Environment-driven Distributed Evolutionary Adaptation for Collective Robotic Systems

Jean-Marc Montanier

UNIVERSITÉ PARIS-SUD XI

T H E S I S

by Jean-Marc Montanier

Environment-driven Distributed Evolutionary Adaptation for Collective Robotic Systems

Prepared at the Laboratoire de Recherche en Informatique (UMR 8623), Université Paris-Sud

Defense on March 1st, 2013

Jury :

Reviewers:	A.E. Eiben	-	VU Amsterdam, Netherlands
	Guillaume Beslon	-	LIRIS/CNRS/INRIA, INSA de Lyon
Examiners:	Stéphane Doncieux	-	ISIR/CNRS, UPMC
	Marc Schoenauer	-	INRIA Saclay IdF
	Philippe Tarroux	-	LIMSI/CNRS, Université Paris-Sud
Advisor:	Nicolas Bredèche	-	ISIR/CNRS, UPMC[1]

[1]Nicolas Bredèche was associate professor at Université Paris-Sud (A&O/TAO team) until sept. 2012

Acknowledgements / Remerciements

Certaines personnes décident de rédiger une section remerciements minimalle et formelle. Cependant, je pense qu'une simple liste ne rendrait pas honneur aux personnes que je souhaite remercier. Par conséquent, le lecteur intéressé devra lire plusieurs paragraphes avant de trouver celui qui le concerne. De plus, l'anglais ou le français sont utilisés en fonction des personnes remerciées.

Some people decide to write minimal and formal acknowledgements. However I feel that a simple list would not do justice to the people i want to thank. Consequently, the interested reader will have to go through multiple paragraphs to find the one that concerns him. Moreover, these acknowledgements are written in different languages depending of the people thanked.

First of all, I'm grateful to the members of the jury who took some of their precious time for my thesis. I add here a particular acknowledgement to the reviewers for having read the thesis from beginning to end and provide meaningful feedbacks. It was also a great pleasure to have such relevant scientific questions asked during my defense (even if sometimes tough).

Merci à Nicolas Bredèche pour avoir encadré cette thèse. Entre mon arrivée pour un stage d'ingénieur, et ma sortie avec un diplôme de docteur, la route a été longue. Il me reste encore beaucoup de chemin à parcourir, c'est une bonne nouvelle.

Je voudrais aussi remercier les équipes de recherches m'ayant accueilli durant mon doctorat : TAO à l'université Paris-Sud et AMAC à Jussieu. Appartenir à ces équipes m'a beaucoup appris sur le plan scientifique. J'ai aussi eu la chance d'y rencontrer des stagiaires, doctorants, et post-docs aux qualités humaines uniques, notamment dans les bureaux FZ (on emportera tous un morceau de Pelé dans notre coeur) et J01 (quand je reviens j'achète une nouvelle cible).

Je tiens à remercier ma famille pour m'avoir soutenu (poussé et tiré aussi) pendant toutes ces années. Les débuts n'ont pas été évidents, mais vous voyez, j'ai fini par étudier. Merci à vous pour m'en avoir donné les moyens et parfois la motivation.

Je voudrais aussi remercier mes amis rencontrés au fil des années. Pendant ma thèse, ces moments où je pouvais penser à autre chose, ou bien recevoir un point de vue extérieur sur la situation ont été très importants. La liste des noms et des souvenirs est malheureusement trop longue pour être écrite ici. Néanmoins, croyez-bien que ces souvenirs existent dans ma mémoire. Among them I would like to add a special word to Sandra Heinzelmann for having been there and hold it through really hard times. Sorry, mein deutsch ist so schlecht wie immer.

Last but certainly not least, thank you Sandra Astete: the right balance between working hard and not getting insane was hard to find, but with you on my side it looked a lot easier. Unfortunately, I have keep my acknowledgments to you short, as I'm afraid I didn't improve my skills in poetry writing during my Ph.D.

Contents

Introduction

This thesis describes some of the work done in the context of the Symbrion project[1]. This project targets the realization of complex tasks which require the cooperation of multiple robots within robotic swarms (at least 100 robots operating together). Among issues studied by the project are the self-assembly of robots to form complex structures and the self-organization of large number of robots toward the realization of a common task. Subjects of interests are thus modular self-adaptive robots with both strong coordination properties, and swarm-level cooperation.

The challenge faced by this project is that robots are used in open environments which remain unknown until their deployment. Since operational conditions can't be predicted beforehand, on-line learning algorithms must be used to design behaviors. In the use of large groups of robots, multiple considerations have to be taken into account: reduced communication abilities, small memory storage, small computational power. Therefore on-line learning algorithms have to be distributed among robots.

Multiple approaches have already been proposed to deal with on-line decentralized learning of robotic behaviors, such as probabilistic robotics, reinforcement learning or evolutionary robotics (all of which will be later described). However, the problem addressed here is even more complex as groups of robots are considered, rather than of a single robot. Moreover, due to the open-endedness and unpredictability of the environment, we can safely assume that the human engineer may lack the background knowledge necessary to sketch directions such that learning is possible.

As a matter of fact, ensuring the integrity of the swarm (e.g. simply surviving through energy recharge) is not only mandatory prior to address any further user defined task, but also an already very challenging problem. Hence, the problem of ensuring integrity should be placed as the first element of the following roadmap, which we assume as a set of necessary steps toward achieving tasks with a group of robots in open environments.

- Step 1: Ensuring the integrity of robots.
- Step 2: Maintaining robots available as a service to the user.
- Step 3: Achieve a user-defined task (optimized or not).

[1] IP, FP7, 2008-2013

In this thesis we address the step 1 of this roadmap, and states the following:

> **Statement:** Collective robotics in open environment requires to perform self-adaptation prior to address user-specified task.

The subject of interest of this thesis, is to design a decentralized algorithmic solution that can be used to guaranty swarm integrity in open environment, using collective robot system with local communication. This is a difficult problem which hasn't been addressed as such in the literature, even if trails exist (see Chapter 3). The main difficulty in the resolution of this problem is the need to take into account the environment. Indeed, robots may have to display a large variety of behaviors at the global scale such as cooperation, specialization, altruism, and division of labor, depending on the environment at hand.

Solving such an environment driven adaptation task can be seen as requiring to satisfy two possibly antagonist motivations. The first motivation is to guaranty the integrity of a maximum of robots within the group in the environment at hand (extrinsic motivation). The second motivation is to allow the necessary interactions between robots and environment to ensure the well functioning of the algorithm (intrinsic motivation). There is therefore a trade-off between conservative approaches (remain stationary, and thus satisfy the intrinsic motivation, event at the risk of preventing the realization of the task), and exploratory approaches (test every possible interactions, and thus satisfy the extrinsic motivation, even at the risk of breaking a robot or getting lost).

In this thesis we introduce and define the problem of Environment-driven Distributed Evolutionary Adaptation. We propose an algorithm to solve this problem, which has been validated both in simulation and on real robots. This algorithm has been used to study self-adaptation problems in specific environments:

- Environments where behavioral consensus is required (see Chapter 4)
- Environments where robustness in front of environmental changes is required (see Chapter 5)
- Environments where altruistic behaviors is required (see Chapter 6)

Organization

The first chapter of this thesis presents a partial review of the robotic field. This review briefly presents the different aspects of the robotic field. A particular focus is given to the design of controllers for robots deployed in challenging environments.

The second chapter aims at presenting the Evolutionary Robotics field. The focus is given to algorithms used in robots during their deployment. We also present a contribution made to this field as an illustrative example.

The third chapter presents the issues faced by the decentralized autonomous optimization of behaviors for robots deployed in unknown environments. Different methods known to partially address these are reviewed. After highlighting the strength and weakness of each method, the EDEA domain is introduced.

In the fourth chapter the minimal EDEA algorithm, termed mEDEA is presented. The ability of mEDEA to reach consensus is presented in simulation and real world experiments.

The fifth chapter is focused on the capacity of mEDEA to address changing environments. Notably, the main evolutionary dynamics of the algorithm are shown.

The sixth chapter shows the evolution of altruistic behaviors by the mEDEA algorithm in front of challenging environments. Moreover, the mechanism at play during the evolution of altruistic behaviors are analyzed.

The seventh chapter conclude this Ph.D. thesis. A discussion on the work done is presented, and perspectives for future works are highlighted.

Chapter 1

Autonomous Robotics

From the Oxford dictionary “Robotics is the branch of technology that deals with the design, construction, operation, and application of robots”. Starting from this broad definition, we will clarify as best as possible the notion of “Robot” (Section 1.1). We will then review main applications for robots (Section 1.2). The challenges met in their conception are presented in Section 1.3. Finally, an overview of the methods known to control robots is given in Section 1.4, with a special emphasis on learning and optimization method.

1.1 Robots

1.1.1 Definition

Even if each has an idea of what a robot is, there is no universal consensus on the definition of the word. In order to illustrate this point, we will take two extreme examples. On the one hand, the programs operating autonomously on Internet in the search of new websites are sometimes called robots. These programs have no incarnation in the real world. They only travel in a virtual world of link between web pages and report their results to a higher level program. However, they perform this task autonomously with few inputs from a human user, and therefore, often take decisions by themselves on which link to follow. On the other hand, demining robots have arguably an impact on the real world. Contrary to their web crawlers counter-part, they are teleoperated by humans, and have consequently no autonomy on the course of their actions. From these two examples, it is clear that systems with diverse properties (an autonomous system acting in a virtual world in one case, and a teleoperated system acting in the real world in the second case) are sharing the same name. This confusion is well summarized in a quote attributed to Joseph Engelberger (Developer of the first industrial robot in 1950 - The Unimate): “I can’t define a robot, but I know one when I see one”.

However some common properties are shared between all the robots. First, all robots are goal-oriented systems. Here we need to point out that the aim followed by a robotic system is in the eye of the observer. The system in itself doesn’t know which aim it is achieving, but rather follows blindly a predefined program. However, the external observer may notice that the behavior resulting from this program tends toward one distinctive goal. The goal of the robot is observed thanks to its impact on its environment. From this perspective, our two extreme examples are sharing the same property, i.e. they modify their own environment. Finally, both systems can be controlled: either directly in the case of demining robots, or by modifying the source code of the program in the case of the web crawler.

Finally the link between robots and humans is a key elements to define them. Robots are often seen has a help to humans by performing the tasks humans don't want to do or can't do. However, the range of tasks a robot should do is always under ethical questions. The sight of robots carrying on always more complex tasks raises also questions on the existence of a boundary between humans and robots that shouldn't be crossed. These aspects will be presented with more details in Section 1.1.3.

1.1.2 History

The history of robots can be divided in two parts, before and after the rise of electronics in the 20^{th} century. The first part is the area of automata which started in the 8^{th} or 7^{th} century B.C. and has reached its apogee in the 19^{th} century. At the beginning, automata were mystical ideas, e.g. the God Hephaestus was reported to have created mechanical servants out of gold. In a matter of centuries, these ideas went to reality in Europe and Asia. Archytas of Tarentum is credited with the creation of a mechanical Pigeon in 400 BC. There are also accounts of flying automata in the Han Fei Zi and other texts, which attributes to Mozi (5th century BC Mohist philosopher) and his contemporary Lu Ban, the invention of artificial wooden birds (ma yuan) that could successfully fly [Needham, 1956]. These automata escalated in complexity from the 10^{th} century with the creation of clocks (at first powered by water and later by mechanical springs) equipped with figures performing complex actions. These works were later reproduced by Leonardo Da Vinci, and Jacques De Vaucanson (creation of a duck eating and digesting food). The apogee of this area was reached with the development of programmable automata, which were used in Japanese's theaters [Ichbiah, 2005].

The development of electronic devices and later of computers constitutes a major milestone in the history of robots. From this area of the end of the 19^{th} century and beginning of the 20^{th} century two major realizations are remembered: the radio-controlled boat of Nicolas Tesla [Tesla, 1898], and the Electronic Dog of Benjamin Miessner [Miessner, 1919; Hammond, 1921]. These where the first autonomous (or at least semi-autonomous) electronic systems, partly controllable by a human. At the same time, the robots appeared in the general culture with the coining of the term Robota (translated to robot in English) in the theater piece R.U.R (Rossum's Universal Robots) written by the Czesh author Karel Capek in 1921. Robota is originally a Czesh and Slovak word carrying the idea of hard work, and slavery. We can therefore see that the idea of performing chores for humans is found in the roots of the word robot. Isaac Asimov made the first use of the word robotic in the "liar!" novel in 1941. The list of "First of" continue on with the creation of the first electronic autonomous robot (later named Turtoise) by William Grey Walter in Bristol, England, in 1948 [Walter, 1950]. While many other robots and scientists should be mentioned, an extensive review of all the robots ever created will be far too long [Ichbiah, 2005]. The recent achievements in robotics are presented in Section 1.2

(organized by application field).

1.1.3 Society's Perception

The advent of robots in modern occidental societies raises multiple questions and concerns. Among them is the fear of the "evil-robot". This image has been associated to robots from the first apparition of the word in the theater piece R.U.R., and is more generally linked to the Frankenstein syndrome (fear of new technologies getting out of control) [Kaplan, 2004]. This image of robots getting out of control of their human master has also been portrayed in multiple movies: from the first apparition of a robot in a movie (Metropolis, 1927), to more recent movies (The Terminator, 1984 ; Matrix, 1999 ; I, robot, 2004). This is a common view of robotics in western countries despite authors such as Asimov presenting robots as loyal companions (Robbie, 1940 ; A.I. Artificial Intelligence, 2001). Noticeable exceptions are Japan and South-Corea, where robots aren't perceived as such a threat to the human-kind. The impact of history, and culture's core foundations on the taming of new technologies in the Japanese culture has been highlighted as an explanation to this difference [Kaplan, 2004].

Moreover, ethical questions are raised on the ever reducing boundary between humans and robots. On one hand, this phenomenon can be observed by observing robots looking more and more like humans [Ishiguro, 2008]. On the other hands, humans are getting equipped with artificial equipments to overcome biological deficiencies (pacemakers, bionic hands[1]). The term "singularity" has been coined by Vernor Vinge to mark the moment when humans will be able to create machines smarter than themselves. Proponents of the singularity are found in the scientific community. For example Ray Kurzweil argue about its benefits for human-kind [Kurzweil, 2005]. On the other side, Joseph Weizenbaum (creator of ELIZA [Weizenbaum, 1966]) has always been critical of the use of computer programs on problem not understood by humans [Weizenbaum, 1976]. The two sides of this debate in the scientific community are shown in the movie "Plug and Pray" [Schanze, 2010], targeted at a general audience.

Finally, the notion of "uncanny of valley" is used to describe the unease provoked by robots imitating humans closely but not perfectly [Mori, 1970]. The first reason used to explain this feeling is linked to the human awareness of its eventual death. The presence of a robot looking like a human, but not subject to death might trigger in the human subjects the terror managements process normally used when reminded of an eventual death [MacDorman, 2005]. A second reason advocated is the difference between the expected behavior of the robot (due its resemblance with a human), and the behavior produced [Saygin *et al.*, 2011]. Logically, in order to avoid the uncanny valley it is advised to match the aspect of a robot and its behavior [Goetz *et al.*, 2003].

[1] https://www.youtube.com/watch?v=y0_a_sbEglw - Retrieved 17 December 2012

1.2 Applications

Nowadays, robots are used in a wide range of applications from hobbyist robotic kits to military robots. We will here review these domains while focusing on the challenges specific to each of them. The presentation will gradually goes from robots away from human life, to robots in daily physical contacts with humans. The last section will be specially dedicated to the areas currently explored as research subjects.

1.2.1 Space Robots

The robots sent in space are of primary use to explore this environment where the human presence is rare and difficult. They are used both to reach physically remote places (e.g. the boundary of the interstellar space for Voyager 1), and to perform measures on even further spatial systems (e.g. observation of galaxies thanks to the hubble telescope). These robots are divided in different categories: space probes, landers, exploratory robots and finally satellites.

Space probes are spacecrafts leaving Earth in order to explore space in order to bring more information on the composition of planets and space. The first and most famous of its kind, Voyager 1, has been launched the 5th of September 1977. It is also the farthest human made object from earth, and is in pass of leaving our solar system to reach the interstellar space. The space probes are mostly autonomous, and regularly send back to earth informations obtained through their sensors. They can also be found orbiting around planets other than earth, such as Mariner 9, arrived on Mars the 13th of November 1977.

Space probes sometimes carry landers in charge of performing analysis on the celestial object's surface. The first robot of this type, Lunokhold 1, landed on the Moon in 1970 and traveled over 10 kilometers in 11 months teleoperated from the Baikonur Cosmodrome. During this time, 25 soil analyses were performed out and 20000 photographs were beamed back to earth. Other famous examples are the Viking 1 and 2 which were the first man-made object on the soil of Mars. These landers were carrying experiments designed to look for the presence of life on Mars. Landers can carry exploratory robots, which perform similar experiments but are also able to move on the planet and therefore perform experiments in more diverse conditions. The latest of this kind is the robot Curiosity, landed on Mars on the 6^{th} of August 2012.

Finally it can be argued that the satellites orbiting Earth are also a special kind of robots, as they are performing a specific task specified by man made instructions. Some are used much like space probes in order to analyze the composition of our planet. Others are used to transmit communication signal (tv, internet, phone), or localizing objects (GPS, Gallileo).

Robots out of Earth are facing highly specific constraints due to their distance from any human being. The farthest away from earth the robot is, the more autonomy is required. Moreover, since humans can't repair these robots if a failure occurs, every possible problems have to be taken into account during the design phase.

1.2.2 Robots in Hazardous Environments

Back on earth, some environments are too dangerous for humans (e.g. deep water, nuclear power plant). In order to conduct human activity in such environments, specifically designed robots are used. Such robots are often teleoperated by humans from a safer remote place. For example, iRobot sent four robots to the Fukushima nuclear power plant after the accident of the 11th March 2011. Robots have also been used after the Deepwater Horizon oil spill in the 20th of April 2010, as humans can't survive at such depth ($\sim$ 1500m).

Recently, investigations have been carried to use robots in rescue missions, for example in building's rumbles after an earthquake, or during a fire [Kamegawa *et al.*, 2004; Baltes and Anderson, 2002]. These robots have to reach autonomously remote places, and be able to help and cooperate with humans [Murphy, 2004].

1.2.3 Industrial Robots

Industrial robots are often kept far away from any human because of their dangerousness. The car industry is known to be the first user of industrial robots, as the first industrial robot has been used in the assembly lines of General Motors [Nof, 1999]. These robots are typically moving heavy objects with great precision and are therefore equipped with powerful motors. Moreover, as they are performing always the same task (moving the same kind of object along a fixed path), they are usually equipped with few to no sensors. Since the combination of these two characteristics (i.e. powerfulness and lack of sensory information) make them a potential threat to any human, such robots are usually confined in areas where no human is allowed. The Baxter robot developed by "Rethink Robotic", and presented in 2012 challenges this problem with an industrial robot able to operate safely close to humans.

The difficulties faced during the conception of this type of robots are mostly due to the need of precision, strength, and robustness in a single device.

1.2.4 Security Robots

Robots are used daily for security and military purposes. Military robots are partly used without weapons for demining missions or recognition missions thanks to Unmanned Aerial Vehicle (UAV). UAVs have also been equipped with weapons and used for offensive missions (e.g. the MQ-1 predator[2]). Ground robots used by the army have also been equipped with weapons (e.g. Gladiator TUGV[3]). For the moment, the decision to fire a weapon is always attributed to a human operator.

[2]http://www.af.mil/information/factsheets/factsheet.asp?id=122 - Retrieved 12 December 2012
[3]http://www.globalsecurity.org/military/systems/ground/gladiator.htm - Retrived 12 December 2012

The use of these robots in the civilian area has recently started with the development of unarmed UAVs at the American-Mexican border from the 1st of September 2010[4]. Even closer to the everyday life, surveillance robots such as Rovio[5] designed by the WowWee company are available to the general public [Sanchez and Alonso, 2012].

These robots are raising ethical questions due to the dangerous combination of autonomy and possibility to kill humans [Lin *et al.*, 1998]. Moreover, the large range of conditions faced by these robots make them difficult to design. Since most of them are at some point teleoperated, the ergonomic of the interface used by the human operator is also of primary importance.

1.2.5 Medical Robots

Medical robots are used as closely as possible to humans but are always under the control of specialists. They are mainly used in two cases: surgical operations, and rehabilitation.

During surgical operations, robots are used to enhance the possibilities of minimaly invasive surgery techniques and the capacity of surgeons. They do so by smoothing the hand movements of surgeons and therefore improving their precision [Guthart and Salisbury, 2000; Al-Ahmad *et al.*, 2005]. In this context a robot is teleoperated. In some cases the surgeon doesn't need to be in the same room as the robot, and can therefore operate from anywhere in the world [Taylor *et al.*, 2008]. However, these procedures require a special training from the surgeon (which increase the length of operation) [Chang *et al.*, 2003], and leads to higher price of operations [Shukla *et al.*, 2010]. Robots are also used to perform routine tasks alongside the surgeon. A key challenge to the design of these systems is to relieve the surgeon, while still remaining under his control [Mettler *et al.*, 1998].

Robotic devices are also used for rehabilitation purposes, targeting two aims : the development of therapies solely based on robotics and the use of robots as therapy aids [der Loos and Reikensmeyer, 2008]. This field has already developed into subfields focused on assistive devices, special needs education, mobility, prosthetics and orthotics, and robot mediated therapy. The newest developments are targeting the creation of exoskeletons to increase strength of movements, quality of movements or to aid the realization of movements [Hillman, 2003].

1.2.6 Domestic Robots

Robots are making their apparition in the domestic area by doing some of the chores of humans. Additionally to the surveillance robots given in example above, robots are now available to clean the floors. The first of this type was Roomba from the iRobot company (2002)[6]. This robot is now available in multiple series, and other companies are producing the same type of robot.

[4]http://www.homelandsecuritynewswire.com/more-uavs-personnel-money-us-mexico-border-protection - Retrieved 12 December 2012

[5]`http://www.wowwee.com/en/products/tech/telepresence/rovio/rovio` - Retrived 5 December 2012

[6]http://www.irobot.com/us/robots/home/roomba.aspx - Retrieved 12 December 2012

Robots are also used for the entertainment of kids and adults. This trend started with the apparition of the Aibo[7] robot of Sony in 1999 (last year of production 2006). Aibo has been one of the first robot used widely in the domestic area and has lead to the creation of multiple hobbyist communities. It has also been used as a research platform, in particular for the Robocup autonomous soccer competition[8]. Nowadays, the Parrot flying robot[9], focuses on a lower price and increased interactions between humans thanks to the use of Augmented Reality games. While being mostly used at the moment for research purpose, the ultimate target of the Nao[10] robot developed by Aldebaran, is the general public, though it is not clear what it will be used for.

Domestic robots have to meet multiple contradictory constraints: low prices, high security, high flexibility. These constraints are imposed on the one hand by the variety of usage conditions (especially for entertainment robots), and on the other hand by their close interactions with a non-roboticist population (which won't take the time to tweak their robot every week).

Finally, the last decade has seen the rise of hobbyist robotic construction kits, such as the Bioloid kit from Robotis[11]. These kits are currently targeted for a technical savvy young adult population looking for affordable parts. For younger populations, adapted products are available, such as the Lego Mindstorm[12](1998). Lego kits are simpler to use but offer less flexibility on the sensors available and the programming methods one can use.

1.2.7 Research Directions

The use of robots in a large number of environments raise multiple scientific questions. We will here present how these questions are addressed by robotic researchers.

Reconfigurable robots. Different tasks might be better solved by robots of different shapes. Reconfigurable robotics explore the possibility to have robots changing their shapes with regards to the task they must solve and the environment they are facing. The works done in this field are based on the use of robotic units able to assemble and disassemble in larger structures. The demonstration of self-reconfigurable robots have been achieved in simulations [Spröwitz, 2010; Sproewitz *et al.*, 2008; Zykov *et al.*, 2008; Stoy *et al.*, 2010b], and in reality [Tolley and Lipson, 2011; Yim *et al.*, 2003]. In the later case, the demonstrations are done with a smaller number of units and are facing a long time of reconfiguration.

[7] `http://www.sony.co.uk/support/en/hub/ERS` - Retrieved 5 December 2012
[8] `http://www.cse.unsw.edu.au/help/students/robocup/robots.html` - Retrieved 5 December 2012
[9] `http://ardrone2.parrot.com/` - Retrived 5 December 2012
[10] `http://www.aldebaran-robotics.com/en/` - Retrived 5 December 2012
[11] `http://www.robotis.com/xe/bioloid_en` - Retrived 5 December 2012
[12] `http://mindstorms.lego.com/en-us/Default.aspx` - Retrived 5 December 2012

Soft robots. Roboticist aim at designing robots able to face a large number of environmental conditions. To this end, a sound choice in the design of a robot's body can reduce greatly the computational effort needed to control it [Pfeifer and Bongard, 2006]. For example, the use of a spring in the leg of a robot remove the need to compute a part of the leg's trajectory. This idea of saving computation time, by using the right structure to represent a problem is also found in the creation of control structures. Thus, by creating artificial models of the oscillatory structures found in biological brains, Isjpeert is able to create a virtual salamander behaving like its biological counterpart realistically on land and in water [Ijspeert *et al.*, 2007].

Micro/Nano robots. In order to modify autonomously systems of small size (under 1 micro-meter) a new type of smaller robots has to be developed. Robots of the micro-scale are already in use in the industry, biology and surgery [Nelson *et al.*, 2008]. Robots at the scale of a nanometer are however harder to design. They are expected to help in medicine by being able to directly interact with viruses, bacteria and blood cells [Kroeker, 2009]. Their capacity to interact with mater at the molecular scale, will be useful in many more applications from the "utility fog" to new materials and objects [Crandall, 1996]. Despite the growth of the field, many challenges remains to be solved before attaining such goals [Nelson *et al.*, 2008].

Swarm robots. Complex problems can be solved by coordinating the work of a high number of simple robots (typically more than 100). The incentive behind this approach is two folds [Sahin, 2005; Sahin *et al.*, 2008]:

- Robotic swarms are suited for tasks where a presence is simultaneously required in multiple places (e.g. surveillance).
- A swarm can continue its task if a robot fails, whereas the failure of a single robot will ruin the mission.

To achieve this goal, swarm robotics take inspiration from behaviors observed in swarm of insects such as ants and bees. Successful realizations have been shown in multiple scenarios detailed in Section 3.3.1. Moreover, robotic swarms are used as a modelization tool for the study of biological questions such as the emergence of altruism [Waibel *et al.*, 2009].

1.3 Challenges

Before analyzing, in the next section, **how** a robot may address objectives, we will clarify **what** problems a robot may have to address. The design of one robot doesn't necessarily implies to solve all these problems, and the list might not be exhaustive.

Navigation. As the examples given in the previous section have shown, the task of navigation is one of the most essential. Even the robots operating in a virtual world (web crawlers), are in some sense moving in their own environment. For most of the missions attributed to robots this aspect is necessary. The noticeable exceptions to this rule are industrial robots, far too heavy to be moved conveniently.

Manipulation. Often, robots are manipulating the environment, at least by pushing objects. In many cases the sole purpose of a robot is to move objects from one place to another, such as the factory robots of the Kiva company in charge moving good from storage to conditioning (this company has been recently bought Amazon)[13]. These manipulations are difficult for robots because they take place in a world designed for humans. Therefore, a robot has to be as precise and dexterous as a human will be. The most recent achievement in this domain have been done by Asimo who is now able to serve food and drinks[14].

Human robot interaction. Interactions between robots and humans are a key to their future developments as domestic partners (Sections 1.1.3 and 1.2.6). To this end, multiple developments are made on the interpretation of speech and human gestures by robots. A strong emphasis is given to the creation of robots whose behaviors can be understood by humans. This has lead to recent researches on the reproduction of facial expressions [Mayer *et al.*, 2010], and the simulation of personalities [Gerlinghaus *et al.*, 2012].

The aforementioned problems can be analyzed and solved independently from each other. However, we have identified another set of problems lying at their intersections.

Perception of the environment. In order to be able to act on their environment, robots should be endowed with perception abilities. These abilities range from the perception of the inner state of robot, to the perception of the world. On the one side, to measure the inner state of a robot, various sensors have been developed: battery state, torque applied by motors. On the other side, in order to perceive the world, vision sensors are usually completed with depth perception devices (telemetric laser, 3d camera), and microphones. In some cases, the perception of the environment or the current status of the robot is performed thanks to sensors situated outside of the robot, such as camera or localization systems.

Autonomy. Some tasks require the presence of mobile robots for long periods of time (e.g. see Sections 1.2.1, 1.2.4, and 1.2.6). For these situations, the robots should be able to harvest energy

[13] http://www.kivasystems.com/ - Retrieved 12 December 2012

[14] https://www.youtube.com/watch?v=1V9XUMCPGF8 - Retrieved 12 December 2012

from the environment (solar panel,robots that eat [Wilkinson, 2000], or power plug).

1.4 Robot Control

In order to see robots performing the tasks identified above in a wide variety of areas, dedicated bodies and software (to control these bodies) are designed. The software component is called a **controller**.

From its control perspective, robots can be divided in four categories, depending on their level of autonomy. **Teleoperated** robots follow orders given by a human operator in a continuous stream. The exact speed of each motor of the robot is assigned by the operator. The controller of the robot activate the motors to match the value given by the operator. Depending on the energy available in the robot and the response time of the motors, this task can be challenging. This approach is used when the robot is in close range of the human, and has few degrees of freedom (e.g. demining robots, robots operating in nuclear power plants). However, teleoperated robotics isn't suitable when the delay needed to transmit an order is too important (e.g. robots in space). Moreover, the manipulation of complex robots can't be achieved by this kind of approach, as the control of a large number of motors will be too difficult.

In the domain of **supervised robotics** [Kan, 1990], a greater level of autonomy is achieved by increasing the mapping's complexity between orders and motors' activation. The operator doesn't assign the value of each motor speed at each step individually, but gives an order of higher level which will be mapped to motors' speed by the controller itself. This approach is used notably to control multiple UAVs by a single user [Ruff *et al.*, 2004; Ryan *et al.*, 2004].

The degree of synchronization needed for teleoperated and supervised robots can't be achieved when a robot is in space. Moreover, a teleoperated robot require the constant care of a human operator. For repetitive, tedious tasks, another degree of autonomy is given to robots by mapping a task order to a sequence of actions (**task-level autonomy** [Giralt *et al.*, 1992]). Thus, an operator orders to a robot to perform a task (e.g. move to a point, retrieve a box, move an object, drill a hole) and save his attention to choose wisely the next task to be performed by the robot.

Finally, **full autonomy** is considered for robots out of reach of humans over an extended period of time, or performing a set of task requiring no supervision from humans. This is the case of robot developed to watch over and clean houses while the landlords are absent. As the field of autonomous computing is developing, new areas can be considered, such as the autonomous vehicles driving safely in towns (Darpa Urban challenge [Buehler *et al.*, 2009]). Moreover, full autonomy is a required property for multiple-robots systems, because they are raising coordination issues too complex to be solved by a single human operator.

1.4.1 Issues

Robustness to failures As an autonomous robot can operate in its environment for an extended period of time, changes in its environment and non-critical failures (due either to the normal wear and tear, or to exceptional conditions) are likely to occur at running time. To address these modifications, a robot should be endowed with robustness capacities, that is to say with the possibility to autonomously modify its behavior (i.e. modifications of the controller).

Multiple approaches have been proposed to realize robust controllers through resilience property. Among them, [Bongard *et al.*, 2006] propose an algorithm where a robotic controller constructs an internal model of its body shape in order to plan the next movement to perform. Later on, when a part of the robot is broken, the controller accurately adapts the model and its behavior. Another way to robustness through resilience is to rely on the redundancy of multiple parts (swarm robotics). With multiple disposable robots, the system performs its task, even when several robots are out-of-order. However, in this case, the sub-tasks have to be re-scheduled among the robots still active [Parker, 2000].

Another way to reach the robustness goal is to add a plasticity property in the controllers designed. Multiple tracks can be followed to obtain this property. A large number of behaviors can be provided at the creation of the controller [Mataric, 1997]. However, this approach is limited by the number of situations conceivable by the designer of the controller. Different learning approaches have been proposed to design controllers with plasticity and able to generalize their behavior in front of multiple environments (see Section 1.4.4).

Performance Measuring robot's performances on a specific task is a key element to assess the quality of a design method. A low performance means that the task would have been better carried by a human, while a high performance leads to benefits for humans at least in terms of time.

The no free lunch theorem state that no method can achieve optimality property on every class of problem [Wolpert and Macready, 1997]. To a certain degree if the optimality is ensured for a specific class of problem, the design method might lead to sub-optimal solutions in different class of problems. Therefore, design methods have to solve a trade-off between resilience and plasticity properties on one side, and optimality property on the other side.

1.4.2 Optimal and Hybrid Architectures

A first approach to the design of robotic controller is to optimize trajectories between two known points. This approach is especially useful to the command of robotic arms performing precise time constraints movements [Zefran, 1994]. In this context, the goal is both physical (reaching the right point), and temporal (reaching finishing the movement on time). This approach has been later extended to the optimization of mobile robots' motions [Laumond, 1998].

The first architecture proposed to design controllers (termed "deliberative" approaches) where based on the optimal control paradigm. Their origins can be traced back to the work of Nilsson Nils, where the controller is divided in three components: a sensing subsystem, a planning subsystem, and an execution subsystem [Nilsson, 1980]. In this approach, the sensing subsystem is in charge of mapping the sensory information into a representation of the world, from which further computations are done. The planning subsystem is in charge of computing a plan based on, the task to solve, and the representation of the world. Finally the execution layer executes the plan given by the planning layer. These principles have been used for example on the robot Shakey [Nilsson, 1984].

While being efficient to bring the robot to one precise position, and carry out long term goals, these methods were criticized as being difficult to apply to real world problems [Firby, 1987; Agre and Chapman, 1987]. Moreover the computation of plans can be computer intensive even for trivial movements, which results in a low response time.

On the one hand, the deliberative approaches provide an efficient way to carry out actions toward the realization of a long term goal. On the other hand, reactive approaches are able to react rapidly to informations from the environment. The hybrid paradigm aim at combining the best of these two approaches [Gat, 1998; Firby, 1989].

These architectures are composed of three layers running in parallel, termed: controller, sequencer, and deliberator. The controller is composed of multiple programs similar to the behaviors of the subsomption architectures (i.e. reactive loop coupling sensors and actuators). An external signal to the controller determines which behavior will be activated at any moment. The deliberator runs the, high level, time consuming, planning algorithms, and interacts with the sequencer. The sequencer choose which behavior to activate based on the up-to-date information from sensors, and the potentially deprecated but highly informative results from the deliberator. A review of the successful implementations of this paradigm, as long as a comparison with the deliberative and reactive paradigms are given in [Nakhaeinia *et al.*, 2011].

Probabilistic robotic approaches are focused on the realization of robots able to deal with the uncertainty of their environment. For this, they rely on the two following statistical tools: bayes rule and bayes filters [Thrun *et al.*, 2005a]. The main idea is to represent the world as probability distributions, and act on it based on this probabilistic information. The perception and motion models (used for all task of the probabilistic robotics) are built thanks to implementations of bayes filters [Sahin *et al.*, 1998; Thrun *et al.*, 2005c]. Probabilistic robotic methods are used to address three typical tasks: Robot localization within a map of the environnement [Ortin *et al.*, 2004; Cox and Leonard, 1994], construction of a map of the environment [Thrun, 2002; Tomatis and Nourbakhsh, 2002], and planning and control of the robot actions [Kaelbling *et al.*, 1998; Thrun *et al.*, 2005b]. These methods are relying on computational intensive algorithms, which results either in the integration of powerful computing unit in the robot, or necessary approximations in the computation of statistics. In known environments or when highly reliable sensors are used, other methods presented above might be preferred.

1.4.3 Reactive Robots

The reactive paradigm is one of the approach proposed to address these issues. Under this paradigm, the robots' actions are computed solely based on the information available through sensors. Therefore these approaches are focused on the design of robust controllers. The term "situated" is also used to name these approaches, placing an emphasis on the grounding of the controller in the robot and therefore in the world. We will see how the performance issues have been addressed by successive contributions. We will also analyze the inherent issues linked to the use of this approach.

The braitenberg vehicles are the first to illustrate this approach even if only by experiment of though [Braitenberg, 1986]. In this work, robots (presented under the term vehicles) are reacting to the presence of light solely based on direct links between sensors and motors.

Subsomption. Subsomption architectures are based on the wiring of simple behaviors together in order to produce higher level behaviors [Brooks, 1986; Brooks, 1990]. Within this paradigm illustrated in figure 1.1 a behavior can either be inhibited (by other behaviors' suppression output), or restarted.

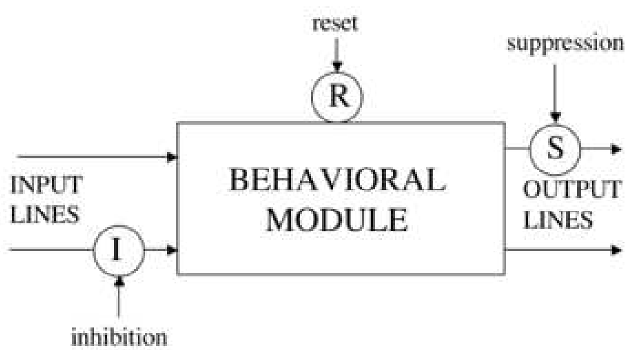


Figure 1.1: Representation of a behavior part of a subsomption architecture. From A Robust Layered Control System for a Mobile Robot (1986).

Within this paradigm, the human engineer can design more complex behaviors. However the design of controllers able to follow long term plans remains a goal out of reach.

Biologically-inspired approaches The biologically-inspired approaches pioneered in 1991 by Wilson [Wilson, 1990] focus on the reproduction of animal behaviors in order to produce an Artificial Intelligence (the artificial animals produced are called animats). The incentive to do so is to use the reproduction of animal intelligence as a proxy to human intelligence. It can be opposed to every other approaches as being a "bottom-up" approach where previous paradigm reviewed where "top-down".

This approach has lead to the reproduction of insects' behavior [Meyer, 1995], automatic evolution of behaviors [Gruau, 1994], and also to produce original behaviors for flying robots [Doncieux *et al.*, 2006].

1.4.4 Learning and Optimization

The works reviewed previously rely on a knowledgeable human engineer able to foresee much of the situations the robot will face during its operating time. Moreover, it is implicitly supposed that the designer is able to cope with the complexity of both the environment and the robot. While these assumptions may sometimes be fulfilled, they are invalid for complex situations (e.g. robot performing a large range of manipulations). A way to solve this issue is to conceive algorithms able to design automatically robot controllers.

Reinforcement Learning problems. Reinforcement Learning (RL) is a field of Machine Learning (ML) concerned with the optimization of an agent's sequential actions, in interaction with a specified environment, in order to maximize a possibly delayed reward [Sutton and Barto, 1998]. Adding learning abilities to a robot's controller is a specific case of RL where: the agent is the robot's controller, the environment is the world as perceived by a robot through its sensors, actions are the movements a robot can do thanks to its motors, and rewards are the quality evaluations given by a human designer (directly or through the design of a reward function). Due to the consideration of a robotic setup, specific issues have to be addressed since, the states, actions, and reward spaces are continuous and noisy.

One way to solve RL problems is to use the TD methods which aim at learning an optimal policy based on Bellman equations [Sutton and Barto, 1998]. These equations are based on the computation of a cumulative reward associated to each state, from which the optimal controller is deduced.

Q-learning algorithms use this method to learn a Q-function, which characterizes the predicted reward of performing a certain action in a certain state. By following the maximal reward path in each state, and updating the Q-value after each action, the controller of a robot will learn to behave optimally.

Evolutionary robotics. Evolutionary Algorithms (EA) are population based stochastic optmization algorithms. Within these methods, a population of solutions to a problem is iteratively evaluated, and re-generate from its best members. These methods are based on the evaluation of a controller candidate, and therefore don't rely on the computation of a model of the environment. They can be viewed as biologically inspired policy search methods [Moriarty *et al.*, 1999; Heidrich-Meisner and Igel, 2008], which are suited to solve Reinforcement Learning problems. Moreover, with these methods different controller's representation can be explored [Back *et al.*, 1997].

The proposition to use these methods in robotic scenarios has led to the creation of the Evolutionary Robotic (ER) domain [Cliff *et al.*, 1993], which has since demonstrated numerous applications and opened the way to other domains [Walker *et al.*, 2003; Gross *et al.*, 2006b]. The different representations, evaluations, and replacement procedures, used in robotics scenarios are detailed in Chapter 2.

Developmental robotic. While being also biologically inspired, the developmental robotic field (also called epigenetic robotics) focuses on the development of the mind by going through Autonomous Mental Development (AMD). The goal of developmental robotic methods is to achieve behavioral capacities similar to the human ones by imitating the incremental cognitive development observed in animals [Weng *et al.*, 2001]. The methodological difference with ER lies in the considered time length of learning: a lifetime for developmental robotic, multiple generations for ER methods.

Since the first system of this type (Cresceptron [Weng *et al.*, 1997]), the field has grown in importance and has achieved major results, such as learning vision guided movements [Olsson *et al.*, 2006], and learning how to navigate in an environment [Provost *et al.*, 2006]. A recent review of these achievements can be found in [Asada *et al.*, 2009]. Naturally, this field grows in close interaction with the ER field and combined approaches have already been proposed [Doncieux *et al.*, 2012].

1.5 Summary and Conclusions

In this chapter we gave a brief overview of the robotic field, which is focused on the design of robots for a large number of applications. Each application field leads to different challenges and require robots tailored to its need.

We have highlighted multiple issues in the design of robotic controllers, namely :

- **Robustness to failures:** Due to the normal wear-and-tear, or exceptional conditions, a robot might be partially broken. Moreover it might be deployed in highly different environments. Controllers should be robust to all these unpredictable events.

- **Performance:** In order to be useful to the end user, a controller should display an optimal behavior for certain class of problems. However this property comes as a trade-off to the property mentioned above.

We have reviewed the strengths and weaknesses of different approaches to the design of robotic controllers. Among the design methods reviewed, learning methods stand as good candidates to address the trade-off between resilience, adaptability and optimality properties.

In the remaining of this thesis, the adaptation of robots to unknown environment will be addressed. As such, learning algorithms stand as good candidates toward ensuring the integrity of a group of robots in this context. In the next chapter the field of Evolutionary Computation will be reviewed, as it as already shown its ability to stand as an efficient learning method for robotic behaviors in unknown environments, including in an on-line setup.

Chapter 2

Evolutionary Robotics

In this thesis the problem of the integrity of a group of robots in unstructured environments is addressed. This is a challenging problem as it is poorly defined, and its resolution can't rely solely on the expertise of the human designer. The most promising method to solve this type of problem is to use Evolutionary Algorithms tailored for robotic scenarios. Indeed this approach has already proven to provide valuable results in context where the objective function is ill or poorly defined, and where few background knowledge is available. In Sections 2.1 and 2.1.1 we will introduces the Evolutionary Computation main principles, how it can be used for the design of autonomous robots, and the issues faced by this approach in relation to our scenario. Questions raised by the design of fitness functions along with solutions proposed in the literature will be presented in Sections 2.2 and 2.3. One of our contribution to this field is presented in Section 2.4.3, as an illustration of some of the challenges faced when addressing on-line adaptation.

2.1 Principles of Evolutionary Computation

Evolutionary Computation (EC) is a research area of Computer Science concerned with the design of biologically inspired optimization methods called Evolutionary Algorithms (EA). These algorithms are based on a population of individual going through a stochastic optimization process [Baeck *et al.*, 1997]. Within the scheme presented in figure 2.1 the population goes through recursive phases of selection and mutation. These two mechanisms (inspired from the two main principles drawn by Charles Darwin in his book "Species Origins") can be defined as follows:

- **Variation:** Variation's operators are used to bring novel solutions in the population. In this phase stochastic variation operators are applied to a restricted set of individuals in order to produce new original individuals.

- **Selection:** The aim of the selection process is to increase the quality of the individuals present in the population. During the selection process each individual is given a fitness value characterizing its performance on the problem to solve. The best individuals (i.e. those with the highest fitness values), are likely to be selected to produce the population at the next generation.

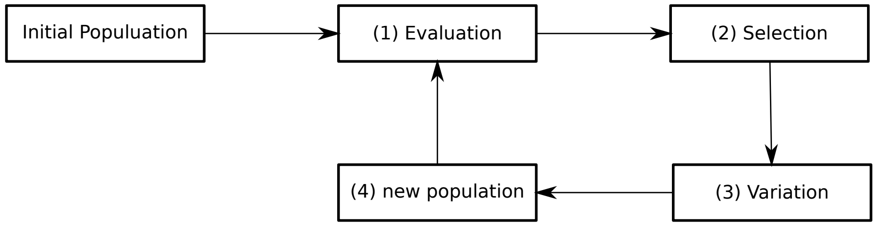


Figure 2.1: Schematic representation of an Evolutionary Algorithm's concept.

Selection operators. Multiple selection scheme exist, and are used differently depending on the problems at hand (details in [Eiben and Smith, 2008a]). We highlight here the most studied mechanisms:

- **Fitness proportional selection:** Within this selection scheme, a roulette wheel mechanism is implemented such that the individuals with the highest fitness are the most likely to be selected. This mechanism has been well studied, and numerous problems exist: premature convergence, lack of selection pressure, and dependence to transposition of the fitness function.

- **Ranking selection:** This variation of the fitness proportional selection uses a roulette wheel mechanism based on the ranking of the solutions. This mechanism avoids the main drawbacks from the fitness proportional selection.

- **Tournament selection:** When the population is very large, or distributed among multiple systems, computing the distribution of individuals performance on the entire population can be hard or even impossible. The tournament scheme has been designed to deal with this issue by relying on small subsets of the population. Within this scheme, k individuals are chosen randomly in the population to be evaluated. The individual with the highest fitness is selected. With this scheme the selection pressure is easily controlled by varying the tournament size k.

Variation Operators. Two different types of variation operator are mainly used in Evolutionary Algorithms:

- **Mutations:** These operators are used on a single individual, from which they produce a (slightly) modified version. Variations are used to bring new original solutions in the population. The implementation of these operators can vary depending on the properties of the genotypic space (continuous or discrete). Independently of the search space, genes under mutation are selected randomly with a probability named "mutation rate".

- **Recombination:** Recombination operators are used on two or more individuals. They are used to recombine the genotype of multiple individuals in a new individual. The main incentive to use this operator is to generate a new individual combining the strength of the original individuals. These variations are also based on stochastic operators.

