-feasible positions* – i.e. the ones that violate the constraints (3) – are neglected and then the new robots' positions are calculated as follows:

$$\left[x_{k+1}^{(1)}, \dots, x_{k+1}^{(M)} \right] = \underset{\substack{j \in \{1, \dots, N\} \\ x_k^{i,j} \text{ not neglected}}}{\operatorname{argmax}} \hat{J}_k \left(x_k^{1,j}, \dots, x_k^{M,j} \right)$$

The idea behind the above logic is simple: at each time-instant a set of many candidate new robots' positions is generated and the candidate – among the ones that provide with a feasible solution – that provides the “best” estimated value \hat{J}_k of the coverage criterion is selected as the new set of robots' positions. The random choice for the candidates is essential and crucial for the efficiency of the algorithm, as such a choice guarantees that \hat{J}_k is a reliable and accurate estimate for the unknown function \mathcal{J} ; see [10], [11] for more details. On the other hand, the choice of a slowly decaying sequence α_k – a typical choice of adaptive gains in stochastic optimization algorithms, see e.g. [13] – is essential for “filtering out” the effects of the noise term ξ_k [cf. (2)]. The next theorem summarizes the properties of the CAO algorithm described above; it has to be emphasized that the proof of this theorem is among the same lines as the main results of [10], [11]:

Theorem 1: Let $x^{(1^*)}, \dots, x^{(M^*)}$ denote any – local – minimum of the constrained optimization problem (4). Assume also that the functions \mathcal{J}, \mathcal{C} are either continuous or discontinuous with a finite number of discontinuities. Then, the CAO-based multi-robot coverage algorithm as described above guarantees that the robots' positions $x_k^{(1)}, \dots, x_k^{(M)}$ will converge to one of the local minima $x^{(1^*)}, \dots, x^{(M^*)}$ with probability 1, provided that the size L of the regressor vector ϕ is larger than a lower bound \bar{L} .

²Here, N is a sufficiently large user-defined positive integer; typically it suffices to choose $N \in \{20, \dots, 30\}$.

III. THE PROPOSED CAO-BASED SOLUTION

The objective of this work is to maximize the monitored area in a given region by using a team of mobile robots, without any assumption on the topology of the environment. We can mathematically define the problem in the following way. Let us consider a planar non-convex environment and let Ω be the region accessible by the robots. Let $\mathcal{P} = \{x_k^{(i)}\}_{i=1}^M$ denote the position of the M robots at the time step k and $C = \{r_i\}_{i=1}^M$ the maximum distances of monitoring for the i -th robot. In our approach, we consider the monitoring of a point $q \in \Omega$ a binary function

$$f(q, \mathcal{P}; C) = \begin{cases} 1 & \text{if } q \text{ is monitored} \\ 0 & \text{otherwise} \end{cases} \quad (9)$$

Let us assume that a robot can monitor the points which satisfy both the following conditions:

- are connected by a line-of-sight with it;
- are at a distance smaller than a given threshold value.

In this paper we consider only the omnidirectional case for simplicity's sake, but a limited monitoring field of view is easy to take into account (see [8]).

Thus, we can define the cost function \mathcal{J} as follows:

$$\mathcal{J}(\mathcal{P}; C) = \frac{1}{V} \int_{\Omega} f(q, \mathcal{P}; C) dq \quad (10)$$

where $V = \int_{\Omega} dq$. Obviously, this is only an implicit expression of the cost function and it is impossible to get an explicit form because of the dependency on the particular environment. However, as explained in the previous section, we just need the numerical value of the cost function for each time step and not its explicit expression. This is the key advantage of CAO which does not require an a priori knowledge of the environment.

A fundamental point for a good behavior of the algorithm is an appropriate choice of the form of the regressor vector ϕ , introduced in equation (5). Several different choices for its explicit expression are admissible. However, for the particular application treated in this paper, it was found that it suffices to choose a third-order polynomial regressor vector (for more details see [8]).

Once the regressor vector ϕ has been set and once the values of the cost function (10) are available for measurement at each time step, it is possible to find at each time step the vector of parameter estimates θ_k and thus the approximation of the cost function \hat{J}_k . The other important choice in order to assure the convergence of the algorithm is the expression of the sequence α_k , defined in equation (7). A typical choice for such a sequence is given by

$$\alpha_k = \frac{c}{(k+1)^\eta}, \quad (11)$$

where c is a positive user-defined constant and $\eta \in (0, 0.5)$.

