Multi-Robot Exploration based on Swarm Optimization Algorithms

Micael S. Couceiro[*], N. M. Fonseca Ferreira[**] and Rui Rocha[**]
[*] *Department of Electrotechnics Engineering, Engineering Institute of Coimbra, Coimbra, Portugal*
[**] *GECAD - Knowledge Engineering and Decision Support Research Center, Institute of Engineering, Polytechnic of Porto (ISEP/IPP), Porto, Portugal*
[***] *Institute of Systems and Robotics, University of Coimbra, Pólo II, Coimbra, Portugal*

Summary. The Darwinian Particle Swarm Optimization (*DPSO*) is an evolutionary algorithm that extends the well-known Particle Swarm Optimization (*PSO*) using natural selection, or survival-of-the-fittest, to enhance the ability to escape from local optima. In this paper, it is explored the effectiveness of using a modified version of both *PSO* and *DPSO*, respectively named as *R-PSO* and *R-DPSO*, on groups of simulated robots performing a distributed exploration task.

Introduction and Motivation

The Particle Swarm Optimization (*PSO*) is an optimization technique which models a set of potential problem solutions as a swarm of particles moving around in a virtual search space. This method developed by [1] was inspired by the movement of flocking birds and their interactions with their neighbours in the group. However, a general problem with the *PSO* and other optimization algorithms is that of becoming trapped in a local optimum such that it may work well on one problem but may fail on another problem.

In search of a better model of natural selection using the *PSO* algorithm, the Darwinian Particle Swarm Optimization (*DPSO*) was formulated by [2], in which many swarms of test solutions may exist at any time. Each swarm individually performs just like an ordinary *PSO* algorithm with some rules governing the collection of swarms that are designed to simulate natural selection.

Just like in Multi-Robot Systems (*MRS*), where groups of robots interact to accomplish their goals [3], both *PSO* and *DPSO* use groups of interacting virtual agents (*aka* particles) in order to achieve their optimization. However, contrarily to virtual agents, robots are designed to act in the real world where, in certain environments or applications such as hostile environments, search & rescue, disaster recovery, battlefields, space and others, the communication infrastructure may be damaged or missing.

Bearing these ideas in mind, this paper explores the possibility to implement adapted versions of both *PSO* and *DPSO*, respectively named as *R-PSO* (Robotic *PSO*) and *R-DPSO* (Robotic *DPSO*), taking into account communication constrains, in a parallel distributed fashion for exploration tasks in *MRS*. Each robot is then responsible for each virtual agent, which it need to evaluate at each iteration. After each set of evaluations, the robots communicate to share the fitness information needed to progress to the next iteration of the algorithm.

This dedicated swarming behaviour of the *MRS* is capable of maintaining a suitable distribution of the robots that could be useful in any application that can be described as a fitness function such as: *i*) search & rescue: a catastrophic scenario wherein the number of victims is randomly positioned and the number of robots must vary accordingly to the number of victims; *ii*) exploration: an unknown and unstructured scenario that needs to be mapped efficiently wherein the number of robots vary accordingly to the complexity of the surroundings; *iii*) *ITS*: an autonomous transport infrastructure wherein the number of robots vary accordingly to the number of passengers; *iv*) nanomedicine: a team of nanorobots is inside the human body in order to repair damaged tissue wherein the number of robots vary accordingly to the most damaged regions.

Robotic Particle Swarm Optimization (*R-PSO*)

The most common behavior-based collective architectures, such as *PSO* [1], where the main purpose is to find the fitness function's global optimum normally ignores the communication issues since it does not offer any advantage. However, one of the major limitations in robotics is the range and signal quality of communications (without preexistent infrastructure).

The main difference between the *R-PSO* and the *PSO* resides in the implementation of an enforcing network connectivity algorithm. Instead of having something similar to broadcasting where all agents can exchange information regardless where they are located in the search space, the algorithm will have the main objective of controlling the robots' position in order to maintain the communication based on constraints such as maximum distance or minimum signal quality.

The *R-PSO*, just like the *PSO*, basically consists on a population of robots that collectively move on the search space (*e.g.*, catastrophic scenario, city) in search of the global optimum (*e.g.*, number of victims, number of passengers) where each robot is characterized by its pose and performance. For instance, if we have a group of mobile olfactory robots that are trying to find a gas leak on an indoor environment, each robot will be characterized by the values of the pose (*i.e.*, position and orientation) and by the corresponding value of the gas density.

Robotic Darwinian Particle Swarm Optimization (*R-DPSO*)

The *DPSO* [2] may be represented by multiple swarms of test solutions where each swarm individually performs just like an ordinary *PSO* algorithm with some rules governing the collection of swarms that are designed to simulate natural selection. The selection process implemented is a selection of swarms within a constantly changing collection of swarms.