Depending on the individuals' representation in the genotypic space, different variation, and selection operators are preferred.

2.1.1 Encoding Schemes

Through the history of Evolutionary Computation different representation scheme of the individuals have been studied, each suited to different type of problems [Eiben and Smith, 2008b].

Genetic Algorithm (GA). Researches in this sub-field aim at solving combinatorial optimization problems. Solutions are represented as strings over a finite alphabet [Holland, 1975]. String of bits are the archetypal example, but other base system can be considered. The commercial software Evolver[1] is based on this approach. Rule based controllers are one example of controllers that can be optimized thanks to Evolutionary Algorithms. Successful applications of this approach are shown in single [Grefenstette *et al.*, 1990; Schultz and Grefenstette, 1992] and multi-robot problems [Grefenstette, 1992; Schultz *et al.*, 1996].

Evolutionary Strategy (ES). ES algorithms are concerned with the optimization of solutions to continuous problems. Individuals are encoded as strings of real values [Schwefel, 1995]. In this context the main variation operator considered is the mutation. This approach can be used over a large number of optimization problems [Herdy and Patone, 1994]. Artificial Neural Networks are one example of robotic controllers optimized thanks to ES algorithms. Since the first works on wall avoidance and photo-taxis [Cliff *et al.*, 1993; Nolfi *et al.*, 1994; Nolfi and Floreano, 2000], this representation has been used for example on multi-robot problems (see Section 3.3.2), and vision based task [Harvey *et al.*, 1994].

Genetic Programming (GP). This methodology uses Evolutionary Algorithms principles to modeling problems, that is to say to optimize models which produce a maximal feet between pairs of input and output. This sub-field is based on individuals represented as tree structures [Koza, 1992]. GP has been applied to the design of robotics controller in multiple contributions, for example: the design of an electronic circuit [Koza *et al.*, 1997], the design of a program controlling a robot [Nordin and Banzhaf, 1997; Lee *et al.*, 1997], and the design of controllers used in multi-robot scenarios [Kala, 2012]. Within this approach the solutions evolved tend to grow in size regardless of the performance. This

[1] `http://www.nytimes.com/1990/08/29/business/business-technology-what-s-the-best-answer-it-s-survival-of-the-fittest.html` - Retrieved 6 December 2012

problem is known as the code bloat [Amil *et al.*, 2009]. Multiple solutions are proposed to face it in [Smith, 2000]. Moreover GP relies often on a `LISP` interpretor to read the encoded solution, which might be too computationally expensive for a robot computation capacities.

Other encoding schemes have been proposed in the literature such as Evolutionary Programming which is originally based on the optimization of Finite State Machines [Fogel, 1962], or developmental systems which are focused on links between biological development, and evolution [Oyama *et al.*, 2001].

2.1.2 EC and Robotics

[Doncieux *et al.*, 2009] have distinguished three use cases for the application of Evolutionary Computation methods to the field of robotics:

- **Parameter tuning:** In this context, robotic systems have been modelized beforehand, and the control parameters of the system have been identified. However the optimal value of these parameters isn't known and can't be found neither by an analytical method (i.e. method not known) nor by an exhaustive search (i.e. too many parameters). Evolutionary Algorithms are then used to explore the space of parameters in search of a solution. A survey of such approaches used for the control of electronic systems in general is found in [Fleming and Purshouse, 2002]. Another survey of these approaches applied to the optimization of bio-inspired Artificial Intelligence systems is found in [Floreano and Mattiussi, 2008].

- **Evolutionary aided design:** Evolutionary Algorithms are also used to characterize the behavior of a system. In this context Evolutionary Computation methods are used to explore the design space and propose a variety of solutions to the human engineer, who can then analyze the results in order to gain a deeper understanding of the system. This has been used for example in the design of UAV's controllers [Hauert *et al.*, 2009], the control of flapping wings [Doncieux, 2009] and lately to the friction stir welding problem [Bandaru *et al.*, 2011]. Multi-objective Evolutionary Algorithms are a special kind of Evolutionary Algorithm designed to find the best trade-offs between multiple objectives [Deb, 2001]. This type of algorithm has been used to find relations between design parameters in a process called "innovization" [Deb and Srinivasan, 2008a; Deb and Srinivasan, 2008b].

- **Evolutionary synthesis:** The use of Evolutionary Algorithms isn't restricted to the optimization of robot's controllers but are also used to optimize their overall design (structure and controller). These approaches take into account the embodied theories by considering the controller of a robot, its morphology, and its interaction with the environment all at once in the design process. This approach is used for example in the Golem project where robot structures and control architectures are optimized at the same time to be latter produced by 3D printer and deployed in the real world [Lipson, 2007]. This field of ER is still recent and work is ongoing to address scalability and robustness issues [Hornby *et al.*, 2005; Gauci and Stanley, 2007; Mouret and Doncieux, 2009a].

2.2 Fitness Functions

The formulation of the fitness function is an important step in the design of an evolutionary robotic experiment. We present here the main types of fitness functions seen in the literature, and take inspiration from [Nelson *et al.*, 2009] to build this classification.

Behavioral fitness function. This class of fitness function, is suited if the human engineer can provide a large amount of knowledge on the problem to solve. Within these methods, the behavior of a robot is closely monitored in order to assess its performance. This type of fitness function was used in the first works of the ER field [Floreano and Mondada, 1996]. It proved its relevance in different locomotion tasks such as object avoidance, object following, wall following, and light avoidance [Banzhaf *et al.*, 1997]. Another example is found in the locomotion of an octopod robots [Gomi and Ide, 1998]. This approach leaves few degrees of freedom to the evolutionary process and implies that the human engineer has a precise idea of how to perform the task.

Tailored fitness function. This approach is based on the measure of task completeness, combined with one or multiple behavioral measure terms detailed in the previous paragraph. For example, in a photo-taxis task, a tailored fitness function could be composed of two main components. The first one rewarding a robot that arrives at the light source, the second one maximized when the robot face the sun. This type of fitness function is task-specific but tailored by the human engineer to accommodate the given problem. It is one of the most used class of fitness function in the ER field. Among the achievements made, one can count ball collection [Hoffmann and Pfister, 1996], coordinated movements [Quinn *et al.*, 2002], sequential tasks [Doncieux and Mouret, 2010], and gait learning for a quadruped robot [Hornby *et al.*, 2005]. Within these approaches, the human engineer should know the elements necessary to the success of the task.

Aggregate fitness function. Aggregate fitness functions reward the accomplishment of a given task or sub-task but uses no information from the human engineer on how to do the task. This type of function aggregates all aspects of a robot's behavior in a single term. For example, if the task is to come closer to an object, the fitness function will be the final distance between the robot and the object. Aggregate fitness functions have been applied successfully in multiple cases such as gait learning [Earon *et al.*, 2000; Augustsson *et al.*, 2002; Zykov *et al.*, 2004; Chernova and Veloso, 2004] and goal soccer tasks [Hornby *et al.*, 2000].

On the one hand, if the human engineer doesn't have any knowledge about the system optimized this method can be used to produce original solutions. On the other hand, at the beginning of an experiment every strategies are performing equally bad. It is then difficult to determine which solutions should be selected to the next generation(*bootstrap problem)* [Kawai *et al.*, 2001].

Another class of fitness function targeting the same class of problems are the multi-objective fitness functions. This approach propose to divide a fitness function in multiple simple objectives and then explore the Pareto front of these objectives [Deb, 2001]. This approach can be used to solve the bootstrap problem by adding a novelty objective (which favor original solutions regardless of their performance [Lehman and Stanley, 2011]) to the main objective [Mouret and Doncieux, 2009b].

Implicit fitness function. Implicit fitness functions are used when the task to perform isn't known beforehand by the human engineer, and might change with time. In this context the optimization process is based on the pressure to survive. That is to say, a survival indicator (often energy) is used as the fitness function of the Evolutionary Algorithm. The optimization of the behavior for a given task is then the implicit result of the maximization of the fitness function.

The maximization of the fitness function will lead to the evolution of different strategies, depending on the environment at hand (possibly composed of other robots). Consequently, this type of fitness function can be used in numerous scenarios without any modifications.

This approach is still in its infancy, and is mainly used in Embodied ER (see Section 3.3.2). We have also found one application investigating the notion of creativity, where [Bird *et al.*, 2008] use this approach to study the traces of robots as drawings resulting from the pressure to harvest energy in order to survive.

2.3 Transfer to Real Robots

While in simulation the world is perfect (no noise), modifiable at will, and the evaluations reliable, the real world introduces noise in sensors and actuators. As a consequence solutions optimized in simulation might not answer to a real problem but only to a problem occurring in simulation [Brooks, 1992]. This results in the observation of significantly different behaviors when control architectures are transfered from simulation to reality (*reality gap* [Jakobi *et al.*, 1995]). Moreover, robots have multiple limitations which make them ill-adapted to run Evolutionary Algorithms on-board [Matarić and Cliff, 1996; Nolfi and Floreano, 2000]:

- **Cost:** Robots remain expensive devices. Unfortunately , the evaluation of poor controllers likely to damage one robot is common at the beginning of an evolutionary run.

- **Speed:** Evolutionary Algorithms need a high number of iterations in order to evaluation one solution. The main incentive is to actually assess the quality of a solution in a significant part of the conditions encountered.

- **Reliability:** Despite the latest improvements, robots remain subject to failure, which results in multiple repairs slowing down Evolutionary Algorithms.

In order to solve these issues two main solutions have been proposed:

- Modify the simulation environment so that the solution optimized have a similar behavior in simulations and in the real world (see Section 2.3.1).

- Produce new Evolutionary Algorithms to carry on evaluations on real robots (see Section 2.3.2) [Pollack *et al.*, 2000].

2.3.1 Simulations

Design of simulation environment. [Miglino *et al.*, 1994] advocates that keeping simulation simple and adding an optimal level of noise is a solution to the reality gap problem. For example, [Jakobi, 1993] propose a framework to design minimal yet efficient simulations, and [Nolfi *et al.*, 1994; Miglino *et al.*, 1995] propose to model the behavior of real sensors in the presence of the object composing the environment. In the later method the behavior of the actuators are modeled from measures made in real conditions. This approach is highly efficient (no decrease of fitness observed), but can be used only in a restricted set of conditions (i.e. when the task is known beforehand and the environment measurable through the sensors used on the real robots).

Learning real world dynamics. In order to avoid the beforehand task of building a simulation, methods have been proposed to learn it based on the measures of the robot itself. This way, an always up-to-date model is built by taking into account the actual shape and behavior of the robot. This approach can be realized by combining an Evolutionary Algorithm with a TD algorithm [Kamio and Iba, 2005](the TD algorithm is used to fill the reality gap). However, it has been later criticized for its need of numerous real-world data [Bongard and Lipson, 2005], and new approaches have been proposed based on the co-evolution of environment model and robotic controllers [Bongard and Lipson, 2005; Bongard *et al.*, 2006; Koos *et al.*, 2009].

2.3.2 Embodied Trials.

In order to address the reality gap problem, the evaluation procedures can be carried on the robots directly. On the one hand, the evaluation times are longer and the work of the human engineer is often increased by the use of real robots. On the other hand, solutions optimized are fit to real world problems, and the human engineer doesn't have to design an adapted simulation environment. This approach has been used in numerous cases where it is difficult to build an accurate simulation, such as gait evolution for legged robots [Gomi and Ide, 1998; Earon *et al.*, 2000; Andersson *et al.*, 2000; Wolff and Nordin, 2001; Okura *et al.*, 2003; Gu *et al.*, 2003; Chernova and Veloso, 2004], legged robots playing football [Hornby *et al.*, 2000], snake-like robots made of shape-memory alloy [Mahdavi and Bentley, 2006], and flying robots [Zufferey *et al.*, 2002].

Within the *embodied trials* methodology, the human engineer has still a role to play by re-setting the initial conditions of each evaluation, running the Evolutionary Algorithm in a computer, preventing any damage done to the robot, or repairing it in the worst case. Moreover, this approach makes the assumption that the operational conditions will remain constant and conform to the conditions used during the learning phase.

2.4 Beyond Classic ER

2.4.1 Morphological and Controller Co-evolution.

Recent works have investigated the possibility to benefit from morphological changes in order to acquire robust behaviors in front of unstructured environments. This approach has been first envisioned in [Sims, 1994]. [Bongard, 2011] has shown that the co-evolution of body and mind leads to faster optimization and more robust solution. This approach has been first used by [Lipson and Pollack, 2000; Lipson *et al.*, 2006], where bodies and controllers are simulated in the same time and then brought to the real world thanks to a 3D printer. It has also been demonstrated by using affordable modules in [Macinnes and Paolo, 2004].

However these works are relying on the simulation of behaviors before there transfer to real robots, and are therefore facing the reality gap problem. One solution to address this problem is to design robots able to modify autonomously their morphologies at running time, that is to say self-reconfigurable robots [Stoy *et al.*, 2010a]. These robots can be used in combination with Evolutionary Algorithms in order to co-evolve the morphologies and controllers of robots [Yim *et al.*, 2007; Eiben *et al.*, 2010b].

2.4.2 On-line On-board Evolution.

In classical use of evolution for the design of robot controllers, the human engineer is deeply involved in the management of evolutionary runs. Moreover, once the a solution has been evolved, the evolutionary process is never used again [Eiben *et al.*, 2010b]. We are presenting here an approach to go beyond these two features, that is to say to evolve robotic controllers without human intervention, and continuously adapting during the operation time. These approaches are called on-line approaches as opposed to off-line approaches used in a typical *design then use* fashion.

Another consideration is given as to *where* evolutionary operators are used, i.e. in a computer (off-board) or in the robot themselves (on-board). In order to free the human engineer from continuous management, the Evolutionary Algorithms are run in the robot.

Specific issues have to be considered for the design of on-line on-board Evolutionary Algorithms:

- **Noisy fitness evaluation:** First, the on-line, on-board evaluation of robot controllers leads inevitably to noisy fitness functions. This is principally due to the variability of the initial conditions used to evaluate the genome. The evaluation of a controller starts with the robot as it was left by the previous controller. Therefore an evaluation can start in a difficult part of the environment or with a partially broken robot.

- **Robotic constraints:** The use of real robots implies a limited computational power. This constraint implies the design of "lightweight" Evolutionary Algorithms, with small population sizes, which reduce the exploratory capacity of the algorithm, and therefore increase the risk of premature convergence.

The on-line on-board approach has been demonstrated in obstacle avoidance and object attraction [Nordin and Banzhaf, 1996] , obstacle avoidance based on vision [Floreano *et al.*, 2002; Marocco and Floreano, 2002], and gait learning in a quadruped robot [Hornby *et al.*, 2005]. However, these contributions have dealt with the issues raised above by tailoring Evolutionary Algorithms to a specific task. Because of the lack of general mechanism to deal with the issues of on-line on-board algorithms, it is difficult to use these contributions on any task.

This approach has been considered for problems involving multiple robots [Watson, 1999]. In this context, the Evolutionary Algorithms can benefit from the possibility to evaluate solution in parallel. However, multiple challenges relative to the decentralization of the algorithm, and the interactions between robots have to be addressed. This approach is reviewed in Section 3.3.2.

2.4.3 An Illustration of Challenges in On-line On-board Evolution: the 1+1-restart-online Algorithm

In this section, one of our contribution to on-line on-board evolutionary robotics is summarized [Montanier and Bredeche, 2009]. The (1+1)-ONLINE adaptation algorithm from [Bredeche *et al.*, 2009] is presented along with a limitation regarding its ability to perform global search in the space of possible solutions. A new algorithm is described, which addresses the trade-off between local and global search by relying on a restart procedure whenever the algorithm is stuck in a local optima.

In the (1+1)-ONLINE algorithm, three mechanisms for the (1+1)-ES algorithm have been designed to deal with issues of on-line on-board evolutionary robotics (presented in the previous section):

- **Local and global search:** The mutation operator is the gaussian distribution $\mathcal{N}(0, \sigma)$. A low value of σ (resp. high) will result in a local (resp. global) search by minor (resp. major) modifications of the original genome. In the (1+1)-ONLINE algorithm the σ parameter is initially set to a low value and increased as long as the champion isn't replaced (gradual shift toward a global search). Each time a challenger replace the champion, σ is reset to its initial low value. Thanks to this mechanism, the search will be broaden to new regions of the search space if a local optima is found.

- **Re-evaluation:** Individuals may get lucky or unlucky during evaluation depending on the environment at hand. This is a typical problem related to fitness noise. An efficient solution is to re-evaluate individuals, as proposed by Beyer [Beyer, 1998].The re-evaluated fitness overwrite the fitness of the champion. This is done to promote individuals with a low variance in their performances. One of the drawback of the overwriting method is that good individuals could be replaced by inferior but lucky individuals. If an individual is lucky during its first evaluation but has a low mean fitness, it will not survive the next re-evaluations. As a consequence, the Evolutionary Algorithm won't be stuck with bad individuals.

- **Recovery:** Since the Evolutionary Algorithm runs without human intervention, the robot isn't repositioned after each evaluation of one individual. For example, a genome may be evaluated starting from completely different initial conditions, such as in front of a wall or in a tight corner. To avoid penalization of good genomes, a *recovery period* is introduced: during this time, the robot behavior is not considered for evaluation (ie. "free of charge"), which favors genomes that display good performance independently from the starting position.

In the (1+1)-ONLINE algorithm an issue has been identified regarding the abilities of the algorithm to perform global search. More precisely, if a really good champion is found, other challengers aren't considered for a long period (until the champion get unlucky in one of its re-evaluation). This hinder the ability of the algorithm to find a large number of solutions to a given problem.

In order to address this issue a restart mechanism [Auger and Hansen, 2005] is proposed based on the number of re-evaluation. The indicator chosen to trigger the restart mechanism is the number of re-evaluation. The incentive to do so is that a champion has to be highly capable and robust in order to obtain good results at each re-evaluation. If the champion fails one of its re-evaluation it will be replaced by the next lucky challenger. Therefore, the number of re-evaluations is a measure of the champion's quality in comparison to the surrounding challengers. A high number of re-evaluations (threshold to be determined experimentally) indicate that the current champion is a local optimum from which the algorithm can't escape. Therefore, once a specified threshold of re-evaluation is reached, the algorithm is restarted. The resulting algorithm is named (1+1)-RESTART-ONLINE and described in Algorithm 1.

The objective function used in experiments to test the (1+1)-RESTART-ONLINE algorithm aim at optimizing a controller for exploration and is close to the one described in [Nolfi and Floreano, 2000]:

$$fitness(x) = \sum_{t=0}^{n} V_t * (1 - V_r) * (1 - D_M)$$

where V_t is the speed factor, V_r is the rotation factor, and D_M the value of the less active sensor, all values are normalized between 0 and 1. Distance sensors returns higher value when they are close to a wall. Therefore, individuals achieve high fitness value while moving fast and forward and avoiding walls.

Algorithm 1 The (1+1)-RESTART-ONLINE Evolutionary Algorithm.

```
for evaluation = 0 to N do
  if random() < P_{re-evaluate} then
    if re - evaluation - count < Max - re - evaluation then
      Recover(Champion)
      Fitness_{Champion} = RunAndEvaluate(Champion)
      re - evaluation - count + +
    else
      σ = σ_{min}
      Champion = RandomGenome()
      Fitness_{Champion} = 0
      Challenger = RandomGenome()
      Fitness_{Challenger} = 0
    end if
  else
    Challenger = Champion + N(0, σ) {Gaussian mutation}
    Recover(Challenger)
    Fitness_{Challenger} = RunAndEvaluate(Challenger)
    if Fitness_{Challenger} > Fitness_{Champion} then
      Champion = Challenger
      Fitness_{Champion} = Fitness_{Challenger}
      σ = σ_{min}
    else
      σ = σ · 2
    end if
  end if
end for
```

The (1+1)-RESTART-ONLINE has been tested in simulation with a micro-controller featuring real hardware of the Symbrion project, and in real world with the popular robotic kit Bioloid[2]. The Figure 2.2 shows the real wold experimental setup, and Figure 2.3 shows the simulation experimental setup.

Results of experiments have shown that the 1+1 RESTART-ONLINE algorithm is actually able to optimize wall avoidance behavior. Moreover, we have demonstrated that the 1+1 RESTART-ONLINE

[2] http://www.robotis.com/xe/bioloid_en - Retrieved 6 December 2012

algorithm address the identified design flow of the 1+1-ONLINE algorithm, which leads to wider exploration of the search space, potentially making it possible to visit more local optima than the previous implementation, and possibly increasing the probability to end up in a global optima. It should also be noted that results from simulations actually show that divers behaviors can be obtained (see [Montanier and Bredeche, 2009] for full details).

Figure 2.2: The experimental setup used with the real robot.

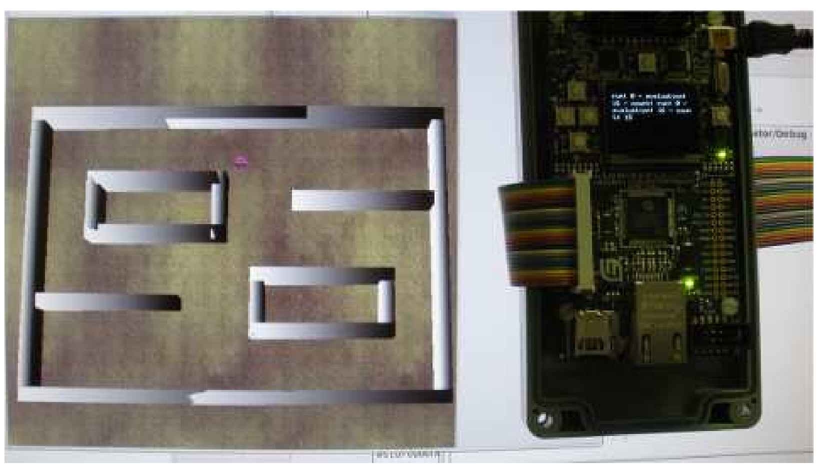


Figure 2.3: The experimental set-up: the Cortex M3 board connected to Symbricator3d.

It is important to remind the reader that this section presents only a summary of a contribution in order to highlight important issues in Embodied ER such as noisy evaluations and lack of help from the human engineer. The bottom line is that whenever on-line ER is considered, specific algorithms must be deployed.

This algorithm has later been extended toward a $(\mu + 1)$ strategy [Haasdijk *et al.*, 2010]. The

$(\mu + 1) - online$ algorithm has then been exented to multi-robot problems in [Eiben *et al.*, 2010a; Haasdijk *et al.*, 2011; Weel *et al.*, 2012].

2.5 Summary and Conclusions

In this chapter the application of Evolutionary Algorithms to the robotic field has been presented. After a general presentation of ER and its main applications, a focus has been done on two issues faced by this field: the design of a fitness function, and the transfer of solutions to real robots. Finally recent approaches beyond classic ER have been presented. Within one of these fields (on-line on-board ER), one of our contribution has been presented as one example to illustrate issues faced and possible solutions.

Notably, in every cases reviewed, the fitness function is designed by the human engineer. Even implicit fitness functions (which are used to address ill defined problems) are designed by the human engineer. This aspect raises issues in front of complex environments. In these contexts the task to perform can become more complex due to environmental constraints. In worst cases the fitness function might not be able to formalize every aspects of the environment and task at hand. One example of such complex situation is the conservation of robots integrity in front of unknown environments. In the next chapter, we investigate how to address such complex situation without relying on possibly misleading background knowledge from the designer, that is, without fitness function.

Chapter 3

Environment-driven Distributed Evolutionary Adaptation

In this chapter we present the challenges faced in the autonomous adaptation of groups of robots for ill defined tasks and unknown environments (see Sections 3.1, and 3.2). A solution to this class of problem is a first step toward the maintenance of a robotic service necessary to the optimization of behaviors. A clear definition of this problem and the challenges faced in its resolution is necessary to the creation of an algorithmic response.

In Section 3.3, we will present the fields related to this problem, and some of the solutions proposed to solve the challenges highlighted. In Section 3.4 we show why these solutions proposed can only address part of the challenges. Finally, this new class of problem is formalized in Section 3.5.

3.1 Scenario

As stated in the introduction, we are interested in large groups of robots deployed in environments unknown a priori to the human engineer [Baele *et al.*, 2009]. This situation happens for example if groups of robots are sold for household usage, as every houses are potentially different. In this context robots will have to solve a task which might be defined in a later phase (e.g. once the final consumer uses the group of robots). The maintenance of robots integrity is a service necessary to the optimization of behaviors toward the realization of a specific task in an unknown environment. We assume that this service has to be provided independently of the task considered (e.g. building monitoring in the search of intruders, or explore building rumbles), and the robot at hand (e.g. wheeled robots, or legged robots). Moreover this has to be done through unpredicted environmental changes (e.g. the group of robots is sold from one consumer to another).

Due to unknown variables (i.e. task to solve and environment at hand), good solutions to maintain integrity can't be pre-programmed by the human engineer beforehand. An auto-adaptive algorithm is a solution to ensure the integrity of robots independently of the environment, the type of robots, or the task. The auto-adaptation process will be solely driven by the environment and its impact on robots integrity.

We are addressing this challenging question in the context of populations of autonomous robots, able to perform local communications, immersed in an unknown, and possibly changing, environment, which implies :

- No external intervention: Human engineers cannot act on the robots once they are deployed. Therefore, if a robot ends up in a difficult situation or even breaks down (partly or completely) two scenarios can be envisaged. On the one hand, part of the remaining robots can repair the damaged robot or bring it back to a place where it can be repaired (e.g. a factor or the human engineer). On the other hand, if a repair isn't possible (e.g. broken robots on mars), other robots can continue to perform the task with the remaining members of the group only.

- No external communication: The deployment of a large number of robots implies the use of low cost robots. As a consequence, the communication abilities of such robots are limited, which implies that robots can only communicate with each other (external informations aren't available to the group of robots).

3.2 Challenges

Since the human engineer doesn't know beforehand **what** the environment will be, behaviors ensuring the integrity of the group of robots can't be predicted. Therefore, efficient strategis (from an integrity point of view) have to be found autonomously. We consider here a scenario where the human engineer isn't present to repair damaged robots, which means that the number of robots can't increase with time but might decrease due to hardware failure. Therefore, the algorithm developed has to solve autonomously the following challenges.

Autonomous exploration: Due to slow environmental changes, evolution of interactions between robots, variation in the number of available robots, different ways to perform the task in the environment can arise gradually. In order to take advantage of such changes, robots must search autonomously for new behaviors. Moreover, the search mechanism should capitalize on successes and failures of previous behaviors.

Restart of the search procedure: The environment might also change abruptly, thus modifying dramatically the most adapted survival strategy. To face such situations the search mechanism must be able to restart without the intervention of the human engineer. Consequently, the search mechanism has to be embedded in each robot and operate continuously and autonomously.

Exploration/Exploitation trade-off: On the one hand, the continuous search of new solutions impacts negatively the integrity of the swarm, as the evaluation of poor solutions can damage robots.On the other hand, the continuous use of the best known behavior will results in the impossibility to adapt after an environmental change. Consequently a mechanism is needed to solve the trade-off between using the best known way to survive (exploitation of the solutions found) and exploring new solutions.

Distributed cooperation: We have to consider the lack of a central entity synchronizing the experiences of all robots currently deployed in the environment. This directly results in the distribution of the learning process in the population of robots. Each robot must base its learning only on information transmitted locally by other robots and its own experience of the environment. In the remaining of this chapter, we will see that multiple issues arise from this challenge (combined with the previously mentioned challenges), such as the emergence of consensus toward strategies promoting survival at the population's scale and not only at the robot's scale.

Real robots' constraints: Due to real world robotics constraints (low computation power, and few sensory informations available), each evaluation of a robot behavior will be noisy. Moreover, since the starting conditions from each evaluation aren't normalized by a human operator, each evaluation is potentially erroneous.

3.3 Related Work

In this section, three approaches are reviewed that could stand as possible candidate to address the challenges at hand. The first one is called *Reinforcement learning applied to multi-agent systems* and is presented in Section 3.3.1. The goal of the learning algorithms studied in this domain is to optimize agents' behavior on problems found in decentralized systems. We will then present in Section 3.3.2 the *Embodied Evolutionary Robotics* (EE) methods. These methods are part of the ER field and aims at addressing the issues relative to the use of real robots. Finally, in Section 3.3.3, we present the *Artificial Life* domain. This domain corresponds to the study of the living systems' properties by relying on their reproduction on artificial substrate, and a special emphasis will be put on open-ended evolution.

3.3.1 Reinforcement Learning for Multi-Agent Systems

There are mainly two ways to deal with learning issues in decentralized systems. The first one, called *distributed problem solving*, use a central instance organizing the work of slave nodes. The second one named *Multi-Agent Systems* (MAS) is based on the interactions between multiple autonomous agents. The use of Temporal Difference methods (also called RL methods, and presented in Section 1.4.4) in MAS defines the domain called Multi-Agent Reinforcement Learning (MARL). MARL methods consider agents learning autonomously behaviors in order to solve a common task. Based on the reviews of [Busoniu *et al.*, 2008; Panait and Luke, 2005; Stone and Veloso, 2000], we present the coordination problems investigated, and one of the main issue arising from the use of MARL methods.

Two classes of problems studied in MARL and Game Theory are of special interest to our scenario:

- The coordination problems focus on the search of nash-equilibrium among multiple strategies [Claus and Boutilier, 1998; Lauer and Riedmiller, 2000].
- This field is also interested in the study of social dilemmas such as the Iterated Prisoners' game or the Tragedy of the Unmanaged Commons [Glance and Huberman, 1994; Lichbach, 1996].

The coordination of multiple robots have been investigated through the following problems:

- **Cooperative Multi-robot Observation of Multiple Moving Targets (CMOMMT)**: This problem is precisely defined in [Parker, 1997]. The aim is to optimize distributed control strategies for multiple autonomous robots collaborating in order to follow multiple moving targets.
- **Area coverage**: Different formulations of the same idea are proposed in the literature [Ahmadi and Stone, 2006; Morales, 2003]. The goal is to reduce as much as possible the time between two successive observations of any part of the environment by a team of robots. Certain flavors of the problem introduce different degrees of priority between different parts of the environment.
- **Robocup Keep-away**: The Keep-away task of the Robocup is defined in [Stone *et al.*, 2006]. A team of 3 robots currently in possession of a ball have to conserve it while 2 robots of an adversarial team try to intercept it. Various formulations of the same task have been proposed, varying notably in the number of robots involved in each team.
- **Object pushing**: Within this problem lies two specifics sub-problems. The first one is concerned with the shifting of one object too big to be moved by a single robot. The second one focus on the aggregation of multiple small objects by a team of robots [Asada *et al.*, 1999].

For MARL methods, the robotics domain constitute a special case where the world is continuous and the observations are partial and noisy. [Brooks and Mataric, 1993] have identified four main types of learning in robotic systems:

- **Learning numerical functions for calibration or parameter adjustment:** In this setup, the problem is to tune parameters in order to compensate the unknown sensor drift, environment changes, and the unmodeled properties of the mechanical system. These parameters are learned autonomously by the controller. Since the effect of each parameter isn't known beforehand, the learning is based solely on the performance of the controllers.
- **Learning about the world:** In this setup the goal is to organize informations from the world in which the agent is placed. This knowledge is then used to learn autonomously how to act in the world. Most of the contributions to the learning field in AI fits in this category.

- **Learning to coordinate behaviors:** In this setup the selection of behaviors is optimized with regards to the current environmental state. As a matter of fact, Reinforcement learning methods are successfully used to learn the mapping from sensory state (environmental conditions) to action state (behavior).

- **Learning new behaviors:** In this setup primitive actions are combined toward the achievement of a specific goal. The goal is to reduce the involvement of human engineers implied by the previous learning methods.

The main difficulty in applying Reinforcement Learning methods in robotic is to be found in the second type of learning (learning about the world). This is because all the Temporal Difference methods (TD methods, presented in Section 1.4.4) are designed to face a discrete world composed of a finite number of states, actions and rewards. As a consequence, a continuous world becomes too large to be tractable by such methods.

Moreover, this problem, already present when a single agent is under study, is getting worse when multiple agents try to synchronize in order to perform the optimal action. For example, in off-line resolution of problems under a finite time horizon, decentralized partially observable Markov decision process are NEXP-complete [Bernstein *et al.*, 2000], and heterogeneous agents' action selection problem is NP-complete [Parker, 2000]. Two approaches have been proposed to solve this type of problems. [Hansen *et al.*, 2004] construct incrementally polices by eliminating iteratively dominated strategies. [Szer *et al.*, 2005] rely on the evaluation of complete sets of strategies (one per agent). The latest methods can exploit additional informations such as the distribution of start state distribution, or the presence of common reward functions.

The problems considered in this thesis are to be solved on-line without a finite time horizon, and are therefore more complex. In order to reduce the complexity of the learning process multiple approaches have been proposed to either structure the problem or introduce heuristics. In the following we present the main approaches developed so far:

- **Policy search** One way to deal with the great number of states and actions is to work directly in the policy space. In this case a policy is created from a set of parameters, its value is computed after a complete trial. The best policy is obtained by following the gradient of the policies' value in the parameters space [Rueckstiess *et al.*, 2010; Sehnke *et al.*, 2008].

  One important contribution has been done in [Mataric, 1997; Mataric, 2001; Ahmadi and Stone, 2006], where policy search algorithms are used to optimize the parameters of hand-coded behaviors. This effectively reduce the search space from large continuous spaces to a space composed of few parameters, and is applied to numerous problems (see Table 3.1). However, this approach is restricted to the space of basic behaviors a priori hand-coded by the human engineer. Consequently, new solutions to unforeseen problems can't be learned with this method.

- **Relational representation:** In the work presented in [Morales, 2003], a new space of relational representation (both in the search space and the action space) is defined. Relational representations are encoding the transition between two states or two actions. Within this approach, reinforcement learning algorithms are used to optimize the mapping from relational state to relational action. While this solution effectively reduces the search space, states and actions have still to be defined by the human engineer.

- **Predictive model:** A predictive model is an estimator of a robot's current state which is based on historical sequences of inputs and outputs. In [Asada *et al.*, 1999] a predictive model is continuously updated thanks to the information given by multiple robots. The state vector produce is a discrete variable (build from observations of the continuous world) that can be used in reinforcement learning algorithms. This model has been tested in a scenario including two robots, which are learning sequentially. However this approach relies on either a central controller, or global communication, which is difficult to obtain with a high number of robots (typically a hundred in our scenarios).

- **Clusters:** Neighbor states in a continuous space can be considered as a unique state in a discrete space. Multiple methods have been used to create discrete states from regions of the continuous space. [Fernandez and Parker, 2001], uses a method called *Vector quantization* which requires a stopping criterion to avoid the raise of many small meaningless clusters. The design of this criterion requires a priori knowledge on the space to cluster.

  Another method, termed *tile coding* [Sutton, 1996], is successfully demonstrated in [Kostiadis and Hu, 2001; Stone *et al.*, 2001]. This method relies on layered grids, each encoding a binary feature of the environment. Therefore, it doesn't need a stopping criterion, but the human engineer knowledge of the problem are used to generate meaningful binary features.

- **Sampling:** A sampling mechanism will reduce the search space by performing the learning only on a restricted set of all the available states and actions initially present in the continuous space. This idea has been explored in the Pessimistic algorithm proposed in [Touzet, 2004]. This algorithm extends the idea of lazy sampling of the state and action spaces (originally proposed in [Sheppard and Salzberg, 1997]), to a team of multiple robots. While this algorithm doesn't rely on knowledges from a human engineer to build its finite action and state space, it does so during an initial sampling phase. Consequently, this algorithm isn't able to deal with modification of the environments without a complete reboot of the whole robotics team.

Each work listed above use its own specific version of Q-learning. In most cases, the modifications come from necessary reformulation due to the use of specific state and action spaces. These algorithms ares listed in Table 3.1.

Article	Space reduction method	Learning method	Experimental problem
[Mataric, 1996; Mataric, 1997]	Hand-coded behavior	Reinforcement Learning	Object pushing
[Ahmadi and Stone, 2006]	Hand-coded behavior	Reinforcement Learning	Area coverage
[Urieli *et al.*, 2011]	Hand-coded behavior	CMA-ES	Robocup
[Morales, 2003]	Hand-coded behavior	rQ-learning	Area coverage
[Asada *et al.*, 1999]	Predictive model	Modular reinforcement learning	Object pushing
[Fernandez and Parker, 2001]	Clustering	Vector Quantization Q-learning	CMOMMT
[Kostiadis and Hu, 2001]	Clustering	Residual Q-learning [Baird, 1995]	Robocup Keep-away
[Touzet, 2004]	Sampling	Lazy q-learning	CMOMMT

Table 3.1: Methods to reduce the search space in MARL

We consider the use of hand-coded behaviors as one of the most promising method to address the challenges of our scenario. This consideration is based at first on the autonomous adaptation capacities demonstrated, which takes into account the absence of central coordinator and the use of local communication. Secondly, the presented algorithms are able to handle changing environments, as shown in [Mataric, 1997] where the behavior learned by agents depends on a cyclic schedule (oscillating between night time and day time). Finally, this approach has been demonstrated numerous time on real robots [Mataric, 1997; Mataric, 1996; Ahmadi and Stone, 2006; Urieli *et al.*, 2011], and has been used in up to 8 robots in simulation [Ahmadi and Stone, 2006], which is the highest number of robots considered in the approaches reviewed.

These approaches are based on the assumption that the environmental conditions are well known to the human engineer. Based on this assumptions, the use of behaviors designed beforehand by a human engineer is a sound choice. However, we aim to address unknown environmental conditions. As a consequence, a new mechanism to produce autonomously original behaviors has to be used.

3.3.2 Embodied Evolutionary Robotics

Evolutionary Robotics is an optimization method based on Evolutionary Computation methods that target the design robots' controllers (presented in Section 2.1). In this context, the behavior of a robot is optimized by modifying the parameters of its controller. All controllers considered are operating directly in continuous state and action spaces. Contributions to this field have been reviewed in Chapter 2. The sub-field of Embodied Evolutionary Robotics (EE) is interested in the automatic design of controllers for large groups of robots. More precisely this field is interested in problems where groups of robots have to evolve and perform tasks in complete autonomy, with similar constraints as on-line on-board methods (see Section 2.4.2), but with group of robots rather than a single robot. On-line decentralized Evolutionary Algorithms are considered to address these challenges, as they will benefit from the parallel use of multiple robots, and avoid the "reality-gap" problem (presented in Section 2.3).

[Watson, 1999] proposed the Embodied Evolution (EE) approach where the Evolutionary Algorithm is run by multiple-robots (on-board). This goes beyond On-line On-board evolution by relying on multiple evaluations of controllers (performed in parallel on multiple robots) in order to get rid of the evaluation noise and therefore free the human engineer from any manipulations. Moreover, the use of multiple robots let the possibility to break a portion of them during the evolutionary process. This approach is useful in scenarios where the task domain can't be simulated, a centralized global coordinator can't be implemented, the robots must learn in the field, the task is interactive, and the reproductive behavior itself is under adaptation [Ficici *et al.*, 1999; Watson *et al.*, 2002]. These ideas are also found in a more restricted framework presented in [Simoes and Dimond, 1999], centered around two phases in the life of robots (working and mating), and relying on a global communication network. We will review the contributions to this field depending on *how* the evolutionary operators are used, that is to say in a distributed or encapsulated fashion [Eiben *et al.*, 2010b].

Encapsulated. Algorithms performing in an encapsulated manner make use of evolutionary operators in the robot themselves. In practice, this result in the constitution of genome's pool, evaluated one after another thanks to a time-sharing system, and ranked accordingly to their performances. Based on this ranking evolutionary operators are applied on the best genomes to produce the new genomes to evaluate at the next generation. This approach doesn't benefit from the parallel evaluations of one genome in multiple robots (and therefore multiple environmental conditions). The other robots are then mere moving obstacles increasing the complexity of the environment.

In the work presented in [König *et al.*, 2008] the focus is placed on the possibility to create a controller based on developmental methods. Therefore, the evolutionary mechanisms used aren't design specifically for the problems faced by Evolutionary Algorithms in multiple robots.

The second implementation known is focusing on the creation of evolutionary mechanisms to face

EE challenges [Haasdijk *et al.*, 2010]. In this contribution a $(\mu + 1)$ algorithm is encapsulated in one robot, and a reevaluation mechanism is added to face the inherent fitness evaluation noise. The use of this algorithm in multiple robot scenarios has been shown in [Haasdijk *et al.*, 2011].

[Huijsman *et al.*, 2011; García-Sánchez *et al.*, 2012] have shown the limits of such algorithm due to the absence of communications between robots. This limitation prevent the evolution of more efficient behaviors since controllers aren't evaluated in parallel on multiple environmental conditions. Moreover, each evolutionary process (one per robot) doesn't benefit from results obtained by other robots (potentially better). Finally, within this approach the evolution of coordination between robots is also hindered by the lack of communications between robots.

Distributed. In distributed EE approaches evolutionary operators rely on the interactions between robots. This approach has been initially envisioned in the original presentation of the EE domain, and the seminal works under the PGTA algorithm [Ficici *et al.*, 1999; Watson *et al.*, 2002]. This algorithm relies on an implicit fitness function (based on energy) to modulate the rate of transmission (proportional to the level of energy) and reception (inversely proportional to the level of energy). The selection mechanism result from the interaction between the robots, which depends on their relative fitnesses. A modified version integrating the use of a learning algorithm within PGTA has been proposed in [Wischmann *et al.*, 2007]. This approach has been also used to study the emergence of self-assembly and self-reproduction by relying on implicit fitness (based on speed) in [Bianco and Nolfi, 2004].

Within the distributed EE field, [Schut *et al.*, 2009b; Schut *et al.*, 2009a] proposed the Situated Evolution approach, which relies on Embryo Based Reproduction mechanism (EBR). EBR is characterized by the use of an embryo always improved by cross-over with encountered genomes. When a robot with a better genome is met, the embryo is replaced by this genome. A well studied example of Situated Evolution algorithm is the ASICO algorithm which make use of EBR and is similar to the contributions of Schut [Prieto *et al.*, 2009; Prieto *et al.*, 2010; Trueba *et al.*, 2011; Trueba *et al.*, 2012].

Finally, [Karafotias *et al.*, 2011] propose to use the distributed evolutionary principles while still relying on more than one genome by accessing a P2P network of communication. This approach has been successfully compared to the encapsulated version of $(\mu + 1)$ on phototaxis, fast forward and patrolling tasks.

Encapsulated and Distributed. On the one hand, these approaches use the same three mechanisms as the encapsulated approaches: the constitution of a genome's pool, the ranking of genomes thanks to a time-sharing evaluation system, and the constitution of new genomes based on the best genomes of the genome's pool. On the other hand, encapsulated and distributed approaches rely partly on communication facilities of robots. Within these approaches genomes of the genome's pool are ranked with relation to fitness value, and a portion of each robot's genome's pool is broadcasted to other robots.

Within this approach [Nehmzow, 2002] proposed an algorithm based on the encapsulation of a (1+1)ES algorithm [Schwefel, 1981] in each robot. [Usui and Arita, 2003] proposed a more complex scheme, where two lists of genomes are saved: one containing the genomes to evaluate, the other containing a list of the best genomes to be send to other robots (the fittest a genome, the highest chance to be send to other robots). From the list of best genomes and the received genomes, evolutionary operators create new genomes which will be later evaluated.

[Huijsman *et al.*, 2011] proposed an algorithm based on the distributed Evolutionary Algorithm `EvAg` [Laredo *et al.*, 2010], and the encapsulated on-line on-bard version of the $(\mu{+}1)$ algorithm [Haasdijk *et al.*, 2010]. This algorithm has since then been applied to the study of multi-cellular robots [Weel *et al.*, 2012], and the emergence of communication in cooperative tasks [Pineda *et al.*, 2012].

[García-Sánchez *et al.*, 2012] have drawn a comparison between three algorithms of this field : $(\mu + 1)$, $(\mu + 1)$ hybrid with `EvAg`, $(\mu + 1)$ hybrid with the `MultiKulti` transmission scheme [Araujo *et al.*, 2008; Araujo and Merelo, 2011]. This work has highlighted the advantages brought by the transmission of genomes between the robots.

Finally, [Elfwing *et al.*, 2011] study the use of learning algorithm in combination with encapsulated Evolutionary Algorithms. Here the learning algorithms $sarsa(\lambda)$ [Theodoridis and Hu, 2006] with potential based shaping rewards [Ng *et al.*, 1999] is used to learn a set of behavior. An encapsulated genetic algorithm similar to [Usui and Arita, 2003] has been designed to evolve: a Neural Network in charge of selecting the behaviors, the learning parameters, and the shaping reward learning parameter. This work relies partly on an implicit function (a genome is removed from the genotypic pool if the virtual level of energy dropped under 0), partly on the environment pressure (new genomes received are randomly selected for evaluation).

Discussion. The EE approaches reviewed in this section are designed to face the challenges meet in real hardware conditions.

- The optimization in a continuous domain is done thanks to evolutionary operators applied to robotic controllers.
- The Evolutionary Algorithms are running on the robots in order to avoid the reality gap problem.
- The controllers' parameters can be exchanged between robots in order to evaluate controllers in parallel on multiple environmental conditions.

EE methods are objective driven evolutionary algorithms. This objective is formalized by the human engineer as a fitness function. As presented in Section 2.2, a large variety of fitness function can be used to describe with various level of precisions the goal targeted. As such, implicit fitness functions are designed by the human engineer but gives few information about the best behavior.

However, we are interested here in problems where the goal thought can't be defined at the individual scale by the human engineer. More specifically the maximization of the integrity depends highly of the environment at hand, and the interaction between robots. Therefore, the selection pressure has to be defined at the population scale, and driven by environmental constraints.

3.3.3 Open-ended Evolution in Artificial Life

Artificial Life (ALife) is a field of study which examine systems related to life, its processes, and its evolution. The questions targeted by the ALife community are numerous and can be summarized in three main categories [Bedau *et al.*, 2000; Bedau, 2003]: How does life arise from the nonliving? What are the potentials and limits of living systems? How is life related to mind, machines, and culture? From a theoretical ground multiple contributions have been made to highlight the minimal properties needed to classify a system (biological or artificial) as alive. The Autopoietic systems [Varela *et al.*, 1974; McMullin, 2004], Chemical Automaton systems [Gánti, 1975; Ganti *et al.*, 2003; Munteanu and Solé, 2006] (Chemoton) are two such formalizations initially published practically simultaneously in the 70's. Some reviews analyze the two theories in a common framework [Bersini, 2010; Bich, 2010].