Remark 2: Please note that the CAO algorithm's computational requirements are dominated by the requirement for solving the least-squares problem (6). As the number of free parameters in this optimization problem is L , most popular algorithms for solving least-squares problems have – in the

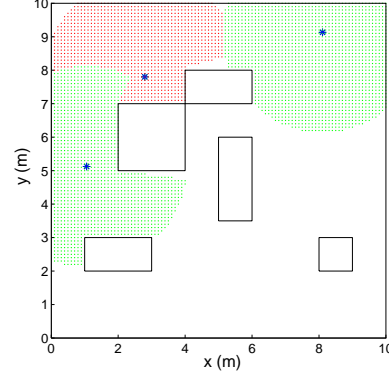


Fig. 1. Example of area monitored by three robots equipped with omnidirectional visual sensor in a non-convex environment (the blue dots correspond to the robots' locations, and the rectangles denote the obstacles). The field of view in red is the effective monitored area (optimization function) of the respective robot.

worst case – $\mathcal{O}(L^3)$ complexity (polynomial complexity with respect to L). For a realistic situation where 3-5 robots are employed for optimal coverage, our simulation investigations indicate that a “good” value for L is around 20. \diamond

A. Distributed algorithm

As already mentioned, the advantages of a distributed solution are several and of crucial importance for a practical application. Even if the optimization function (10), which we want to maximize, depends on all the state vector's variables, it is reasonable to think that the coverage problem could be expressed in a decentralized way. Indeed, the aim of each robot is to deploy itself in order to minimize the overlapping of its field of view with the obstacles of the environment and with the fields of view of the other robots. In other words, for each robot the problem is like maximizing the surface monitored while it is moving in an environment with both static obstacles and dynamic obstacles, which are the fields of view of the other robots (see fig. 1). In practice, we can write the new optimization function for the robot i as follows:

$$J_i(\mathcal{P}) = V_i - \sum_{j \neq i} V_i \cap V_j, \quad (12)$$

where V_j is the total area monitored by the j -th robot.

In other words, instead of a single cost function, the problem is now characterized by N cost functions, being N the number of robots. The CAO approach is adopted to independently maximize each optimization function. In particular, each function will be characterized by a different approximator. Then the CAO approach will perform the optimization by only perturbing the position of the relative robot, while the others are considered static.

It is important to note that at each time step this new single-robot optimization function depends only on the position of the robots which have an overlapping field of view, which may be identified as the neighbors. However, to obtain the approximator, the CAO algorithm uses all the data stored

and so it is necessary to consider also robots that have been neighbors in past time steps.

The proposed distributed method can also account communication constraints. Indeed, each robot has to know only the real positions of the other robots when they are neighbors. For robots which are not neighbors, i.e. for the robots whose monitored area does not overlap with the area of the considered robot, knowing the actual positions is unnecessary. Practically, the considered robot generates for them fictitious regular trajectories. These false positions do not influence the construction of the approximator because it does not depend on them. Therefore, every robot has to communicate only with its neighbors and even if there is a failure in this communication the algorithm continues to work.

IV. SIMULATIONS

To evaluate the efficiency of the proposed algorithm, several simulations with varying number of robots and different monitoring constraints, have been performed in a variety of environments. We show the results of the simulations comparing the performance of the centralized approach (i.e. the one which maximizes (10) by simultaneously perturbing the whole state vector) and the distributed approach introduced in section III-A. The teams are considered to be homogeneous since the maximum distance of monitoring for each robot is the same, although it is not the same in all simulated scenarios. This assumption has been made for simplification purposes and easier comprehension of the results.

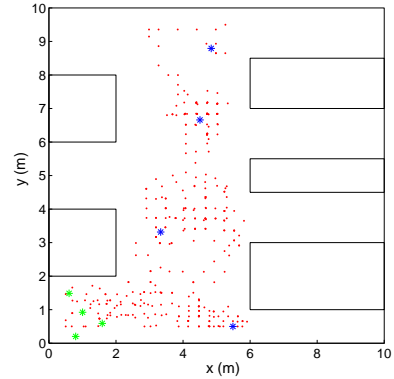
Fig. 2(a) shows a typical scenario in which four robots have to cover a nonconvex environment. The behaviour of the cost function during the task is shown in fig. 2(b). It is possible to see that the final deployment is such that the team is able to monitor the whole environment. For further results on the centralized approach, even considering a limited monitoring field of view for each robot, see [8].

