# Co-adapting Mobile Sensor Networks to Maximize Coverage in Dynamic Environments

François-Michel De Rainville, Christian Gagné, and Denis Laurendeau
Laboratoire de vision et systèmes numériques
Département de génie électrique et de génie informatique
Université Laval, Québec (Québec), Canada G1V 0A6
francois-michel.de-rainville.1@ulaval.ca,
{christian.gagne,denis.laurendeau}@gel.ulaval.ca

## ABSTRACT

With recent advances in mobile computing, swarm robotics has demonstrated its utility in countless situations like recognition, surveillance, and search and rescue. This paper presents a novel approach to optimize the position of a swarm of robots to accomplish sensing tasks based on cooperative co-evolution. Results show that the introduced cooperative method simultaneously finds the right number of sensors while also optimizing their positions in static and dynamic environments.

## Categories and Subject Descriptors

I.2.9 [**Artificial Intelligence**]: Robotics—*sensors, autonomous vehicules*; G.1.6 [**Numerical Analysis**]: Optimization—*global optimization*

## General Terms

Algorithms, Performance

## Keywords

Cooperative co-evolution, swarm robotics, mobile sensor network, static and dynamic environment, visibility

## 1. INTRODUCTION

Swarm robotics is a research area of growing interest where a group of relatively simple robots cooperate to achieve a given goal that a single robot could not reach alone. Swarm robotics is inspired from the fascinating world of social insects, which demonstrates three valuable characteristics for multi-robot systems: robustness, flexibility, and scalability [5]. When applied to inspection and surveillance, swarm robotics has a great potential to accomplish tasks in a myriad of scenarios. For example, a swarm of robots could be deployed to investigate hazardous or inaccessible environments, such as defective nuclear power plants (e.g. Fukushima) or spatial exploration (e.g. surveying planet Mars).

This work focuses on automatic placement of a swarm of mobile sensors in hostile and/or remote environments. More specifically, we are motivated by telepresence applications where an external user whishes to observe, from a specific point of view, a scene that is inaccessible to humans for different reasons. A swarm of robots should then be able to automatically configure itself so that the desired *virtual* point of view can be reconstructed from the robots' different observation position of the environment. Sensor placement optimization is thus essential if good coverage of the required surfaces with a minimum number of sensors is desired.

To achieve this goal, we propose to use a cooperative co-evolutionary algorithm to configure the swarm of robots to observe the desired surfaces of the environment. The algorithm should then be able to produce a configuration that not only optimizes the position of the swarm but also estimates how many robots are required to achieve our goal for static and dynamic environments.


*GECCO'12 Companion,* July 7–11, 2012, Philadelphia, PA, USA.
ACM 978-1-4503-1178-6/12/07.

## 2. CO-ADAPTING MOBILE SENSORS

Co-adaptation of multiple species in an evolutionary algorithm is a way to break down a hard problem into several sub-problems that are resolved cooperatively by multiple sub-solutions [4]. Additionally, the sub-problems and sub-solutions emerge naturally from the evolutionary process. As the number of sensors needed to cover a general environment is rarely known exactly a priori, cooperative co-evolution offers a simple mechanism to adapt the number of species to the problem at hand.

In the co-evolutionary algorithm, the sensors are encoded by three real numbers: the two position values $(x, y)$ and the orientation (angle $\theta$). Evolved sensors are homogeneous, as they share intrinsic parameters such as field of view, maximum range, etc. The individuals are thus represented as a vector of three floating-point numbers. As pointed out by Potter and De Jong [4], the inner loop of the co-evolutionary algorithm can be of any type. We chose a simple real-valued vector genetic algorithm with $\alpha$-blend crossover, gaussian mutation, and tournament selection.

The fitness of an individual is given as the coverage of a group of collaborating sensors in the environment computed through Monte-Carlo simulations. The coverage of a single sensor is estimated by projecting rays in an occupancy grid representation of the world and computing the footprint area from the distance travelled by the rays and the orientation of the surface hit. The computed collaboration fitness is assigned only to the individual being evaluated, the fitness of the other species representatives remain unchanged.