In the common *DPSO*, "punish" means the deleting of particles and swarms, while "reward" means the spawning of new particles and swarms. However, in the *R-DPSO* the deleting and spawning of a robot represents the social exclusion and inclusion of a robot in the swarm, respectively. In other words, the *R-DPSO* will also be represented by multiple swarms (group of robots) where each swarm individually performs just like the *R-PSO* in search for the solution and some rules governs the whole population of robots. However, any robot may be used as a relay node independently of their swarm. The number of times a swarm is evolved without finding an improved fitness is tracked with a search counter, *SC*. If the swarm's search counter exceeds a maximum critical threshold, SC^{max}, the swarm is punished by excluding the worst performing robot adding it to a socially excluded group. If the number of robots falls below the minimum number of accepted robots to form a swarm, the swarm is punished by being dismantled and all the robots that were from that swarm are added to the socially excluded group. On the other hand, if the swarm improves its fitness then it is rewarded with the best performing robot in the socially excluded group. If a swarm has been more often rewarded than punished it has a small probability $p = f / N_S$ of spawning a new swarm, where f is a uniform random number on [0,1] and N_S is the number of active swarms. The group of robots of this new swarm will be the best performing robots in the socially excluded group.

The key issue in this novel approach is the answer to the question: What does the robots of the socially excluded group do? In fact the answer is the same that we would give if asking about a group excluded from our society: they don't do anything. Instead of wandering in the scenario searching for the fitness function's global optimum like the other robots in the active swarms do, they basically randomly wander in the scenario. Note, however, that they are always aware of their individual solution and the global solution of the socially excluded group. As previously explained, when a swarm is rewarded with a new member or a new swarm is spawned, the best performing robots in the socially excluded group are chosen. On the other hand, this previously socially excluded robots becomes now aware of their main objective (*i.e.*, finding the fitness function's global optimum) and they will then share their own solution with the rest of the swarm. This approach improves the algorithm making it less susceptible of becoming trapped in a local optimum.

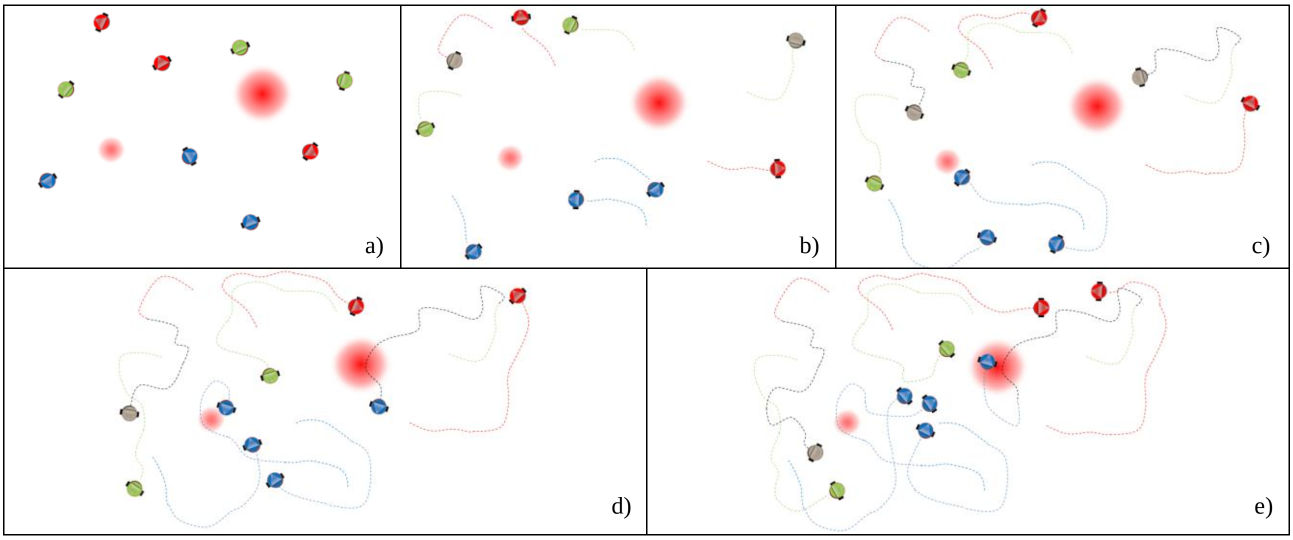

Fig 1. Illustrative sequence of a *MRS* exploration using the *R-DPSO* algorithm. *a*) The population is divided in 3 swarms of 3 robots each; *b*) Swarm 1 and 2 (red and green robots, respectively) can't improve their fitness for SC^{max} iterations and they are punished by excluding the worst performing robot of each swarm and adding them to the socially excluded group; c) The socially excluded robots randomly wanders in the scenario memorizing their individual best solution and the global best solution of the socially excluded group; d) Swarm 3 improves its solution, since it finds a local optimum, and it is rewarded with the best performing robot in the socially excluded group; e) The new member of swarm 3 communicates its best individual solution to the other members which is better than their best global solution inducing them to move toward this new solution.

References

[1] Kennedy, J. and Eberhart, R. (1995) A new optimizer using particle swarm theory. In Proceedings of the IEEE Sixth International Symposium on Micro Machine and Human Science, pp. 39-43.

[2] Tillett, J., Rao, T. M., Sahin, F., Rao, R. and Brockport, S. (2005) Darwinian Particle Swarm Optimization. In Proceedings of the 2nd Indian International Conference on Artificial Intelligence, pp. 1474-1487.

[3] Rocha, R. (2006) Building Volumetric Maps with Cooperative Mobile Robots and Useful Information Sharing: a Distributed Control Approach based on Entropy. PhD Thesis, Faculty of Engineering of University of Porto.