Apart from the main theories on the origin of life and its definition, experimentations have been done to answer questions on the behavior of living systems, and the conditions necessary to obtain these behaviors in artificial substrate. These experimentations are classified in three fields : the wet, the soft and the hard [Bedau, 2003]. The wet ALife is mainly concerned with the study of the structures, functions and interactions of molecular components at the heart of life [Couzin, 2002; Blight *et al.*, 2000]. The hard field aims at the reproduction of selected key-properties of living system in hardware systems such as robots. This field is discussed at length in this manuscript (an overview can be found in Chapters 1, and 2). The soft community aims at studying the key properties of life by re-creating it in digital simulations. The silico substrate of computers is chosen for its easy manipulation, ease of experimentation and low risk. Multiple behaviors are target for reproduction in artificial substrat, such as: self-organization and self-replication phenomenons, evolution of complexity (with the practical case of the origin of multi-cellularity), and the evolution of complex behaviors such as language or cooperation.

More precisely the soft ALife aims at studying the key elements for the evolution of these specific phenomenons observed in the biological realm. To study such issues, we can highlight two methods. Firstly, analytical systems are based on the modelization of a population as a set of equation. The resolution of these equations allow the prediction of outcomes (such as the presence of different strategies) with relation to experimental settings. They have been used for example to study the efficiency of different strategies [Axelrod and Hamilton, 1981] the impact of noise level on these strategies [Nowak and Sigmund, 1993; McNamara *et al.*, 2004]. However these methods aren't well suited to study the impact of the spatial environmental conditions such as terrain topography, and the locality of food resources.

Secondly, agent-based systems are based on a population of heterogeneous agents simulated step by step and interacting with each other. These methods are designed to study the emergence of high level phenomenon resulting from these interactions in different environments [Macal and North, 2005]. These systems can be combined with Evolutionary Algorithms, i.e. the strategies used are modified by random mutations, and a selection mechanism favor the most successful strategies. These systems are used to study the key mechanisms and conditions necessary for the evolution of complex high-level behaviors (such as cooperation and communication). Notably, one goal targeted by these systems is the realization of open-ended evolutionary dynamics [Maley, 1999; Bedau *et al.*, 1998]. Among the contributions to this field, we present two main approaches:

- **Modelization of biological dynamics:** Contributions to this approach focus on the study of evolutionary dynamics through different simplification's level of biological systems. The most known simulator addressing open-ended issues is Tierra [Ray, 1992], which has shown the evolution of parasites, hyper-parasites, and sociality. Other works not focusing on open-ended issues have addressed for example the evolution of shapes [Yaeger, 1994], ecological systems [Gras *et al.*, 2009; Golestani *et al.*, 2012], and the interplay between three adaptation mechanisms (evolutionary, individual, and social) [Gilbert *et al.*, 2006; Eiben *et al.*, 2007].

- **Modelization of biological systems:** This approach is focused on the modelization of real biological systems, and notably the comparison between data measured in the real world and obtained by simulation. Within this approach the open-ended evolution properties have been studied based for example on historical data [Epstein *et al.*, 1996]. Other works are used to study the properties of bacteria and viruses. [Adami *et al.*, 1994; Knibbe *et al.*, 2007; Misevic *et al.*, 2012].

Studies in ALife have shown mechanisms to reach open-ended evolution properties. Part of these contributions are focused on the interplay between survival of agents and population dynamics. These systems are therefore a source of inspiration to address problems relative to the integrity of a group of robots, though they have not been designed, and used, for this particular purpose.

3.4 Issues

In Section 3.3.1 we have seen that MARL methods are able to learn near optimal solution, while solving the trade-off between exploration and exploitation. However we have also highlighted that MARL methods are ill-adapted to face the on-line search of large continuous search spaces used in robotics. Within this field, all the methods known to reduce the search space to tractable size relies on a human engineer with a priori knowledges of the problem at hand. Therefore, the resolution of the "robotic constraints" challenge reduces the ability to solve the "autonomous exploration" challenge.

The Evolutionary Robotics methods (Chapter 2) propose to optimize the parameters of different type of controllers which can express a large number of behaviors (different type of controllers are reviewed in Section 2.1.1). While this allow the exploration of new solution unforeseen by a human engineer, in most cases this exploration isn't autonomous. Indeed Evolutionary Robotics methods are typically used in a "*design, then use*" fashion which rely on a centralized process.

The approaches reviewed under on-line on-board evolution and EE brings us close to a solution to the scenario presented in Section 3.1. They are all concerned with embodied evaluations which makes sure that the solutions found correspond to real world problems as perceived by the robot. On-line on-board approaches propose methods to overcome the problems of noisy fitness evaluation function and restricted capacities of real robots when only one robot is used. While addressing similar problems in multi-robots setup, Embodied trials methods are also challenging self-organization issues. However, these methods are still relying on the knowledge of a human engineer to measure the survival skills of a controller, that is: to define a fitness function.

The soft ALife domain, by studying the way life appears and is sustained, is looking for reliable mechanisms to ensure the survival of a system (biological or artificial). The methods of the ALife domain propose to use evolutionary operators relying on the reproductive and survival capacity of the agents. If an agent is able to survive longer and transmit more often its genome, the next generation of agents will most likely inherit from it. By this mechanism the population is always ensuring its integrity by selecting the most adapted agents. The adaptation method is independent from any a priori knowledge of the human engineer, and can therefore face the "autonomous exploration" challenge. It is important to note here that these methods address primarily modelization problems and not design problems. These methods have been used in virtual environments where the addition ex-nihilo of new agents, or the deletion of ill-adapted agents, is common place. The production of new robots require a great amount of human and can't be done autonomously by other robots. Likewise, the destruction of a robot is an irreversible action and will be avoided as much as possible. As a consequence, we need a way to abstract the addition and removal of agents when considering a robotic substrat. In practice, this will result in the emulation of selection and mutation mechanisms in real robots while maintaining the number of functioning robots as high as possible.

3.5 Proposal: Environment-driven Distributed Evolutionary Adaptation

In order to solve all the challenges mentioned earlier, we propose a new method called Environment-driven Distributed Evolutionary Adaptation (EDEA). The Table 3.2 proposes a classification of the works reviewed in this thesis and highlight the positionning of the EDEA problem. EDEA methods are environment-driven algorithms aiming at ensuring the integrity of a group of robots based on the survival and reproduction abilities of each robot. That is to say, the more efficient is a strategy

Objective-driven (user-defined metric to compare individuals)		**Environment-driven** (the environment does the selection)	
explicit fitness	implicit fitness	biased natural selection	unbiased natural selection
ER/EE [Watson 02 ; Nehmoz 02 ; Usui 03 ; Haasdijk 10]	**ER/EE,ALife** [Araujo 11 ; Weel 12]	**ALife** [Adami 94 ; Parsons 10]	**Alife, EDEA** E.g.: TIERRA's natural selection [Ray 92]

Table 3.2: Classification of Embodied Evolution (EE), Open-ended Evolution in Artificial Life (ALife) and Environment-driven Distributed Evolutionary Adaptation (EDEA): a perspective from the nature of the selection pressure

to ensure the survival of one robot, and transmit itself to other robots, the more it should be used by other robots. By this mean variations of strategies efficiently maintaining integrity are used and improved at all time. This approach is directly inspired from the mechanism of natural selection as modeled in ALife systems.

As in the EE method, robots aren't added and removed, but controllers' parameters are exchanged between robots. The main different with EE algorithms, is that the success of a genome isn't computed thanks to a function relying on the agent's activity (e.g. sensor values, and actuator values). In EDEA methods (alike some ALife systems), the success of a genome is measured only based on its capacity to survive and spread in the current environment.

It follows that the key to EDEA is the environment-driven nature of the adaptation process. This may be seen as the result of two possibly conflicting motivations:

- **Extrinsic motivation:** An agent must cope with environmental constraints in order to maximize survival, which results solely from the interaction between the agent and its environment (possibly including other agents).

- **Intrinsic motivation:** A set of parameters (i.e. "genomes") must spread across the population to survive, in accordance with the algorithmic nature of the evolutionary process. Therefore, genomes are naturally biased toward producing efficient mating behaviors as the larger the number of agents met, the greater the opportunity to survive. By this mean the whole evolutionary process is continuously driven by the environment toward efficient and sustainable strategies.

The level of correlation between these two motivations impacts problem complexity to a significant degree: a high correlation implies that the two motivations may be treated as one while a low correlation implies conflicting objectives. An efficient EDEA algorithm must address this trade-off between extrinsic and intrinsic motivations as the optimal genome should reach the point of equilibrium where genome spread is maximum (e.g. looking for mating opportunities) with regards to survival efficiency (e.g. ensuring energetic autonomy).

Figure 3.1 proposes a simplified illustration of these principles by a fictive simulation of one generation. The first image shows the beginning of a simulation: each robot and start with an empty genome list. The second and third images show the robots moving in their environment (each robot's behavior is controlled by its own active genome), and exchanging genomes when they are close enough. The forth image shows the situation at the end of the generation: the red genome has spread more and thus has a higher probability of begin selected. In this case the red genome has a probability p = 1 to be selected by two robots, and the two other genomes only get p = 0.5 in one robot. The next generation will contain slightly mutated copies of the original genomes. This example highlights the environment-driven nature of the evolutionary process. Indeed, the selection probability of each genomes depends only on the behavior resulting from its expression. The behavior produced depends of the environment in which the genome is found. This point of view is highly different from works in objective-driven ER, where the selection probability depends of the evaluation of the behavior by a fitness function.

The overall motivation behind the work presented in the remaining of this thesis is the study of general evolutionary adaptation algorithms that can be implemented on real robotic hardware. The next chapter present a novel distributed evolutionary adaptation algorithm for use in a population of autonomous agents. The remaining chapters are dedicated to the study of the evolutionary dynamics of this algorithm in different environments (real or simulated).

3.6 Summary and Conclusions

In this chapter we detailed the challenges raised by a scenario where a group of autonomous robots have to maintain their integrity in an unknown and possibly changing environment. After having reviewed relevant work in related domains we have further defined the issues faced in this scenario. Three important issues have been identified: a) The lack of information on how to survive in an environment, b) The lack of measure for the survival skills of a controller, c) A population composed of a limited number of robots (upper bound). To solve these issues we have shown the importance of relying on the environmental pressure to drive the learning process. Such environmental pressure is the sole information available to assess the quality of one genome with regards to integrity issues. These aspects have been formalized in a new class of problem, the Environment-driven Distributed Evolutionary Adaptation (EDEA). Finally, we have highlighted in an illustrative example the mechanisms needed to the realization of EDEA algorithms.

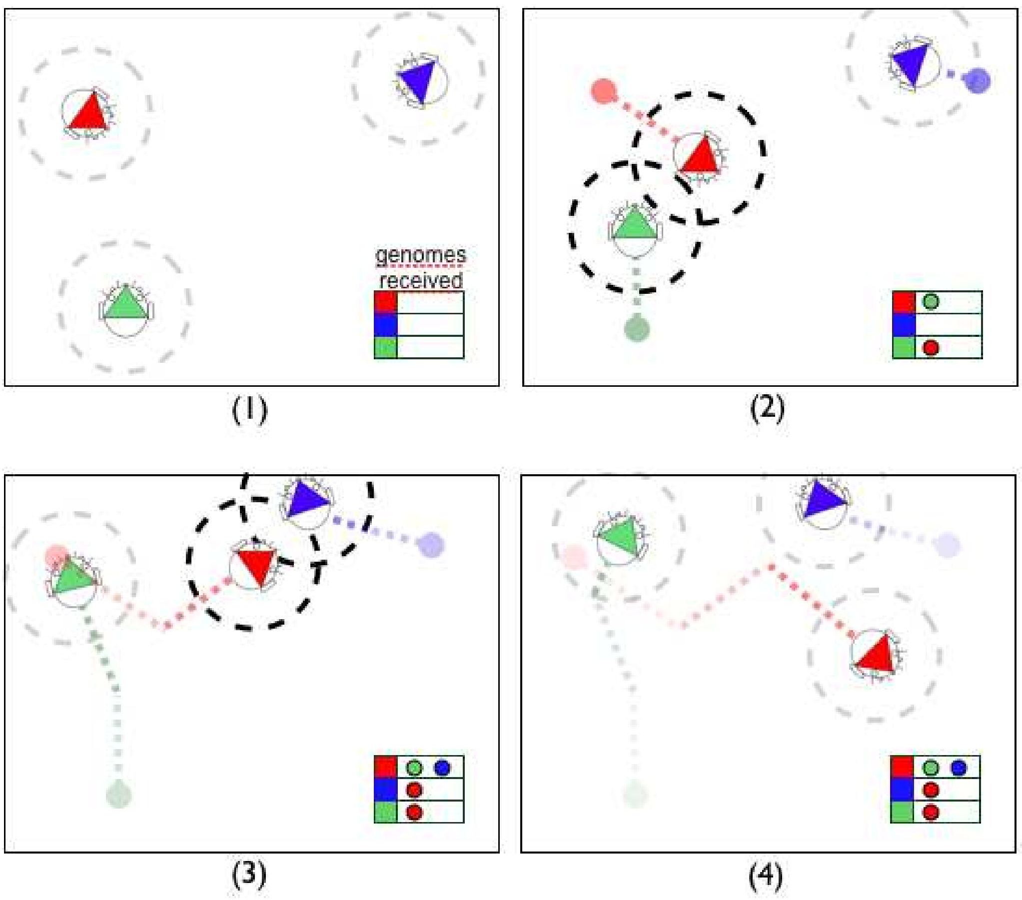


Figure 3.1: Illustration of EDEA principles.

Chapter 4

mEDEA: A Minimal Algorithm for EDEA

In the following chapters we address the problem of self-adaptation of a robotic swarm to unknown environments. To this aim we propose a minimal Environment-driven Distributed Evolutionary Adaptation algorithm, termed mEDEA. Our goal is to study the in-depth evolutionary dynamics of this algorithm under various environmental conditions.

The thinking behind the mEDEA algorithm is to consider strategies as atomic elements and the population of agents as a distributed substrate on which strategies compete with one another. This approach is better illustrated using the Selfish Gene metaphor [Dawkins, 1976]: one specific strategy (or set of parameters, or genome) is "successful" if it manages to spread across the population, which implicitly requires to both minimize the risk for its "vehicle" (i.e. extrinsic motivation) and to maximize the number of mating opportunities (i.e. intrinsic motivation), although the two may be contradictory. The mEDEA algorithm is original since it is the first algorithm (to the best of our knowledge) to address the trade-off between extrinsic and intrinsic motivations in collective robotic systems within open environments.

The detailed description of the mEDEA algorithm is given in Section 4.1. The evolutionary dynamics of this algorithm are studied, with a particular emphasis on the evolution of behavioral consensus in Section 4.2. Lastly, the algorithm is implemented and tested with real robots, thus illustrating the transferability to real world problems (Section 4.3). The work presented in this section is based on our publication [Bredeche *et al.*, 2012].

4.1 Algorithm

The mEDEA algorithm ("minimal EDEA"), described in Algorithm 2, is distributed among all agents. A given agent is driven by a controller, whose behavior depends of a set of parameter (e.g. Artificial Neural Network). A set of controller's parameters is called a genome. At any moment, the parameters of an agent's controller are extracted from a specific genome, termed "active" genome. Each agent is endowed with a communication device which is solely used to broadcast information within a predefined communication radius. In this work the communication abilities of each agent is used to broadcast its active genome. Finally each agent has a memory unit, which is used to store the genomes received thanks to the communication device. This is very similar to the illustration shown in the previous chapter (see Section 3.5).

Algorithm 2 The mEDEA algorithm

```
genome.randomInitialize()
while forever do
  if genome.notEmpty() then
    agent.load(genome)
  end if
  for iteration = 0 to lifetime do
    if genome.notEmpty() then
      agent.move()
      broadcast(genome)
    end if
  end for
  genome.empty()
  if genomeList.size > 0 then
    genome = applyVariation(select_random(genomeList))
  end if
  genomeList.empty()
end while
```

This algorithm implements several simple but crucial features, that can be interpreted from the viewpoint of a traditional Evolutionary Algorithm structure:

- **Selection operator**: The selection operator is a random sampling selection among the list of imported genomes. It is important to note that no fitness value can even be computed, which makes random selection the only option here. Consequently, there is no selection pressure *on a local agent basis*. In fact, the selection pressure is active at the global level (population) rather than at the local level (agent): the most widespread genomes *on a global population basis* are more likely to be randomly sampled *on average*.

- **Variation operator**: The variation operator is chosen to be rather conservative to ensure continuity during evolution. Generating altered copies of a genome only makes sense if there is some continuity in the genome lineage: with no variation the algorithm will simply converge on average toward the best genome initially existing in the population. In the following, we assume a Gaussian random mutation operator, inspired by Evolutionary Strategies [Beyer and Schwefel, 2002], with which the locality of mutation can be easily tuned through a σ parameter.

- **Replacement operator**: Lastly, the replacement of the current active genome to control a given agent is achieved by (1) local deletion of the active genome at the end of one generation (i.e. a limited pre-defined amount of time) and (2) the random selection of a new active genome among the imported genome list (cf. selection operator). On a population level, this implies that surviving genomes are likely to be correlated with efficient mating strategies, as a given genome will only survive in the long run through more or less slightly different copies of itself.

Impacts of environmental fluctuations on genomes' performances is smoothed by the definition of the variation operator, as newly created genomes are always more or less closely related to their parent. As a consequence, genotypic traits result from a large number of parallel evaluations, on two scales:

- **The spatial scale:** Closely related copies sharing the same ancestor may be evaluated in different agents of the swarm. It is very unlikely that two agents will experience the same evaluation conditions. Therefore copies of the same genome will be evaluated under multiple environmental conditions.

- **The temporal scale:** Each genomes is strongly related to its ancestors. Therefore, the same lineage of genome, carrying the same main behavioral traits, will be evaluated in multiple conditions through time.

A single poorly efficient genome may be picked by chance as the "active" genome of an agent once in a while. Therefore the following question arise: does a family of closely related genomes manage to survive in the population, despite the presence of more efficient competitors ? We hypothesize that larger swarm will result in lower sampling bias. In other words, better than average genomes will get a better chance of surviving if the population is large. This hypothesis is investigated in the experiments presented in this chapter.

A fundamental requirement for ensuring selection pressure is that there is a strict constraint on the maximum number of genomes N_g to choose from during selection with regards to the population size N_p. $N_g < N_p - 1$ must hold for at least one agent (i.e. the received genome list should contain *strictly less* than all genomes from the population minus the local genome). Otherwise, no selection pressure will apply as genome survival probabilities would be uniform over the swarm. Then again, this worst-case scenario does not necessarily imply failure of the algorithm: the lack of selection pressure may imply a random walk in the genotypic space thanks to the variation operator (i.e. *"random genetic drift"* [Futuyma, 2009]), possibly leading to a new – possibly more interesting – region of the genotypic space, despite a temporary loss of selection pressure.

4.2 Preliminary Experiments: Evolution of Behavioral Consensus

In this section we study how the agents behave as a population, that is to say, we study the population-wide dynamics resulting from agents local interactions. While the local interactions between the agents are well known, the resulting global dynamics might be difficult to predict. This difficulty arises from the fact that new strategies appear continuously (represented by genomes) during a simulation, and that their survival in the population depends of other strategies present at that moment. Moreover, the survival of a genome is achieved by a trade-off between intrinsic and extrinsic motivations. Thus, the global outcome of all the local interactions can be unpredictable and dynamic.

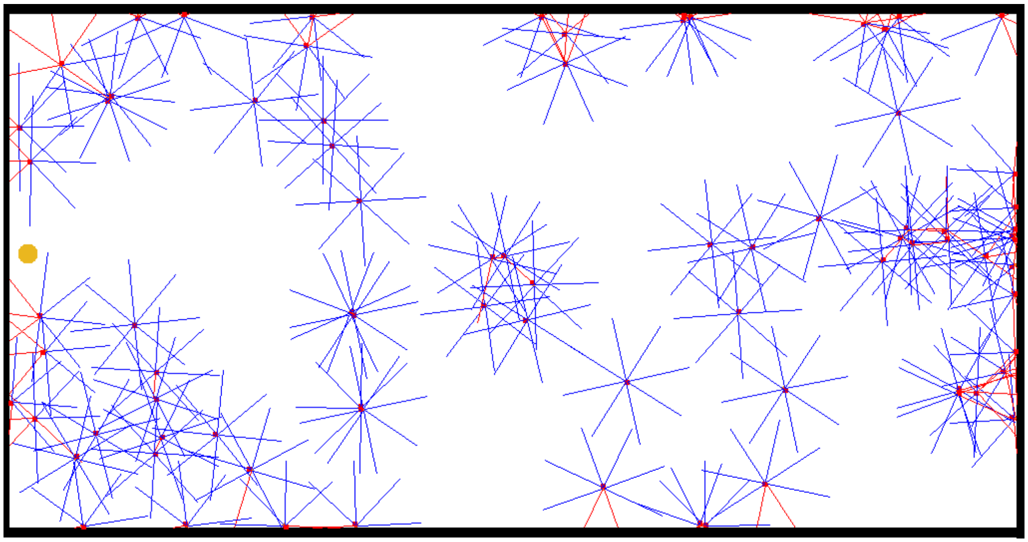


Figure 4.1: The "two-suns" setup: a swarm of agents is dropped into an empty environment where a "sun" is present (the circle on the left of the image). Each agent gets information about its distance and orientation with respect to the sun, but the sun itself gives no direct advantage whatsoever (e.g. no energy). Every 50 generations, the current sun disappears and another sun appears at the opposite end of the arena.

In this section, we are interested in the following questions: does the whole population converge to one specific "canonical" behavior ? If not, how different are the behaviors ? Are there transient behaviors ? Is it possible for two sub-populations with different behaviors to co-exist (i.e. multiple stable attractors) ?

Given a swarm of agents, each running mEDEA, how are efficient behaviors acquired and how are they distributed across the population. In practice, we study the emergence of consensus, where a population of agents may choose to exploit, or not, singularities in the environment (e.g. a particular element arbitrarily located in the environment).

4.2.1 Problem and Experimental Setting: the "Two-suns" Setup

Figure 4.1 provides a snapshot of 80 agents within the arena, where a particular landmark is present (located on the East side of the arena). We refer to this landmark as "the sun". A population of autonomous agents is placed in an empty arena, and there is no environmental constraint except the pressure to wander around mating with one another. The sun is an intangible object that provides no advantage whatsoever - i.e. there is no energy in this setup. Two sensory inputs give the orientation (relative angle) and distance to the sun to the agent's controller. Once in a while (every 50 generations), the sun changes location from East to West (and back).

The motivation for this setup is to provide an environment where several possible behaviors are likely to emerge and might be observed. Indeed the sun may here be considered either as a distraction and ignored, or as an environmental feature from which to take advantage. Our hypothesis is that in the latter case we shall be able to observe behaviors at the level of the swarm, with possibly one, or several, emergent strategies. In fact we might expect that the sun will be considered as a compass and thus used as a mating location by some agents, even though there is no way for the agents to *explicitly* reach a consensus. Of course offspring of a particular genome are likely to conserve a given mating strategy, resulting in a more or less widespread behavioral consensus.

In order to evaluate the emergence of consensus two experimental parameters have been taken into account: population size and range of radio communication. Population sizes vary from 10 to 80 agents and radio communication takes three possible values: $r = 64$, $r = 32$ and $r = 16$, resulting in a total of 213 experimental settings. All experiments described in this section result from 24 independent runs for each parameter setting (i.e. a total of $(213 * 24 =)5112$ runs), performed on a cluster of PCs with two AMD Opteron dual-core 1.8GHz processors running Ubuntu Linux. Each experiment lasts for 150 generations and the sun switches location every 50 generations (either East or West of the arena). Detailed parameters used for experiments presented here are given in Table 4.1.

On a practical viewpoint, one experiment takes approx. 15 minutes to be performed using one core of a AMD Opteron dual-core 1.8GHz processor. The home-brew agent simulator roborobo is programmed in C++ and features basic robotic-inspired agent dynamics with friction-based collision (available at: https://code.google.com/p/roborobo/). This simulator is used for all the simulations presented in this thesis. The source code and parameters for all experiments presented here is available on-line in the Evolutionary Robotics Database[1].

4.2.2 Representation

Specifications for the autonomous agents are inspired by a traditional robot configuration, with 8 proximity sensors arranged around the agent body and 2 motor outputs (translational and rotational speeds). Two additional setup dependent inputs are considered (utility explained in section 4.2.1). Each agent is controlled by a multi layer perceptron (MLP) with 5 hidden neurons, which means a total of 67 weights[2]. Note that the mEDEA algorithm is independent of any particular control architecture implementation, even though Artificial Neural Networks provide a convenient, flexible and well-established representation formalism. Thus, in experiments, we will focus on the dynamics of the mEDEA algorithm rather than providing in-depth analysis of the particular internal properties of the evolved neural networks.

The variation operator is a Gaussian mutation with one σ parameter: a small (resp. large) σ tends to produce similar (resp. different) offspring. This is a well known scheme from Evolution

[1] Evolutionary Robotics Database: http://www.isir.fr/evorob_db/

[2] 10 input neurons ; 5 hidden neurons ; 2 output neurons ; 1 bias neuron. The bias neuron value is fixed to 1.0 and projects onto all hidden and output neurons.

Parameter	Value
arena width and length	$1024 * 530$ pixels
sun location switch	every 50 generations
lifetime (i.e. generation duration)	400 steps per generation
population size	10 to 80 agents
proximity sensor range	64 pixels
radio broadcast signal	16, 32, and 64 pixels
agent rotational velocity	0.52 rad/s
agent translational velocity	2 pixels/step
genome length	68 real values (67 MLP weights + σ)
variation operator	Gaussian mutation with σ parameter
$\sigma_{minValue}$	0.01
$\sigma_{maxValue}$	0.5
$\sigma_{initialValue}$	0.1
α (ie. σ update parameter)	0.35

Table 4.1: Parameters for experiments.

Strategies where continuous values are solely mutated using parameterized Gaussian mutation, where the σ parameter may be either fixed, updated according to pre-defined heuristics or evolved as part of the genome. In the scope of this work we rely on self-adaptive mutation where σ is part of the genome [Beyer and Schwefel, 2002] (i.e. the full genome contains 68 real values).

As with other genome parameters, the σ value responds to environmental pressure: a genome survives in the population if it leads to efficient agent behavior *and* if it is able to produce comparable or fitter offspring. In some cases this requires a fine tuning of existing genome parameters (i.e. local search), while in other cases it requires very different genomes (i.e. escaping local optima).

The current implementation of the σ update rule is achieved by introducing α, a σ update value, which is used to either decrease ($\sigma_{new} = \sigma_{old} * (1-\alpha)$) or increase ($\sigma_{new} = \sigma_{old} * (1+\alpha)$) the value of σ whenever a genome is transmitted. In the following, α is a predefined value set prior to the experiment so that it is possible to switch from the largest (resp. smallest) σ value to the smallest (resp. largest) in a minimum of approx. 9 (resp. 13) iterations (i.e. there is a bias toward local search).

4.2.3 Results and Analysis

As a general overview, Figure 4.2 tracks the impact of population size and radio communication settings on evolutionary adaptation. As expected, smaller populations with a short radio communication distance are more prone to extinction. On the other hand, large populations produce a successful outcome which is independent of the communication range. Figure 4.3 provides much more detail at the

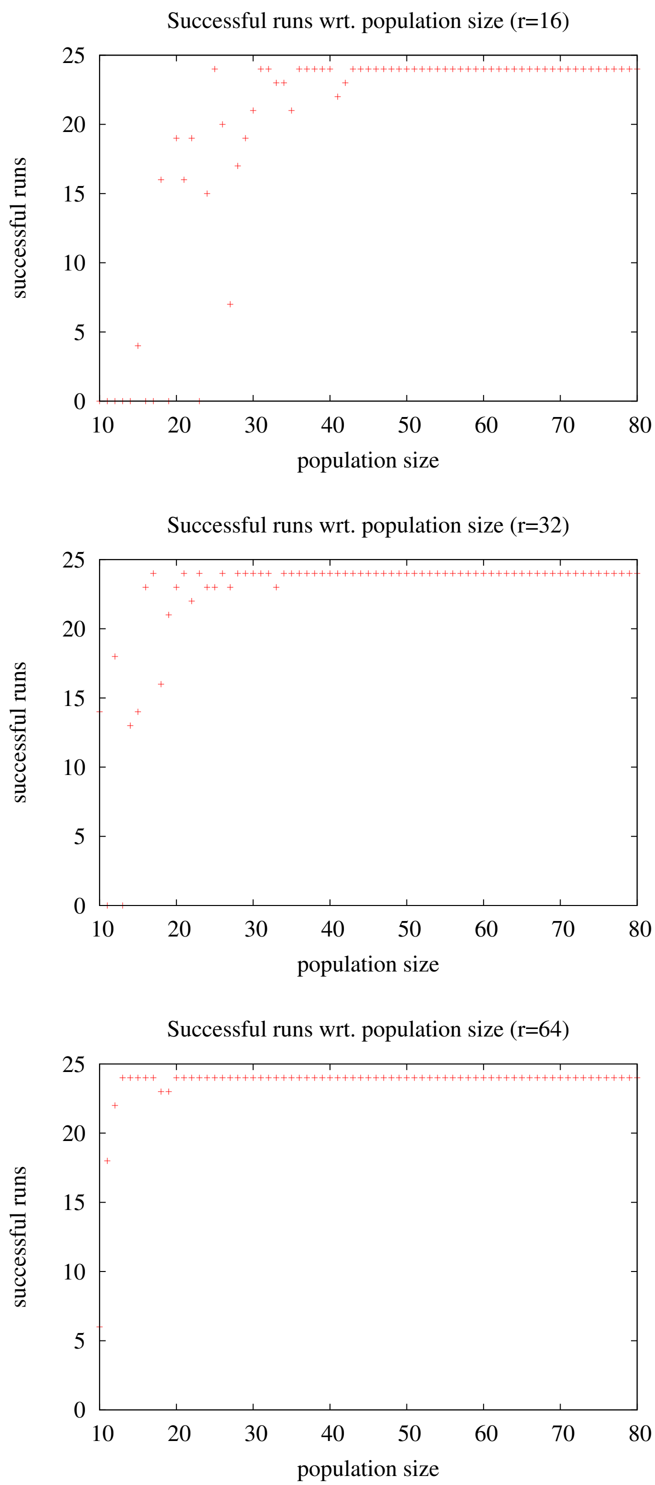


Figure 4.2: Number of successful runs (out of 24, for each population size) for different values of radio communication length ($r = 16$, $r = 32$ or $r = 64$). A run is successful whenever at least one agent is still alive at the end of the last generation. In practice however, successful runs always feature healthy population (ie. most agents are alive). These graphs illustrate an abrupt phase transition in the parameter space (considering population size and radio communication length) before which agents do not survive (ie. mating becomes too difficult).

toward sun (see Figure 4.4)	**away from sun** (see Figure 4.5)	**double consensus** (see Figure 4.6)	**misc** (see Figure 4.7)	**extinction** (see Figure 4.8)
59%	8%	3%	10%	20%

Table 4.2: Various types of consensus.

behavioral level: each graph accounts for the final (i.e. the last step of the last generation) positions of all agents from the 24 independent runs for each population size. Both agents' final orientation and distance to the center of the arena are plotted. Distance and orientation are given with regards to an imaginary vector at the center of the arena (position (0,0)), facing North (i.e. upwards). Distance is normalized between 0 (center of the arena) and 100 (farthest possible location). As an example, (orientation= *West*, distance=80) corresponds to an agent standing near, but not close to, the right border of the arena.

Graphs from Figure 4.3 illustrate that in most experiments, the vast majority of agents stand very close to the sun (west, far away from center), and that the (few) remaining agents are located in the region of the arena opposite the sun with a preference for corners (either North-East-East or South-East-East, far away from the center). From these graphs, we can see that at least two kinds of consensus have emerged, which we will call: “toward sun” and “away from sun”. However, questions remain open as to the existence of other kinds of consensus and/or the possibly simultaneous occurrence of several types of consensus. These questions were addressed by taking a closer look at the experiments from the qualitative viewpoint (i.e. hand analysis of the logs) to identify the kind of consensus reached. Table 4.2 summarizes the results and points to Figures 4.4, 4.5, 4.6, 4.7, and 4.8 for illustration purposes. Results shown in the table were aggregated by randomly sampling one run of each parameter setting and qualitatively evaluating the outcome of the simulation by a human expert.

The vast majority of all runs feature *at least* one consensus among the agents (this was expected, as mentioned in Section 4.2.1). There were remarkably few runs where no clear consensus could be identified: even the “misc” class included either transient behaviors shifting between different consensus or different kind of stable consensus, such as following walls and ignoring the sun. Also, looking at larger population sizes (i.e. ≥ 50 agents) results in observing 85% of the runs ending up with a unique consensus “toward the sun” for all agents. This tends to suggest that increasing population size introduces more stability in the evolutionary dynamics. We also conclude that a general agreement to go toward the sun is an efficient consensus in this scenario. This can be easily explained, as adopting this strategy is the best way for agents to meet at a precise and robust location in the environment, thus maximizing the genome's chances of survival.

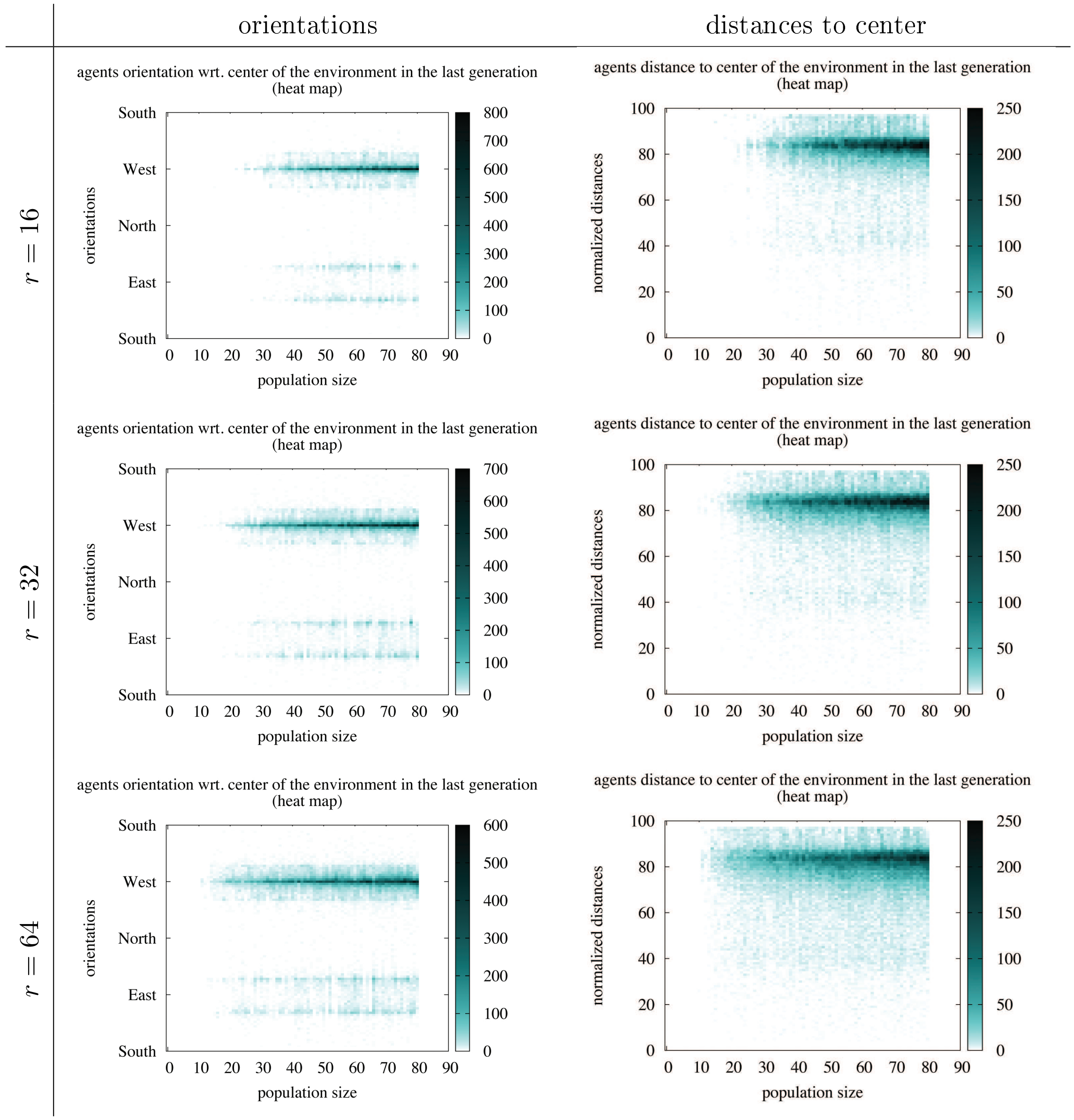


Figure 4.3: Tracking agents' orientation and distance to the centre of the environment for different population sizes and experimental setups, in the final iteration of the experiment (i.e. snapshot of the last moment of the experiment). Three different communication radii are considered. Graphs in the left column show the orientations of agents with regards to the centre of the environment. Graphs in the right column show agents' distances to the centre of the environment. The three pairs of graphs correspond respectively to experiments where the communication radius is set to 16 (first row), 32 (second row) and 64 (third row). For each graph, darker regions in the graphs indicate the most commonly observed orientation/distance w.r.t. the centre at the end of experiment (y-axis) for each population size parameter (x-axis) with a given communication radius value (table rows) – this representation scheme is commonly known as a heat map. Each row is the result of aggregation of 1704 independent runs.

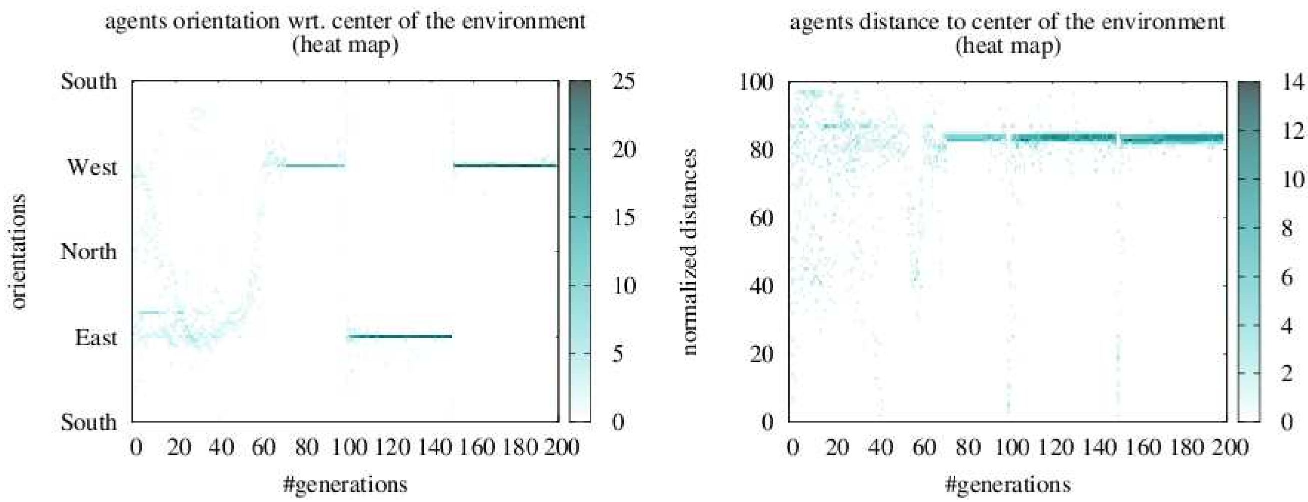


Figure 4.4: Example of consensus toward the sun (extracted from one run with $pop_{size} = 29$, $r = 32$). Distances vary between 0 (agent in the center of the environment) and 100 (agent in a corner).

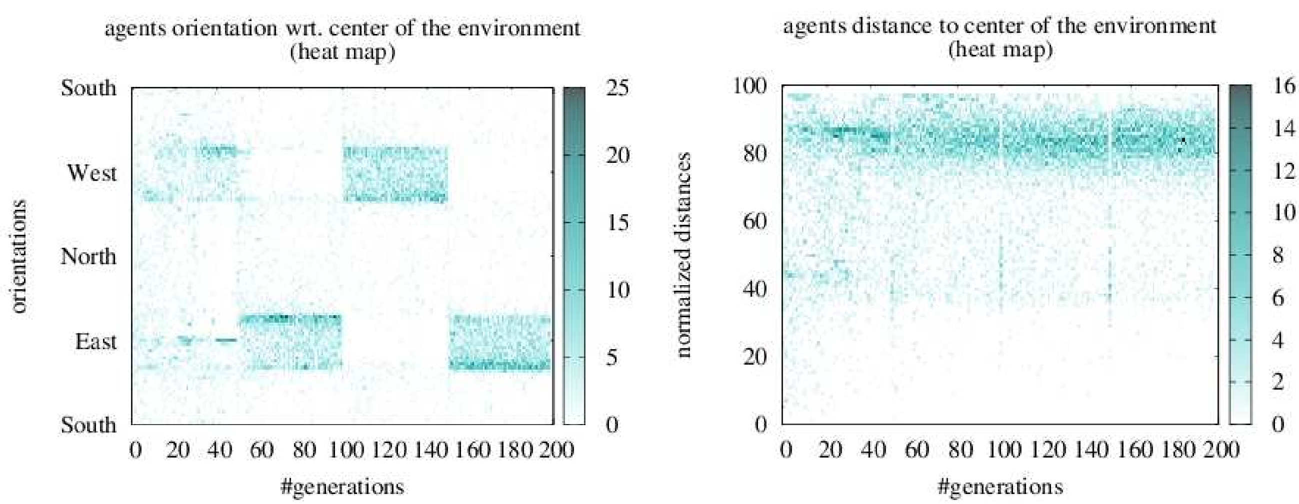


Figure 4.5: Example of consensus toward the side *opposite* to the sun (extracted from one run with $pop_{size} = 80$, $r = 32$)

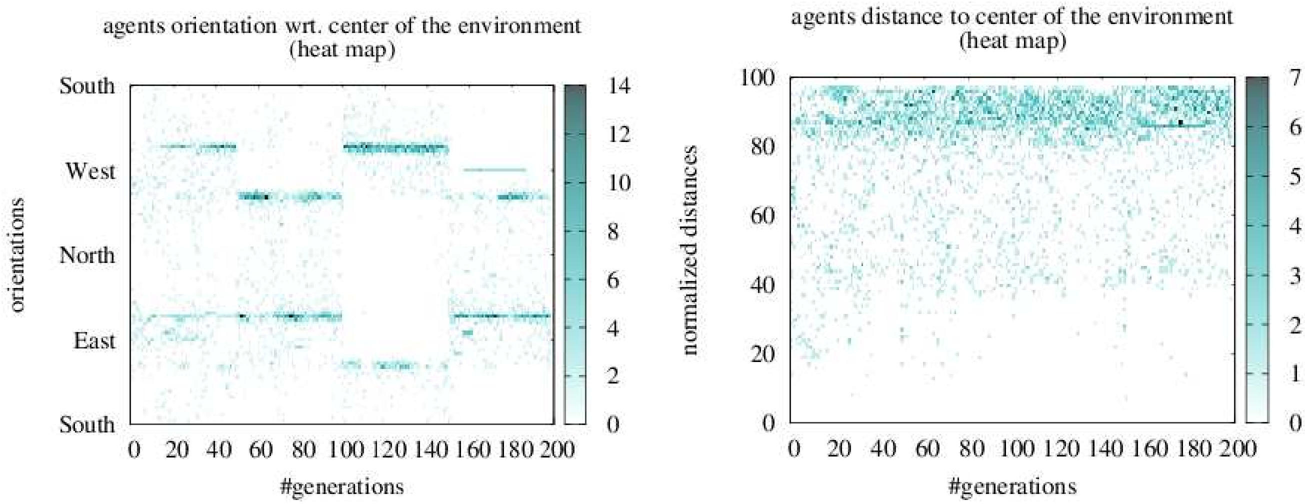


Figure 4.6: Example of several simultaneous consensus during one run (extracted from one run with $pop_{size} = 30$, $r = 32$)

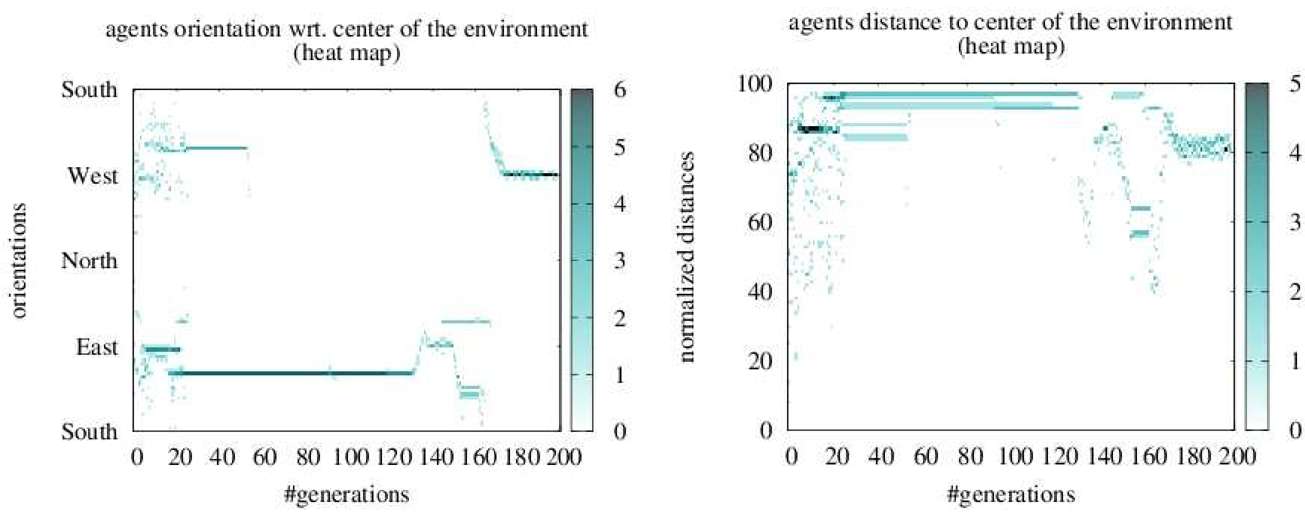


Figure 4.7: Example of a run where agents ignore the sun, except at the very end (extracted from one run with $pop_{size} = 26$, $r = 32$)

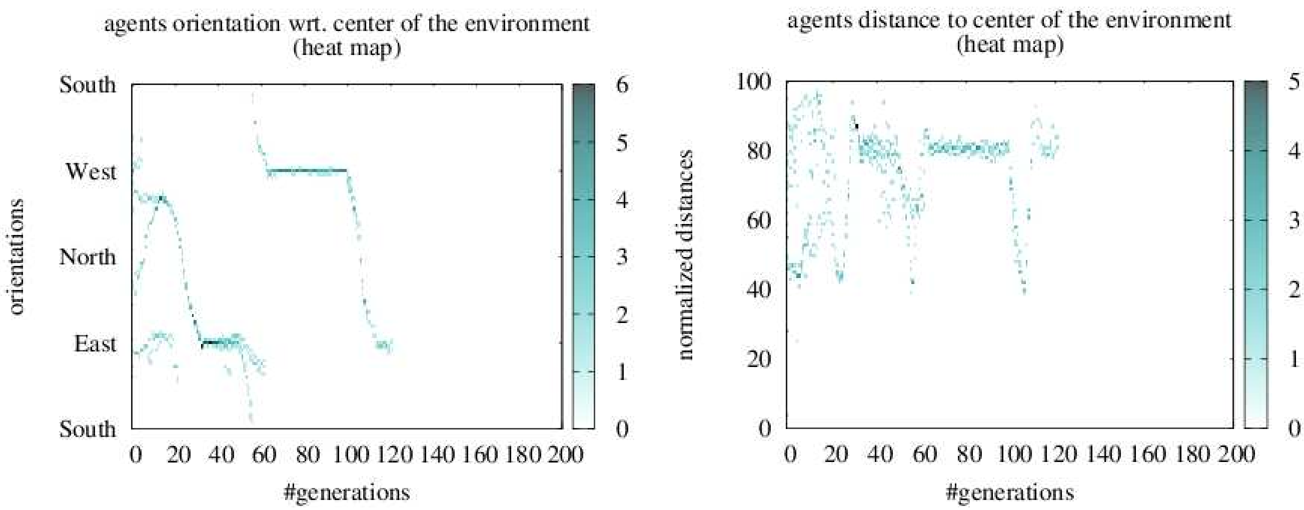


Figure 4.8: Example of run where the population faces extinction (extracted from one run with $pop_{size} = 24$, $r = 32$). This is but one particular example of run leading to extinction and is displayed here for illustration purpose. In this run, the agents were actually able to reach a consensus for some time. However, the relatively low population and medium communication range created a context within which a series of unsuccessful mutation (around generation 120) could not be recovered from and quickly lead to complete extinction.