Now we want to show that it is possible to obtain the same results achieved with the centralized algorithm by using the distributed solution. In fig. 3(a) there is the result obtained starting from the same configuration than for the centralized case previously presented. It is worth to note that, as it can be seen in fig. 3(b), the time of convergence can be even less than for the previous approach.

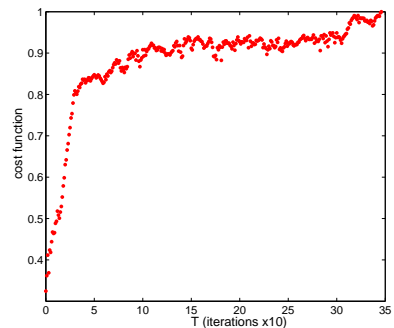
In the last proposed simulation (fig. 4) the team is composed by six robots with a monitoring radius of $r = 3m$ and the environment has a different topology. Also in this case the team is able to find a final configuration which allows monitoring almost all the free space and the distributed algorithm is faster to converge (see fig. 4(b)).

V. CONCLUSION AND FUTURE WORK

A new method for obtaining distributed and scalable multi-robot controls for a cooperative surveillance problem, in a 2D non-convex environment with unknown obstacles has been proposed. The goal is to maximize the area monitored by all the robots through their visual sensors. The optimization



(a)



(b)

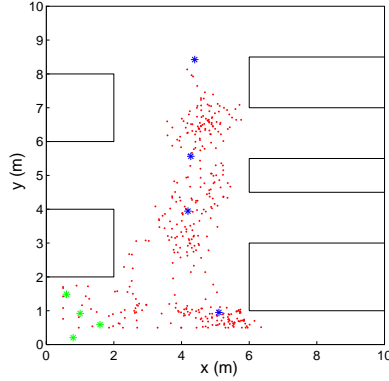
Fig. 2. Centralized approach. Four robots with a maximum monitoring distance $r = 6m$. In fig. (a) it is shown the robots' motion: the green points show the initial positions of the robots, the final ones are in blue, in red the trajectories. In fig. (b) the cost function $\mathcal{J}(\mathcal{P})$.

problem is solved by using a new stochastic method: the cognitive-based adaptive optimization algorithm.

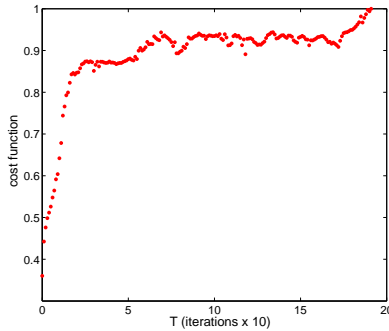
The proposed approach has the following key advantages with respect to previous works:

- it can solve the problem in a distributed way;
- it does not require any a priori knowledge on the environment;
- it works in any environment, without the necessity to make assumptions about its topology;
- it can incorporate any kind of constraints, for instance regarding a possible existing threshold on the maximum distance on the monitored region;
- it does not require a knowledge about these constraints since they are learnt during the task execution;
- its complexity is low allowing real time implementations.

The advantages of the proposed methodology make it suitable for real implementations and the results obtained through numerical simulations give us the motivation to adopt the CAO also in other frameworks. We are also



(a)



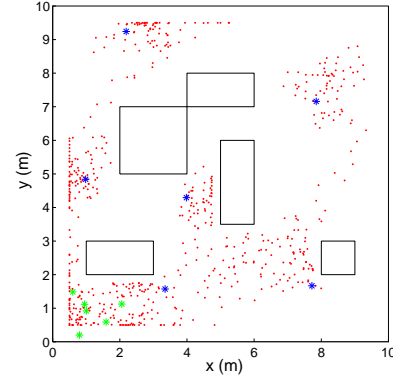
(b)

Fig. 3. Distributed approach. Four robots with a maximum monitoring distance $r = 6m$.