Each species contribution to the cooperative solution is evaluated by the difference in coverage with and without the representative of that species, allowing the identification of species that are unproductive and should be removed. The

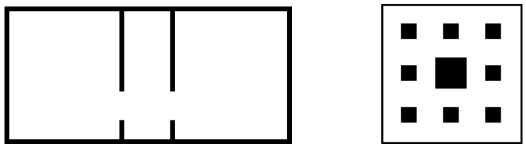

**Figure 1: Static (left) and dynamic (right) worlds.**

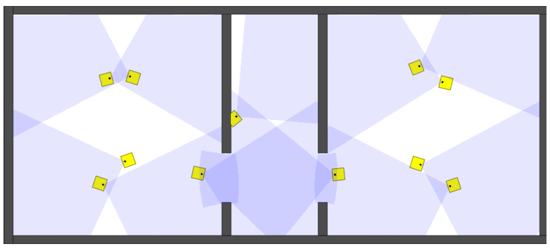

**Figure 2: Co-adapted configuration of 11 sensors.**

improvement of the algorithm is monitored by the quality of the set of representatives at each ecosystem generation. If the set of representatives has not improved after a number of generations, then the mechanism to delete unfit species and add a new one is executed. This mechanism allows for an efficient adaptation of the number of sensors required to accomplish the task. The co-evolutionary algorithm presented so far will allow: (1) a good enough configuration of sensors to be found so that the entire environment is sensed; (2) the number of sensors naturally from the geometry of the environment to be adapted; and (3) the found configuration when movements occur in the environment to be adjusted.

## 3. EXPERIMENTS

The experiments consist of optimizing, with the proposed algorithm, the configuration of a homogeneous sensor network starting with a single species (sensor) in a static and a dynamic environment, shown in Figure 1. We wish to observe the walls of the environment entirely. For the dynamic scenario, only two small blocks are present at the same time, beginning with the north-west and south-east positions and moving counter-clockwise.

Figure 2 presents the final configuration of the representatives found by one of the ten experiments conducted with the co-evolution. We observe that the optimization process is able to find a configuration covering most of the complex environment. In fact, this process guarantees that the network is composed only of sensors that are contributing to the swarm coverage. More complete coverage can be achieved by reducing the contribution threshold, with a risk of producing more unfit species [4]. In Figure 3, the convergence to an optimal solution is compared with the Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [2], and a canonical Particle Swarm Optimization (PSO) [3], each with 11 sensors. It is clear that cooperative co-evolution is able to adapt the number of sensors and configure them in a suitable position in a reasonable amount of time.

Figure 4 presents the average fitness of the species representatives fitness for ten successive changes in a dynamic environment over five runs. From the algorithms point of view, only the fitness is different between two changes, i.e. the algorithms preserve all of their parameters and individuals. Sensors are not rewarded for the surface observed on the moving blocks. For PSO, we included a quantum restart technique on the main swarm [1]. CMA-ES and PSO evolve seven sensors, while co-evolution evolves up to seven species.

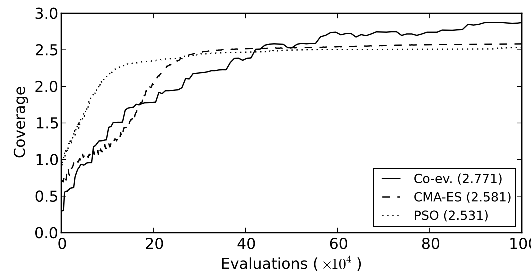


**Figure 3: Evolution of coverage in static mode.**

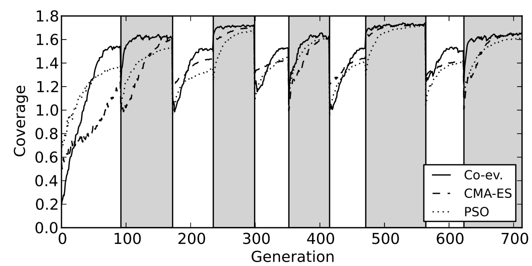


**Figure 4: Evolution of coverage in dynamic mode.**

As we observe, the co-evolution is able to reconfigure the sensors conveniently and rapidly after a change, while CMA-ES and PSO seem to have more difficulties to do so. We also note that the most important mechanism of the co-evolutionary technique in dynamic environments is the stagnation detection that acts as a restart for sensors that are the most affected by a change. This mechanism, which deletes unfit sensors and adds a new one, helps the algorithm to reach a configuration that suits the current environment geometry quickly.

## Acknowledgements

This work is funded by the FQRNT (Québec) and the NSERC (Canada).

## 4. REFERENCES

[1] T. M. Blackwell and J. Branke. Multi-swarm optimization in dynamic environment. In *Application of Evolutionary Computation*, LNCS 3005, pages 489–500. 2004.

[2] N. Hansen and A. Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 9(2):159–195, 2001.

[3] J. Kennedy and R. C. Eberhart. *Swarm Intelligence*. Morgan Kaufmann, 2001.

[4] M. A. Potter and K. A. De Jong. Cooperative coevolution : An architecture for evolving co-adapted subcomponents. *Evolutionary Computation*, 8(1):1–29, 2001.

[5] E. Şahin. Swarm robotics: From sources of inspiration to domains of application. In *Swarm Robotics*, LNCS 3342, pages 10–20. 2005.