4.2.4 Summary

In this section we have shown a first demonstration of the mEDEA algorithm presented in Section 4.1. From a general perspective, we have highlighted the impact of the radius of communication, and the number of agents, on the survival of the population. The higher the radius of communication, and the higher the number of agents, the more likely the survival of the population.

Our results aren't restricted to the demonstration of a proof of concept, but give also answers to the questions relative to the emergence of consensus:

- **Emergence of consensus:** When using an experimental set-up with an environmental singularity, we have been able to show the emergence of behavioral consensus. This confirms our hypothesis that consensus doesn't need to evolve *explicitly*, but can rely on the behavioral similarities between one genome and its ancestors.

- **Non-canonical consensus:** Even if the consensus to move toward the sun was the most present among the evolutionary runs studied, other types of consensus have been observed such as going away from the sun.

- **Co-existence of consensus:** In some cases multiple consensus have been shown to evolve and sustain in the population. However, this is possible only when the two sub-populations don't meet each other. If this condition isn't fullfilled, one behavior will be favored by the population.

- **Transient consensus:** In some of the runs studied transition between different types of consensus was observed.

Finally, thanks to the experiments presented in this section, we have show that a smaller radius of communication leads to more stable behavioral consensus.

4.3 Implementation on Real Robots

One of the main challenges in Evolutionary Robotics is to address the so called reality gap [Jakobi *et al.*, 1995], i.e. going from simulation to the real world. We aim here at answering the following question: is the mEDEA algorithm robust to the reality gap ? In traditional Evolutionary Robotics, the reality gap refers to the validation of solutions, i.e. evolved control architecture with a specific behavior, in the real world [Brooks, 1992]. In this work the methodological step is somewhat different as the reality gap concerns the experimental validation of the process (the mEDEA algorithm). We are therefore looking at the comparison between evolutionary dynamics obtained in simulation and on real robots. This section presents the implementation of the mEDEA algorithm on real robots.

To provide experimental validation within a real robot setup, the mEDEA algorithm has been implemented within a population of e-puck mobile robots extended with a Linux board, running at the Bristol Robotics Lab. In this section, the robotic environment is described, as well as considerations regarding implementing mEDEA in this context. Then, results from experimental trials are described and conclusions are drawn.

4.3.1 Technical Overview

Research on swarm robotics has gained much attention in recent decades as a novel biologically-inspired approach to the coordination of large groups of relatively simple robots, following simple rules [Dorigo and Şahin, 2004; Şahin and Spears, 2005; Şahin and Winfield, 2008]. Generally, in order to carry out real robot experiments in research labs we require a robot which is small, reliable and inexpensive in order to minimise physical space and maintenance for running a relatively large number (several tens) of robots. Traditionally research labs have designed and built their own robot platforms for swarm robotics research, such as the Linuxbot [Winfield and Holland, 2000], Alice [Caprari *et al.*, 2002], Jasmine [Kornienko *et al.*, 2005] and Swarm-Bot [Gross *et al.*, 2006a]. There are also a number of commercially available mobile robots suitable for swarm robotics research, such as the widely used Khepera II and III from K-Team, Lego Mindstorms from the Lego company and Create from iRobot. However, the open-hardware e-puck educational mobile robot developed at the *École Polytechnique Fédérale de Lausanne* (EPFL) has become very popular within the swarm robotics research community within the last three years [Mondada *et al.*, 2009]. The e-puck combines small size – it is about 7cm in diameter, 6cm tall and 660g weight – with a simple and hence reliable design[3]. Despite its small size the e-puck is rich in sensors and input-output devices.

The basic configuration e-puck is equipped with 8 Infra-Red (IR) proximity sensors, 3 microphones, 1 loudspeaker, 1 IR remote control receiver, a ring of 9 red LEDs + 2 green body LEDs, 1 3D accelerometer and 1 CMOS camera. Bluetooth provides the facility for wirelessly uploading programs and general monitoring and debugging. All sensors and motors are processed and controlled with a dsPIC30F6014A microprocessor from Microchip™. Extension sockets provide for connecting additional sensors or actuators.

In order to ease the porting of our algorithm to real robots, we relied on the Linux board for e-puck developped at BRL [Liu and Winfield, 2011]. An embedded Linux system is installed on this board as the primary operating system of the whole robot. Our algorithm is run on this board, while the lower level sensor processing and motor control is executed on the e-puck DSP. One of the key advantages is the introduction of the WiFi into the system to provide fast and topologically flexible communication. This also provides a convenient way for wirelessly accessing and controlling the robot. In terms of productivity and ease of use, the board also allows us to use a wide range of development tools in addition to the C/ASM language development environment. The linux board for e-puck offers not only enhanced processing power, memory and communications, but a powerful control architecture for the robot. For instance, it provides flexibility in how programs may be compiled inside the robot natively (instead of cross-compiled on PC), with standard Linux tools and frameworks, including Player/Stage [Vaughan, 2008].

Figure 4.9 illustrates the modified e-puck robot and its environment. The primary function of the yellow 'hat' at the top of the robot is to allow us to mount reflective markers for the visual tracking system, but additionally the USB WiFi card (with its cover removed), is fitted into a slot on the underside of the hat.

Programming, initializing, starting and stopping experimental runs of a large swarm of mobile robots, then monitoring and logging data from those runs, is problematical if it has to be done manually. However, with the Linux extended e-pucks and wireless networking, we have been able to set up a powerful infrastructure for programming, controlling and tracking swarm experiments much more conveniently. Figure 4.10 illustrates the overall structure of the experimental infrastructure. Each e-puck robot is configured and identified with a static IP address. They connect to the LAN through a wireless router and can be accessed from any desktop computer connected to the network using the SSH protocol. A 'swarm lab server' is configured as a central code repository and data logging pool for the swarm of robots. The server also functions as a router to bridge the swarm's wireless subnet and the local network. In addition, as there is no battery-backed real time clock (RTC) on the extension board, the server may provide a time server for synchronisation of the robots' clocks and time stamping log data.

[3]The open-hardware design can be found at http://www.e-puck.org

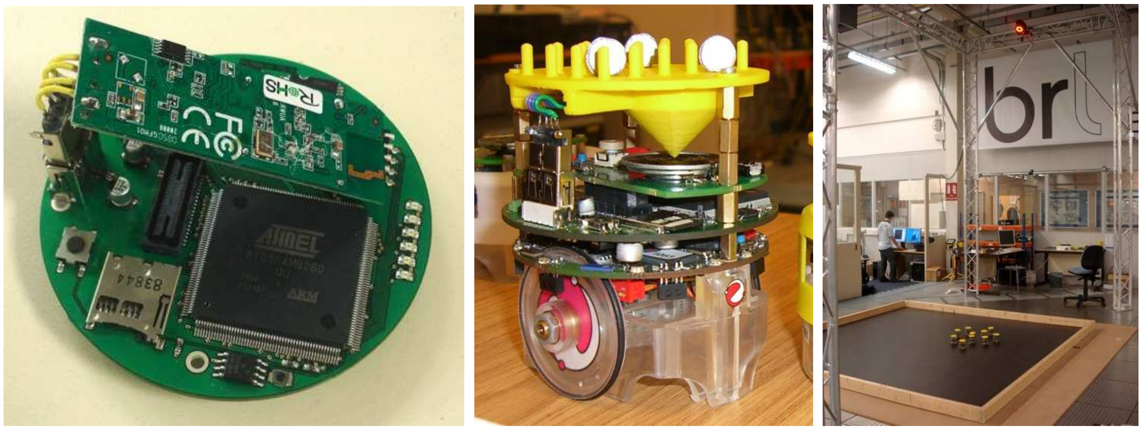


Figure 4.9: Left: e-puck Linux extension board with USB WiFi card (casing removed). Middle: An e-puck with Linux board fitted in between the e-puck motherboard (lower) and the e-puck speaker board (upper). Also note the yellow 'hat' which here serves three different functions: (1) it provides a matrix of pins for the reflective spheres which allow the tracking system to identify and track each robot, and (2) it provides a mounting for the USB WiFi card which slots in horizontally (the wires connecting to the WiFi card are above the USB connector). Right: experimental swarm robotics arena with 10 Linux extended e-pucks.

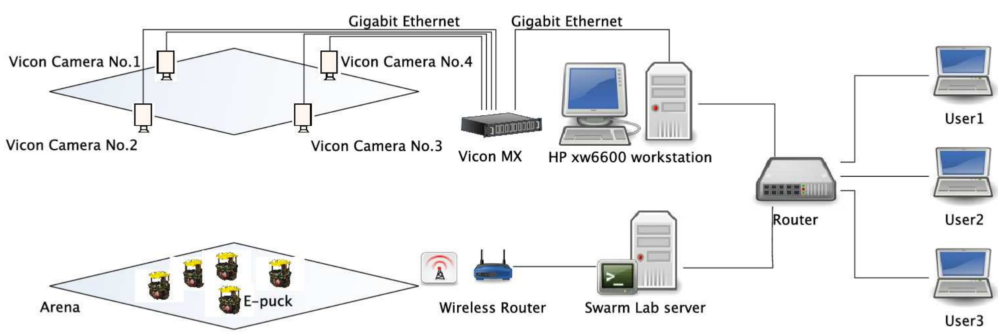


Figure 4.10: Experimental infrastructure for swarm robotics research based on the Linux extended e-puck. The Swarm Lab Server provides a data logging capability that combines and time stamps position tracking data collected by the Vicon™system with robot status and sensor data from the e-pucks via WiFi, into a log file for post-analysis of experimental runs.

A visual tracking system from Vicon™[4] provides high precision position tracking for robot experiments. This consists of four Vicon™MX400 cameras, one Vicon™MX and one HP xw6600 workstation. Each robot is uniquely identified by the visual tracking system from the pattern of reflective markers mounted on the matrix pins of the yellow hat, as shown in Figure 4.9:center. The tracking system is connected to the local network and broadcasts the real-time position of each tracked robot through a standard TCP/IP port. We use the position tracking data for logging and post-analysis of experimental runs.

4.3.2 Implementing the Two-suns Experiment

In order to validate the mEDEA algorithm an experimental setting, strongly inspired by the two-suns setup described in section 4.2, was designed. A population of up to 20 robots is placed in a $280\,cm * 230\,cm$ empty arena. Limited-range communication is emulated by using a WiFi network combined with the Vicon™tracking system. An additional object is introduced into the arena and act as a landmark which is referred to as the **sun**: each robot in the simulation knows the sun's relative direction and distance thanks to the Vicon™system. The experimenter may arbitrarily change the sun's location from time to time during the experiment, switching the sun location from one end of the arena to the other. All experiments described in this section lasted for at least 30 min (duration of experiments was restricted due to the limited autonomy of the robots), and at least 25 generations (each generation lasts approx. 1 min and 10 sec, corresponding to 400 time steps – this time is sufficient for a controller to cross the whole area, and therefore meet most of the robots). The refresh rate for a robot controller is limited to 5 updates per second due to various technical limitations of the setup (cf. section 4.3.4).

Moreover, a set of technical issues were addressed, and are listed here along with their diagnosis and solution:

- Lack of selection pressure:
  - diagnosis: given a small arena and few robots, it is likely that one robot can meet all other robots. This would imply that selection pressure is eliminated (see section 4.1).
  - solution: when the maximum number of robots is known it is possible to limit the number of genomes that can be imported within one agent so that $N_g < N_p - 1$ (with N_g the maximum size of the genome list size and N_p the population size). In this particular setup, the genome list is arbitrarily limited to 17 for each robot, and is filled on a first-met-first-served basis to enforce a pressure toward fast mating. Note that an extreme case would be to limit the genome list size to 1, i.e. the first genome imported. However, the selection pressure would then be particularly aggressive, and may not be suitable for the

[4] http://www.vicon.com

current experimental setting which is already prone to high variability because of the small population.

- Slow convergence:
  - diagnosis: because we are running physical robots in the real world, time becomes an expensive luxury.
  - solution: the search space has been limited to a simple two-layer perceptron with no hidden layer, in contrast with previous experiments. Moreover, a large initial mutation rate is chosen. As few robots are available, large values of σ should promote the discovery of new solutions. Thus σ_{init} is set to 0.1, σ_{max} to 0.4 and σ_{min} is fixed at 0.05.
- Risk of extinction due to small populations:
  - diagnosis: as shown in section 4.2 small populations are more prone to extinction, implying time consuming interventions from the human supervisor.
  - solution: a restart procedure is introduced into the algorithm. Whenever a robot stands inactive for 5 consecutive generations it simply picks a new genome with random values. This simple feature thus makes it possible to avoid extinction and start possibly from more promising regions in the search space.
- Lack of global synchronisation:
  - diagnosis: generation count among robots cannot be synchronised on a global basis, resulting in robots with genomes from different generations.
  - solution: we assume that the mEDEA algorithm is naturally robust to asynchronous generation count as it does not provide any survival advantage for any genome[5]. Therefore no further modification is performed.

4.3.3 Results

An initial set of 21 experiments were conducted to explore the performance of mEDEA under various parameter settings: number of robots (from 9 to 20), generation duration, mutation values, communication radius, etc. Results from these preliminary experiments lead to several important conclusions. Firstly, the population size was found to be the most critical parameter: switching from 9 to 20 robots completely changed the observed outcome of the experiments as mEDEA clearly benefits from larger populations (cf. section 4.1). Secondly, the communication radius was also identified as a key parameter: small radii (10 cm) implied versatile populations, with rare convergence toward specific

[5] As a counter example, generation times do need to be equal across robots as longer generation times would lead to more opportunities for a genome to spread.

behaviours while larger radii (20 cm or more) more often led to the emergence of more stable consensus within the population.

A practical conclusion is that the small number of robots available can at least be partially compensated by a larger communication radius. Indeed, experiments with a population of 20 robots and a communication radius of 10 cm would not result in more than 17 active robots at the same time while extending the radius to 20 cm regularly led to the whole population of robots being active at the same time. On the other hand, a population of 20 robots still remains relatively small with regards to experiments performed in simulation (see previous sections). A direct consequence is that in all experiments consensus were occasionally lost, switching from one kind of consensus (e.g. following the sun) to another (e.g. ignoring the sun).

Following these preliminary experiments, 8 experiments with 20 robots based on similar parameters were performed using the experimental settings described in section 4.3.2. Each experiment lasted from 30 to 45 min depending on the energy consumption. The sun was moved after approx. 25 minutes and emergence of consensus was studied both from camera recordings and experimental data recorded using the Vicon™tracking system as well as the internal data logs recorded by each robot. Figure 4.11 illustrates one of the 8 experiments run with similar settings using 19 robots[6]. Both the emergence of consensus to go toward the sun (above) and the effect of changing the sun location (below) are visible on this figure. The images are extracted from a video summarising the main results from these experiments [7].

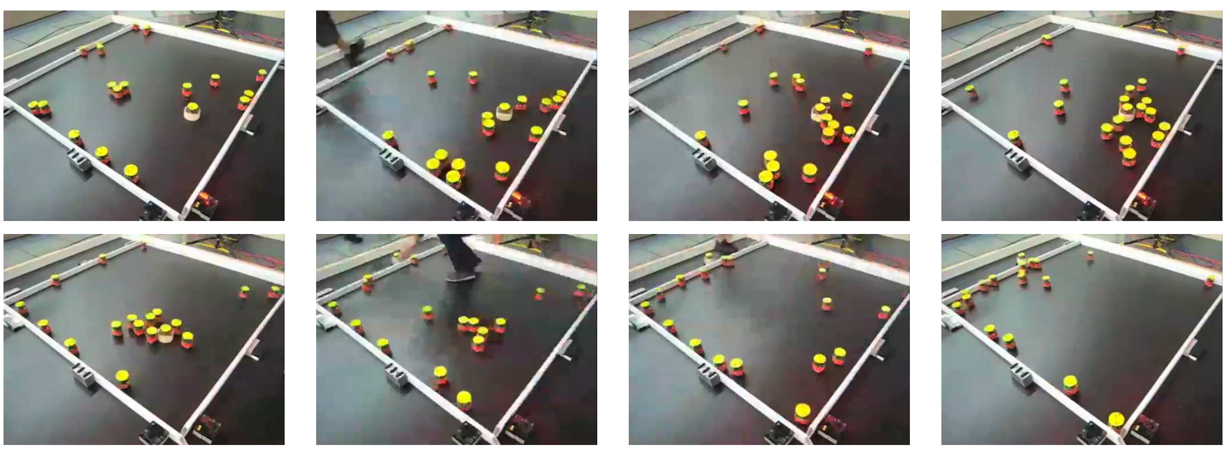


Figure 4.11: Illustration from the experiment described in the text. Above: emergence of consensus to go toward the sun. Below: impact of changing the sun location (caption 2 shows the human changing the sun location) and convergence of the robots to the new location.

The following issues are considered: the emergence of consensus over time, the number of active robots during the run, and robustness to environment changes. Figure 4.12 gives the number of active

[6]one robot was removed due to technical failure in a previous experiment.

[7]Video publicly available at: http://www.youtube.com/watch?v=_ilRGcJN2nA - Retrieved 19 December 2012)

robots during the experiment. Figure 4.13 tracks the distance between each robot and the sun and is represented as a density map (or "heat map", i.e. darker regions indicate the most represented distance value). Darker regions close to zero indicate that robots stand close to the sun, and are likely to be a good indication of the occurrence of sun-follower genomes in the population. Lastly, figure 4.14 features the weights (i.e. gene values) of the neural links connecting the sun-orientation and sun-distance sensor values to the motor rotational value of each robot (later referred to as sun-orientation gene and sun-distance gene). The weight of the sun-orientation sensor input is particularly interesting as it provides a good indication of the possible correlation between the sun position and the robot behaviour, even though the exact nature of the correlation may be difficult to guess *a priori* (since motor outputs also depend on other values including sensors and NN weights).

From these data, it is possible to analyse the course of the experiment:

- t=[0sec,400sec]: 14 to 18 robots (fig. 4.12) are active and a consensus to go toward the sun is emerging, while not exclusive (the dark region near 0 in fig. 4.13);

- t=[400sec,800sec]: The consensus is suddenly lost at t=400sec (possibly because too few robots were involved). This can be observed in both fig. 4.12) and fig. 4.13. The number of active robots drops to 8 and we can also observe that the genotypic signature shown in fig. 4.14 does indeed change with the disappearance of sun-orientation gene values around 0.

- t=[800sec,1400sec]: the number of active robots increases to the maximum of 19 robots, and is correlated with most robots being located near the sun (fig. 4.13), possibly implying an even stronger consensus than before. Moreover, this remains stable over time until the sun location is changed at t=1400sec.

- t=[1400sec,2000sec]: Shortly after changing the sun location, the number of active robots suffers from a short decrease (to 14 robots) followed a quick recovery (to 18 robots). This suggests good robustness in the population, an inference which is reinforced by looking at the two other figures: the robots slowly converge back to the sun and the genotypic signature remains unchanged. We can also observe that sun-follower genomes are likely to be correlated with sun-orientation gene values close to zero (but positive).

The 7 other experiments with similar settings provided comparable results. All the runs displayed the emergence of various types of consensus as well as occasional time periods in which no consensus could be identified. Also, changing the sun location had an important impact during experiments, with the vast majority of robots adopting the same "go toward the sun" consensus. This can be explained by the fact that sun-follower genomes spread while their robot hosts were moving toward the sun. The following general conclusions can be drawn from these experiments:

- The robot restart procedure is mostly used in the initial generations, which implies the population becomes self-sustaining (i.e. there are enough encounters between robots to avoid requiring the restart procedure);

- The largest number of simultaneously active robots was obtained with a population of sun-followers (from 15 to 19 active robots having reached a consensus to go toward the sun).
- Changing the sun location had different effects depending on the existence (or non-existence) of consensus in a population. Populations of robots ignoring the sun did not suffer from such a change, while sun-follower populations first suffered (ie. some agents are lost and stop) from the sun changing location followed by a quick recovery.

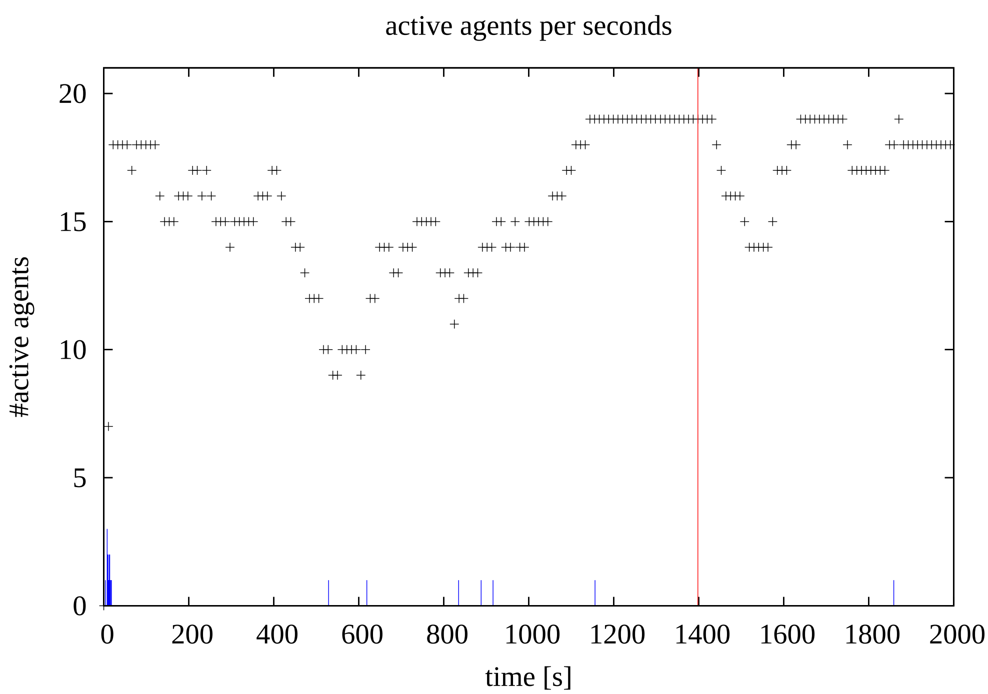


Figure 4.12: Number of active robots during a selected experiment. Small bars correspond to one of the robots restarting. The high bar corresponds to the sun location changing event.

4.3.4 Discussion of the Reality Gap

Implementation of the mEDEA algorithm within a population of robots reveals a number of technical issues unfamiliar to experiments in simulation. Together, they comprise the reality gap between simulation and real world experiments [Jakobi *et al.*, 1995]. Here, the reality gap is studied from the perspective of the evolutionary dynamics observed during a trial. These issues are articulated as follows:

- Proximity sensors are unreliable: the low quality of the infra-red sensors makes it very difficult to detect obstacles and/or other robots (with a binary positive response only for distances under

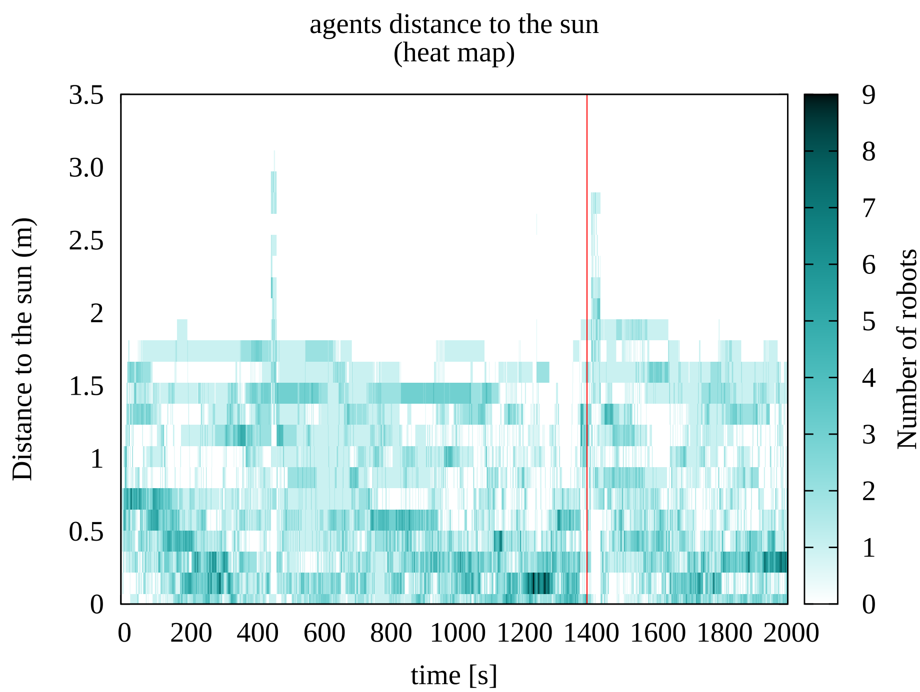


Figure 4.13: Distance of the robots to the sun at every time step during a selected experiment. Darker regions imply more robots are located at this particular distance from the sun. The high bar corresponds to the sun location changing event.

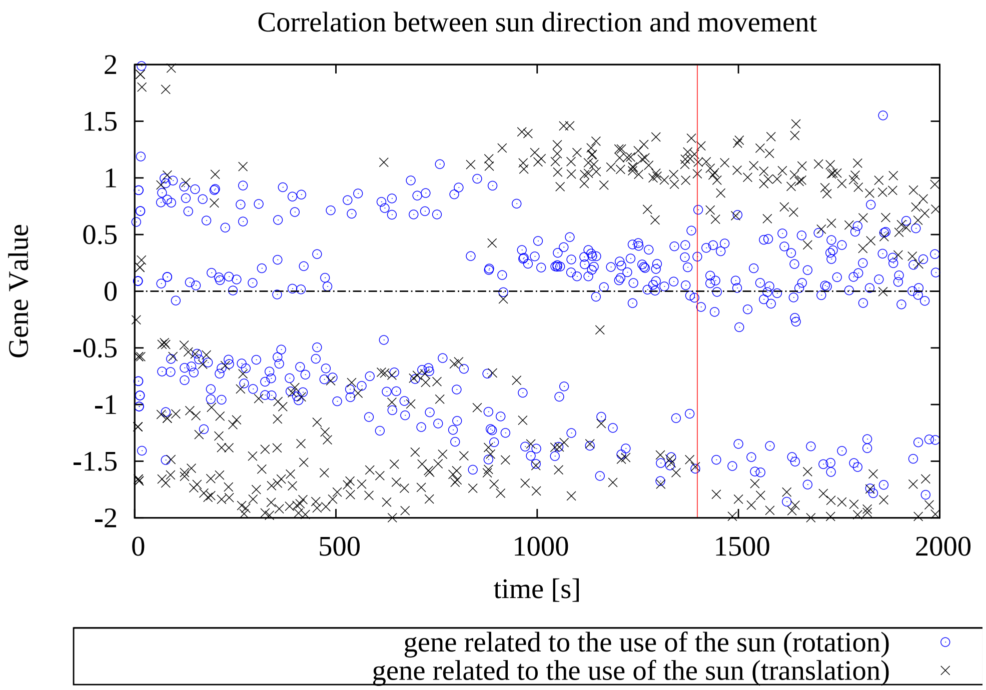


Figure 4.14: Tracking the values for each robot at each generation of genes related to the sun orientation: NN weights connecting the sun's orientation to (a) agent rotational speed ("sun-orientation gene") and (b) agent translation speed ("sun-distance gene"). The high bar corresponds to sun location changing event.

1cm). Also the e-puck body is more or less transparent, making it almost invisible in some cases. Adding a coloured plastic skirt to the robots only partly solves the problem as the proximity sensors are occasionally blinded by the skirt. As a consequence, the proximity sensors can be disregarded because of their unreliability.

- Colliding robots are regularly unable to send/receive genomes to their neighbours. This is due to our particular setup where local communication is emulated using the Vicon™system for computing the local communication network based on distance between robots. While this approach was originally motivated by the lack of local robot-robot communication, it ended up as quite a problem as robots could not participate in the evolutionary adaptation process during collisions. In practice robots were occasionally lost from the emulated communication network (average of ~5% of robots lost per minute) but always recovered when moving away from one another.

- On-board processing is slow: the combination of on-board computation with limited hardware and the particular setup for emulating local communication has a negative impact on speed of execution. In fact 5 updates/sec was observed, with often asynchronous updates of sun distance and orientation from the tracking system (i.e. out-of-date information).

Two conclusions can be drawn from these issues. Firstly, simulation and real world experiments do differ, as expected. Secondly, however, results from the experiments showed that mEDEA is remarkably efficient to cross the reality gap, as it manages to deal with all of the issues outlined above and still demonstrates interesting behavior that manages to survive in the environment. As a matter of fact, the most critical issue from these experiments in the real world is the small population size, whose importance was already discussed in Section 4.2 as it impacts the stability of the algorithm: larger populations display more stable behaviors, while very small populations (i.e. less than 20 robots in this context) are prone to extinction. Preliminary experiments using less than 10 robots have confirmed this analysis: few runs converged toward a behavioral consensus, and extinction was observed in some runs. In the present context, the negative effect of a small population was mainly counter-balanced by two modifications:

- **Large communication range:** As shown in Section 4.2, on top of increasing the behavioral stability, larger communication radius reduce the risk of extinction. After preliminary experiments, the communication radius has been increased up to 20cm.

- **Restart mechanism:** A restart procedure is triggered when a robots stands inactive for 5 consecutive generations. This mechanism aims at avoiding the extinction of the population, and explores possibly more promising region of the search space.

4.4 Summary and Conclusions

In this chapter the problem of self-adaptive maintenance of robots integrity in unknown environments has been addressed by the design of a new algorithm. This light-weight algorithm, termed mEDEA, is suitable for implementation within hardware/software setups with limited computational power such as robotic agents. The mEDEA algorithm has been tested both in simulation and on real worlds with regards to its ability to evolve successful strategies as well as achieving behavioral consensus.

In simulations, large groups of robots were shown to converge to stable, mostly unique, behavioral strategies. When using smaller populations size (i.e. facing similar environmental conditions with fewer resources), dissimilar strategies co-existing at the same time were observed in few simulations.

Experiments with real robots have illustrated the applicability of the algorithm to hardware implementation. Moreover, it has been shown that the mEDEA algorithm converges also to behavioral consensus in these conditions. However, due to difficult environmental conditions, co-existence of dissimilar behavioral consensus wasn't observed. To sum it up, it appears that one key element for the success of the algorithm is to run within a large enough population.

The experiments presented in this chapter provide a first understanding of the global behavior of the mEDEA algorithm. In the next chapters, the mEDEA algorithm is evaluated in more challenging situations. The next chapter assess the performances of mEDEA in front of changing environments. The Chapter 6, is concerned by the use of mEDEA to optimize the population's global welfare rather than individuals' welfare.

Chapter 5

Robustness to Environmental Changes

In this chapter we address the following question: How to achieve self-adaptation for robots deployed in unknown and changing environments ? This class of problems typically arises when the environment remains unknown to the human designer until the population of agents is actually made operational in the real situation, and the environment changes during operation, without any indication on when and how these changes will impact the agent's strategies.

In robotic context, methods proposed so far to address this challenge can be divided in two main categories:

- **Design of efficient and reliable controllers:** Multiple methods have been proposed to design efficient and reliable controllers for homogeneous group of robots [McLurkin and Smith, 2004; Pettinaro *et al.*, 2005; Nembrini, 2005]. These methods are proposed to address the design of controllers suitable for a large range of tasks, without involving any learning mechanism.

  Multiple authors have proposed new paradigms based on the observation of physical systems [Shimizu *et al.*, 2003; Pugh and Martinoli, 2008], biological systems [Schmickl *et al.*, 2009], or hybrid of both [Meng *et al.*, 2007]. The latest works in this field known to us are challenging the robust coordination of heterogeneous groups of robots [Dorigo *et al.*, 2012].

  These design methods are useful if the human engineer has enough knowledge about the task at hand and the robots used. Another assumption is that the possible environmental changes are known beforehand. If all these conditions are met, the conception and test of a robust controller will still requires a deep investment from the human engineer.

- **Automatic design of controllers:** Mutltiple contributions have been made to address the challenges of the autonomous design of robust controllers. Among them we have identified two categories. The firsts aim at learning the proper coordination of behaviors hand-designed by a human engineer [Parker, 2000; joon Sun *et al.*, 2001; Dias *et al.*, 2004]. These methods are robust to environmental changes and partial failure of the robots, but can be used only if the human engineer has enough knowledge about the task at hand.

  The seconds are challenging the autonomous design of robotic controllers without human intervention [Trianni *et al.*, 2008]. However, within these methods the conception phase is often performed in simulation, which leads to the previously presented reality-gap problem (see Section 2.3). Moreover, using simulations during the design phase would imply the possibility to simulate environmental changes (which isn't the case here).

In all approaches reviewed above, a human engineer is mandatory, either to design the robots' controllers, or to design a reward function necessary to the autonomous design of controllers. Therefore, these methods can be used only if the environment is known to the human engineer before the deployment of the robots, which isn't the case considered here. As a consequence another method is needed to design continuously and autonomously controllers suitable to an environment unknown to the human engineer and possibly changing.

In this chapter experiments mEDEA is evaluated in unknown and possibly changing environments. More specifically, the resulting behaviors are studied both at the individual scale and population scale. A new environmental setup is presented in Section 5.1, and the results obtained are detailed in Section 5.2. The work presented here is based on our publication [Bredeche and Montanier, 2010].

5.1 Experimental Setting

5.1.1 The Problem: Surviving in a Dynamic Unknown Environment

As already stated in the previous chapter, no information whatsoever about the task and the environment at hand can be used to tailor the algorithm. Quite the opposite, the exact same algorithm is used in all cases without external intervention.

In practice, the experimental setting is based on the succession of two different experimental setups (which share few similarities in order to avoid populations' extinction). During one run no intervention whatsoever is performed on the agents (change of genome, restart of the algorithm, modification of the learning algorithm).

The two environmental setups are detailed hereafter:

1. **The "free-run" setup:**
    - **Description:** A population of autonomous mobile agents is immersed within an environment with few obstacles. As a consequence, an agent dies only if it was not able to meet at least one other agent - ie. the current genome is lost for sure as it does not get a chance to survive within any other agents.
    - **Motivation:** This setup makes it possible to evaluate the mechanisms of the mEDEA algorithm as environmental pressure should be limited. Some key behavioral elements should be learned during this phase that might be useful for the second environmental setup.

2. **The "energy" setup:**

   - **Description:** A set of energy resources ("food items") is spread all over the environment, which can be harvested by the agents. Agents are endowed with an energy level, which depends on harvested food items and power consumption. If the energy level reaches 0, agent dies and genome information is lost. Moreover, harvested food items only "grow" back after a given number of iterations.
   - **Motivation:** In this setup, genomes also compete for agent resources but have to deal with environmental pressure as maximizing mating encounters may not be fully compatible with energy self-sustainability. The added constraints might put the agents in difficult situations as they have never been experienced before. However, the behaviors evolved in the first experimental setup might be modified to the new experimental setting.

5.1.2 Implementation

Figure 5.1 shows the environment used for the experiment: a 2D arena with obstacles, possibly containing food items. The figure also illustrates 100 autonomous mobile agents loosely inspired from the ePuck mobile robot specifications. These agents are similar to the ones used in the previous chapter:

- Each agent has two motor outputs controlling the translational and rotational speed, and 8 proximity sensors arranged around the agent body,
- Each agent is controlled by a MLP with 5 hidden neurons,
- The weights of the MLP are evolved locally by using mEDEA on each robots,
- The variation operator is a Gaussian mutation whose σ parameter is added to the genome and increased or decreased randomly.

However, for this experiment three additional sensory inputs are considered: the angle and direction toward the nearest food item and the current energy level (which is set to a fixed value in the "free-ride" setup). Note that these additional sensor values are useless in the first setup, and may even be considered as distractors.

The full experimental setup considers starting with the "free-run" setup, and then suddenly switching to the "energy" setup after a pre-defined fixed number of generations. In the meantime, agents are of course unaware of such a change in the environment and keep on running the same unchanged mEDEA algorithm. The whole experiment lasts for 150 generations, switching from the "free-run" setup to the "energy" setup at generation 75.

As said before, the energy level is not used during the "free-run" setup. During the course of evolution some agents may come to a halt because they did not meet any other agents, thus failing

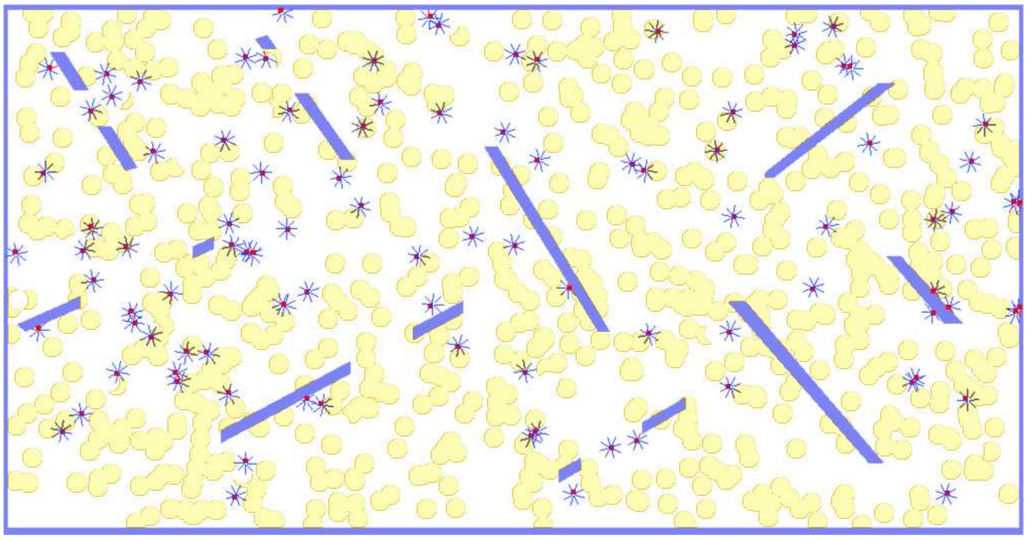


Figure 5.1: Snapshot from the simulator with 100 agents. Yellow: food items. Red: agents, modeled after an e-puck robot. Blue: range of proximity sensors (the range of proximity sensors is reduced in this snapshot for visibility issues).

to import a new genome for use in the next generation. In the "free-run" setup, the agents without a genome remain still (or "inactive", i.e. without genome), waiting for new genomes imported from "active" agents that eventually come into contact[1].

In the "energy" setup, the agents can also become inactive because they ran out of energy during the "energy" setup (each agent can store a maximum of 400 energy units and consumes 1 unit/step, one generation lasts 400 steps, each harvested food item gives 100 units of energy). These agents remain stationary until the end of the current generation. After this time they are automatically refilled with enough energy for surviving through slightly more than one generation, so that when an agent become active it has enough energy to reach a food item. These agents remain inactive, and wait for a new imported genome that may be used for the next generation. While the reviving procedure makes it possible to avoid progressive extinction, extinction is nevertheless possible whenever all agents in the population fail to meet any other agents during one generation, whatever the cause (bad exploratory or harvesting strategies).

Therefore, monitoring the number of active agents in a population provides a reliable indicator of the performance of the algorithm as external intervention is considered as impossible (except for energy refill if agent runs out of energy). Detailed parameters used for the experiment presented in the next section are given in Tables 5.1 and 5.2.

In order to provide a challenging environment, the "energy" setup is designed so that the number of food items in the environment depends on the actual number of active agents. A food item grows

[1]Note that the simulation begins with each agent containing a randomly initialised genome.

Parameter	**Value**
arena width and length	$1024 * 530$ pixels
"free-run" setup duration	75 generations
"energy" setup duration	75 generations
lifetime (i.e. generation duration)	400 steps per generation
population size	100 agents
agent diameter	1 pixel
proximity sensor range	64 pixels
radio broadcast signal	32 pixels
agent rotational velocity	0.52 rad/s
agent translational velocity	2 pixels/step
genome length	79 real values (78 MLP weights + σ)
variation operator	Gaussian mutation with σ parameter
$\sigma_{minValue}$	0.01
$\sigma_{maxValue}$	0.5
$\sigma_{initialValue}$	0.1
α (ie. σ update parameter)	0.35

Table 5.1: Parameters for experiments.

"energy" setup only:	
food items	2000
food item diameter	10 pixels
food item regrow delay	btw 400 and 4000 steps (see text)
energy per food item	100 energy units
agent energy consumption	1 energy unit per step
agent maximum energy level	400 energy units
agent initial energy level	400 energy units

Table 5.2: Parameters for experiments - Specific to the "energy" setup.

back whenever harvested, but only after some delay. If the number of active agents is less than half the population size, then $delay_{regrow}$ is set to 400 steps. However, if the number of active agents is between 50 and 100, then the delay linearly increases from 400 steps (fast regrowing) to 4000 steps (slow regrowing, aggressive environment), as follows: $delay_{regrow} = 72 * nb_{agents} - 3200$.

Clearly the smaller the population the easier it is to harvest enough food items for survival. On the other hand, a 4000 steps regrow delay implies that a given food item is available only once every ten generations, which gives a setup of the utmost difficulty. A population of 80 altruistic agents (i.e. agents harvesting only *what* is necessary) with perfect coordination (i.e. agents harvesting only *when* necessary) would not even be able to fully survive in this environment with these parameters. Combining the regrow delay update scheme with the equation relating the required number of food items and population size leads to a quadratic equation with one positive solution. In this setup, the optimal population size is strictly below 80 agents. However, as soon as 50 agents are active the regrow delay increase quickly (72 iterations) for each new active agent, which makes this setup very aggressive when more than 50 agents are active.

In the particular setup described here, switching from a possibly efficient population of 100 agents from the “free-run” setup to the “energy” setup will have a possibly disastrous impact as the number of agents at the beginning of the second setup implies longer regrow delays.

All experiments have been run in the roborobo simulator already presented in Section 4.2.1.

5.2 Results and Analysis

The lack of explicit objective function makes it difficult to compare performance during the course of evolution. However, the two conflicting motivations presented in Section 3.4 can be evaluated thanks to computationally cheap measures:

- The evolutionary pressure for the survival of the agents affects the integrity of the swarm, and is therefore estimated thanks to the number of active agents. Moreover, the behavior evolved by the agents regarding this issue can be analyzed from the point of view of their energy balance (i.e. the difference between the energy harvested, and the energy consumed by one agent during one generation).

- The evolutionary pressure for a genome to duplicate itself will tend to create many mating opportunities between agents. Thus, the average number of imported genomes per generation gives a good hint of how the algorithm performs regarding this motivation. The σ mutation parameter provides also interesting informations: a low value results in the conservation of most of the genomes (even if it is suboptimal), a high value results in the creation of completely new behaviors.

5.2.1 Results

The four graphs in Figure 5.2 give a synthetic view of the results over 100 independent runs obtained with mEDEA on the experimental scenario described in the previous section. These graphs compile values of the selected parameters, or "indicators", over generations: number of active agents, ratio of imported genomes per agent (normalized by the number of active agents), energy balance per agent, and σ mutation parameter values.