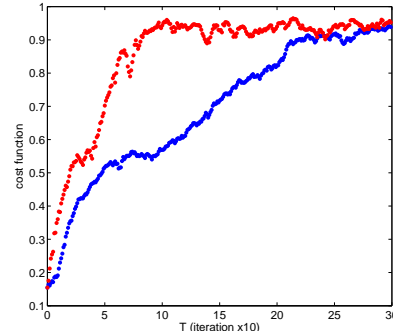
interested in extending the problem to the 3D case to develop a strategy for the surveillance of urban-like environments with a MAV swarm.

REFERENCES

- [1] J. Cortés, S. Martínez, T. Karataş and F. Bullo, "Coverage Control for Mobile Sensing Networks," *IEEE Transactions on Robotics and Automation*, vol. 20, no. 2, pp. 243-255, 2004.
- [2] M. Schwager, J. McLurkin and D. Rus, "Distributed Coverage Control with Sensory Feedback for Networked Robots," in *Proceedings of Robotics: Science and Systems*, Philadelphia, USA, 2006.
- [3] L. Pimenta, V. Kumar, R.C Mesquita and G. Pereira, "Sensing and coverage for a network of heterogeneous robots," in *47th IEEE Conference on Decision and Control*, Cancun, Mexico, 2008.
- [4] A. Breitenmoser, M. Schwager, J. Metzger, R. Siegwart and D. Rus, "Voronoi Coverage of Non-Convex Environments with a Group of Networked Robots," in *IEEE International Conference on Robotics and Automation (ICRA)*, Anchorage, USA, 2010.
- [5] A. Ganguli, J. Cortés and F. Bullo, "Maximizing visibility in nonconvex polygons: nonsmooth analysis and gradient algorithm design," in *American Control Conference*, vol. 2, pp. 792-797, 2005.
- [6] A. Ganguli, J. Cortés and F. Bullo, "Visibility-based multi-agent deployment in orthogonal environments," in *American Control Conference*, New York, USA, pp. 3426-3431, 2007.
- [7] A. Renzaglia, L. Doitsidis, A. Martinelli and E.B. Kosmatopoulos, "Cognitive-based Adaptive Control for Cooperative Multi-Robot Coverage," in *IEEE International Conference on Robotics and Intelligent System (IROS)*, Taipei, Taiwan, 2010.
- [8] A. Renzaglia, L. Doitsidis, A. Martinelli and E.B. Kosmatopoulos, "Adaptive-based, Scalable Design for Autonomous Multi-Robot Surveillance," in *49th IEEE Conference on Decision and Control (CDC)*, Atlanta, Georgia, USA, 2010.
- [9] E.B. Kosmatopoulos, Papageorgiou, M., Vakouli, A. Kouvelas, A., "Adaptive fine-tuning of nonlinear control systems with application to the urban traffic control strategy TUC," *IEEE Transactions on Control Systems Technology*, Vol. 15, no. 6, pp. 991-1002, 2007.
- [10] E.B. Kosmatopoulos, "An adaptive optimization scheme with satisfactory transient performance," *Automatica*, Vol. 45, No. 3, pp. 716-723, 2009.
- [11] E.B. Kosmatopoulos and A. Kouvelas, "Large-Scale Nonlinear Control System Fine-Tuning through Learning," *IEEE Transactions Neural Networks*, Vol. 20, No. 6, pp. 1009-1023, 2009.
- [12] J.C. Spall, "Multivariate stochastic approximation using a simultaneous perturbation gradient approximation," *IEEE Trans. Autom. Control*, vol. 37, no. 3, pp. 332-341, 1992.
- [13] D. P. Bertsekas and J. N. Tsitsiklis, "Gradient convergence in gradient methods with errors," *SIAM Journal in Optimization*, vol. 10, no. 3, pp. 627-642, 2000.
- [14] A. Howard, M.J. Matarić and G.S. Sukhatme, "An incremental deployment algorithm for mobile robot teams," in *IEEE International Conference on Robotics and Intelligent System (IROS)*, Lausanne, Switzerland, pp. 2849-2854, 2002.
- [15] P.K. Agarwal and M. Sharir, "Efficient algorithms for geometric optimization," *ACM Computing Surveys*, vol. 30, no. 4, 1998.



(a)



(b)

Fig. 4. Six robots with a maximum monitoring distance $r = 3m$. In fig. (b) the cost functions obtained by using the two version of the algorithm: in red the distributed method, in blue the centralized one.