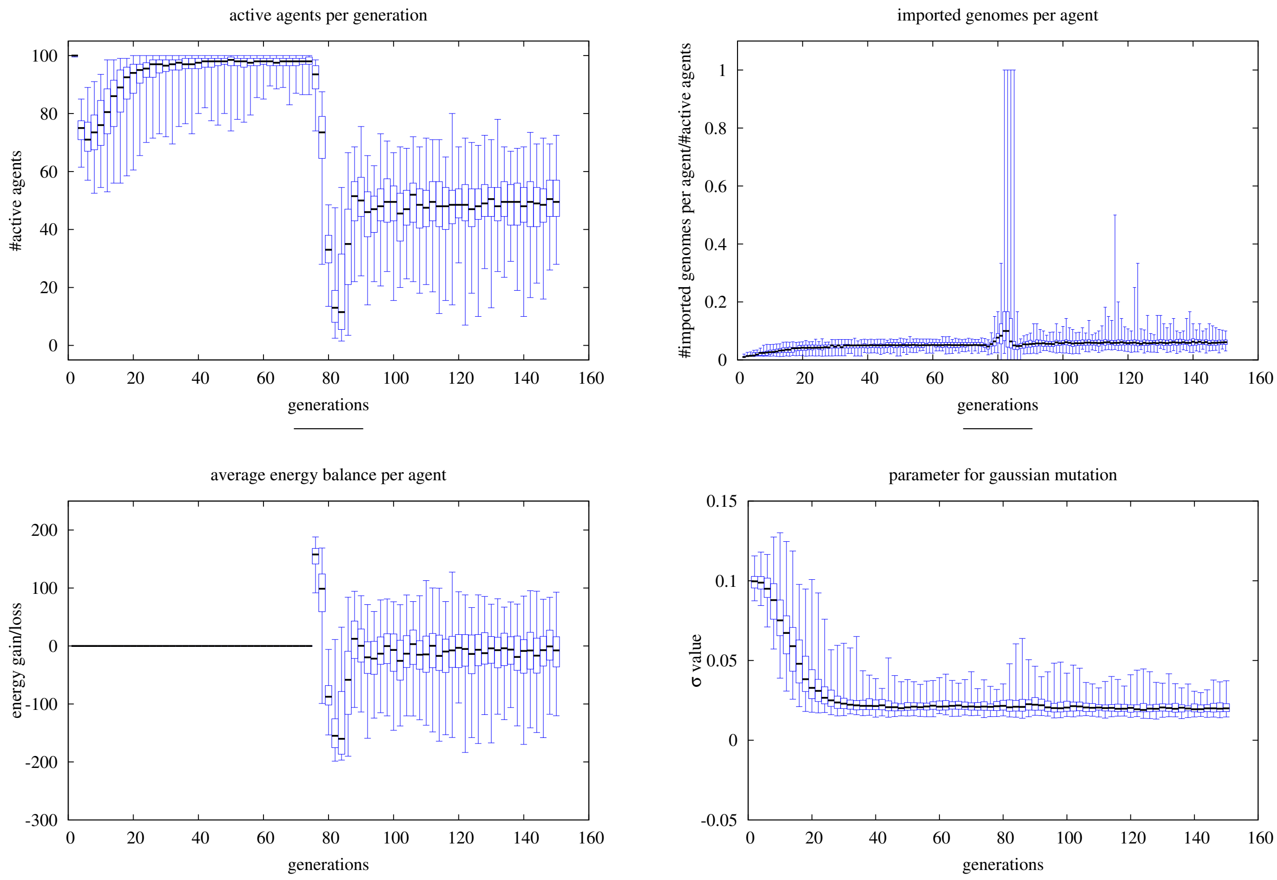


Figure 5.2: Results over 100 runs for the change of environment experiment: number of active agent, size of genomes list, number of energy point harvested, σ mutation parameters

In both setups, the number of active agents rises to reach stable average values, which show an increase in the capacity of the agents to survive in their environment. Moreover, this indicator drops after the switch of setup, and the recovers to a stable higher value. This shows the capacity of the mEDEA algorithm to address the pressure for the survival of agents in front of a changing environment. Moreover these observations illustrate the ability of mEDEA to maintain the integrity of an agent's swarm in changing environments. This interpretation is supported by the increasing value of the energy balance which is a key element for the survival of agents in the second setup.

The ratio of imported genomes increases in the first setup, fluctuates when the environment is switched, and then reaches back a stable level (this value is actually slightly increasing in the second part of the experiment). The strong fluctuations are due to the small number of active agents right after the switch of environment. An increase in the number of imported genomes is interpreted as an increase in the mating opportunities which corresponds to the intrinsic motivation. Therefore, in this scenario, the mEDEA algorithm address the evolutionary pressure for genomes to duplicate themselves in the same time as the pressure for the survival of agents.

The Gaussian mutation parameter is the only indicator not really influenced by the change of environmental setups (except for a slight increase in maximum values). The constant decrease of the σ parameter close to minimal values comes from two aspects: a)the σ update scheme is biased toward small values) b) slightly mutated genomes have a higher chance to produce coherent behaviors.

The ability to recover after a change of environment comes from the spread of one of the few surviving, yet fit, genomes facing few competitors. This genome is later adapted to the environment by the process of mutation, mating and selection used also in the first setup. Moreover, complete extinctions were observed after switching to the "energy" setup only in 3 of the 100 runs (results not shown).

While the results may vary among runs, with a great difference between minimal and maximal values for each indicator, values between the upper and lower quartiles are remarkably close given the noise inherent in this kind of experiment.

5.2.2 Analysis of Single Runs

Examples of randomly chosen runs, limited to tracking the number of active agents, are given in Figure 5.3 for further illustration. These graphs highlight the course of evlution in both setups. In the "free-run" setup, the number of active agents is maximized up to the total number of agents. However in the "energy" setup, the number of active agents is always oscillating between a low value (easy environments), and a high value (difficult environments). These oscillations are smoothed and shadowed in Figure 5.2 because of the large number of runs considered. The number of active agents oscillates around the stable solution for mEDEA in this experiment (i.e. a larger population cannot survive in the environment, while a smaller population does not exploit all available resources. The oscillations are essentially due to the design of the experiment. When the number of active agent is lower than 50, the environment is easy and a high number of agents become active. However, as soon as the number of active agent is higher than 50, the environment becomes much more difficult, and the number of active agents drops. The question of cooperation among agents, which could improve the population integrity, is studied in the next chapter.

The efficiency of the algorithm can be observed by looking at the resulting behavioral strategies. Examples of behaviors observed in both the free-run and energy setups are shown in Figures 5.4

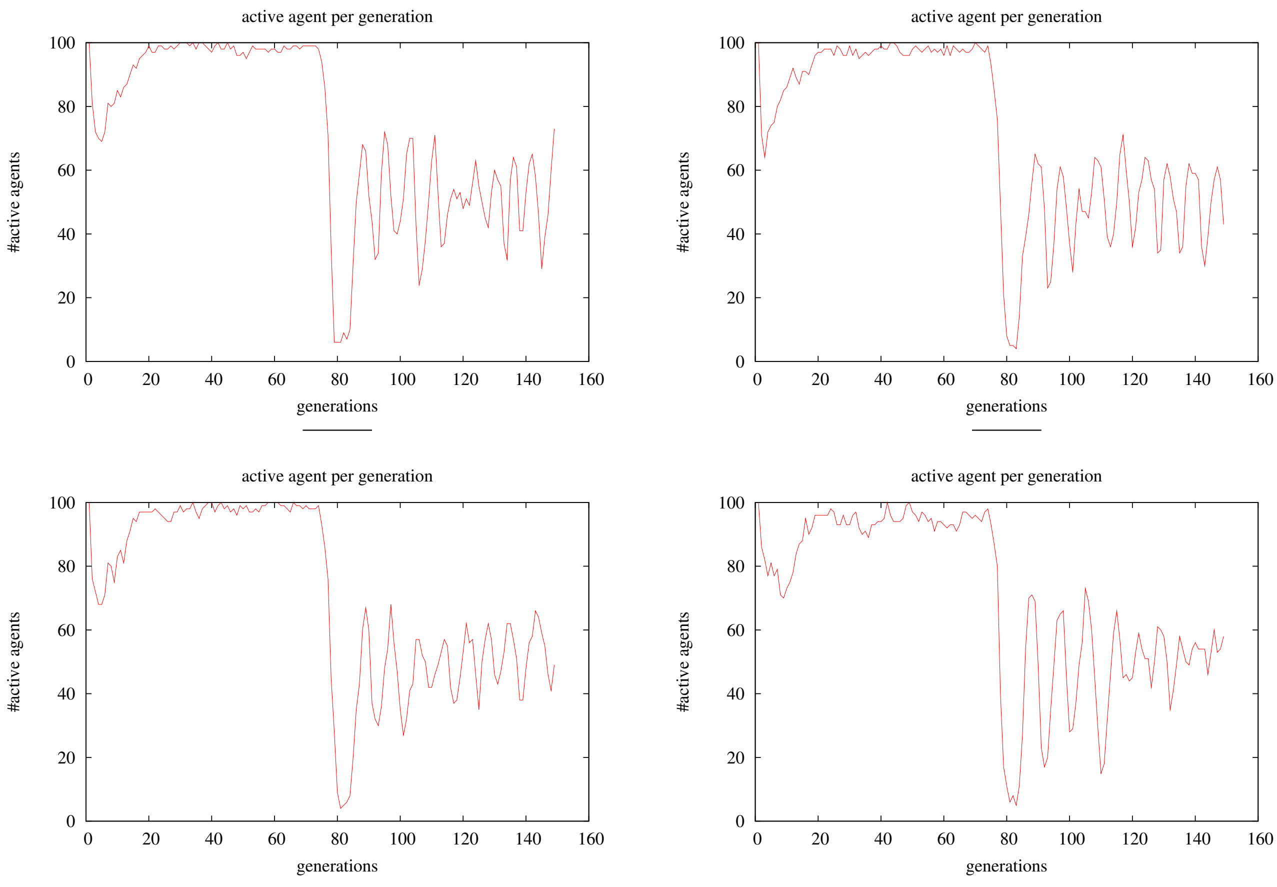


Figure 5.3: Experimental results for four (randomly chosen) specific runs.

and 5.5, resulting from agents driven by genomes obtained in the late generations of both setups. In the "free-run" setup, genomes tend to lead to rather conservative behaviors, with obstacle avoidance but limited exploratory behavior. On the other hand, genomes from the later generations of the "energy" setup show a different behavioral pattern, favoring long distance travels or behaviors as circling. This is an efficient strategy to avoid being stuck in a depleted area. Moreover, a closer look at trajectories (including, but not limited to what is shown here) show that agents acquired the ability to drive toward detected food items under certain conditions, such as favoring safe areas with few obstacles whenever energy levels are low. The middle picture of each figure shows exception to the most commonly observed behaviors. The next section presents a quantitative analysis of behaviors' efficiency evolved on both setups.

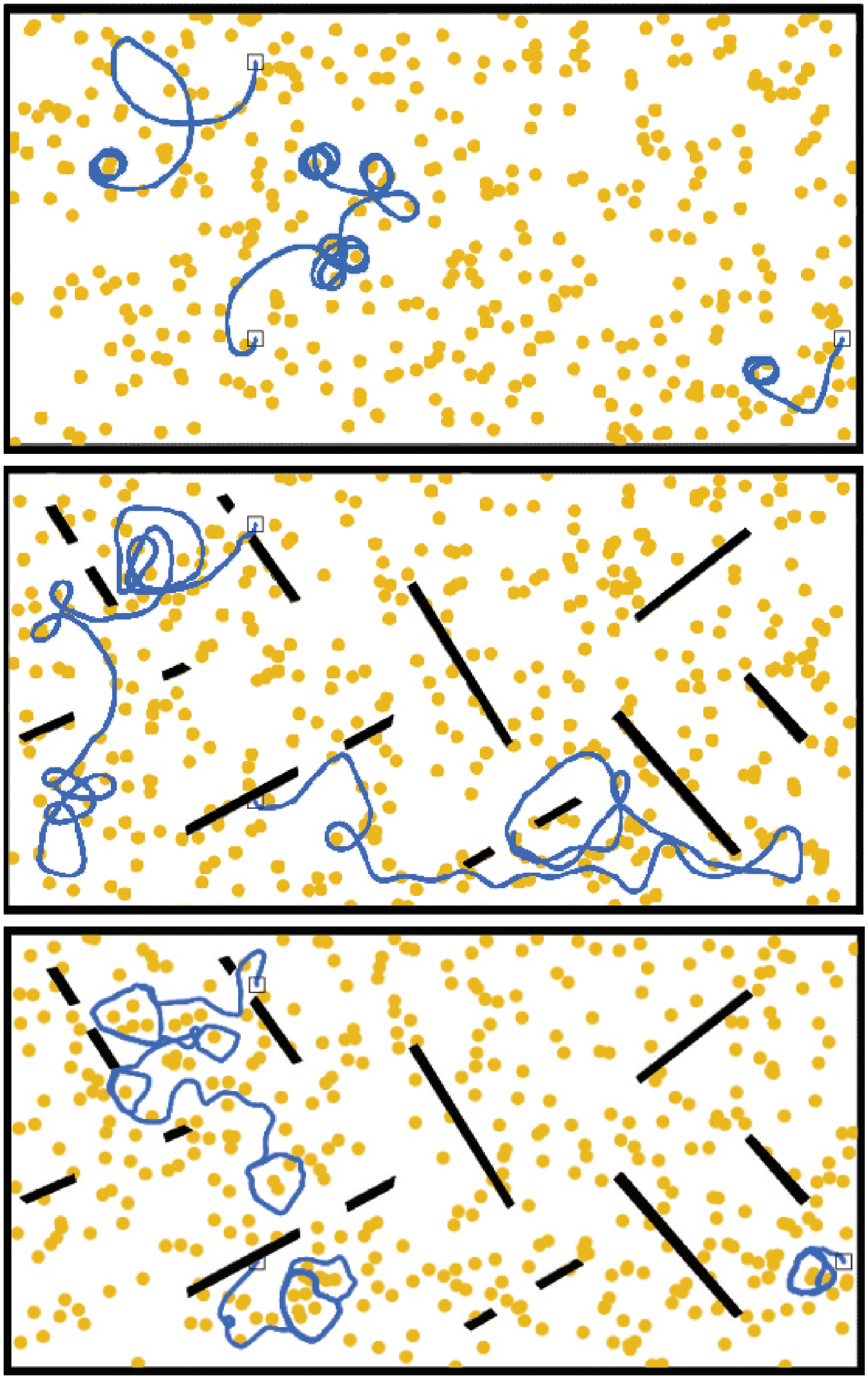


Figure 5.4: Typical examples of agents' behavioral traces in the free-run setup. The square symbol shows the agent starting points. Agents are tested in environments with or without walls (i.e. black bars).

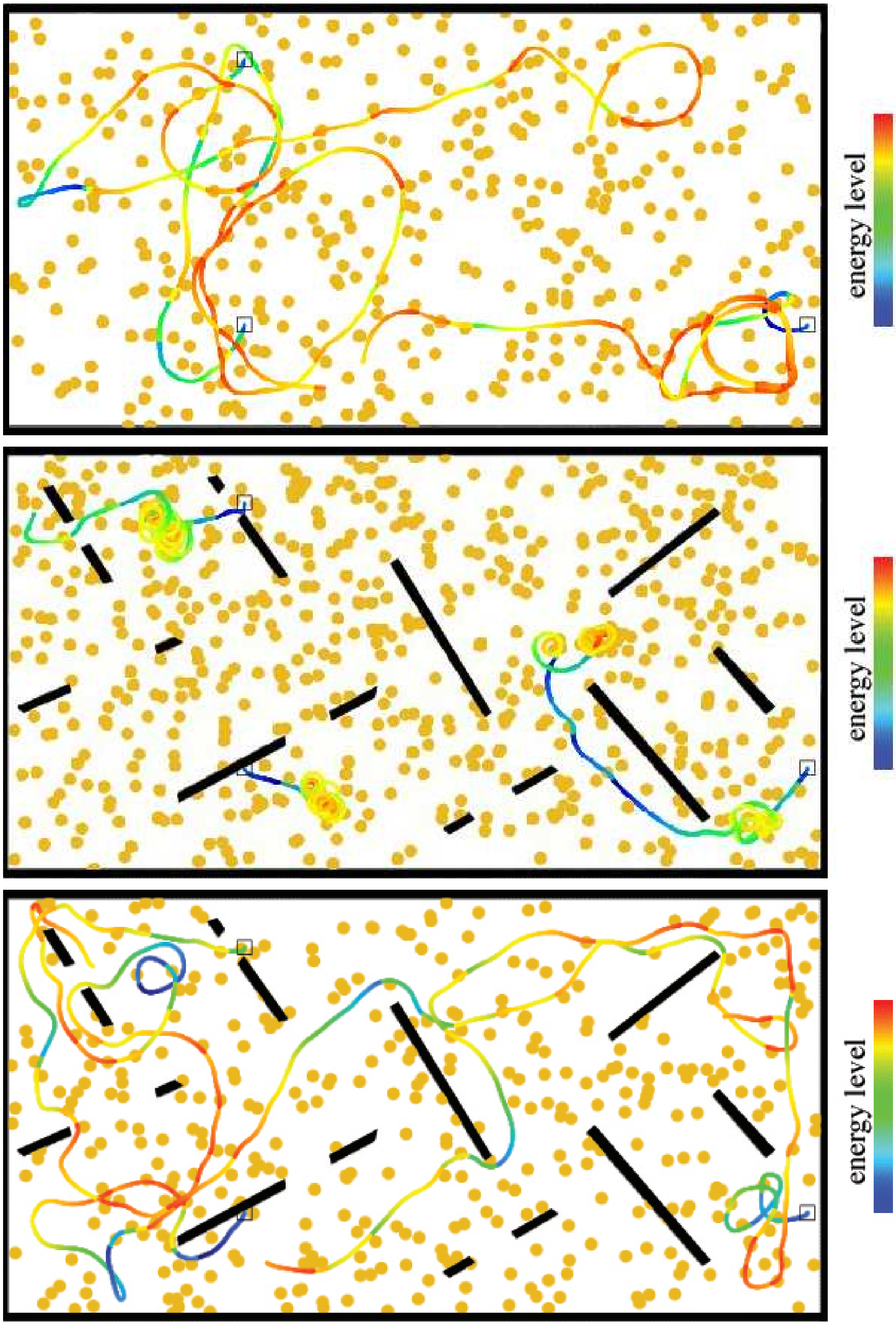


Figure 5.5: Typical examples of agents' behavioral traces in the energy setup. The coloured traces account for the current energy level of the agent (see legend). The square symbol shows the agent starting points. Agents are tested in environments with or without walls.

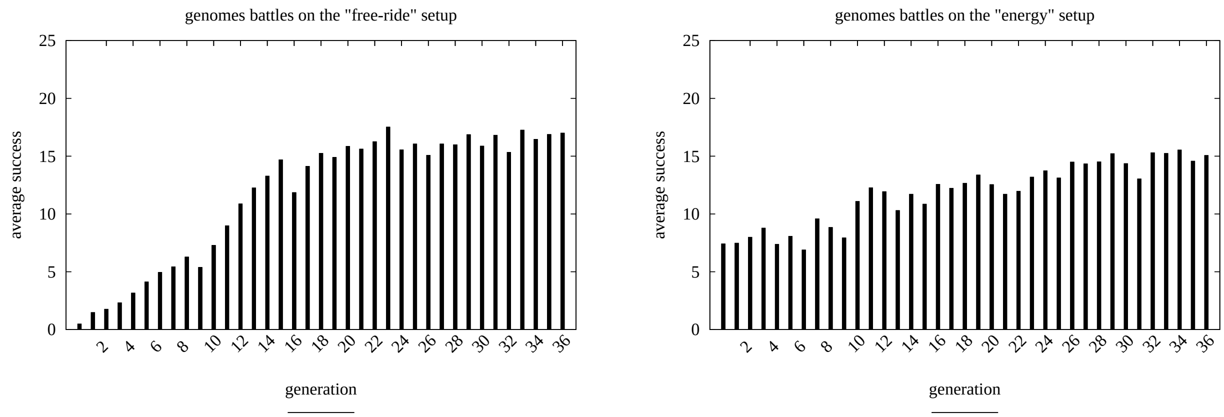


Figure 5.6: Genome battles on both setups (*(1)* free-run setup ; *(2)* energy setup). Average success scores for each generation – both histograms are the results of 1000+ battles. See text for details.

5.2.3 Quantitative Analysis

The results presented above must be interpreted with great caution. For example, the quality of the equilibrium between maximizing mating opportunity and coping with environmental constraints (i.e. avoiding walls, avoiding collisions with other agents and harvesting) is difficult to estimate as such an equilibrium may (and appears to) imply sub-optimal values for both related indicators. In fact all interpretations provided so far rely on the assumption that values monitored in the experiment are actually correlated with genome survival.

In order to test this assumption an experimental setup is defined from the results obtained so far. This setup is called the *post-mortem* battle experiment, or battle experiment, for short. The battle experiment is loosely inspired by competitive coevolution, where each individual competes against a hall-of-fame of the best individuals from every past generation [Rosin and Belew, 1996]. This is used to estimate *a posteriori* the fitness rank of one individual (by a comparison mechanism) within all possible (or at least, all available) situations.

For the current experiment, one "battle" is achieved by randomly picking 10 generations from the same setup and extracting one random genome from each of these generations. Then, each genome is copied into ten different agents, resulting in 100 agents that are immersed in the same setup they evolved in. Variation is turned off, and evolution is re-launched. After 100 generations of random selection and replacement the number of copies of each genome is accounted for and used to compute a "survival score". It is important to note that only the performance of the original genome is measured, since the mutation is turned off. As an example, one genome gets a maximal score if it succeeds in taking control of all the agents.

Average results over 1000 battles are given in Figure 5.6. In both setups genomes from later generations display better survival than early genomes. Moreover, battles on the second setup show the population recovers very quickly following environmental change, possibly to stable but limited strategies as the number of active agents is far from the maximum. Also, these histograms lack the misleading artifacts observed in previous graphs regarding the early generations in both setups: genomes from generation 0 do not benefit anymore from uniform sampling of starting location, and genomes from generation 75 start with the same initial energy level as genomes from every other generation.

5.3 Summary and Conclusions

In this chapter, the robustness of the mEDEA algorithm has been assessed in front of unpredicted changes in the environment. Analysis of evolutionary dynamics have illustrated that the selection pressure foreseen in Chapter 4 effectively favors the selection of more efficient individuals.

Surviving in aggressive environments may require more complex behavioral patterns than the ones studied here (such as coordination). Sharing similar concerns, previous works in collective intelligence and reinforcement learning have already stressed the issue of the price of anarchy [Wolpert and Tumer, 2001], ie. the cost of efficient selfish behavior with regards to population global welfare. Addressing this issue remains an open problem, especially if there is no explicit objective function to decompose. In the next chapter, the possibility to evolve cooperative behaviors with mEDEA is tested.

Chapter 6

Evolution of Altruistic Behaviors

Altruistic behavior stresses the sacrifice of one agent for the benefit (in terms of survival) of other agents, without short term benefit for the donor (during its lifetime). As such, it is a non trivial behavior reducing the individual welfare of an altruistic agent in order to enhance the population's welfare. Therefore the evolution of altruistic behaviors, notably by populations facing adversarial environments, is an important problem for algorithms targeting the population's integrity. This chapter aims at answering the following question: Is it possible to evolve altruistic behaviors with mEDEA ?

In order to study the evolution of altruism in mEDEA, a particular setup is presented which is inspired from the Tragedy of the Unmanaged Commons. "The tragedy of the Unmanaged Commons" is the consequence of the overexploitation of a common resource by a group of agents.

Altruistic behaviors and the tragedy of unmanaged commons are presented in more details in Section 6.1. The methods used to study the evolution of altruism are presented in Section 6.2. Finally, the evolution of altruistic behaviors by the mEDEA algorithm is studied in Sections 6.3 and 6.4.

6.1 Introduction

6.1.1 Definitions

In the literature, the terms cooperation and altruism are often found side by side. Both phenomena have been the focus of a particular attention from many research fields, including Game Theory [Fletcher and Zwick, 2007] and Biology [Piliavin and Charng, 1990]. Both cooperative acts are performed either thanks to a social act (e.g. slime molds sacrificing themselves to promote the reproduction of others when resources are scares), or the refraining from selfishness (e.g. competition for light results in plants investing in growth rather than in productivity) [Rankin *et al.*, 2007].

However, a clear distinction is drawn between cooperation with mutual benefit[1] [West *et al.*, 2007] and "strong" altruism (termed altruism from now on) depending on the nature of the fitness benefit at the level of *either* the individual *or* the population [Lehmann and Keller, 2006b].

[1] Cooperation is also sometimes used as a synonym for altruism (e.g. cooperation in the prisoner's dilemma corresponds to altruism [Sober, 1992]). In this chapter, we assume the restricted and well-accepted definition of cooperation as a behavior leading to mutual benefit.

Cooperation. Cooperation implies that a given individual benefits from its behavior during its lifetime (called direct benefit), either through direct or delayed (i.e. through repeated interactions) reciprocity. In these scenarios, the rise of free-riders within the population can be hindered thanks to the following mechanisms [Lima, 1989; West *et al.*, 2007]:

- **Reciprocity:** When agents use the reciprocity mechanism a cooperation between two agents is maintained as long as both agents continue to perform cooperative actions. As soon as one of the two agents behave selfishly, the other agent will retaliate by displaying selfish behaviors. This approach is used to favor the emergence of cooperation in scenarios where interactions between individuals are repeated without finite time horizon. It is at the core of the tit-for-tat strategy [Axelrod and Hamilton, 1981], and has been shown to evolve in artificial systems under specific conditions [Trivers, 1971; Lima, 1989; Nowak and Sigmund, 1993; Suzuki and Akiyama, 2007].

- **Punishment:** The repression of free rider's within groups is another mechanism proposed to increase the level of cooperation between agents [Frank, 2003]. Within this strategy, cooperating individuals have the opportunity to impose a fine to selfish ones. Therefore the evolution of cooperation relies on the knowledge of the previous agent's actions. Different works on this approach are reviewed in [Lehmann and Keller, 2006a].

Biological experiments on the evolution of cooperation are rare because of their length and difficulty to set up. It is for example impossible to study various payoffs, or different fitness mechanisms. However experiments on biological systems don't suffer from modeling issues, and offer first hand perspectives on the emergence of cooperation in nature [Dugatkin and Wilson, 1992].

Altruism Altruism characterizes the sacrifice of (part of) one own's fitness for the benefit of others. Therefore, an altruistic behavior benefits other individuals and possibly has a positive impact on longer time-scale (e.g. more than a single lifetime). This can be illustrated by the death of Amoeba cells in order to favor the reproductions of others [Queller *et al.*, 2003]. In altruistic contexts the mechanisms enforcing social behavior don't rely on lifetime benefit, but on the longterm genetic advantage. Interactions between agents sharing altruistic genes are enforced in order to increase the survival rate of these genes.

This idea has been formalized under the name inclusive fitness proposed by [Hamilton, 1964]. It is now widely accepted to account for the emergence of altruism: inclusive fitness considers the fitness of a particular individual to depend both on its own success and the success of its close relatives. The basic idea is to consider individuals as vehicles for genes. Therefore the welfare of a gene is considered rather than the sole interest of one individual/vehicle. Of course, sacrificing one's vehicle depends on several parameters such as the expected fitness loss (from sacrifice) and benefit (for other copies of the genome) as well as the genotypic relatedness of the individuals concerned (closer relatives may imply increased altruistic behaviors).

Hamilton formalized the relationship between cost, benefit and relatedness in the following equation:

$$C/B < r \tag{6.1}$$

The cost C is the amount of fitness lost by an altruistic individual. The benefit B is the amount of fitness gained by the recipient that benefits from the altruistic behavior. And r is the genotypic relatedness between the two individuals. This equation has been extensively studied since its first introduction in 1964, and the methods to compute its parameters often discussed [Queller, 1992]. [Lehmann and Keller, 2006a] discuss this point based on recent contributions.

Multiple mechanisms favoring the evolution of higher level of altruism have been proposed since the pioneering work of Hamilton. All these mechanisms aim at increasing the likelihood of altruistic actions between closely related individuals. We are presenting here a partial review of the most recognized mechanisms.

Kin selection. The term kin selection has been introduced by [Maynard Smith, 1964] to describe the mechanisms maximizing the inclusive fitness of an individual. If an individual has a gene for altruism, its kins are likely to have it as well. Therefore, an altruistic individual willing to sacrifice itself for closely related individuals may spread its gene for altruism through natural selection.

[Sober, 1992] studies the impact of cost, benefit and population size on the evolution of altruism in the iterated prisoner's dilemma. From these experiments, multiple conditions are presented, recalling the one found in the hamilton's rule. Multiple aspects of this mechanism have since then been studied in details, such as the impact of time [Eshel and Shaked, 2001], and group sizes [Leticia *et al.*, 2004]. Moreover, this mechanism has also been studied in off-line ER by varying the genetic team composition and level of selection in public good dilemma experiments [Waibel *et al.*, 2009]. Finally, mechanisms aiming at detecting kin are named kin recognition (cognitive process) and kin discrimination (observable behavioral pattern) [Byers and Bekoff, 1986; Ostrowski *et al.*, 2008]. These mechanisms increase the effect of kin selection in the evolution of altruism, even if their utility in Biology is discussed [Tang-Martinez, 2001; Mateo, 2004].

Group selection. The group selection mechanism is used to increase the likelihood of interactions between individuals with high level of altruism, without relying on the genotypic relatedness [Wynne-Edwards, 1986]. Groups are created randomly between altruistic and egoistic individuals and individuals can only interact between members of the same group. After a fixed amount of time, the groups are dissolved and new random groups are formed. If groups are maintained too long, altruistic individuals will be wiped out. However, if specific constraints are respected, groups with higher ratio of altruistic individuals will grow faster than other groups. By reforming groups regularly, the most altruistic agents will benefit from the growth of their group.

This theory is debated due to its similitudes with the kin selection mechanism, and its lack of realism with regards to biological systems. A review of the previous contributions to the study of this mechanism is given in [Frank, 2003], and the presentations of arguments relative to the surrounding controversy are presented in [West *et al.*, 2007]. Currently, it is argued that this mechanism is equivalent to the kin selection mechanism [Queller, 1992; Rousset and Ronce, 2004].

Tag recognition. The tag recognition mechanism is based on the perception of genotypic characteristic through phenotype's features [Holland, 1996]. Thanks to this association, individuals with similar inclination toward altruistic behavior can recognize each other, and have therefore a selective advantage to engage in altruistic behavior with each other. This mechanism is proposed as an explanation to high level of altruism in simple creatures lacking the capacity to recognize kins. Two similar instantiations of this mechanism have been proposed: the green beard effect (relying on sight), and the armpit effect (relying on smell).

The green beard effect has been described by [Hamilton, 1964] and famously named in [Dawkins, 1976]. Within this scheme two individuals recognize the cooperative tendency of the other thanks to a visible feature (a green beard for example). The armpit effect is a similar mechanism dubbed in [Dawkins, 1982] which doesn't rely on mere preference toward individuals having a specific feature, but toward similar individuals. In the thought experiment presented, Dawkins imagines animals using the odor of their own armpit as a basis to decide with which individuals they will collaborate. The closer the other individual armpit's odor, the more likely the cooperation.

One of the first demonstration of tag mechanisms in a virtual environment has been proposed in [Nowak and Sigmund, 1998]. Since then, these mechanisms have been demonstrated in multiple contexts [Riolo *et al.*, 2001; Antal *et al.*, 2009], and their stability in virtual environments has been demonstrated [Roberts and Sherratt, 2002; Spector *et al.*, 2004; Spector and Klein, 2006]. Finally, both of these mechanisms have been observed in biological systems [Queller *et al.*, 2003; Haig, 1996; Keller and Ross, 1998; Hauber *et al.*, 2000; Mateo and Johnston, 2000; Isles *et al.*, 2001].

Environment viscosity. The viscosity is an environmental parameter impacting the dispersion of agents (e.g. moving speed). Therefore an increased viscosity will result in the presence of related individuals close to each other, and thus enforce kin-selection. For example when goods produced by altruistic individuals benefits locally to other individuals, an increased viscosity will favor the creation of cluster of altruistic agents. This mechanism originally presented in [Hamilton, 1964], has been demonstrated in multiple works [Wilson, 1980; Queller, 1985; Wilson, 1987; Wilson *et al.*, 1992; Mitteldorf and Wilson, 2000]. However [Hauert and Doebeli, 2004] has recently criticized this approach by stating that small clusters can be more favorable for the emergence of altruism in specific setups.

6.1.2 The Tragedy of the Unmanaged Commons

The Tragedy of the Unmanaged Commons [Hardin, 1968; Hardin, 1994] is a particular kind of social dilemma arising when a population of individuals has access to a finite common resource pool. In this context, each individual may temporarily increase its benefit through selfish behavior, but this inevitably leads to exhaust the common resource pool, ultimately ending with population extinction. For concerned individuals, one possible way to address this tragedy is to display altruistic behaviors.

The classic example features a pasture open to all, and herdsmen seeking their own interest. By owning as many cows as possible, each herdsman increases its personal benefit without regards for the common pasture the cows harvest. This situation quickly leads to the overexploitation of the pasture, ending with cows dying from starvation. Pastures are only one out of many scenarios where the Tragedy of the Unmanaged Commons can happen: fish stocks and timber are other well-known study scenarios.

The Tragedy of the Unmanaged Commons has been widely studied in Evolutionary Biology [Rankin *et al.*, 2007], Economics [Mankiw, 2009], and Historical geography [Diamond, 2005]. From a formal point of view, the Tragedy of the Unmanaged Commons appears in scenarios featuring a common good, which is characterized by two properties:

- **Non-excludable access:** The resource is freely accessible to anyone. There is no physical restriction to access the resource, and no cost is associated to its access.

- **Rivalrous access:** Once the resource has been consumed by an individual it doesn't exist anymore. However, this resource can regenerate, in the right conditions, and after a specific time.

The combination of these two properties imply a competition among individuals to access the resource. A hypothesis to explain the tragedy of the unmanaged common is the presence of free-riders [Rankin *et al.*, 2007]. Free-riders are selfish agents, taking advantage of the altruistic behavior of other agents. Since they are favored by natural selection, they out-grow altruistic agents and soon become the only type of agent in the population. At this point, the presence of only selfish agents will result in the overexploitation of the common good, leading to the Tragedy of the Unmanaged Commons.

In economics the solutions proposed to the Tragedy of the Unmanaged Commons are ranging from two extremes: the privatization of the common good, and the public management of the common good [Hardin, 1994]. The Tragedy of the Unmanaged Commons can also arise in evolutionary scenarios where agents compete for a resource in order to improve their fitness. Altruistic behavior has been proposed as a solution to these scenarios [Dionisio and Gordo, 2006].

6.2 Experimental Setup: Implementing the Tragedy of the Unmanaged Commons

In the remaining of this chapter, we will look for answers to the following questions:

- Is it possible to evolve altruistic behaviors in MEDEA as a response to a scenario where the Tragedy of the Unmanaged Commons is bound to occur ?
- What are the important properties to evolve altruism by the MEDEA algorithm ?
- What are the mechanisms responsible for the evolution of altruistic behaviors ?
- Is it possible to impact the level of altruism by modifying the local interactions between agents ?

6.2.1 Setup

In order to account for the existence of altruism, a foraging task has been defined where a population of autonomous agents must eat food items to maintain a positive energy level. The experimental setup used in this chapter is illustrated in Figure 6.1, with food items (circles), agents (small dots) and obstacles. The environment and task depends on the following elements:

- **Self-sustainability:** foraging is necessary to survive, as each food items gives a small amount of battery energy. However, an agent's battery is limited to a maximum amount of energy, and foraging may end up in wasting resource.
- **Foraging behavior:** an agent may choose to harvest *all* or *part* of a food item.
- **Re-grow rate:** whenever a food item is harvested, it is removed from the environment until it grows back after some delay. The time to re-grow depends linearly on the quantity of energy harvested from the food item.

As a consequence, the environment features a common good (non-excludable but rivalerous access) for which agents compete. In other words, in this setup the Tragedy of the Unmanaged Commons is bound to occur. The probability to observe the Tragedy of the Unmanaged Commons is dependent of the delay needed by a food item to grow back. A short re-grow delay will result in a high level of food available (low probability to observe the Tragedy of the Unmanaged Commons), while a long re-grow delay will result in scarce resources (high probability to observe the Tragedy of the Unmanaged Commons).

This is achieved by setting the **maximum** re-grow delay for a food item ($EP_{Lag_{Max}}$, with EP as in "Energy Point"), which in turn will be used to compute on-the-fly the re-grow delay of a food item that was just harvested (EP_{Lag}). This is described in Equation 6.2, which also takes into account the amount of energy harvested by an agent from the food item ($E_{harvested}$) and the amount of energy available in each food item ($EP_{e_{Max}}$). On the one hand, in the case where $EP_{Lag_{Max}}$ is equal to the duration of an evaluation, the re-grow delay of a food item (EP_{Lag}) will range between 0 (if no energy is harvested from the food item), to one generation (if all energy is harvested from the food item). On the other hand, a short $EP_{Lag_{Max}}$ will result in a smaller range of variations of EP_{Lag}.

$$EP_{Lag} = E_{harvested}/EP_{e_{Max}} * EP_{Lag_{Max}} \tag{6.2}$$

Within this setup, it is expected that altruistic agents in difficult environments shall harvest the minimum amount of energy from each food items, therefore increasing the availability of the resource (short re-grow delay, no wasted energy). On the other hand, selfish behaviors are likely to be fitted for small values of $EP_{Lag_{Max}}$, but are expected to become more and more critical as the value of $EP_{Lag_{Max}}$ increases.

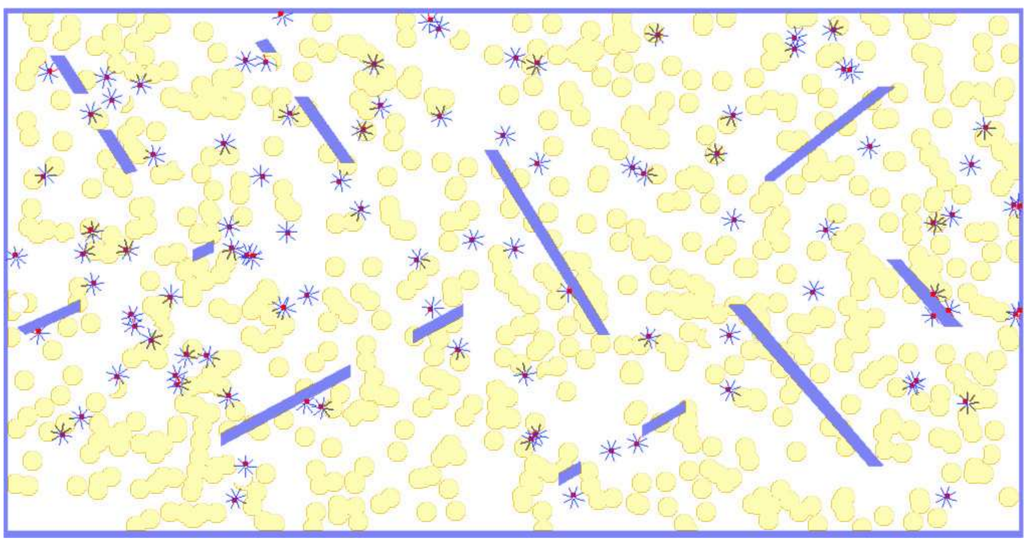


Figure 6.1: Snapshot from the simulator: food items (circles), agents (dots) and obstacles

6.2.2 Methodology

All experiments are conducted with 100 robotic agents in the environment described and illustrated in the previous section. All agents are using the mEDEA algorithm (see Chapter 4). The environment contains 800 food items and an agent may harvest a maximum of 50 units from a food item. Each agent consumes 1 unit of energy per step, and can store up to 800 energy units (harvesting surplus is lost). If the agent's battery level drops to zero, the agent stops and its genome is lost. It remains then inactive for one generation without neither recording any genome nor transmitting any information.

Parameter	Value
arena width and length	$1024 * 530$ pixels
number of energy points	800
size of energy points	10
maximum energy from energy points	50
maximum energy in one robot	800
lifetime (i.e. generation duration)	400 steps per generation
population size	100 agents
proximity sensor range	64 pixels
radio broadcast signal	32 pixels
agent rotational velocity	0.52 rad/s
agent translational velocity	2 pixels/step
genome length	79 real values (78 MLP weights + σ)
variation operator	Gaussian mutation with σ parameter
$\sigma_{minValue}$	0.01
$\sigma_{maxValue}$	0.5
$\sigma_{initialValue}$	0.1
α (ie. σ update parameter)	0.35

Table 6.1: Parameters for experiments.

This is done to prevent the survival of poorly efficient strategies switching from one agent to another. The agent then switch to the listening state, for one generation, during which it doesn't move but records the genomes broadcasted within radio range by other robots. If the agent has received more than one genome during the listening state, a new controlling genome is chosen thanks to the random selection mechanism. Otherwise, the agent remains in listening state for one more generation. To sum it up, this is the exact same algorithm as was presented in the previous chapter.

The control architecture is a Multilayer Perceptron (MLP) with 5 hidden neurons, 11 inputs (8 proximity sensors, battery level and orientation/distance to the closest food item) and 3 outputs (left/right motor and proportion of energy to be harvested from a food item, if any). The weights of the MLP are decoded from the active genome of the agent. Each agent broadcasts a mutated copy of its own genome and receives genomes from neighbors within a limited range (roughly $1/10th$ of the length of the larger side of the environment). The mutation operator used in the mEDEA algorithm is defined as a gaussian mutation with a σ parameter. σ is included into the genome (i.e. similar to a self-adaptive Evolution Strategy) and ranges from 0.01 (low mutation rate) to 0.5 (large mutation rate), as presented in Section 4.2.2. Detailed parameters used for experiments presented here are given in Table 6.1.

6.2.2.1 Measuring altruism

In order to account for altruism, a measure for monitoring the *cost of altruism* for one foraging agent is defined. This corresponds to measuring the amount of energy that *could be* consumed when harvesting a food item, but which is actually *not consumed* by the agent. This is formally defined in Equation 6.3.

$$Cost = max(0, min(EP_{e_{Max}}, r_{E_{max}} - r_{E_{now}}) - E_{harvested}) \tag{6.3}$$

Where $EP_{e_{Max}}$ is defined as before (i.e. maximal energy in a food item), $r_{E_{max}}$ is the maximal energy level of an agent, $r_{E_{now}}$ is the current energy level of the agent and $E_{harvested}$ is the energy harvested by the agent from the food item.

While a selfish agent shall have a cost of zero, an altruistic agent should be able to perform a trade-off between its altruistic nature and its survival needs. Therefore, the cost of altruism can be seen as the agent's level of sacrifice which is continuous (a quantity of energy) rather than discrete (eat or dont eat).

6.2.2.2 Characterizing agents' behaviors

In order to characterize the behavior of the evolved behaviors (independently of their level of altruism), the area covered by one agent in one generation is measured. By this way a numerical value is associated to the movements performed by the agents. Thanks to this the variations in agents behaviors are observed for experimental setups.

The agent movement is monitored within the environment, by counting the number of location visited. To do so, the environment is divided into cells, of resolutions 4x4 pixels. At the end of a generation, an approximation of the area covered by an agent is computed thanks to the Formula 6.4 (the formula is applied only on the grid of the concerned agent).

$$AreaCovered = \frac{\#VisitedCells}{\#Cells} \tag{6.4}$$

6.2.2.3 Fixed levels of altruism

In order to investigate the impact of different level of altruisms an alternative setup is designed where the level of altruism is fixed beforehand. In practice, this is done by assigning the energy harvested by an agent regardless of the output of its neural network. The energy harvested is linked to a fixed cost of altruism set for the course of the experiment. Based on the fixed cost ($Cost_{Fixed}$) the energy harvested ($E_{harvested}$) is computed thanks to the Formula 6.5.

$$E_{harvested} = max(0, min(EP_{e_{Max}}, r_{E_{max}} - r_{E_{now}}) - Cost_{Fixed}) \tag{6.5}$$

This formula is deduced from Formula 6.5, and the same notations are used.

6.3 Natural Emergence of Altruism

In this section we study the behavior evolved by mEDEA under different pressures of the environment. The goal is to evaluate whether the mEDEA algorithm is capable of evolving altruistic behavior, and to study the impact of environmental pressure on such evolutionary dynamics. A large set of experiments was performed under various environmental pressures by setting a specific value of $EP_{Lag_{Max}}$ for each run, ranging from 25 steps to regrow ("easy" environment) to 400 steps to regrow ("difficult" environment), for a total of 16 setups. For each setup (i.e. a fixed value of $EP_{Lag_{Max}}$), at least 200 independent runs were performed (total of more than 3200 experiments) and the results were aggregated to extract two indicators: the number of active agents, and the cost of altruism. In all experiments, the course of evolution is similar: the number of active agents quickly increases to a stable value while costs start from random values and stabilize to (possibly) positive values. While the increasing number of active agents is expected from evolutionary adaptation, the second observation is of primary importance regarding the possibility of altruistic behavior: a positive cost of altruism would imply that agents do not systematically harvest all possible energy from the food items.

Results are summarized in Figures 6.2, 6.3(a) and 6.3(b) (resp. ratio of extinguished runs, number of active agents, and cost of altruism). A run is classified as *successful* if more than one agent is active at the 400000^{th} iteration (the last one), that is to say after 1000 theoretical generations. A theoretical generation is a generation in the classical sense of Evolutionary Algorithms (when the replacement of an old population is done synchronously). The number theoretical generations is obtained by dividing the number of iterations by the length of one lifetime. When the number of active agents drops to 0 before the last iteration, the run is labeled as *extinguished*. In order to obtain comparable results the same number of successful runs has been generated for each environmental pressure. However, for high environmental pressures, most of the runs are extinguished. The Figure 6.2 shows the number of extinguished runs, divided by the number successful runs. The Figures 6.3(a) and 6.3(b) are produced by taking into consideration the last 10 generations of all successful runs for each setup (i.e. after convergence to stable behaviors).

With stronger environmental pressures (larger values of $EP_{Lag_{Max}}$), the number of active agents decreases ($p - value < 2.2 * 10^{-16}$ between $EP_{Lag_{Max}} = 25$ and $EP_{Lag_{Max}} = 400$), which confirms that the environment is becoming more and more challenging. Moreover, the number of extinguished runs is increasing with the environmental pressure, which shows the Tragedy of the Unmanaged Commons: when the amount of food is reduced, the risk to deplete completely the resource is increased. The next question is to investigate the behaviors evolved by the surviving populations.

Altruistic behavior in the context of increasing environmental pressure can be observed by looking at the cost of altruism, which converges to a stable value, while the number of active agents decreases and the extinction rate increases (i.e. the limit for survival). Indeed, altruistic behaviors are first observed with environment pressure $EP_{Lag_{Max}} = 100$ ($p - value = 0.01857$ between $EP_{Lag_{Max}} = 25$

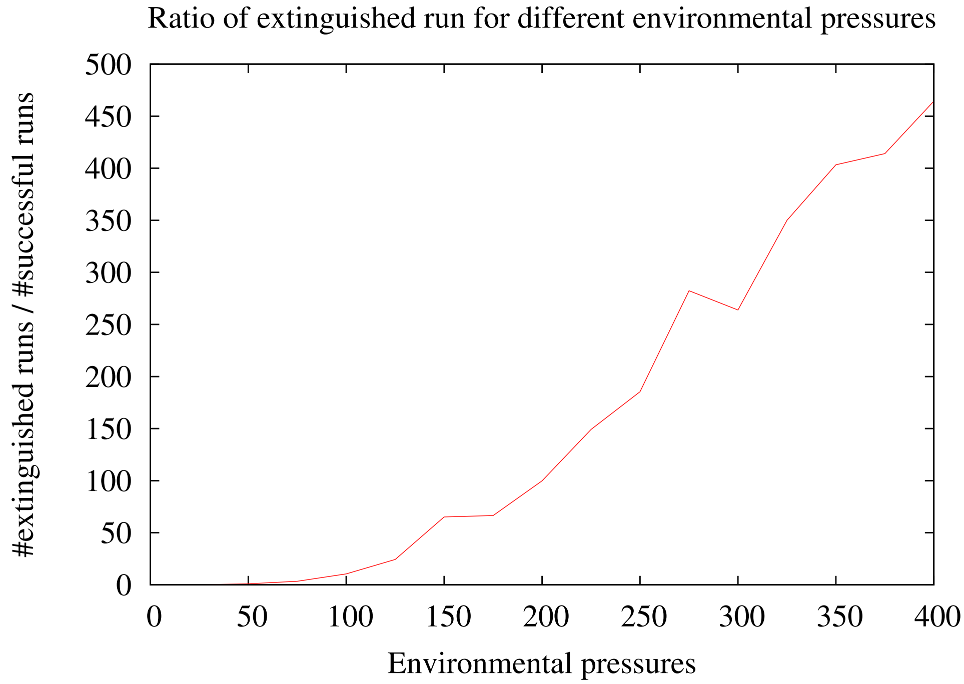


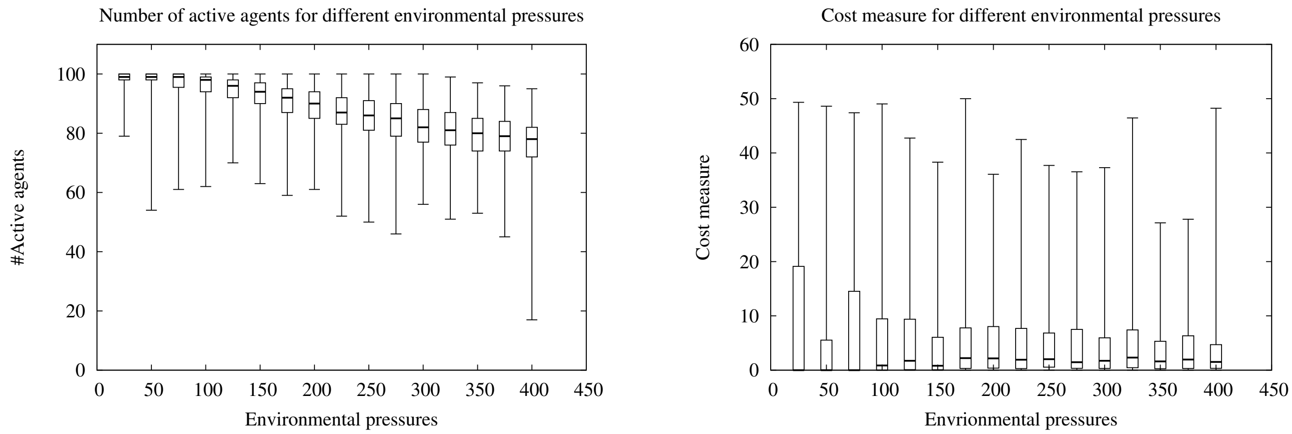
Figure 6.2: Results with $EP_{Lag_{Max}}$ between 25 and 400: Extinction rate (data: value from each run)

and $EP_{Lag_{Max}} = 100$), and remains constant afterwards ($p{-}value = 0.05865$ between $EP_{Lag_{Max}} = 100$ and $EP_{Lag_{Max}} = 400$). Altruistic behaviors are difficult to observe when environmental pressure is low and the Tragedy of the Unmanaged Commons not bound to occur (extinction rate is close to its minimum for values of $EP_{Lag_{Max}}$ under 100 iterations).

The greedy nature of the algorithm has been shown in easy environments: without environmental pressure, altruism does not emerge spontaneously. This shows a link between environmental pressure and the emergence of altruism. However, altruistic behaviors remain stable in the population even though the environmental pressure increases and the number of active agents starts to drop, implying limited correlation between the level of altruism and environmental pressure. These observations imply that some dynamics of mEDEA promote greedy strategies, while others linked to the environmental pressures (but not only) favor altruistic strategies.

6.3.1 Effect of Environmental Pressure

The characterization of the impact of evolved altruistic on the integrity of the group of robots, is necessary to further understand the phenomenon highlighted above. The harvesting strategy is one degree of freedom in the search space that can be exploited by the mEDEA algorithm. However, its importance with regards to the integrity of a group of robots remains to be evaluated. To investigate this, results obtained when mEDEA explicitly "control" part of the harvesting strategy (i.e. the



(a) (b)

Figure 6.3: Results with $EP_{Lag_{Max}}$ between 25 and 400: *a)* Number of active agent (data: value from each run) ; *b)* Cost measure (data: median values from each run)

amount of restrain in the harvesting is evolved), are compared to results obtained by using mEDEA when the amount of restrain in the harvesting is fixed (i.e. fixed cost of altruism). Even if the harvesting strategy is fixed other degree of freedom can be exploited by mEDEA to evolve altruistic behaviors. For example, agents can intentionally avoid food items.

Experiments are run in 14 different setups: 13 using fixed costs of altruism (from egoist ($Cost = 0$) to extremely altruist ($Cost = 45$)), and one using an evolved harvesting strategy (as presented before). In order to observe results on a large range of environmental pressures experiments are run in environments where the pressure is at first easy ($EP_{Lag_{Max}} = 25$ *iterations*) until the 400000^{th} iteration, and then continuously increasing by step of 80 iterations every 4000 iterations (10 theoretical generations) until the extinction of the population. That is to say, at the 404000^{th} iteration the environmental pressure is equal to $EP_{Lag_{Max}} = 105$ *iterations* , and at the 408000^{th} iteration the environmental pressure is equal to $EP_{Lag_{Max}} = 185$ *iterations*. Only runs where extinctions occurred after the 400000^{th} iteration are taken in consideration. For each setup, at least 500 runs were performed (total of more than 7000 runs).

The Figure 6.4 shows the ratio of successful runs (number of successful runs divided by the total number of runs) for every environmental pressure, and every cost strategies. The Figure **??** shows the ratio of successful runs measured between environmental pressure $EP_{Lag_{Max}} = 5000$ *iterations* and $EP_{Lag_{Max}} = 8100$ *iterations*.

The benefit of fixed harvesting strategies for environmental pressure lower than $EP_{Lag_{Max}} = 4000$ *iterations* is observed in Figure 6.4: evolving the harvesting strategy is one of the worst strategy tested (one of the strategy for which the ratio of successful run is the lowest). However, after the

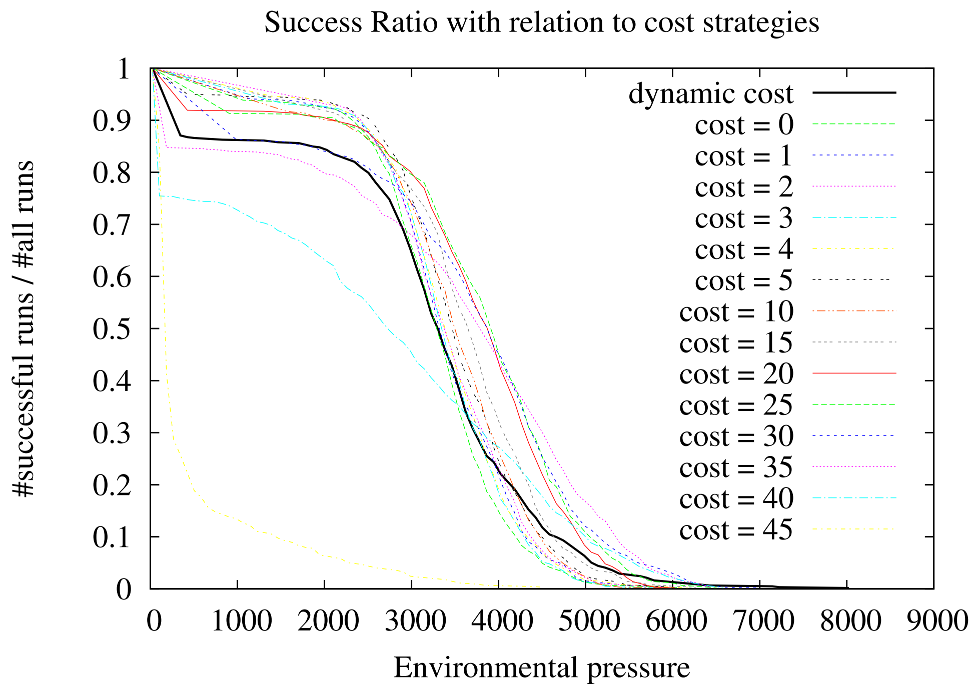


Figure 6.4: Ratio of successful runs for different harvesting strategies (evolved and *Cost* between 0 and 45) measured when the pressure of the environment increases by step of 80 iterations every 4000 iterations (from an easy environmental pressure, i.e. $EP_{Lag_{Max}} = 25\ iterations$)

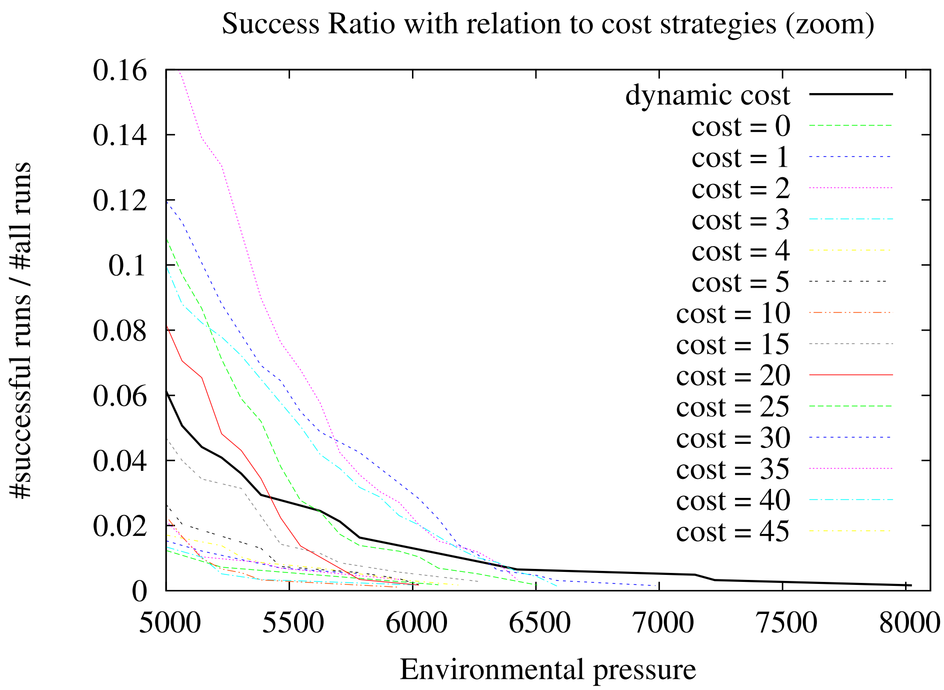


Figure 6.5: Ratio of successful runs for different harvesting strategies (evolved and $Cost$ between 0 and 45) measured when the pressure of the environment increases by step of 80 iterations every 4000 iterations (from an easy environmental pressure, i.e. $EP_{Lag_{Max}} = 25\ iterations$. Presentation of the ratios measured between $EP_{Lag_{Max}} = 5000\ iterations$ and $EP_{Lag_{Max}} = 8100\ iterations$

$EP_{Lag_{Max}} = 4000$ *iterations* threshold, the evolved harvesting strategy becomes more and more successful in comparison with fixed harvesting strategies. Figure 6.5 highlights the advantage of the evolved harvesting strategy at the highest environmental pressure. The only populations surviving in environmental pressures higher than $EP_{Lag_{Max}} = 7000$ *iterations* are produced by the evolved harvesting strategy.

As a summary, on the one hand the added dimension in the search space reduces the performances of the mEDEA algorithm in easy environments. On the other hand this dimension seems to be necessary for mEDEA to deal with extremely high environmental pressure.

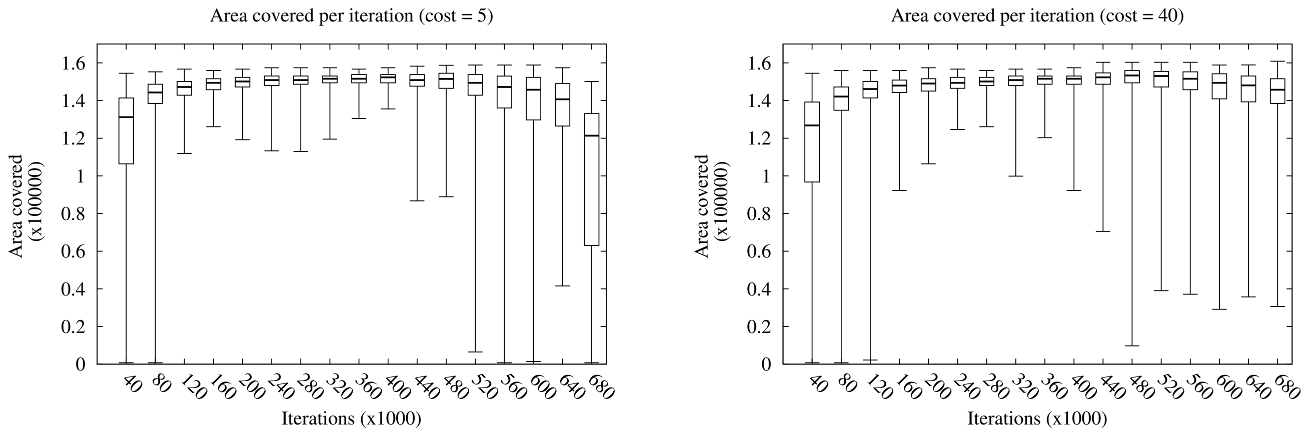
6.3.2 Altruism and Dispersion

Even when the harvesting uses a fixed cost, other behavioral strategies can display altruistic behaviors, such as particular locomotion behavior. On the one hand, if an agent covers a large part of the environment, its actions will impact a large number of individuals, some of them not closely related. Therefore, altruistic actions have a small probability to increase significantly the inclusive fitness of the individuals. Nevertheless, an agent covering a large part of the environment has a high probability to transmit its genome. On the other hand, if an agent covers a small part of the environment, its actions will impact a small number of agents located within its vicinity. These agents are likely to use genomes closely related to the one of the considered agent. Therefore, an altruistic action will benefit to relatives of this agent and increase its inclusive fitness. It is therefore interesting to investigate if the mEDEA algorithm is using this possibility to regulate the level of altruism evolved.

The area covered by agents has been measured in three different setups: one where the harvesting strategy is fixed to a strongly altruistic behavior ($Cost = 40$), one where the harvesting strategy is fixed to a moderately altruistic behavior ($Cost = 5$), and one where the harvesting strategy is evolved by the mEDEA algorithm (dynamic cost). In the third setup, the cost of altruism evolved has also been measured. Experiments are performed in environments where the pressure is at first easy ($EP_{Lag_{Max}} = 25$ *iterations*) until the 400000^{th} iteration, and then continuously increasing by step of 80 iterations every 4000 iterations (10 theoretical generations) until the extinction of the population. Only runs for which the extinction occurs after the 400000^{th} iteration are taken in consideration. For each setup, at least 500 runs are performed (total of more than 1500 runs).

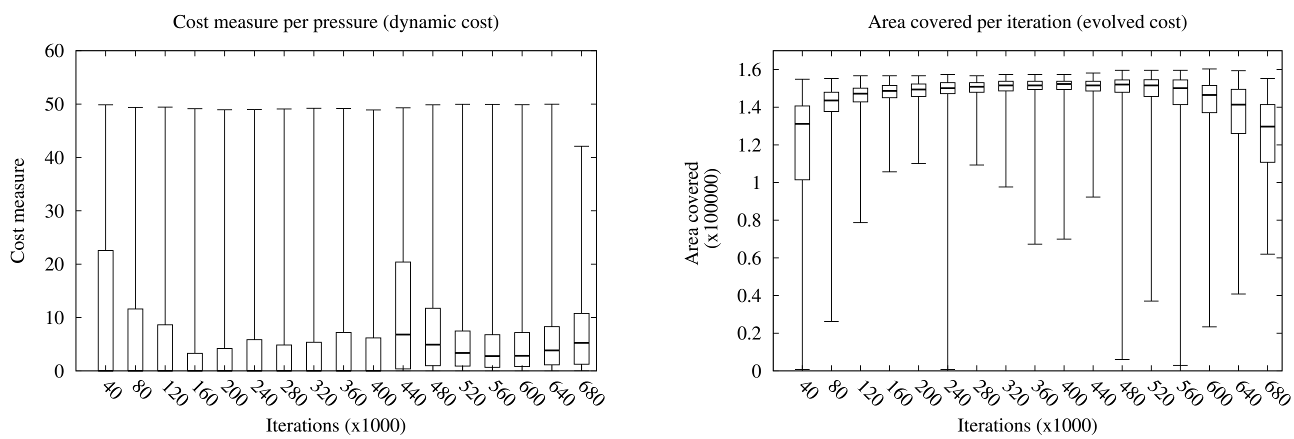
Figures 6.6(a), and 6.6(b) show respectively the area covered by agents when the harvesting strategy is fixed to a moderately altruistic behavior and when the harvesting strategy is fixed to a strongly altruistic behavior. Figures 6.7(a), and 6.7(b) show respectively the costs of altruisms and areas covered when the harvesting strategy is evolved by the mEDEA algorithm.

Figure 6.7(b) shows clearly the impact of environmental pressures on the behaviors of agents when the harvesting strategy is evolving. When the environmental pressure increases (from the 400000^{th} iteration), the dispersion rates are always decreasing ($p - value < 0.05$ for comparisons between iteration 400000 and all iterations from 560000).



(a) (b)

Figure 6.6: Area covered by agents measured when the pressure of the environment increases by step of 80 iterations every 4000 iterations from iteration 400000 (before the environmental pressure is $EP_{Lag_{Max}} = 25\ iterations$): *a)* With harvesting strategy fixed to $Cost = 5$; *b)* With harvesting strategy fixed to $Cost = 40$.



(a) (b)

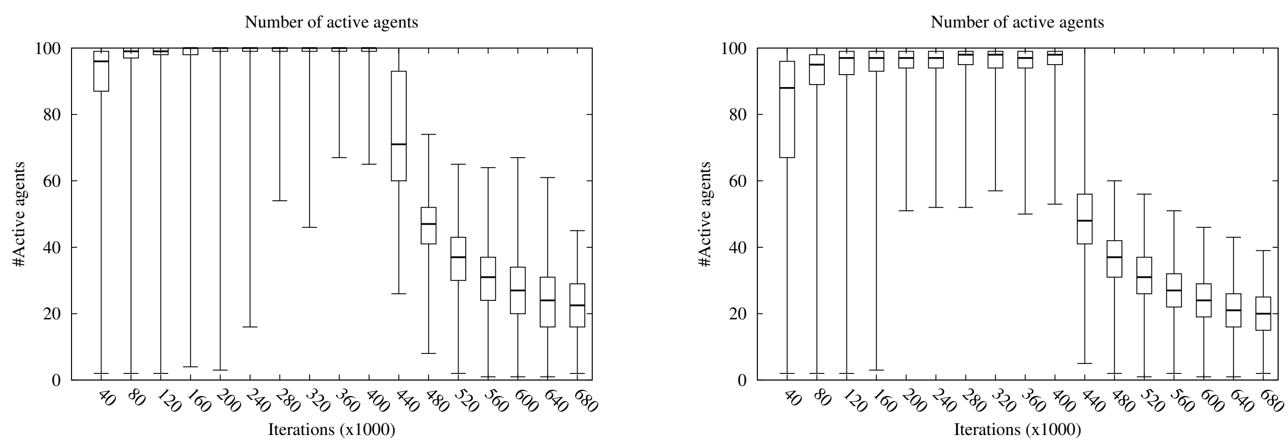
Figure 6.7: Results when the harvesting strategy evolves and the pressure of the environment increases by step of 80 iterations every 4000 iterations from iteration 400000 (before the environmental pressure is $EP_{Lag_{Max}} = 25\ iterations$) : *a)* Costs of altruism ; *b)* Area covered by agents.

Figure 6.6(a) shows similar evolutionary dynamics as the one displayed in Figure 6.7(b). Moreover dispersions rate of agents on the two last iterations of Figures 6.6(a) and 6.7(b) are similar ($p-value >$ 0.05). This shows that the emergence of spatial behaviors is influenced by the harvesting strategy, even if it is a fixed one.

Following previous arguments, the highest level of altruism should be associated with spatial behaviors displaying small dispersion rates. However, higher levels of dispersion are associated with the highest level of altruism tested. Indeed, values displayed in Figure 6.6(b) are significantly higher than the ones presented in Figure 6.7(b) from iteration 440000 ($p - value < 0.05$).

An hypothesis to explain this difference is the following. When a high level of altruism is used each agent harvest few energy from each energy point. As a consequence the inclusive fitness of each agent can be hardly maximized. Therefore the success of a gene in the population is favored by its capacity to spread itself, that is to say to display spatial behavior with high dispersion rates.

In order to test this hypothesis the performance as the number of active agents is measured on each run. Figures 6.8(a), and 6.8(b) show the number of active agents respectively when the harvesting strategy is fixed to a moderately altruistic behavior, and when the harvesting strategy is fixed to a strongly altruistic behavior. Figure 6.9 shows the number of active agents when the harvesting strategy is evolved.



(a) (b)

Figure 6.8: Number of active agents measured when the pressure of the environment increases by step of 80 iterations every 4000 iterations from iteration 400000 (before the environmental pressure is $EP_{Lag_{Max}} = 25$ *iterations*) : *a)* With harvesting strategy fixed to $Cost = 5$; *b)* With harvesting strategy fixed to $Cost = 40$.

On these figures, the number of active agents for a fixed cost of 5 is similar to the number of active agents for an evolved cost at the latest iteration (p-value = 0.8821 for iteration 680000). However, the number of active agents is always lower when a fixed cost of 40 is used ($p - value < 0.05$ for every

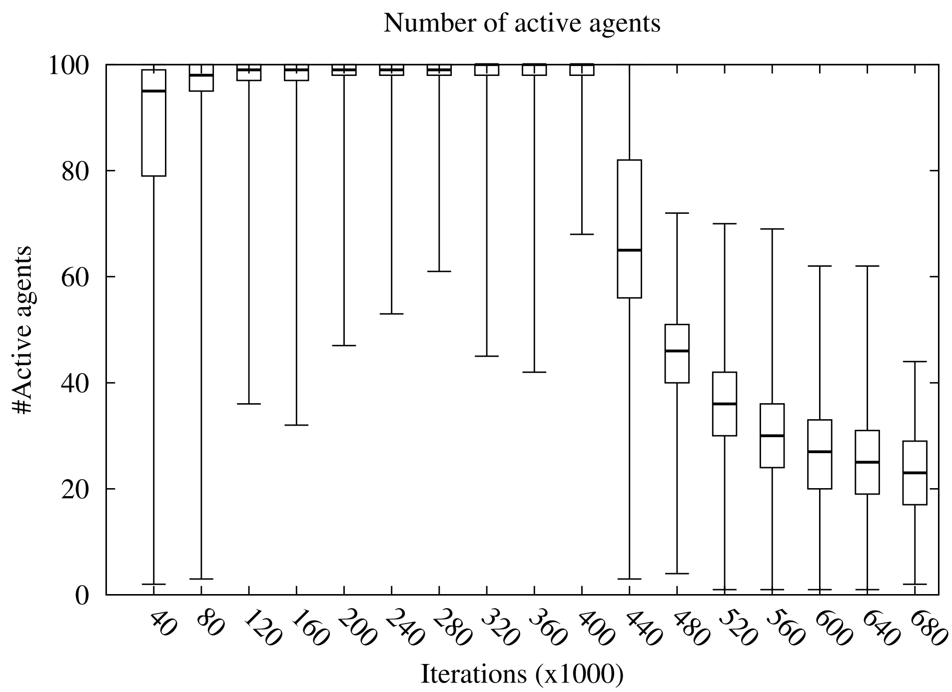


Figure 6.9: Number of active agents measured when the harvesting strategy evolves and the pressure of the environment increases by step of 80 iterations every 4000 iterations from iteration 400000 (before the environmental pressure is $EP_{Lag_{Max}} = 25\ iterations$)

iterations). This difference in the number of active agents confirms the hypothesis made. When highly altruistic behaviors are forced, the survival capacities of each agent are diminished. As a consequence spatial behaviors with small dispersion rates have a small influence on the welfare of their relatives. However spatial behaviors with large dispersion allow the transmission of their genomes to a large portion of the population.

In this section, the nature of locomotion behavior was studied and dispersion was shown as emerging from altruistic behaviors. When moderately altruistic behaviors are used (evolved or fixed), locomotion strategies with small dispersion rates emerge as a consequence of the maximization of the inclusive fitness by the mEDEA algorithm. Moreover, experiments have shown that when a (too) strong level of altruism is artificially enforced agents are acting so as to minimize relatedness with other agents.

6.3.3 Effect of Environmental Setup

The environment itself has an impact in the selection pressure applied to the genomes by modifying the opportunity for two agents to meet. Researchers in evolutionary ecology have termed this property of the environment "viscosity" [Mitteldorf and Wilson, 2000]. A larger viscosity results in fewer genomes transmissions and higher level of relatedness between neighbor agents. It is therefore expected that a large environment viscosity will lead to the evolution of large level of altruism.

In order to test the influence of the environment, the radius of communication is increased to its maximal value (that is to say greater than the diagonal of the environment). A large set of experiments is performed under various environmental pressures by setting a specific value of $EP_{Lag_{Max}}$ for each run, ranging from 25 steps (easy environment) to 400 steps (difficult environment), for a total of 16 setups. For each setup (i.e. a fixed value of $EP_{Lag_{Max}}$), at least 200 independent runs were performed (for a total of more than 3200 experiments) and the results are aggregated to extract two indicators: the number of active agents, and the cost measure. The ratio of extinguished runs was also extracted (number of extinguished runs divided by the number of successful runs).

The aggregated results of the last 10 generations are summarized in Figures 6.10, 6.11(a) and 6.11(b) (resp. ratio of extinguished runs, number of active agents, and cost measure).

Before analyzing the impact of diminishing environment viscosity on the level of altruism evolved, two elements can already be highlighted:

- **Difficulty of the tasks:** Figure 6.10 shows that, for every environmental pressures greater than 50, the extinction rate is at least one order of magnitude lower than on Figure 6.2. This shows that the task is easier with a maximal radius of communication. This is coherent with the observations we made in Chapter 4, i.e. the radius of communication has a high impact on the selection pressure. A large radius of communication results in the transmission of one genome to every agents present in the population, therefore relaxing the pressure to ensure the

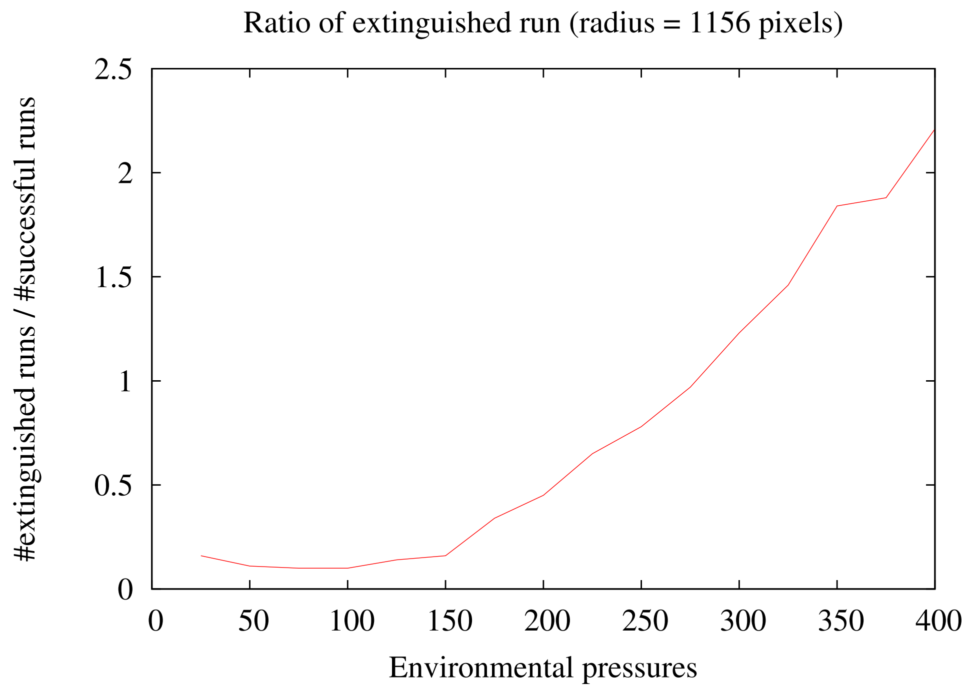


Figure 6.10: Results with $EP_{Lag_{Max}}$ between 25 and 400, and a communication radius of 1156 pixels: Extinction rate (data: value from each run)

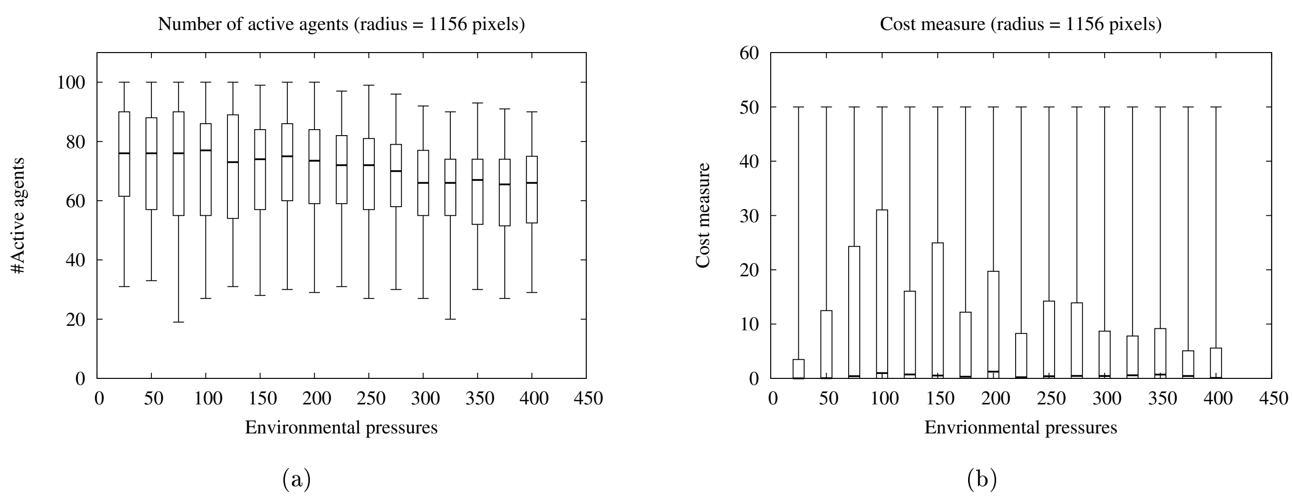


Figure 6.11: Results with $EP_{Lag_{Max}}$ between 25 and 400, and a communication radius of 1156 pixels: *a)* Number of active agent (data: value from each run) ; *b)* Cost measure (data: median values from each run)

propagation of a genome by the agent.

- **Performance of the algorithm:** The number of active agents displayed on Figure 6.11(a) is lower for larger radii of communication ($p - value < 2.2 * 10^{-16}$ for every environmental pressures). This shows effectively that relaxing the pressure to propagate genomes impacts negatively the efficiency of the algorithm.

When the environmental pressure is lower than $EP_{Lag_{Max}} = 200$ the distinction between the level of altruism evolved with each communication radius can't be made with certainty (detailed statistical tests in Table 8.1 in annex). However, above this threshold the level of altruism evolved is always lower with the largest communication radius. This result tend to confirm our hypothesis on the impact of genotypic relatedness on the evolution of altruism by the mEDEA algorithm: large radii of communications impact negatively the relatedness between neighbor agents, thus reducing the impact of altruistic actions on inclusive fitness.

6.4 Altruism and Relatedness

From the perspective of Hamilton's inclusive fitness [Hamilton, 1964], closely related individuals tend to display larger levels of altruism. The intrinsic mechanisms in the algorithm, in particular conservative mutation, already imply a strong genotypic relation between one genome and its offsprings. This close link between genomes in the population might explain the evolution of higher level of altruism. In this section, mEDEA is modified in order to control the level of altruism thanks to genotypic relatedness.

As presented in Section 6.1.1, kin recognition is a mechanism used in artificial and biological systems to increase the effect of kin selection. Kin recognition mechanism is implemented in mEDEA with few modifications to the algorithm. The impact of such operator on the genotypic relatedness, and the evolution of altruistic behaviors is studied.

6.4.1 Effect of Kin-recognition

In practice, the random selection mechanism of mEDEA is replaced by a tournament selection [Miller and Goldberg, 1995] (also embedded in each agent). In this implementation the ranking is based on the genotypic (euclidian) distance between the previously active genome and the locally available genomes (the closer, the better). Thus the mEDEA algorithm is modified to:

- Given: a list of imported genomes
- Randomly select K genomes
- Rank genomes with relation to active genome (genotypic distance)

- Deterministically select the genome closest to the active genome
- Replace the active genome with a mutated selected genome

Tournament selection combined with genotypic distance (termed kin-tournament from now on) makes it possible to introduce a kin recognition mechanism, where the parameter K modifies the exploitation rate (the larger K, the more exploitative mEDEA is).

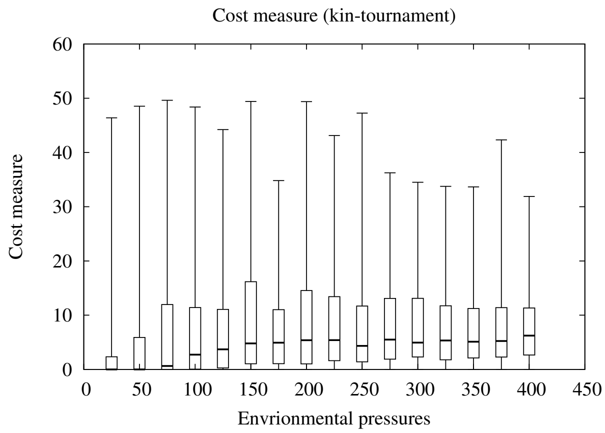
Experiments with a tournament size (K) of 3 (roughly corresponding to medium pressure toward kin selection in this setup) have been achieved with 16 setups, with the environmental pressure ranging from easy ($EP_{Lag_{Max}} = 25$) to strong ($EP_{Lag_{Max}} = 400$). For each setup, at least 200 successful runs were performed (total of more than 3200 experiments), and statistical test are computed with Mann-Whitley's Test to clearly establish the difference in performance [Wilcoxon, 1945; Mann and Whitney, 1947]. The costs of altruism evolved are presented in the Figure 6.12(a) by taking into consideration the last 10 generations of all successful runs for each setup (i.e. after convergence to stable behaviors). For comparison purpose the costs of altruism evolved when the pure-random selection scheme is used are recalled in Figure 6.12(b). In the same fashion, Figure 6.13(a) presents the number of active agents of the last 10 generations when using the kin-tournament selection scheme, and Figure 6.13(b) recall the number of active agents of the last 10 generations when the pure-random selection scheme is used. The ratio of extinguished run for every environmental pressures and both selection schemes are presented in Figure 6.14.

Performing kin recognition increases the level of altruism from the environmental pressure ($p-value < 0.003$ for $EP_{Lag_{Max}} >= 150$, see Table 8.2 in annex for detailed statistical tests). While the number of runs with extinctions is roughly similar (see Table 8.4 in annex for detailed statistical tests), kin recognition suffers from a smaller number of active agents under low and high environmental pressures (see Table 8.3 in annex for detailed statistical tests).

Kin-recognition (implemented thanks to the kin-tournament mechanism) is shown to artificially increase the already existing level of altruism, at the cost of a decreased overall performance with regards to individual survival. This shows that a modification in the agents' behavior impacts the evolution of behaviors observed at the scale of the population. It also highlights the importance of population homogeneity to the evolution of altruism.

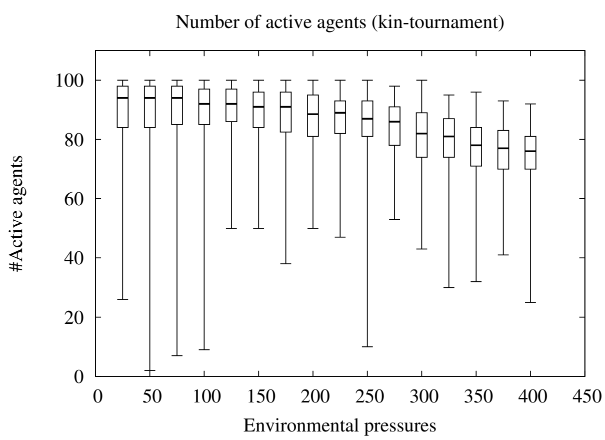
6.4.2 Kin-recognition and Relatedness

By modifying the way genomes are selected, the kin-tournament selection operator is expected to effect the population homogeneity. It has also been considered that increased levels of altruism observed at higher environmental are linked to population homogeneity. Measuring the probability of transmission of one genome to the next generation can help to confirm the relation between population homogeneity and altruism. The rationale is as follow:



(a) (b)

Figure 6.12: Results with $EP_{Lag_{Max}}$ between 25 and 400: *a)* Cost measure with kin-tournament selection scheme (data: median values from each run) ; *b)* Cost measure with pure random selection scheme (data: median values from each run)



(a) (b)

Figure 6.13: Results with $EP_{Lag_{Max}}$ between 25 and 400: *a)* Number of active agent with kin-tournament selection scheme (data: value from each run) ; *b)* Number of active agent with pure random selection scheme (data: value from each run)

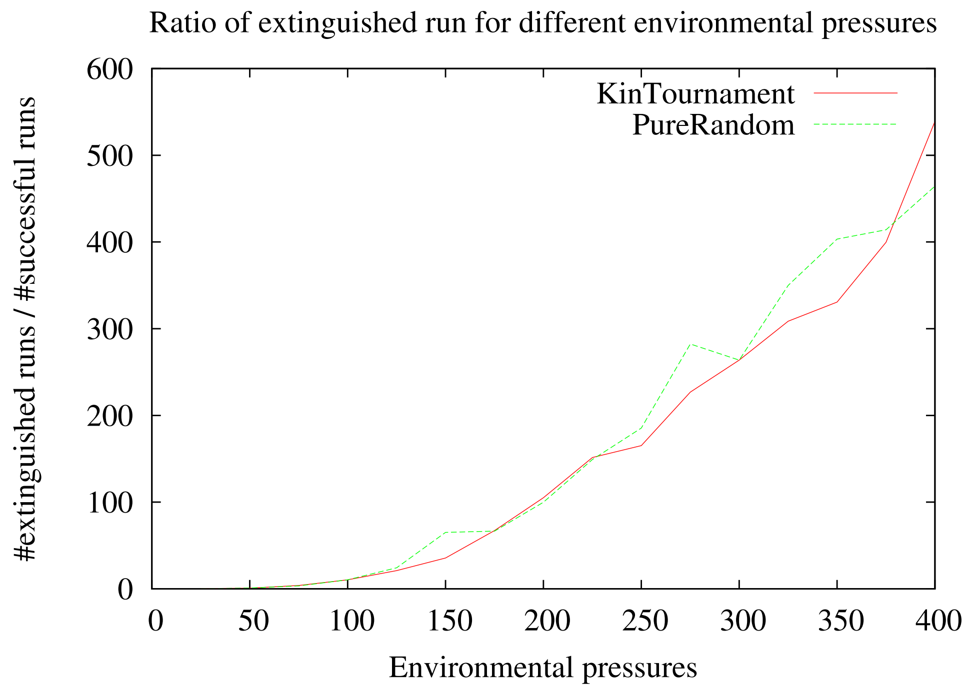


Figure 6.14: Extinction rate for different environmental pressures and the pure-random and kin-tournament selection scheme.

- **High transmission probability:** When every genome has a high probability to be selected by at least one other agent in the next generation, each will propagate to a small number of agents. This leads to an increased heterogeneity in the population.

- **Low transmission probability:** When few genomes are selected at the next generation, selected genomes will propagate to a high number of agents. Therefore, soon every agents will inherit from only one genome found a few generations earlier (homogeneous population).

The global probability of transmission is measured *a posteriori* by counting the number of generations to the Most Related Common Ancestor (MRCA), that is to say the first ancestor from which all current genomes are inheriting. For example, in Figure 6.15, the genome G1 (generation N-3) is the MRCA of all genomes at the generation N, and the number of generations to the MRCA is 3. When the number of generation to the MRCA is low (few genomes propagate quickly in many agents), the selection pressure is high. Reciprocally, when the number of generation to the MRCA is high (many genomes propagate to few agents), the selection pressure is low.

The number of generations to the MRCA is computed for all successful experiments of each of the 16 setups and for both selection scheme (pure-random and kin-tournament selection). Figures 6.16(a), and 6.16(b) shows the number of generations to the MRCA for the last 10 generations of all successful runs for each setup (i.e. after convergence to stable behaviors). Three main observations are drawn

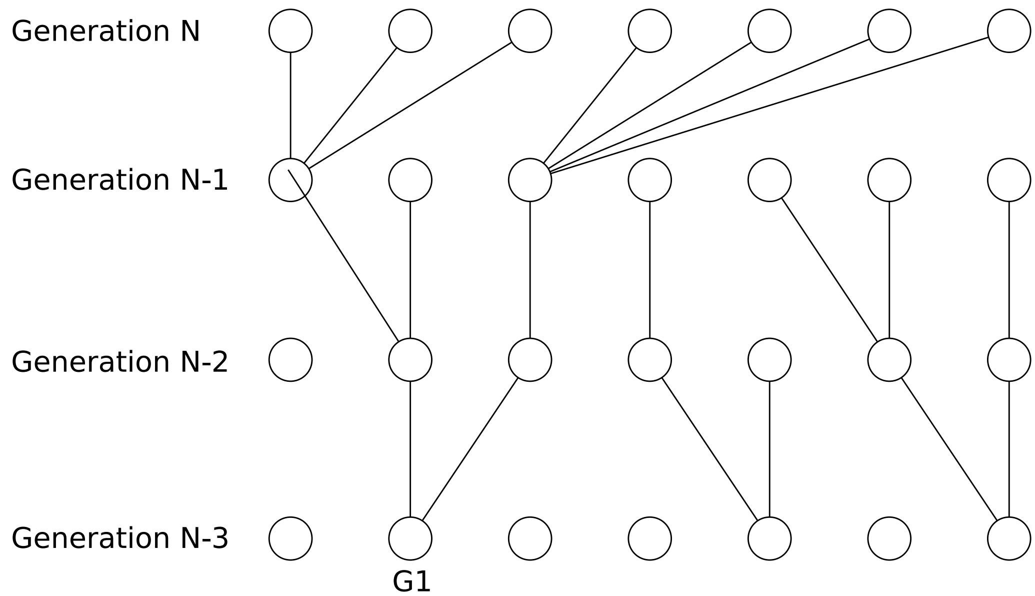


Figure 6.15: Example of selection events in an evolutionary process.

from these graphs.

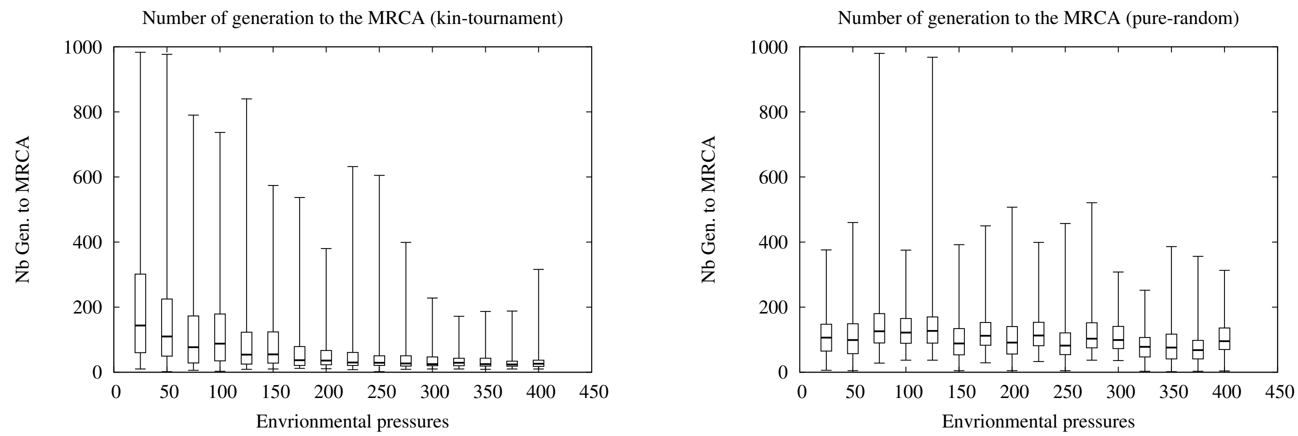
The selection pressure is similar in easy environment for both selection scheme ($p - value > 0.25$), but highly different when the environment is becoming more difficult ($p - value < 10^{-16}$). This confirms the impact of kin-recognition on the homogeneity of the population, which is expected to be the main explanation for the evolution of altruism.

Moreover, variations of altruism's levels (shown in Figure 6.12) are roughly correlated with the variations of selection pressures. This statement is supported by the following observations: a) when the selection pressure increases (from an environmental pressure of $EP_{Lag_{Max}} = 100\ iterations$), the level of altruism increases b) when the selection reaches a stable level (from an environmental pressure of $EP_{Lag_{Max}} = 200\ iterations$), the level of altruism reaches also a stable level. This confirms the link between population homogeneity and altruism when the environmental pressure increase.

6.5 Summary and Conclusions

In this chapter we have detailed how the mEDEA algorithm faces scenarios where the Tragedy of Unmanaged Commons occurs. The evolution of altruism has been seen as a solution, even if not optimal (in terms of number of active agent and ratio of successful runs) in the scenario tested.

In order to obtain a better understanding of the mEDEA algorithm and the reasons for the emergence of altruism, multiple measures on the individuals and their interactions have been performed while varying the selection forces at hand. Based on these experiments a clear explanation for the evolution of altruism can be drawn. A robotic scenario has been chosen where robots are using a limited



(a) (b)

Figure 6.16: Results with $EP_{Lag_{Max}} = 200$: *a)* Number of generations to the most related common ancestor when using kin-tournament selection scheme (data: median values from each run) ; *b)* Number of generations to the most related common ancestor when using pure random selection scheme (data: median values from each run)

radius of communication, as a consequence related genomes are located in the same neighborhood. The importance of local interactions on the evolution of altruism has been highlighted in experiments modifying the viscosity of the environment, and measuring the behavior evolved by agents.

Therefore, kin-selection is expected to be an effective mechanism to explain the natural emergence of altruism. In our experiments, the natural kin-selection is faced with two different selection schemes:

- **Pure-random:** With this selection scheme, a successful reproductive strategy is to meet as much agents as possible, i.e. cover a large proportion of the environment. In this context the selection pressure is low, which produces a relatively heterogeneous population. This means that the agents met don't share necessarily the same genes, so an altruistic act is likely to help a none related genome. In this context, the reproduction strategy goes against the altruistic pressure (leaving energy for its relatives).

- **Kin-tournament:** With this selection scheme, a successful reproductive strategy is to remain in the same neighborhood in order to meet closely related agents (which are more likely to select the genome). This behavior results in high selection pressures and therefore in homogeneous populations. As a consequence, by being altruist an agent has a high probability to help its relatives and enhance the propagation of its own genes. The reproduction strategy goes then along the altruistic pressure.

Moreover, the impact of altruism's level on the performances of the algorithm has been assessed in front of a large range of environmental pressures. From these experiments two main cases are found:

- **Low environmental pressure:** In this context it is usually better to fix beforehand the harvesting strategy. Even if the fixed cost of altruism chosen isn't the best one, the mEDEA algorithm will exploit behavioral strategies as another way to display altruism. Therefore, diminishing the number of possible behaviors will result in a simpler optimization task, without hindering the integrity capacities.

- **High environmental pressure:** Evolving harvesting strategies leads to a higher adaptability in front of highly challenging environments. However this implies the resolution of a trade-off between harvesting strategy and behavioral strategy by the mEDEA algorithm. Experiments with increasing environmental pressures have shown the ability of the mEDEA algorithm to evolve such trade-off in difficult environments, which results in a higher probability to see the population survive.

Chapter 7

Conclusion

7.1 Summary

The work presented in this thesis targets the self-adaptation of a population of robots to unknown and possibly changing environments. This problem is the first step toward the maintenance of a robotic service necessary to perform tasks assigned by the human engineer. As environments considered remain largely unknown, the integrity of robots can't be optimized by the human engineer. Therefore, the maintenance of integrity has to be autonomously addressed by robots by relying solely on the pressure of the environment. These issues are formalized under a new class of problem termed Environment-driven Distributed Evolutionary Adaptation (EDEA).

The minimal set of mechanisms necessary to the resolution of this trade-off is implemented in an algorithm termed mEDEA. Performances of this algorithm have been evaluated in several environments, with different levels of challenge.

In Chapter 4, the mEDEA algorithm has been evaluated in a simple setup, in order to assess its ability to ensure robots integrity. In this context, the mEDEA algorithm evolves efficiently behaviors to maintain the integrity of robots. In this context, the mEDEA is shown to converge toward behavioral consensus when large populations are considered. The applicability of the mEDEA algorithm on real robots has also been validated. Notably, evolutionary dynamics observed are similar to the ones obtained in simulation.

In Chapter 5, the robustness of mEDEA to unpredictable environmental changes has been tested. In this context mEDEA has shown the optimization of behaviors toward a maximization of robots' integrity. To do so, mEDEA achieves a trade-off between possibly antagonistic motivations from the genomes view point (survival of the robots, and spreading of the genome).

Finally, adversarial environments have been used to test the possibility to evolve altruistic behaviors with mEDEA (see Chapter 6). Experiments have shown that in the mEDEA algorithm, the selfish interest of genes may result in behaviors maximizing the global welfare of the group of robots through altruistic behaviors. The evolution of sub-optimal behaviors by the mEDEA algorithm have also been analyzed and explained by the importance of the locality of interactions.

7.2 Discussion

Limitations of the algorithm: In order to ease the study of evolutionary dynamics displayed by the mEDEA algorithm, the following parameters have been fixed: size of the radius of communication, evaluation duration and parameters of the controller architecture. The restriction of these parameters naturally hinder the variety of behaviors evolved by the mEDEA algorithm. Evolving or setting different values for these parameters may be necessary in particular contexts. This could be done by adding the parameters to the genome evolved (as it has been done with the σ parameter). Another way would be to add parameter tunning algorithms embedded in each robot [Tabatabaee *et al.*, 2005]. This requires further experiments to evaluate the sensitivity of each of these parameters.

Division of labor: Scenarios where a high number of different tasks has to be performed in parallel constitute a difficult class of problem. One answer to this problem is the generalization of behaviors to multiple tasks. However such controllers might be too complex to be designed autonomously. The other way to solve this problem is the division of labor. In this context, groups of robots are divided into sub-groups, each optimized for the realization of a sub-task.

Within EDEA algorithms, the division of groups of robots in sub-groups (in order to specialize for the resolution of a task), could be seen as speciation. Preliminary experiments have shown the possibility for the mEDEA algorithm to evolve multiple species within one experiment. These experiments were based on the exploitation of multiple energy sources for the survival of agents. The evolution of species was possible depending on the degree of communication between agents. Three cases are differentiated: allopatric, sympatric, and parapatric speciation. Allopatric speciation occurs when different species are physically separated one from another. This case is trivial from the point of view of the mEDEA algorithm, and multiple species have been evolved in control experiments.

Sympatric speciation occurs when one species divides in two species while the two remain in the same environment. This case is highly difficult to solve by the mEDEA algorithm as evolutionary dynamics naturally tend toward the homogeneity of the population. Preliminary experiments have shown that in a context where two energy sources are available (each able to sustain half of the population), the population will remain homogeneous and exploit only one source.

The parapatric speciation is a phenomenon occuring when species don't completely overlap, but have some interactions. This situation has been modeled in multiple experiments including fixed and moving agents. Preliminary experiments have shown that parapatric speciation can evolve in a narrow range of environmental constraints in a setup with fixed agents, and in few simulations in a setup with moving agents. Further experimentations will be performed to target the evolution of parapatric speciation.

Toward achieving tasks In this thesis we have tackled the challenges raised by the evolution of behaviors targeting the survival of agents in unknown and possibly changing environments. This is a first step to ensure the availability of robots whose final goal is to address a task assigned by the human engineer.

The next step in the resolution of the global challenge is the maintenance of the quality of service. While the survival of robots is a required element to solve this problem, others problems have to be considered, such as for example the maintenance of a radio network. Indeed, the possibility to reach every robots of a group may be necessary to transmit new orders, or new informations essentials to the integrity of robots. Another additional problem is to ensure that robots are physically able to perform a task. A robot currently surviving in the environment but with an important tool broken, is as useless for the Evolutionary Algorithm as an inactive robot. The integration of services' constrains within EDEA algorithms can be performed by the ranking of solutions with regards to the quality of service. This mechanism would favor strategies providing a better quality of service among those able to ensure robot's integrity.

On top of ensuring the integrity of the swarm and maintaining the required quality of service from robots, the final step is to actually solve a task assigned by the human engineer. This implies to integrate a bias from a human engineer in the so far autonomous selection process. This integration should solve a trade-off between selecting strategies good for the quality of service and strategies good to solve the task. This trade-off might actually be solved by the implementation of a decentralized version of a multi-objective algorithms such as NSGA-II [Deb *et al.*, 2002] within an EDEA algorithm. Another issue is the design of the representation scheme used to store a robot design. On the hand, this representation scheme should be able to store the multiple behaviors and design choice resulting from three different need (integrity, quality of service, resolution of a task). On the other hand, the representation should be simple enough to be manipulable by the evolutionary process embedded in each robot. If these two constraints are antagonist, a trade-off has to be set during the design of the representation scheme.

7.3 Perspectives

7.3.1 Communication

Application to the mEDEA algorithm In order to optimize the resolution of a task specified by the human engineer, the availability of robots has to be proposed as a service. The optimization of task requiring strong cooperation between robots, may benefit from a robotic service with communication properties. However, in order to be efficient, this communication should be tailored to the robots, task and environment at hand. Therefore, communicative behaviors should be designed autonomously as part of solutions to maintain integrity.

The use of communication by animals has been widely observed in different contexts, and multiple contributions have investigated for the conditions necessary to its evolution [Roberts and Sherratt, 2002; Mitri *et al.*, 2009]. We provide here an overview of the conclusion drawn by these studies, and provide insight on the applicability of there founding in evolutionary robotics setup.

Biological interpretation A mechanism for the emergence of signals from cues has been proposed in [Tinbergen, 1964]: ritualization. This mechanism is based on the perception of cues by one agent involuntarily emitted by another agent. From this point the receiver evolves a reaction for this cue, and the emitter evolves a behavior for the impact the cue has on the receiver. The cue is then ritualized in a signal, and its impact on behaviors might be unrelated to its origin.

This notion has been reinterpreted in the mind-reader/manipulator framework [Krebs and Dawkins, 1984]. This notion is illustrated by two wolfs fighting against each other (for mating or food). The fight will escalate up to the point where one will bite the other. To do so, the attacking wolf has to bare it's teeth right before bitting. This can be considered as a clue given to other wolf which can read it in order to guide its next action (e.g. avoid, flee, attack). This means that the reaction to the bitting cue can evolve since it will provide a fitness bonus to the wolf able to avoid a dangerous attacker. From this point, the tooth-baring by the attacker can be selected for the alteration of behavior it produces on the receiver. The tooth-baring is then considered as a signal since both the receiver's and emitter's response, have evolved because of there outcome.

In the mind-reader/manipulator framework, the emitter of the signal (tooth-baring) is considered as a manipulator since it manipulates the response of the other wolf thanks to its behavior. On the other side, the receiver of the signal is considered as a mind-reader since it read the intention of the first wolf from its behavior. Therefore, even if wolfs are competing, the signaling can emerge from a an arm-race between mind-reader and manipulator.

In-silico experimentation From the best of our knowledge, five works have studied the ritualization of clues into signals in virtual environments. [Quinn *et al.*, 2002; Williams *et al.*, 2008] are using setups where physical agents benefited directly from the evolution of communication (either the same genome in all agents, or the maximization of the fitness function required a high level of communication). The work done in [Quinn, 2001] has been reproduced by [Arranz *et al.*, 2011], where Quinn's conclusion have been challenged. From the results obtained by [Arranz *et al.*, 2011], no ritualization emerges since the signal isn't amplified and repeated. The last contribution in [Mitri *et al.*, 2009] is concerned with the evolution of communication in adversarial environments. The results obtained have shown that when agents can control the intensity of cues emitted, they would reduce it. This evolutionary dynamic shows an absence of ritualization and therefore a tendency to inhibit communication in competitive scenarios.

From the previously exposed literature we can make the assumption that the ritualization of cues is a good explanation for the emergence of communication when agents have conflicting interest in the biological world. However, experiments performed in competitive scenarios have shown the inhibition of cues rather than their ritualization. An hypothesis to explain these differences is the lack of consideration of genotypic relatedness in virtual environments.

7.3.2 Toward New EDEA Algorithms

The mEDEA algorithm is a minimal demonstration of EDEA algorithms. As such, only the basic features have been implemented. Multiple aspects of this algorithm could be improved by relying on the contributions made to the ER field. The list of contributions of possible interest to the design of better EDEA algorithms given here, isn't meant to be complete.

Covariance matrix Covariance matrix have proved to be an efficient evolutionary method in the design of ES algorithms such as CMA-ES [Hansen, 2006]. This mechanism is based on the association of one mutation parameter σ for each dimension of the search space. The covariance matrix is updated based on the distribution of the best individual of a given generation. Thanks to this, mutations are preferably performed toward most promising parts of the search space. This algorithm has shown its efficiency in a large range of problems [Hansen and Ostermeier, 2001; Hansen *et al.*, 2010]. Moreover its use doesn't require the tunning of parameters as they are all handled autonomously by the algorithm (except the population size).

The integration of covariance matrix mutation scheme in EDEA algorithms could reduce the number of generations required to find suitable solutions. However its implementation might be a challenge as it requires evaluations of multiple solutions in comparable environments. In our context, the environment might change anytime, and the evaluation of two individuals may occur in different environments. As a consequence the implementation of such algorithm should be able to deal with highly noisy fitness functions. In addition a decentralized flavor of the algorithm should be used, in order to take advantage of the use of multiple robots.

Novelty mechanism Evolutionary algorithms are known to evolve sub-optimal but simple behaviors in front of challenging tasks. The novelty mechanism is proposed as a way to promote the evolution of more complex solutions which might be more suited [Lehman and Stanley, 2008]. Authors' underlying goal is to observe open-ended evolution (as no particular area of the search space is targeted). The novelty mechanism is based on the selection of solutions able to show a behavior never seen before even if not better. It's success has been demonstrated in bipedal walking [Lehman and Stanley, 2011].

In complex environments the most basic behavior to maintain integrity might be too complex to be found by chance (bootstrap problem). The novelty mechanism may be then useful to promote genomes with original behaviors, until integrity can be ensured. However, the implementation of a novelty mechanism is challenging in the EDEA context, as comparison between multiple solutions in similar contexts should be provided. This challenge could be solved by maintaining a list of observed behavior individually within each robot, and then compute locally the novelty score of each genome received. The evaluation time necessary for the computation of a novelty score can be saved if a representation of the behavior expressed is sent along each genome.

Diversity mechanism The diversity mechanism aims at addressing the common bootstrap problem in Evolutionary Algorithms. This problem is encountered when all solutions perform equally bad during the first generations. When this situation arise the selection mechanism can't select the most promising behavior as all of them looks bad. The diversity mechanism proposed in [Mouret and Doncieux, 2009b] aim at conserving behaviorally different solutions until a solution significantly above the others is found.

In the experiments performed with the mEDEA algorithm the bootstrap problem hasn't been observed, probably due to the choice of setups. Facing more challenging situation the bootstrap problem is bound to occur in EDEA problems. As a consequence the diversity mechanism could be a solution to address more challenging scenarios. Moreover the maintenance of diversity within the solutions used might help to cope faster with environmental changes.

Behavioral complexity The addition of memory to robotic controllers might help in environments where the integrity depends on a sequence of tasks (e.g. pushing a button to open a door). This research track has been explored by works on recurrent networks such as Elman networks [Elman, 1990], and Echo State Network [Jaeger and Haas, 2004]. Another property useful to answer complex tasks is the use of modularity within one controller (functional or structural). Modularity is the clear identification of the localization of an element either functional or structural [Lipson, 2007]. This property allows the evolution to re-use lower level blocks to compose high level behaviors. It's positive impact on the evolution of robots' controller has been shown in [Mouret and Doncieux, 2008; Cazenille *et al.*, 2012].

The implementation of controllers endowed with such properties might be challenging because of the computational limitation of robots. In order to face this challenge new implementations have to be designed, targeting similar properties but remaining suited to a use on real hardware.

7.3.3 A tool for Evolutionary Ecology

The evolutionary ecology field studies the interplays between evolutionary history and ecological systems. The main questions targeted by this field are the following [Fox *et al.*, 2001]: understanding the condition in which natural selection operate, predicting whether and how the natural selection will favor a genotypic trait on others, understanding why a given phenotype influence an agent's fitness, and evaluate if phenotypic variations observed represent long-term outcome of evolutionary process.

The mEDEA algorithm can be used as a tool modeling evolving populations in order to address part of evolutionary ecology questions. As it has been already shown in this thesis, the mEDEA algorithm may be suited to study the impact of local interactions on the outcome of evolutionary processes which is not trivial with standard approach in evolutionary ecology.

Chapter 8

Appendix

Environmental pressure considered	Statistical comparison thanks to the Mann-Whitney test
$EP_{Lag_{Max}} = 25$	p-value = 0.2928
$EP_{Lag_{Max}} = 50$	p-value = 1.004e-05
$EP_{Lag_{Max}} = 75$	p-value = 2.109e-05
$EP_{Lag_{Max}} = 100$	p-value = 0.01541
$EP_{Lag_{Max}} = 125$	p-value = 0.5613
$EP_{Lag_{Max}} = 150$	p-value = 0.9125
$EP_{Lag_{Max}} = 175$	p-value = 0.0003253
$EP_{Lag_{Max}} = 200$	p-value = 0.42
$EP_{Lag_{Max}} = 225$	p-value = 5.028e-05
$EP_{Lag_{Max}} = 250$	p-value = 0.005418
$EP_{Lag_{Max}} = 275$	p-value = 0.01030
$EP_{Lag_{Max}} = 300$	p-value = 0.0003287
$EP_{Lag_{Max}} = 325$	p-value = 0.0004553
$EP_{Lag_{Max}} = 350$	p-value = 0.01736
$EP_{Lag_{Max}} = 375$	p-value = 5.877e-06
$EP_{Lag_{Max}} = 400$	p-value = 2.923e-07

Table 8.1: Statistical comparison of cost of altruism distribution obtained with two different radius of communication and an environmental pressure range between easy ($EP_{Lag_{Max}} = 25$) and hard ($EP_{Lag_{Max}} = 400$)

Comparison between the kin-tournament and pure-random selection scheme	Statistical comparison thanks to the Mann-Whitney test
$lag = 25$	$p - value = 0.000596$
$lag = 50$	$p - value = 0.765$
$lag = 75$	$p - value = 0.9455$
$lag = 100$	$p - value = 0.08075$
$lag = 125$	$p - value = 0.1983$
$lag = 150$	$p - value = 2.289 * 10^{-07}$
$lag = 175$	$p - value = 0.002193$
$lag = 200$	$p - value = 7.301 * 10^{-05}$
$lag = 225$	$p - value = 1.236 * 10^{-06}$
$lag = 250$	$p - value = 6.621 * 10^{-05}$
$lag = 275$	$p - value = 1.353 * 10^{-08}$
$lag = 300$	$p - value = 1.746 * 10^{-09}$
$lag = 325$	$p - value = 9.699 * 10^{-07}$
$lag = 350$	$p - value = 6.183 * 10^{-12}$
$lag = 375$	$p - value = 3.846 * 10^{-09}$
$lag = 400$	$p - value = 8.422 * 10^{-15}$

Table 8.2: Statistical comparison of costs of altruism distribution obtained by two selections scheme (kin-tournament and pure-random) under 16 environmental pressures

Comparison between the kin-tournament and pure-random selection scheme	Statistical comparison thanks to the Mann-Whitney test
$lag = 25$	$p - value < 2.2 * 10^{-16}$
$lag = 50$	$p - value < 2.2 * 10^{-16}$
$lag = 75$	$p - value < 2.2 * 10^{-16}$
$lag = 100$	$p - value = 3.138 * 10^{-14}$
$lag = 125$	$p - value = 2.818 * 10^{-09}$
$lag = 150$	$p - value = 4.582 * 10^{-06}$
$lag = 175$	$p - value = 0.3505$
$lag = 200$	$p - value = 0.05039$
$lag = 225$	$p - value = 0.2181$
$lag = 250$	$p - value = 0.803$
$lag = 275$	$p - value = 0.7693$
$lag = 300$	$p - value = 0.5159$
$lag = 325$	$p - value = 0.638$
$lag = 350$	$p - value = 0.008253$
$lag = 375$	$p - value = 0.008763$
$lag = 400$	$p - value = 0.00748$

Table 8.3: Statistical comparison of the number of active agents distribution obtained by two selections scheme (kin-tournament and pure-random) under 16 environmental pressures

Comparison between the kin-tournament and pure-random selection scheme	Statistical comparison thanks to the Chi-square test
$lag = 25$	$p - value = 0.2723$
$lag = 50$	$p - value = 0.9702$
$lag = 75$	$p - value = 0.9073$
$lag = 100$	$p - value = 0.9859$
$lag = 125$	$p - value = 0.72$
$lag = 150$	$p - value = 0.7071$
$lag = 175$	$p - value = 0.9986$
$lag = 200$	$p - value < 2.2 * 10^{-16}$
$lag = 225$	$p - value < 2.2 * 10^{-16}$
$lag = 250$	$p - value < 2.2 * 10^{-16}$
$lag = 275$	$p - value = 0.9284$
$lag = 300$	$p - value < 2.2 * 10^{-16}$
$lag = 325$	$p - value = 0.9594$
$lag = 350$	$p - value = 0.9456$
$lag = 375$	$p - value < 2.2 * 10^{-16}$
$lag = 400$	$p - value < 2.2 * 10^{-16}$

Table 8.4: Statistical comparison of the number of extinguished runs obtained by two selections scheme (kin-tournament and pure-random) under 16 environmental pressures

Bibliography

[Adami *et al.*, 1994] C. Adami, C. T. Brown, and W. K. Kellogg. Evolutionary Learning in the 2D Artificial Life System Avida. In *Artificial Life IV*, pages 377–381, 1994. (Cited on page 44.)

[Agre and Chapman, 1987] P. E. Agre and D. Chapman. Pengi: an implementation of a theory of activity. In *Proceedings of the sixth National conference on Artificial intelligence - Volume 1*, AAAI'87, pages 268–272. AAAI Press, 1987. (Cited on page 15.)

[Ahmadi and Stone, 2006] M. Ahmadi and P. Stone. A multi-robot system for continuous area sweeping tasks. In *In ICRA*, 2006. (Cited on pages 36, 37 et 39.)

[Al-Ahmad *et al.*, 2005] A. Al-Ahmad, J. Grossman, and P. Wang. Early experience with a computerized robotically controlled catheter system. *Journal of Interventional Cardiac Electrophysiology*, 12:199–202, 2005. (Cited on page 10.)

[Amil *et al.*, 2009] M. Amil, N. Bredeche, C. Gagné, S. Gelly, M. Schoenauer, and O. Teytaud. A statistical learning perspective of genetic programming. In *EuroGP*, Tuebingen, Allemagne, 2009. Springer. (Cited on page 23.)

[Andersson *et al.*, 2000] B. Andersson, P. Svensson, M. Nordahl, and P. Nordin. On-line evolution of control for a four-legged robot using genetic programming. In S. Cagnoni, editor, *Real-World Applications of Evolutionary Computing*, volume 1803 of *Lecture Notes in Computer Science*, pages 322–329. Springer Berlin / Heidelberg, 2000. (Cited on page 27.)

[Antal *et al.*, 2009] T. Antal, H. Ohtsuki, J. Wakeley, P. D. Taylor, and M. A. Nowak. Evolution of cooperation by phenotypic similarity. *Proceedings of the National Academy of Sciences*, 106(21):8597–8600, 2009. (Cited on page 90.)

[Araujo and Merelo, 2011] L. Araujo and J. Merelo. Diversity through multiculturality: Assessing migrant choice policies in an island model. *Evolutionary Computation, IEEE Transactions on*, 15(4):456–469, aug. 2011. (Cited on page 41.)

[Araujo *et al.*, 2008] L. Araujo, J. J. M. Guervós, C. Cotta, and F. F. de Vega. Multikulti algorithm: Migrating the most different genotypes in an island model. *CoRR*, abs/0806.2843, 2008. (Cited on page 41.)

[Arranz *et al.*, 2011] J. Arranz, J. Noble, and E. Silverman. The origins of communication revisited. In *Eleventh European Conference on the Synthesis and Simulation of Living Systems (ECAL'11)*, 2011. (Cited on page 118.)

[Asada *et al.*, 1999] M. Asada, E. Uchibe, and K. Hosoda. Cooperative behavior acquisition for mobile robots in dynamically changing real worlds via vision-based reinforcement learning and development. *Artificial Intelligence*, 110:275–292, 1999. (Cited on pages 36, 38 et 39.)

[Asada *et al.*, 2009] M. Asada, K. Hosoda, Y. Kuniyoshi, H. Ishiguro, T. Inui, Y. Yoshikawa, M. Ogino, and C. Yoshida. Cognitive Developmental Robotics: A Survey. *IEEE Transactions on Autonomous Mental Development*, 1(1):12–34, April 2009. (Cited on page 18.)

[Auger and Hansen, 2005] A. Auger and N. Hansen. A restart cma evolution strategy with increasing population size. In *Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2005.*, 2005. (Cited on page 30.)

[Augustsson *et al.*, 2002] P. Augustsson, K. Wolff, and P. Nordin. Creation of a learning, flying robot by means of evolution. In *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO '02, pages 1279–1285, San Francisco, CA, USA, 2002. Morgan Kaufmann Publishers Inc. (Cited on page 25.)

[Axelrod and Hamilton, 1981] R. Axelrod and W. D. Hamilton. The evolution of cooperation. *Science*, 211(4489):1390–1396, March 1981. (Cited on pages 43 et 88.)

[Back *et al.*, 1997] T. Back, D. B. Fogel, and Z. Michalewicz, editors. *Handbook of Evolutionary Computation*. IOP Publishing Ltd., Bristol, UK, UK, 1st edition, 1997. (Cited on page 18.)

[Baeck *et al.*, 1997] T. Baeck, D. Fogel, and Z. Michalewicz. *Handbook of Evolutionary Computation*. Taylor & Francis, 1997. (Cited on page 21.)

[Baele *et al.*, 2009] G. Baele, N. Bredeche, E. Haasdijk, S. Maere, N. Michiels, Y. Van De Peer, T. Schmickl, C. Schwarzer, and R. Thenius. Open-ended on-board evolutionary robotics for robot swarms. In *Proceedings of the Eleventh conference on Congress on Evolutionary Computation*, CEC'09, pages 1123–1130, Piscataway, NJ, USA, 2009. IEEE Press. (Cited on page 33.)

[Baird, 1995] L. Baird. Residual algorithms: Reinforcement learning with function approximation. In *In Proceedings of the Twelfth International Conference on Machine Learning*, pages 30–37. Morgan Kaufmann, 1995. (Cited on page 39.)

[Baltes and Anderson, 2002] J. Baltes and J. Anderson. A pragmatic approach to robot rescue: The keystone fire brigade. In W. D. Smart, T. R. Balch, and H. A. Yanco, editors, *AAAI Mobile Robot Competition*, volume WS-02-18 of *AAAI Technical Report*, pages 38–43. AAAI Press, 2002. (Cited on page 9.)

[Bandaru *et al.*, 2011] S. Bandaru, C. Tutum, K. Deb, and J. Hattel. Higher-level innovization: A case study from friction stir welding process optimization. In *Evolutionary Computation (CEC), 2011 IEEE Congress on*, pages 2782–2789, june 2011. (Cited on page 24.)

[Banzhaf *et al.*, 1997] W. Banzhaf, P. Nordin, and M. Olmer. Generating adaptive behavior using function regression within genetic programming and a real robot. In *In 2nd International Conference on Genetic Programming*, pages 35–43. Morgan Kaufmann, 1997. (Cited on page 25.)

[Bedau *et al.*, 1998] M. A. Bedau, E. Snyder, and N. H. Packard. A Classification of Long-Term Evolutionary Dynamics. In C. Adami, R. K. Belew, H. Kitano, and C. Taylor, editors, *Artificial Life VI: Proc. of the Sixth Int. Conf. on Artificial Life*, pages 228–237, Cambridge, MA, 1998. The MIT Press. (Cited on page 43.)

[Bedau *et al.*, 2000] M. A. Bedau, J. S. McCaskill, N. H. Packard, S. Rasmussen, C. Adami, D. G. Green, T. Ikegami, K. Kaneko, and T. S. Ray. Open problems in artificial life. *Artificial Life*, 6:363–376, 2000. (Cited on page 42.)

[Bedau, 2003] M. A. Bedau. Artificial life: Organization, adaptation and complexity from the bottom up. In *Trends in Cognitive Sciences*, pages 505–512, 2003. (Cited on pages 42 et 43.)

[Bernstein *et al.*, 2000] D. S. Bernstein, R. Givan, N. Immerman, and S. Zilberstein. The complexity of decentralized control of markov decision processes. In *Mathematics of Operations Research*, page 2002, 2000. (Cited on page 37.)

[Bersini, 2010] H. Bersini. Software replica of minimal living processes. *Origins of Life and Evolution of Biospheres*, 40:121–130, 2010. (Cited on page 42.)

[Beyer and Schwefel, 2002] H.-G. Beyer and H.-P. Schwefel. Evolution strategies – A comprehensive introduction. *Natural Computing*, 1:3–52, 2002. (Cited on pages 50 et 54.)

[Beyer, 1998] H.-G. Beyer. Evolutionary algorithms in noisy environments: Theoretical issues and guidelines for practice. In *Computer Methods in Applied Mechanics and Engineering*, pages 239–267, 1998. (Cited on page 29.)

[Bianco and Nolfi, 2004] R. Bianco and S. Nolfi. Toward open-ended evolutionary robotics: evolving elementary robotic units able to self-assemble and self-reproduce. *Connection Science*, 4:227–248, 2004. (Cited on page 41.)

[Bich, 2010] L. Bich. Biological autonomy and systemic integration. *Origins of Life and Evolution of the Biospheres*, 40:480–484, 2010. (Cited on page 42.)

[Bird *et al.*, 2008] J. Bird, P. Husbands, M. Perris, B. Bigge, and P. Brown. Implicit fitness functions for evolving a drawing robot. In *Proceedings of the 2008 conference on Applications of evolutionary computing*, Evo'08, pages 473–478, Berlin, Heidelberg, 2008. Springer-Verlag. (Cited on page 26.)

[Blight *et al.*, 2000] K. J. Blight, A. A. Kolykhalov, and C. M. Rice. Efficient initiation of hcv rna replication in cell culture. *Science*, 290(5498):1972–1976, 2000. (Cited on page 43.)

[Bongard and Lipson, 2005] J. Bongard and H. Lipson. Nonlinear system identification using coevolution of models and tests. *Evolutionary Computation, IEEE Transactions on*, 9(4):361–384, aug. 2005. (Cited on page 27.)

[Bongard *et al.*, 2006] J. Bongard, V. Zykov, and H. Lipson. Resilient machines through continuous self-modeling. *Science*, 314:1118–1121, 2006. (Cited on pages 14 et 27.)

[Bongard, 2011] J. Bongard. Morphological change in machines accelerates the evolution of robust behavior. *Proceedings of the National Academy of Sciences*, 2011. (Cited on page 28.)

[Braitenberg, 1986] V. Braitenberg. *Vehicles: Experiments in Synthetic Psychology.* MIT Press, 1986. (Cited on page 16.)

[Bredeche and Montanier, 2010] N. Bredeche and J.-M. Montanier. Environment-driven Embodied Evolution in a Population of Autonomous Agents. In *The 11th International Conference on Parallel Problem Solving From Nature (PPSN 2010)*, pages 290–299, 2010. (Cited on page 74.)

[Bredeche *et al.*, 2009] N. Bredeche, E. Haasdijk, and A. E. Eiben. On-line, on-board evolution of robot controllers. In *Proceedings of Artificial Evolution / Evolution Artificielle (EA'09)*, 2009. (Cited on page 29.)

[Bredeche *et al.*, 2012] N. Bredeche, J.-M. Montanier, W. Liu, and A. F. Winfield. Environment-driven Distributed Evolutionary Adaptation in a Population of Autonomous Robotic Agents. *Mathematical and Computer Modelling of Dynamical Systems, Special Issue: Modelling the swarm – analysing biological and engineered swarm systems*, 18(1):101–129, 2012. (Cited on page 49.)

[Brooks and Mataric, 1993] R. Brooks and M. J. Mataric. Real robots, real learning problems. In J. H. Connell and S. Mahadevan, editors, *Robot Learning*, volume 233 of *The Springer International Series in Engineering and Computer Science*, chapter 8, pages 193–213. Springer, 1993. (Cited on page 36.)

[Brooks, 1986] R. A. Brooks. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, 2(10), 1986. (Cited on page 17.)

[Brooks, 1990] R. A. Brooks. Elephants don't play chess. *Robot. Auton. Syst.*, 6(1-2):3–15, June 1990. (Cited on page 17.)

[Brooks, 1992] R. A. Brooks. Artificial life and real robots. In F. J. Varela and P. Bourgine, editors, *Toward a Practice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life*, pages 3–10, Cambridge, MA, USA, 1992. MIT Press. (Cited on pages 26 et 61.)

[Buehler *et al.*, 2009] M. Buehler, K. Iagnemma, and S. Singh. *The DARPA Urban Challenge: Autonomous Vehicles in City Traffic*. Springer Tracts in Advanced Robotics. Springer, 2009. (Cited on page 14.)

[Busoniu *et al.*, 2008] L. Busoniu, R. Babuska, and B. De Schutter. A comprehensive survey of multiagent reinforcement learning. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 38(2):156–172, 2008. (Cited on page 35.)

[Byers and Bekoff, 1986] J. A. Byers and M. Bekoff. What does "kin recognition" mean? *Ethology*, 72(4):342–345, 1986. (Cited on page 89.)

[Caprari *et al.*, 2002] G. Caprari, T. Estier, and R. Siegwart. Fascination of Down Scaling — Alice the Sugar Cube Robot. *Journal of Micro-Mechatronics*, 1(3):177–189, 2002. (Cited on page 61.)

[Cazenille *et al.*, 2012] L. Cazenille, N. Bredeche, H. Hamann, and J. Stradner. Impact of neuron models and network structure on evolving modular robot neural network controllers. In *GECCO*, pages 89–96, 2012. (Cited on page 120.)

[Chang *et al.*, 2003] L. Chang, R. M. Satava, C. A. Pellegrini, and M. N. Sinanan. Robotic surgery: identifying the learning curve through objective measurement of skill. *Surgical Endoscopy And Other Interventional Techniques*, 17:1744–1748, 2003. (Cited on page 10.)

[Chernova and Veloso, 2004] S. Chernova and M. Veloso. An evolutionary approach to gait learning for four-legged robots. In *Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, volume 3, pages 2562–2567 vol.3, sept 2004. (Cited on pages 25 et 27.)

[Claus and Boutilier, 1998] C. Claus and C. Boutilier. The dynamics of reinforcement learning in cooperative multiagent systems. In *Proceedings of the fifteenth national/tenth conference on Artificial intelligence/Innovative applications of artificial intelligence*, AAAI '98/IAAI '98, pages 746–752, Menlo Park, CA, USA, 1998. American Association for Artificial Intelligence. (Cited on page 36.)

[Cliff *et al.*, 1993] D. Cliff, P. Husbands, and I. Harvey. Explorations in evolutionary robotics. *Adapt. Behav.*, 2(1):73–110, July 1993. (Cited on pages 18 et 23.)

[Couzin, 2002] J. Couzin. Active poliovirus baked from scratch. *Science*, 297(5579):174–175, 2002. (Cited on page 43.)

[Cox and Leonard, 1994] I. J. Cox and J. J. Leonard. Modeling a dynamic environment using a bayesian multiple hypothesis approach. *Artificial Intelligence*, 66(2):311–344, 1994. (Cited on page 16.)

[Crandall, 1996] B. Crandall. *Nanotechnology: Molecular Speculations on Global Abundance*. MIT Press, 1996. (Cited on page 12.)

[Şahin and Spears, 2005] E. Şahin and W. Spears, editors. *Swarm Robotics Workshop: State-of-the-art Survey*, volume 3342 of *Lecture Notes in Computer Science*. Springer, New York, 2005. (Cited on page 61.)

[Şahin and Winfield, 2008] E. Şahin and A. F. T. Winfield, editors. *Swarm Intelligence: Special Issue on Swarm Robotics*, volume 2. Springer, 2008. (Cited on page 61.)

[Dawkins, 1976] R. Dawkins. *The Selfish Gene*, volume 32. Oxford University Press, 1976. (Cited on pages 49 et 90.)

[Dawkins, 1982] R. Dawkins. *The extended phenotype: the gene as the unit of selection*. Oxford Univ., Oxford, 1982. (Cited on page 90.)

[Deb and Srinivasan, 2008a] K. Deb and A. Srinivasan. Innovization: Discovery of innovative design principles through multiobjective evolutionary optimization. In J. Knowles, D. Corne, K. Deb, and D. R. Chair, editors, *Multiobjective Problem Solving from Nature*, Natural Computing Series, pages 243–262. Springer Berlin Heidelberg, 2008. (Cited on page 24.)

[Deb and Srinivasan, 2008b] K. Deb and A. Srinivasan. Innovization: Discovery of innovative design principles through multiobjective evolutionary optimization. In J. Knowles, D. Corne, K. Deb, and D. Chair, editors, *Multiobjective Problem Solving from Nature*, Natural Computing Series, pages 243–262. Springer Berlin Heidelberg, 2008. (Cited on page 24.)

[Deb *et al.*, 2002] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *Trans. Evol. Comp*, 6(2):182–197, April 2002. (Cited on page 117.)

[Deb, 2001] K. Deb. *Multi-Objective Optimization using Evolutionary Algorithms*. Wiley Interscience Series in Systems and Optimization. Wiley, 2001. (Cited on pages 24 et 25.)

[der Loos and Reikensmeyer, 2008] H. F. M. V. der Loos and D. J. Reikensmeyer. Rehabilitation and health care robotics. In *Springer Handbook of Robotics*, Gale virtual reference library, pages 1199–1218. Springer, 2008. (Cited on page 10.)

[Diamond, 2005] J. M. Diamond. *Collapse : how societies choose to fail or survive / Jared Diamond*. Penguin Group, Camberwell, Victoria :, 2005. (Cited on page 91.)

[Dias *et al.*, 2004] M. B. Dias, M. B. Zinck, R. M. Zlot, and A. T. Stentz. Robust multirobot coordination in dynamic environments. In *IEEE International Conference on Robotics and Automation*, volume 4, pages 3435–3442, April 2004. (Cited on page 73.)

[Dionisio and Gordo, 2006] F. Dionisio and I. Gordo. The tragedy of the commons, the public goods dilemma, and the meaning of rivalry and excludability in evolutionary biology. 2006. (Cited on page 91.)

[Doncieux and Mouret, 2010] S. Doncieux and J.-B. Mouret. Behavioral diversity measures for evolutionary robotics. In *WCCI 2010 IEEE World Congress on Computational Intelligence, Congress on Evolutionary Computation (CEC)*, pages 1303–1310, 2010. (Cited on page 25.)

[Doncieux *et al.*, 2006] S. Doncieux, J. B. Mouret, A. Angeli, R. Barate, J. A. Meyer, and E. de Margerie. Building an artificial bird: Goals and accomplishments of the ROBUR project. In *Proceedings of the European Micro Aerial Vehicles (EMAV 2006) conference*, 2006. (Cited on page 17.)

[Doncieux *et al.*, 2009] S. Doncieux, J.-B. Mouret, and N. Bredeche. Exploring New Horizons in Evolutionary Design of Robots. In *Workshop on Exploring new horizons in Evolutionary Design of Robots at IROS 2009*, pages 5–12, Saint Louis, United States, 2009. (Cited on page 24.)

[Doncieux *et al.*, 2012] S. Doncieux, Y. Jin, and J.-B. Mouret. Evo-devo-robo workshop program. In *GECCO (Companion)*, pages 355–356, 2012. (Cited on page 18.)

[Doncieux, 2009] S. Doncieux. Evolutionary algorithms as exploration and analysis helper tools, application to a flapping wing aircraft. In *Proceedings of IROS Workshop "Exploring New Horizons in the Evolutionary Design of Robots"*, 2009. (Cited on page 24.)

[Dorigo and Şahin, 2004] M. Dorigo and E. Şahin, editors. *Autonomous Robots: Special issue on Swarm Robotics*, volume 17. Springer, 2004. (Cited on page 61.)

[Dorigo *et al.*, 2012] M. Dorigo, D. Floreano, L. Gambardella, F. Mondada, S. Nolfi, T. Baaboura, M. Birattari, M. Bonani, M. Brambilla, A. Brutschy, D. Burnier, A. Campo, A. L. Christensen, A. Decugnière, G. Di Caro, F. Ducatelle, E. Ferrante, A. Förster, J. Guzzi, V. Longchamp, S. Magnenat, J. M. Gonzales, N. Mathews, M. Montes de Oca, R. O'Grady, C. Pinciroli, G. Pini, P. Rétornaz, J. Roberts, V. Sperati, T. Stirling, A. Stranieri, T. Stützle, V. Trianni, E. Tuci, A. E. Turgut, and F. C. Vaussard. Swarmanoid: a novel concept for the study of heterogeneous robotic swarms. *IEEE Robotics & Automation agazine*, 2012. (Cited on page 73.)

[Dugatkin and Wilson, 1992] L. A. Dugatkin and D. S. Wilson. The prerequisites for strategic behaviour in bluegill sunfish, lepomis macrochirus. *Animal Behaviour*, 44(Part 2):223–230, 1992. (Cited on page 88.)

[Earon *et al.*, 2000] E. J. P. Earon, T. D. Barfoot, and G. M. T. D'Eleuterio. From the sea to the sidewalk: The evolution of hexapod walking gaits by a genetic algorithm. In *Proceedings of the Third International Conference on Evolvable Systems: From Biology to Hardware*, ICES '00, pages 51–60, London, UK, UK, 2000. Springer-Verlag. (Cited on pages 25 et 27.)

[Eiben and Smith, 2008a] A. Eiben and J. Smith. *Introduction to Evolutionary Computing*. Natural Computing Series. Springer, 2008. (Cited on pages 22 et 134.)

[Eiben and Smith, 2008b] A. E. Eiben and J. Smith. *What is in this book*, chapter 15. In *Natural Computing Series* [2008a], 2008. (Cited on page 23.)

[Eiben *et al.*, 2007] A. Eiben, A. Griffioen, and E. Haasdijk. Population-based adaptive systems: concepts, issues, and the platform new ties. In *Proceedings of European Conference on Complex Systems, Dresden, Germany*, 2007. (Cited on page 43.)

[Eiben *et al.*, 2010a] A. E. Eiben, G. Karafotias, and E. Haasdijk. Self-adaptive mutation in on-line, on-board evolutionary robotics. In *SASO Workshops'10*, pages 147–152, 2010. (Cited on page 32.)

[Eiben *et al.*, 2010b] A. Eiben, E. Haasdijk, and N. Bredeche. *Symbiotic Multi-Robot Organisms: Reliability, Adaptability, Evolution*, volume 7 of *Cognitive Systems Monographs*, chapter Embodied, On-line, On-board Evolution for Autonomous Robotics, pages p.361–382. Springer, 2010. (Cited on pages 28 et 40.)

[Elfwing *et al.*, 2011] S. Elfwing, E. Uchibe, K. Doya, and H. I. Christensen. Darwinian embodied evolution of the learning ability for survival. *Adaptive Behavior - Animals, Animats, Software Agents, Robots, Adaptive Systems*, 19(2):101–120, April 2011. (Cited on page 42.)

[Elman, 1990] J. Elman. Finding structure in time. *Cognitive science*, 14(2):179–211, 1990. (Cited on page 120.)

[Epstein *et al.*, 1996] J. Epstein, R. Axtell, and 2050 Project. *Growing Artificial Societies: Social Science from the Bottom Up*. Complex Adaptive Systems Series. Mit Press, 1996. (Cited on page 44.)

[Eshel and Shaked, 2001] I. Eshel and A. Shaked. Partnership. *Journal of Theoretical Biology*, 208(4):457–474, 2001. (Cited on page 89.)

[Fernandez and Parker, 2001] F. Fernandez and L. E. Parker. Learning in large cooperative multi-robot domains, 2001. (Cited on pages 38 et 39.)

[Ficici *et al.*, 1999] S. G. Ficici, R. A. Watson, and J. B. Pollack. Embodied evolution: A response to challenges in evolutionary robotics. In *Eighth European Workshop on Learning Robots*, pages 14–22, 1999. (Cited on pages 40 et 41.)

[Firby, 1987] R. J. Firby. An investigation into reactive planning in complex domains. In *Proceedings of the sixth National conference on Artificial intelligence - Volume 1*, AAAI'87, pages 202–206. AAAI Press, 1987. (Cited on page 15.)

[Firby, 1989] R. J. Firby. *Adaptive execution in complex dynamic worlds*. PhD thesis, Yale University, New Haven, CT, USA, 1989. (Cited on page 16.)

[Fleming and Purshouse, 2002] P. Fleming and R. Purshouse. Evolutionary algorithms in control systems engineering: a survey. *Control Engineering Practice*, 10(11):1223–1241, 2002. (Cited on page 24.)

[Fletcher and Zwick, 2007] J. A. Fletcher and M. Zwick. The evolution of altruism: Game theory in multilevel selection and inclusive fitness. *Journal of Theoretical Biology*, 245(1):26–36, 2007. (Cited on page 87.)

[Floreano and Mattiussi, 2008] D. Floreano and C. Mattiussi. *Bio-Inspired Artificial Intelligence: Theories, Methods, and Technologies*. The MIT Press, 2008. (Cited on page 24.)

[Floreano and Mondada, 1996] D. Floreano and F. Mondada. Evolution of Homing Navigation in a Real Mobile Robot. *IEEE Transactions on Systems, Man and Cybernetics Part B : Cybernetics*, 26(3):396–407, 1996. (Cited on page 25.)

[Floreano *et al.*, 2002] D. Floreano, N. Schoeni, G. Caprari, and J. Blynel. Evolutionary Bits'n'Spikes. In R. K. Standish, M. A. Beadau, and H. A. Abbass, editors, *8th International Conference on the Simulation and Synthesis of Living Systems (Alife 8)*. MIT Press, 2002. In R. K. Standish, M. A. Beadau and H. A. Abbass (eds). (Cited on page 28.)

[Fogel, 1962] L. Fogel. Autonomous automata. *Industrial Research*, 4(2):14–19, 1962. (Cited on page 24.)

[Fox *et al.*, 2001] C. Fox, D. Roff, and D. Fairbairn. *Evolutionary Ecology: Concepts and Case Studies*. Oxford University Press, 2001. (Cited on page 120.)

[Frank, 2003] S. A. Frank. Repression of competition and the evolution of cooperation. *Evolution*, 57(4):693–705, 2003. (Cited on pages 88 et 90.)

[Futuyma, 2009] D. Futuyma. *Evolution*. Sinauer Associates Inc., 2nd edition, 2009. (Cited on page 51.)

[Ganti *et al.*, 2003] T. Ganti, J. Griesemer, and E. Szathmary. *The Principles of Life*. Oxford Biology. Oxford University Press, USA, 2003. (Cited on page 42.)

[Gánti, 1975] T. Gánti. Organization of chemical reactions into dividing and metabolizing units: The chemotons. *Biosystems*, 7(1):15–21, 1975. (Cited on page 42.)

[García-Sánchez *et al.*, 2012] P. García-Sánchez, A. E. Eiben, E. Haasdijk, B. Weel, and J.-J. Merelo-Guervós. Testing diversity-enhancing migration policies for hybrid on-line evolution of robot controllers. In *Proceedings of the 2012t European conference on Applications of Evolutionary Computation*, EvoApplications'12, pages 52–62, Berlin, Heidelberg, 2012. Springer-Verlag. (Cited on pages 40 et 41.)

[Gat, 1998] E. Gat. On three-layer architectures. In D. Kortenkamp, P. R. Bonnasso, and R. Murphy, editors, *Artificial Intelligence and Mobile Robots*, pages 195–210, 1998. (Cited on page 16.)

[Gauci and Stanley, 2007] J. Gauci and K. Stanley. Generating large-scale neural networks through discovering geometric regularities. In *Proceedings of the 9th annual conference on Genetic and evolutionary computation*, GECCO '07, pages 997–1004, New York, NY, USA, 2007. ACM. (Cited on page 24.)

[Gerlinghaus *et al.*, 2012] F. Gerlinghaus, B. Pierce, T. Metzler, I. Jowers, K. Shea, and G. Cheng. Design and emotional expressiveness of gertie (an open hardware robotic desk lamp). In *Proceedings of the 2012 IEEE International Symposium on Robot and Human Interactive Communication*, September 2012. (Cited on page 13.)

[Gilbert *et al.*, 2006] N. Gilbert, M. den Besten, A. Bontovics, B. G. Craenen, F. Divina, A. Eiben, R. Griffioen, G. Hévízi, A. L. orincz, B. Paechter, S. Schuster, M. C. Schut, C. Tzolov, P. Vogt, and L. Yang. Emerging artificial societies through learning. *Journal of Artificial Societies and Social Simulation*, 9(2):9, 2006. (Cited on page 43.)

[Giralt *et al.*, 1992] G. Giralt, R. Alami, and R. Chatila. Task level programming and robot autonomy. In *Intelligent Robots and Systems, 1992., Proceedings of the 1992 lEEE/RSJ International Conference on*, volume 1, pages 27–34, jul 1992. (Cited on page 14.)

[Glance and Huberman, 1994] N. S. Glance and B. A. Huberman. The dynamics of social dilemmas. *Scientific American*, 270(3):76–81, 1994. (Cited on page 36.)

[Goetz *et al.*, 2003] J. Goetz, S. Kiesler, and A. Powers. Matching Robot Appearance and Behavior to Tasks to Improve Human-Robot Cooperation. In *Proceedings of the 12th IEEE International Workshop on Robot and Human Interactive Communication (RO-MAN 2003)*, pages 55–60, 2003. (Cited on page 7.)

[Golestani *et al.*, 2012] A. Golestani, R. Gras, and M. Cristescu. Speciation with gene flow in a heterogeneous virtual world: can physical obstacles accelerate speciation? *Proceedings of the Royal Society B: Biological Sciences*, 2012. (Cited on page 43.)

[Gomi and Ide, 1998] T. Gomi and K. Ide. Evolution of gaits of a legged robot. In *Fuzzy Systems Proceedings, 1998. IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference on*, volume 1, pages 159–164 vol.1, may 1998. (Cited on pages 25 et 27.)

[Gras *et al.*, 2009] R. Gras, D. Devaurs, A. Wozniak, and A. Aspinall. An individual-based evolving predator-prey ecosystem simulation using a fuzzy cognitive map as the behavior model. *Artif. Life*, 15(4):423–463, October 2009. (Cited on page 43.)

[Grefenstette *et al.*, 1990] J. J. Grefenstette, C. L. Ramsey, and A. C. Schultz. Learning sequential decision rules using simulation models and competition. *Mach. Learn.*, 5(4):355–381, October 1990. (Cited on page 23.)

[Grefenstette, 1992] J. J. Grefenstette. The evolution of strategies for multiagent environments. *Adaptive Behavior*, 1(1):65–90, 1992. (Cited on page 23.)

[Gross *et al.*, 2006a] R. Gross, M. Bonani, F. Mondada, and M. Dorigo. Autonomous Self-assembly in Swarm-Bots. *IEEE Transactions on Robotics*, 22(6):1115–1130, 2006. (Cited on page 61.)

[Gross *et al.*, 2006b] R. Gross, M. Bonani, F. Mondada, and M. Dorigo. Autonomous self-assembly in swarmbots. *IEEE Trans. Robot*, pages 1115–1130, 2006. (Cited on page 18.)

[Gruau, 1994] F. Gruau. Automatic definition of modular neural networks. *Adapt. Behav.*, 3(2):151–183, September 1994. (Cited on page 17.)

[Gu *et al.*, 2003] D. Gu, H. Hu, J. Reynolds, and E. Tsang. Ga-based learning in behaviour based robotics. In *Computational Intelligence in Robotics and Automation, 2003. Proceedings. 2003 IEEE International Symposium on*, volume 3, pages 1521–1526 vol.3, july 2003. (Cited on page 27.)

[Guthart and Salisbury, 2000] G. Guthart and J. J. Salisbury. The intuitivetm telesurgery system: overview and application. In *Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on*, volume 1, pages 618–621, 2000. (Cited on page 10.)

[Haasdijk *et al.*, 2010] E. Haasdijk, A. E. Eiben, and G. Karafotias. On-line evolution of robot controllers by an encapsulated evolution strategy. In *IEEE Congress on Evolutionary Computation'10*, pages 1–7, 2010. (Cited on pages 32, 40 et 41.)

[Haasdijk *et al.*, 2011] E. Haasdijk, A. A. ul Qayyum, and A. E. Eiben. Racing to improve online, on-board evolutionary robotics. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, GECCO '11, pages 187–194, New York, NY, USA, 2011. ACM. (Cited on pages 32 et 40.)

[Haig, 1996] D. Haig. Gestational drive and the green-bearded placenta. *Proceedings of the National Academy of Sciences*, 93(13):6547–6551, 1996. (Cited on page 90.)

[Hamilton, 1964] W. Hamilton. The genetical evolution of social behaviour. i. *Journal of Theoretical Biology*, 7(1):1–16, July 1964. (Cited on pages 88, 90 et 107.)

[Hammond, 1921] J. H. Hammond. System of radiodirective control, 08 1921. (Cited on page 6.)

[Hansen and Ostermeier, 2001] N. Hansen and A. Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 9(2):159–195, 2001. (Cited on page 119.)

[Hansen *et al.*, 2004] E. A. Hansen, D. S. Bernstein, and S. Zilberstein. Dynamic programming for partially observable stochastic games. In *Proceedings of the 19th national conference on Artifical intelligence*, AAAI'04, pages 709–715. AAAI Press, 2004. (Cited on page 37.)

[Hansen *et al.*, 2010] N. Hansen, A. Auger, R. Ros, S. Finck, and P. Pošík. Comparing results of 31 algorithms from the black-box optimization benchmarking bbob-2009. In *Proceedings of the 12th annual conference companion on Genetic and evolutionary computation*, GECCO '10, pages 1689–1696, New York, NY, USA, 2010. ACM. (Cited on page 119.)

[Hansen, 2006] N. Hansen. The CMA evolution strategy: a comparing review. In J. Lozano, P. Larranaga, I. Inza, and E. Bengoetxea, editors, *Towards a new evolutionary computation. Advances on estimation of distribution algorithms*, pages 75–102. Springer, 2006. (Cited on page 119.)

[Hardin, 1968] G. Hardin. The tragedy of the commons. *Science*, 162:1243–1248, December 1968. (Cited on page 90.)

[Hardin, 1994] G. Hardin. The tragedy of the unmanaged commons. *Trends in Ecology & Evolution*, 9(5):199–199, 1994. (Cited on pages 90 et 91.)

[Harvey *et al.*, 1994] I. Harvey, P. Husbands, and D. Cliff. Seeing the light: artificial evolution, real vision. In *Proceedings of the third international conference on Simulation of adaptive behavior : from animals to animats 3: from animals to animats 3*, SAB94, pages 392–401, Cambridge, MA, USA, 1994. MIT Press. (Cited on page 23.)

[Hauber *et al.*, 2000] M. E. Hauber, P. W. Sherman, and D. Paprika. Self-referent phenotype matching in a brood parasite: the armpit effect in brown-headed cowbirds (molothrus ater). *Animal Cognition*, 3:113–117, 2000. (Cited on page 90.)

[Hauert and Doebeli, 2004] C. Hauert and M. Doebeli. Spatial structure often inhibits the evolution of cooperation in the snowdrift game. *Nature*, 428(6983):643–646, 2004. (Cited on page 90.)

[Hauert *et al.*, 2009] S. Hauert, J.-C. Zufferey, and D. Floreano. Reverse-engineering of artificially evolved controllers for swarms of robots. In *Evolutionary Computation, 2009. CEC '09. IEEE Congress on*, pages 55–61, may 2009. (Cited on page 24.)

[Heidrich-Meisner and Igel, 2008] V. Heidrich-Meisner and C. Igel. Evolution strategies for direct policy search. In *Proceedings of the 10th international conference on Parallel Problem Solving from Nature: PPSN X*, pages 428–437, Berlin, Heidelberg, 2008. Springer-Verlag. (Cited on page 18.)

[Herdy and Patone, 1994] M. Herdy and G. Patone. Evolution strategy in action : 10 es-demonstrations. Technical report, Technische Universität Berlin, October 1994. (Cited on page 23.)

[Hillman, 2003] M. Hillman. Rehabilitation robotics from past to present-a historical perspective. In *International Conference on Rehabilitation Robotics*, Daejeon, Korea, April 2003. (Cited on page 10.)

[Hoffmann and Pfister, 1996] F. Hoffmann and G. Pfister. Evolutionary learning of a fuzzy control rule base for an autonomous vehicle. In *Proceedings of the Fifth International conference IPMU: Information Processing and Management of Uncertainty in Knowledge-Based Systems*, pages 659–664, July 1996. (Cited on page 25.)

[Holland, 1975] J. Holland. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. University of Michigan Press, 1975. (Cited on page 23.)

[Holland, 1996] J. Holland. *Hidden Order: How Adaptation Builds Complexity*. Basic Books, 1996. (Cited on page 90.)

[Hornby *et al.*, 2000] G. Hornby, S. Takamura, O. Hanagata, M. Fujita, and J. B. Pollack. Evolution of controllers from a high-level simulator to a high dof robot. In *Proceedings of the Third International Conference on Evolvable Systems: From Biology to Hardware*, ICES '00, pages 80–89, London, UK, UK, 2000. Springer-Verlag. (Cited on pages 25 et 27.)

[Hornby *et al.*, 2005] G. S. Hornby, S. Takamura, T. Yamamoto, and M. Fujita. Autonomous evolution of dynamic gaits with two quadruped robots. *Trans. Rob.*, 21(3):402–410, June 2005. (Cited on pages 24, 25 et 28.)

[Huijsman *et al.*, 2011] R.-J. Huijsman, E. Haasdijk, and A. Eiben. *An On-line On-board Distributed Algorithm for Evolutionary Robotics*. Springer, 2011. (Cited on pages 40 et 41.)

[Ichbiah, 2005] D. Ichbiah. *Robots: from science fiction to technological revolution*. Harry N. Abrams, 2005. (Cited on page 6.)

[Ijspeert *et al.*, 2007] A. J. Ijspeert, A. Crespi, D. Ryczko, and J.-M. Cabelguen. From swimming to walking with a salamander robot driven by a spinal cord model. *Science*, 315:1416–1420, 2007. (Cited on page 12.)

[Ishiguro, 2008] H. Ishiguro. Studies on humanlike robots - humanoid, android and geminoid. In *SIMPAR*, page 2, 2008. (Cited on page 7.)

[Isles *et al.*, 2001] A. R. Isles, M. J. Baum, D. Ma, E. B. Keverne, and N. D. Allen. Genetic imprinting: Urinary odour preferences in mice. *Nature*, 409:783–784, 2001. (Cited on page 90.)

[Jaeger and Haas, 2004] H. Jaeger and H. Haas. Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. *Science*, 304(5667):78–80, 2004. (Cited on page 120.)

[Jakobi *et al.*, 1995] N. Jakobi, P. Husband, and I. Harvey. Noise and the reality gap: The use of simulation in evolutionary robotics. In F. Moran, A. Moreno, J. Merelo, and P. Chancon, editors, *Advances in Artificial Life: Proceedings of the Third European Conference on Artificial Life*. Berlin: Springer Verlag, 1995. (Cited on pages 26, 61 et 68.)

[Jakobi, 1993] N. Jakobi. Half-baked, ad-hoc, and noisy: Minimal simulations for evolutionary robotics. In P. Husbands and I. Harvey, editors, *Proceedings of the Fourth International Workshop on the Synthesis and Simulation of Living Systems (Artificial Life IV)*, pages 348–357. Morgan Kaufmann, 1993. (Cited on page 27.)

[joon Sun *et al.*, 2001] S. joon Sun, D. wook Lee, and K. bo Sim. Artificial immune-based swarm behaviors of distributed autonomous robotic systems. In *International Conference on Robotics and Automation*, volume 4, pages 3993–3998, 2001. (Cited on page 73.)

[Kaelbling *et al.*, 1998] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101(1-2):99–134, May 1998. (Cited on page 16.)

[Kala, 2012] R. Kala. Multi-robot path planning using co-evolutionary genetic programming. *Expert Systems with Applications*, 39(3):3817–3831, 2012. (Cited on page 23.)

[Kamegawa *et al.*, 2004] T. Kamegawa, T. Yamasaki, H. Igarashi, and F. Matsuno. Development of the snake-like rescue robot. In *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*, volume 5, pages 5081–5086 Vol.5, april-1 may 2004. (Cited on page 9.)

[Kamio and Iba, 2005] S. Kamio and H. Iba. Adaptation technique for integrating genetic programming and reinforcement learning for real robots. *Evolutionary Computation, IEEE Transactions on*, 9(3):318–333, june 2005. (Cited on page 27.)

[Kan, 1990] E. Kan. Supervised robotics for space servicing. In *Intelligent Robots and Systems '90. 'Towards a New Frontier of Applications', Proceedings. IROS '90. IEEE International Workshop on*, pages 437–443, jul 1990. (Cited on page 14.)

[Kaplan, 2004] F. Kaplan. Who is afraid of the humanoid? investigating cultural differences in the acceptance of robots. *International Journal of Humanoid Robotics*, 1(3):1–16, 2004. (Cited on page 7.)

[Karafotias *et al.*, 2011] G. Karafotias, E. Haasdijk, and A. E. Eiben. An algorithm for distributed on-line, on-board evolutionary robotics. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, GECCO '11, pages 171–178, New York, NY, USA, 2011. ACM. (Cited on page 41.)

[Kawai *et al.*, 2001] K. Kawai, A. Ishiguro, and P. Eggenberger. Incremental evolution of neurocontrollers with a diffusion-reaction mechanism of neuromodulators. In *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on*, volume 4, pages 2384–2391, 2001. (Cited on page 25.)

[Keller and Ross, 1998] L. Keller and K. G. Ross. Selfish genes: a green beard in the red fire ant. *Nature*, 394:573–575, 1998. (Cited on page 90.)

[Knibbe *et al.*, 2007] C. Knibbe, A. Coulon, O. Mazet, J.-M. Fayard, and G. Beslon. A long-term evolutionary pressure on the amount of noncoding dna. *Mol. Biol. Evol.*, 24(10):2344–2353, 2007. (Cited on page 44.)

[König *et al.*, 2008] L. König, K. Jebens, S. Kernbach, and P. Levi. Stability of on-line and on-board evolving of adaptive collective behavior. In *European Robotics Symposium 2008*, volume 44/2008 of *Springer Tracts in Advanced Robotics*, pages 293–302. Springer Berlin / Heidelberg, MAR 2008. (Cited on page 40.)

[Koos *et al.*, 2009] S. Koos, J.-B. Mouret, and S. Doncieux. Automatic system identification based on coevolution of models and tests. In *Proceedings of the Eleventh conference on Congress on Evolutionary Computation*, CEC'09, pages 560–567, Piscataway, NJ, USA, 2009. IEEE Press. (Cited on page 27.)

[Kornienko *et al.*, 2005] S. Kornienko, O. Kornienko, and P. Levi. Collective AI: context awareness via communication. In *IJCAI'05: Proceedings of the 19th international joint conference on Artificial intelligence*, pages 1464–1470, San Francisco, CA, USA, 2005. Morgan Kaufmann Publishers Inc. (Cited on page 61.)

[Kostiadis and Hu, 2001] K. Kostiadis and H. Hu. Kabage-rl: Kanerva-based generalisation and reinforcement learning for possession football. In *In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS*, 2001. (Cited on pages 38 et 39.)

[Koza *et al.*, 1997] J. R. Koza, I. F. H. Bennett, M. A. Keane, and D. Andre. Automatic programming of a time-optimal robot controller and an analog electrical circuit to implement the robot controller by means of genetic programming. In *Proceedings of the 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation*, CIRA '97, pages 340–346, Washington, DC, USA, 1997. IEEE Computer Society. (Cited on page 23.)

[Koza, 1992] J. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. Complex Adaptive Systems. MIT Press, 1992. (Cited on page 23.)

[Krebs and Dawkins, 1984] J. R. Krebs and R. Dawkins. *Animal signals: Mind-reading and manipulation*, pages 380–402. Sinauer Associates, Sunderland, Massachussetts, 1984. (Cited on page 118.)

[Kroeker, 2009] K. L. Kroeker. Medical nanobots. *Commun. ACM*, 52(9):18–19, September 2009. (Cited on page 12.)

[Kurzweil, 2005] R. Kurzweil. *The Singularity Is Near: When Humans Transcend Biology*. A Penguin Book: Science. Viking, 2005. (Cited on page 7.)

[Laredo *et al.*, 2010] J. L. Laredo, A. E. Eiben, M. Steen, and J. J. Merelo. Evag: a scalable peer-to-peer evolutionary algorithm. *Genetic Programming and Evolvable Machines*, 11(2):227–246, June 2010. (Cited on page 41.)

[Lauer and Riedmiller, 2000] M. Lauer and M. Riedmiller. An algorithm for distributed reinforcement learning in cooperative multi-agent systems. In *Proceedings of the Seventeenth International Conference on Machine Learning*, pages 535–542. Morgan Kaufmann, 2000. (Cited on page 36.)

[Laumond, 1998] J. Laumond. *Robot motion planning and control*. Lecture notes in control and information sciences. Springer, 1998. (Cited on page 15.)

[Lee *et al.*, 1997] W.-P. Lee, J. Hallam, and H. Lund. Applying genetic programming to evolve behavior primitives and arbitrators for mobile robots. In *Evolutionary Computation, 1997., IEEE International Conference on*, pages 501–506, apr 1997. (Cited on page 23.)

[Lehman and Stanley, 2008] J. Lehman and K. Stanley. Exploiting open-endedness to solve problems through the search for novelty. In S. Bullock, J. Noble, R. Watson, and M. A. Bedau, editors, *Artificial Life XI: Proceedings of the Eleventh International Conference on the Simulation and Synthesis of Living Systems*, pages 329–336. MIT Press, Cambridge, MA, 2008. (Cited on page 119.)

[Lehman and Stanley, 2011] J. Lehman and K. Stanley. Abandoning objectives: Evolution through the search for novelty alone. *Evolutionary computation*, 19(2):189–223, 2011. (Cited on pages 26 et 119.)

[Lehmann and Keller, 2006a] L. Lehmann and L. Keller. The evolution of cooperation and altruism – a general framework and a classification of models. *Journal of Evolutionary Biology*, 19(5):1365–1376, 2006. (Cited on pages 88 et 89.)

[Lehmann and Keller, 2006b] L. Lehmann and L. Keller. The evolution of cooperation and altruism - a general framework and a classification of models. *Journal of Evolutionary Biology*, 19(5):1365–1376, 2006. (Cited on page 87.)

[Leticia *et al.*, 2004] A. Leticia, A. Fletcher Jeffrey, and A. D. Cutter. The kin composition of social groups: trading group size for degree of altruism. *Am Nat*, 164(2):132–44, 2004. (Cited on page 89.)

[Lichbach, 1996] M. Lichbach. *The Cooperator's Dilemma*. Economics, Cognition, and Society. University of Michigan Press, 1996. (Cited on page 36.)

[Lima, 1989] S. L. Lima. Iterated Prisoner's Dilemma: An Approach to Evolutionarily Stable Cooperation. *American Naturalist*, 134:828–834, 1989. (Cited on page 88.)

[Lin *et al.*, 1998] P. Lin, G. Bekey, and K. Abney. Autonomous military robotics: Risk, ethics and design, 1998. (Cited on page 10.)

[Lipson and Pollack, 2000] H. Lipson and J. Pollack. Automatic design and manufacture of robotic lifeforms. *Nature*, 406:974–978, 2000. (Cited on page 28.)

[Lipson *et al.*, 2006] H. Lipson, J. Bongard, V. Zykov, and E. Malone. Evolutionary Robotics for Legged Machines: From Simulation to Physical Reality. *Intelligent Autonomous Systems 9: IAS-9*, 2006. (Cited on page 28.)

[Lipson, 2007] H. Lipson. Principles of modularity, regularity, and hierarchy for scalable systems. *The Journal of Biological Physics and Chemistry*, 7(4):125–128, 2007. (Cited on pages 24 et 120.)

[Liu and Winfield, 2011] W. Liu and A. F. Winfield. Open-hardware e-puck linux extension board for experimental swarm robotics research. *Microprocessors and Microsystems*, 35(1):60–67, 2011. (Cited on page 62.)

[Macal and North, 2005] C. M. Macal and M. J. North. Tutorial on agent-based modeling and simulation. In *Proceedings of the 37th conference on Winter simulation*, WSC '05, pages 2–15. Winter Simulation Conference, 2005. (Cited on page 43.)

[MacDorman, 2005] K. F. MacDorman. Mortality salience and the uncanny valley. In *Humanoids*, pages 399–405, 2005. (Cited on page 7.)

[Macinnes and Paolo, 2004] I. Macinnes and E. D. Paolo. Crawling out of the simulation: Evolving real robot morphologies using cheap reusable modules. In *In Artificial Life IX: Proc. Ninth Intl. Conf. on the Simulation and Synthesis of Life*, pages 94–99. MIT Press, 2004. (Cited on page 28.)

[Mahdavi and Bentley, 2006] S. H. Mahdavi and P. J. Bentley. Innately adaptive robotics through embodied evolution. *Auton. Robots*, 20(2):149–163, 2006. (Cited on page 27.)

[Maley, 1999] C. Maley. Four steps toward open-ended evolution. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-1999)*, volume 2, pages 1336–1343, 1999. (Cited on page 43.)

[Mankiw, 2009] Mankiw. *Principles of economics, 3rd edition*. South-Western College Pub, february edition, 2009. (Cited on page 91.)

[Mann and Whitney, 1947] H. B. Mann and D. R. Whitney. On a test of whether one of two random variables is stochastically larger than the other. *Annals of Mathematical Statistics*, 1947. (Cited on page 108.)

[Marocco and Floreano, 2002] D. Marocco and D. Floreano. Active Vision and Feature Selection in Evolutionary Behavioral Systems. In *From Animals to Animats 7: Proceedings of the Seventh International Conference on Simulation of Adaptive Behavior*, pages 247–255, 2002. (Cited on page 28.)

[Matarić and Cliff, 1996] M. J. Matarić and D. Cliff. Challenges in evolving controllers for physical robots. *Robotics and Autonomous Systems*, 19(1):67–83, 1996. (Cited on page 26.)

[Mataric, 1996] M. J. Mataric. Using communication to reduce locality in distributed multi-agent learning. *Journal of Experimental and Theoretical Artificial Intelligence*, 10:357–369, 1996. (Cited on page 39.)

[Mataric, 1997] M. J. Mataric. Reinforcement learning in the multi-robot domain. *Autonomous Robots*, 4:73–83, 1997. (Cited on pages 15, 37 et 39.)

[Mataric, 2001] M. J. Mataric. Learning in behavior-based multi-robot systems: Policies, models, and other agents. *Cognitive Systems Research*, 2:81–93, 2001. (Cited on page 37.)

[Mateo and Johnston, 2000] J. M. Mateo and R. E. Johnston. Kin recognition and the 'armpit effect': evidence of self–referent phenotype matching. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 267(1444):695–700, 2000. (Cited on page 90.)

[Mateo, 2004] J. M. Mateo. Recognition systems and biological organization: the perception component of social recognition. *Ann Zool Fenn*, 41:729–745, December 2004. (Cited on page 89.)

[Mayer *et al.*, 2010] C. Mayer, S. Sosnowski, K. Kuhnlenz, and B. Radig. Towards robotic facial mimicry: System development and evaluation. In *RO-MAN, 2010 IEEE*, pages 198–203, sept. 2010. (Cited on page 13.)

[Maynard Smith, 1964] J. Maynard Smith. Group selection and kin selection. *Nature*, 201:1145, 1964. (Cited on page 89.)

[McLurkin and Smith, 2004] J. McLurkin and J. Smith. Distributed algorithms for dispersion in indoor environments using a swarm of autonomous mobile robots. In *Distributed Autonomous Robotic Systems Conference*. Springer, Berlin, Germany, 2004. (Cited on page 73.)

[McMullin, 2004] B. McMullin. Thirty years of computational autopoiesis: a review. *Artif. Life*, 10(3):277–295, June 2004. (Cited on page 42.)

[McNamara *et al.*, 2004] J. M. McNamara, Z. Barta, and A. I. Houston. Variation in behaviour promotes cooperation in the prisoner's dilemma game. *Nature*, 428(6984):745–8, 2004. (Cited on page 43.)

[Meng *et al.*, 2007] Y. Meng, O. Kazeem, and J. Muller. A hybrid aco/pso control algorithm for distributed swarm robots. In *Swarm Intelligence Symposium, 2007. SIS 2007. IEEE*, pages 273–280, April 2007. (Cited on page 73.)

[Mettler *et al.*, 1998] L. Mettler, M. Ibrahim, and W. Jonat. One year of experience working with the aid of a robotic assistant (the voice-controlled optic holder aesop) in gynaecological endoscopic surgery. *Hum Reprod*, 13(1O):2748–2758, 1998. (Cited on page 10.)

[Meyer, 1995] J.-A. Meyer. The animat approach to cognitive science. In H. Roitblat and J.-A. Meyer, editors, *Comparative Approaches to Cognitive Science*, pages 27–44. The MIT Press, 1995. (Cited on page 17.)

[Miessner, 1919] B. F. Miessner. The electric dog. *Scientific American Supplement*, (2267):376, June 1919. (Cited on page 6.)

[Miglino *et al.*, 1994] O. Miglino, K. Nafasi, and C. E. Taylor. Selection for wandering behavior in a small robot. *Artif. Life*, 2(1):101–116, January 1994. (Cited on page 27.)

[Miglino *et al.*, 1995] O. Miglino, H. H. Lund, and S. Nolfi. Evolving mobile robots in simulated and real environments. *Artif. Life*, 2(4):417–434, January 1995. (Cited on page 27.)

[Miller and Goldberg, 1995] B. L. Miller and D. E. Goldberg. Genetic algorithms, tournament selection, and the effects of noise. *Complex Systems*, 9:193–212, 1995. (Cited on page 107.)

[Misevic *et al.*, 2012] D. Misevic, A. Frénoy, D. Parsons, and F. Taddei. Effects of public good properties on the evolution of cooperation. In *Artificial Life XIII*, pages 218–225, July 2012. (Cited on page 44.)

[Mitri *et al.*, 2009] S. Mitri, D. Floreano, and L. Keller. The evolution of information suppression in communicating robots with conflicting interests. *Proceedings of the National Academy of Sciences*, 106(37):15786–15790, September 2009. (Cited on page 118.)

[Mitteldorf and Wilson, 2000] J. Mitteldorf and D. S. Wilson. Population viscosity and the evolution of altruism. *Journal of Theoretical Biology*, pages 481–496, 2000. (Cited on pages 90 et 105.)

[Mondada *et al.*, 2009] F. Mondada, M. Bonani, X. Raemy, J. Pugh, C. Cianci, A. Klaptocz, S. Magnenat, J.-C. Zufferey, D. Floreano, and A. Martinoli. The e-puck, a robot designed for education in engineering. In *Proceedings of the 9th Conference on Autonomous Robot Systems and Competitions*, volume 1, pages 59–65, 2009. (Cited on page 61.)

[Montanier and Bredeche, 2009] J.-M. Montanier and N. Bredeche. Embedded evolutionary robotics: the (1+1)-restart-online adaptation algorithm. In *IEEE IROS Workshop on Exploring new horizons in Evolutionary Design of Robots (Evoderob09)*, pages 37–43, October 2009. (Cited on pages 29 et 30.)

[Morales, 2003] E. F. Morales. Scaling up reinforcement learning with a relational representation. In *In Proc. of the Workshop on Adaptability in Multi-agent Systems*, pages 15–26, 2003. (Cited on pages 36, 37 et 39.)

[Mori, 1970] M. Mori. The uncanny valley. *Energy*, 7(4):33–35, 1970. (Cited on page 7.)

[Moriarty *et al.*, 1999] D. E. Moriarty, A. C. Schultz, and J. J. Grefenstette. Evolutionary algorithms for reinforcement learning. *Journal of Artificial Intelligence Research*, 11:199–229, 1999. (Cited on page 18.)

[Mouret and Doncieux, 2008] J.-B. Mouret and S. Doncieux. Mennag: a modular, regular and hierarchical encoding for neural-networks based on attribute grammars. *Evolutionary Intelligence*, 1:187–207, 2008. (Cited on page 120.)

[Mouret and Doncieux, 2009a] J.-B. Mouret and S. Doncieux. Evolving modular neural-networks through exaptation. In *Proceedings of the Eleventh conference on Congress on Evolutionary Computation*, CEC'09, pages 1570–1577, Piscataway, NJ, USA, 2009. IEEE Press. (Cited on page 24.)

[Mouret and Doncieux, 2009b] J.-B. Mouret and S. Doncieux. Overcoming the bootstrap problem in evolutionary robotics using behavioral diversity. In *Proceedings of the Eleventh conference on Congress on Evolutionary Computation*, CEC'09, pages 1161–1168, Piscataway, NJ, USA, 2009. IEEE Press. (Cited on pages 26 et 120.)

[Munteanu and Solé, 2006] A. Munteanu and R. V. Solé. Phenotypic diversity and chaos in a minimal cell model. *Journal of Theoretical Biology*, 240(3):434–442, 2006. (Cited on page 42.)

[Murphy, 2004] R. Murphy. Human-robot interaction in rescue robotics. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 34(2):138–153, may 2004. (Cited on page 9.)

[Nakhaeinia *et al.*, 2011] D. Nakhaeinia, S. H. Tang, S. B. M. Noor, and O. Motlagh. A review of control architectures for autonomous navigation of mobile robots. *International Journal of the Physical Sciences*, 6(2):169–174, January 2011. (Cited on page 16.)

[Needham, 1956] J. Needham. *History of Scientific Thought*, volume 2 of *Science and Civilisation in China*, page 54. Cambridge University Press, 1956. (Cited on page 6.)

[Nehmzow, 2002] U. Nehmzow. Physically embedded genetic algorithm learning in multi-robot scenarios: The pega algorithm, 2002. (Cited on page 41.)

[Nelson *et al.*, 2008] B. J. Nelson, L. Dong, and F. Arai. Micro/nanorobots. In B. Siciliano and O. Khatib, editors, *Springer Handbook of Robotics*, pages 411–450. Springer, Berlin, Heidelberg, 2008. (Cited on page 12.)

[Nelson *et al.*, 2009] A. L. Nelson, G. J. Barlow, and L. Doitsidis. Fitness functions in evolutionary robotics: A survey and analysis. *Robotics and Autonomous Systems*, 57:345–370, 2009. (Cited on page 25.)

[Nembrini, 2005] J. Nembrini. *Minimalist coherent swarming of wireless networked autonomous mobile robots*. PhD thesis, University of the West of England, Bristol, January 2005. (Cited on page 73.)

[Ng *et al.*, 1999] A. Y. Ng, D. Harada, and S. Russell. Policy invariance under reward transformations: theory and application to reward shaping. In *Proc. 16th International Conf. on Machine Learning*, pages 278–287. Morgan Kaufmann, San Francisco, CA, 1999. (Cited on page 42.)

[Nilsson, 1980] N. Nilsson. *Principles of artificial intelligence*. Tioga Pub. Co., 1980. (Cited on page 15.)

[Nilsson, 1984] N. J. Nilsson. Shakey the robot. Technical Report 323, AI Center, SRI International, 333 Ravenswood Ave., Menlo Park, CA 94025, Apr 1984. (Cited on page 15.)

[Nof, 1999] S. Y. Nof. *Handbook of Industrial Robotics*. John Wiley & Sons, Inc., New York, NY, USA, 2nd edition, 1999. (Cited on page 9.)

[Nolfi and Floreano, 2000] S. Nolfi and D. Floreano. *Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-Organizing Machines*. Cambridge, MA: MIT Press/Bradford Books, 2000. (Cited on pages 23, 26 et 30.)

[Nolfi *et al.*, 1994] S. Nolfi, D. Floreano, O. Miglino, and F. Mondada. How to Evolve Autonomous Robots: Different Approaches in Evolutionary Robotics. In R. A. Brooks and P. Maes, editors, *Artificial life IV: Proceedings of the 4th International Workshop on Artificial Life*, pages 190–197. MA: MIT Press, 1994. R. A. Brooks and P. Maes (eds.). (Cited on pages 23 et 27.)

[Nordin and Banzhaf, 1996] P. Nordin and W. Banzhaf. An on-line method to evolve behavior and to control a miniature robot in real time with genetic programming. *Adapt. Behav.*, 5(2):107–140, September 1996. (Cited on page 28.)

[Nordin and Banzhaf, 1997] P. Nordin and W. Banzhaf. Real time control of a khepera robot using genetic programming. *Cybernetics and Control*, 26(3):533–561, 1997. (Cited on page 23.)

[Nowak and Sigmund, 1993] M. Nowak and K. Sigmund. Chaos and the evolution of cooperation. *Proceedings of the National Academy of Sciences*, 90:5091–5094, 1993. (Cited on pages 43 et 88.)

[Nowak and Sigmund, 1998] M. Nowak and K. Sigmund. Evolution of indirect reciprocity by image scoring. *Nature*, 393:573–7, June 1998. (Cited on page 90.)

[Okura *et al.*, 2003] M. Okura, A. Matsumoto, H. Ikeda, and K. Murase. Artificial evolution of fpga that controls a miniature mobile robot khepera. In *SICE 2003 Annual Conference*, volume 3, pages 2858–2863 Vol.3, aug. 2003. (Cited on page 27.)

[Olsson *et al.*, 2006] L. A. Olsson, C. L. Nehaniv, and D. Polani. From unknown sensors and actuators to actions grounded in sensorimotor perceptions. *Connection Science*, 18(2):121–144, June 2006. (Cited on page 18.)

[Ortin *et al.*, 2004] D. Ortin, J. Neira, and J. Montiel. Relocation using laser and vision. In *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*, volume 2, pages 1505–1510, may 2004. (Cited on page 16.)

[Ostrowski *et al.*, 2008] E. A. Ostrowski, M. Katoh, G. Shaulsky, D. C. Queller, and J. E. Strassmann. Kin discrimination increases with genetic distance in a social amoeba. *PLoS Biol*, 6(11):e287, 11 2008. (Cited on page 89.)

[Oyama *et al.*, 2001] S. Oyama, P. Griffiths, and R. Gray. *Cycles of Contingency: Developmental Systems and Evolution*. Bradford books. Mit Press, 2001. (Cited on page 24.)

[Panait and Luke, 2005] L. Panait and S. Luke. Cooperative multi-agent learning: The state of the art. *Autonomous Agents and Multi-Agent Systems*, 11(3):387–434, November 2005. (Cited on page 35.)

[Parker, 1997] L. Parker. Cooperative motion control for multi-target observation. In *In Proceedings of the 1997 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1591–1598, 1997. (Cited on page 36.)

[Parker, 2000] L. E. Parker. Lifelong adaptation in heterogeneous multi-robot teams: Response to continual variation in individual robot performance. *Auton. Robots*, 8(3):239–267, June 2000. (Cited on pages 15, 37 et 73.)

[Pettinaro *et al.*, 2005] G. Pettinaro, L. Gambardella, and A. Ramirez-Serrano. Adaptive distributed fetching and retrieval of goods by a swarm-bot. In *Advanced Robotics, 2005. ICAR'05. Proceedings., 12th International Conference on*, pages 825–832, 2005. (Cited on page 73.)

[Pfeifer and Bongard, 2006] R. Pfeifer and J. C. Bongard. *How the Body Shapes the Way We Think: A New View of Intelligence (Bradford Books)*. The MIT Press, 2006. (Cited on page 11.)

[Piliavin and Charng, 1990] J. Piliavin and H. Charng. Altruism: A review of recent theory and research. *Annual review of sociology*, pages 27–65, 1990. (Cited on page 87.)

[Pineda *et al.*, 2012] L. E. Pineda, A. E. Eiben, and M. van Steen. Evolving communication in robotic swarms using on-line, on-board, distributed evolutionary algorithms. In *Proceedings of the 2012t European conference on Applications of Evolutionary Computation*, EvoApplications'12, pages 529–538, Berlin, Heidelberg, 2012. Springer-Verlag. (Cited on page 41.)

[Pollack *et al.*, 2000] J. Pollack, H. Lipson, S. Ficici, P. Funes, G. Hornby, and R. Watson. Evolutionary techniques in physical robotics. In J. Miller, A. Thompson, P. Thomson, and T. Fogarty, editors, *Evolvable Systems: From Biology to Hardware*, volume 1801 of *Lecture Notes in Computer Science*, pages 175–186. Springer Berlin Heidelberg, 2000. (Cited on page 26.)

[Prieto *et al.*, 2009] A. Prieto, F. Bellas, A. Faina, and R. J. Duro. Asynchronous situated coevolution and embryonic reproduction as a means to autonomously coordinate robot teams. In *Proceedings of the 13th International Conference on Knowledge-Based and Intelligent Information and Engineering Systems: Part I*, KES '09, pages 351–359, Berlin, Heidelberg, 2009. Springer-Verlag. (Cited on page 41.)

[Prieto *et al.*, 2010] A. Prieto, J. A. Becerra, F. Bellas, and R. J. Duro. Open-ended evolution as a means to self-organize heterogeneous multi-robot systems in real time. *Robot. Auton. Syst.*, 58(12):1282–1291, December 2010. (Cited on page 41.)

[Provost *et al.*, 2006] J. Provost, B. Kuipers, and R. Miikkulainen. Developing navigation behavior through self-organizing distinctive-state abstraction. *Connection Science*, 18:159–172, June 2006. (Cited on page 18.)

[Pugh and Martinoli, 2008] J. Pugh and A. Martinoli. Distributed adaptation in multi-robot search using particle swarm optimization. In *Proceedings of the 10th International Conference on the Simulation of Adaptive Behavior*, Lecture Notes in Computer Science, pages 393–402, 2008. (Cited on page 73.)

[Queller *et al.*, 2003] D. C. Queller, E. Ponte, S. Bozzaro, and J. E. Strassmann. Single-gene greenbeard effects in the social amoeba dictyostelium discoideum. *Science*, 299(5603):105–106, 2003. (Cited on pages 88 et 90.)

[Queller, 1985] D. C. Queller. Kinship, reciprocity and synergism in the evolution of social behaviour. *Nature*, 318:366–367, November 1985. (Cited on page 90.)

[Queller, 1992] D. C. Queller. A general model for kin selection. *Evolution*, 46(2):pp. 376–380, 1992. (Cited on pages 89 et 90.)

[Quinn *et al.*, 2002] M. Quinn, L. Smith, G. Mayley, and P. Husbands. Evolving team behaviour for real robots. In *Proc. Artificial Life VIII*, pages 302–311. MIT Press, 2002. (Cited on pages 25 et 118.)

[Quinn, 2001] M. Quinn. Evolving communication without dedicated communication channels. *Advances in Artificial Life*, pages 357–366, 2001. (Cited on page 118.)

[Rankin *et al.*, 2007] D. Rankin, K. Bargum, and H. Kokko. The tragedy of the commons in evolutionary biology. *Trends in Ecology & Evolution*, 22(12):643–651, December 2007. (Cited on pages 87 et 91.)

[Ray, 1992] T. S. Ray. Evolution, ecology and optimization of digital organisms. Technical report, Santa Fe Institute, 1992. (Cited on page 43.)

[Riolo *et al.*, 2001] R. L. Riolo, M. D. Cohen, and R. Axelrod. Evolution of cooperation without reciprocity. *Nature*, 414:441–443, November 2001. (Cited on page 90.)

[Roberts and Sherratt, 2002] G. Roberts and T. N. Sherratt. Behavioural evolution (communication arising): Does similarity breed cooperation? *Nature*, 418:499–500, 2002. (Cited on pages 90 et 118.)

[Rosin and Belew, 1996] C. Rosin and R. Belew. New methods for competitive coevolution. *Evolutionary Computation*, 5:1–29, 1996. (Cited on page 84.)

[Rousset and Ronce, 2004] F. Rousset and O. Ronce. Inclusive fitness for traits affecting metapopulation demography. *Theoretical Population Biology*, 65(2):127–141, 2004. (Cited on page 90.)

[Rueckstiess *et al.*, 2010] T. Rueckstiess, F. Sehnke, T. Schaul, D. Wierstra, Y. Sun, and J. Schmidhuber. Exploring parameter space in reinforcement learning. *Paladyn*, 1(1):14–24, March 2010. (Cited on page 37.)

[Ruff *et al.*, 2004] H. A. Ruff, G. L. Calhoun, M. H. Draper, J. V. Fontejon, and B. J. Guilfoos. Exploring automation issues in supervisory control of multiple uavs. In *Proceedings of the Human Performance, Situation Awareness, and Automation Technology Conference*, pages 218–222, March 2004. (Cited on page 14.)

[Ryan *et al.*, 2004] A. Ryan, M. Zennaro, A. Howell, R. Sengupta, and J. Hedrick. An overview of emerging results in cooperative uav control. In *Decision and Control, 2004. CDC. 43rd IEEE Conference on*, volume 1, pages 602–607, dec. 2004. (Cited on page 14.)

[Sahin *et al.*, 1998] E. Sahin, P. Gaudiano, E. S. Ahin, and P. Gaudiano. Mobile robot range sensing through visual looming. In *Proceedings of the ISIC/CIRA/ISAS*, pages 370–375, 1998. (Cited on page 16.)

[Sahin *et al.*, 2008] E. Sahin, S. Girgin, L. Bayindir, and A. E. Turgut. Swarm robotics. In C. Blum and D. Merkle, editors, *Swarm Intelligence*, Natural Computing Series, pages 87–100. Springer Berlin Heidelberg, 2008. (Cited on page 12.)

[Sahin, 2005] E. Sahin. Swarm robotics: From sources of inspiration to domains of application. In E. Sahin and W. Spears, editors, *Swarm Robotics*, volume 3342 of *Lecture Notes in Computer Science*, pages 10–20. Springer Berlin / Heidelberg, 2005. (Cited on page 12.)

[Sanchez and Alonso, 2012] V. G. Sanchez and I. G. Alonso. Dh compliant adapter to integrate a surveillance robot in the digital home. *International Journal of Engineering and Innovative Technology*, 2(3), September 2012. (Cited on page 10.)

[Saygin *et al.*, 2011] A. P. Saygin, T. Chaminade, H. Ishiguro, J. Driver, and C. Frith. The thing that should not be: predictive coding and the uncanny valley in perceiving human and humanoid robot actions. *Soc Cogn Affect Neurosci*, 2011. (Cited on page 7.)

[Schanze, 2010] J. Schanze. Plug & pray, 2010. (Cited on page 7.)

[Schmickl *et al.*, 2009] T. Schmickl, R. Thenius, C. Moeslinger, G. Radspieler, S. Kernbach, M. Szymanski, and K. Crailsheim. Get in touch: cooperative decision making based on robot-to-robot collisions. *Autonomous Agents and Multi-Agent Systems*, 18:133–155, 2009. (Cited on page 73.)

[Schultz and Grefenstette, 1992] A. C. Schultz and J. J. Grefenstette. Using a genetic algorithm to learn behaviors for autonomous vehicles. In *In Proceedings of the AiAA Guidance, Navigation, and Control Conference Hilton Head, SC*, pages 739–749, 1992. (Cited on page 23.)

[Schultz *et al.*, 1996] A. C. Schultz, J. J. Grefenstette, and W. Adams. Roboshepherd: Learning a complex behavior. In *Proceedings of the FLAIRS'96*, volume 6, pages 763–768, 1996. (Cited on page 23.)

[Schut *et al.*, 2009a] M. C. Schut, E. Haasdijk, and A. E. Eiben. What is situated evolution? In *Proceedings of the Eleventh conference on Congress on Evolutionary Computation*, CEC'09, pages 3277–3284, Piscataway, NJ, USA, 2009. IEEE Press. (Cited on page 41.)

[Schut *et al.*, 2009b] M. C. Schut, E. Haasdijk, and A. Prieto. Is situated evolution an alternative for classical evolution? In *Proceedings of the Eleventh conference on Congress on Evolutionary Computation*, CEC'09, pages 2971–2976, Piscataway, NJ, USA, 2009. IEEE Press. (Cited on page 41.)

[Schwefel, 1981] H.-P. Schwefel. *Numerical Optimization of Computer Models.* John Wiley & Sons, Inc., New York, NY, USA, 1981. (Cited on page 41.)

[Schwefel, 1995] H. Schwefel. *Evolution and optimum seeking.* Number vol.1 in Sixth-generation computer technology series. Wiley, 1995. (Cited on page 23.)

[Sehnke *et al.*, 2008] F. Sehnke, C. Osendorfer, T. R. A. Graves, J. Peters, and J. Schmidhuber. Policy gradients with parameter-based exploration for control. In *Proceedings of the 18th international conference on Artificial Neural Networks, Part I*, ICANN '08, pages 387–396, Berlin, Heidelberg, 2008. Springer-Verlag. (Cited on page 37.)

[Sheppard and Salzberg, 1997] J. W. Sheppard and S. L. Salzberg. A teaching strategy for memory-based control. *Artif. Intell. Rev.*, 11(1-5):343–370, February 1997. (Cited on page 38.)

[Shimizu *et al.*, 2003] M. Shimizu, A. Ishiguro, T. Kawakatsu, Y. Masubuchi, and M. Doi. Coherent swarming from local interaction by exploiting molecular dynamics and stokesian dynamics methods. In *Proceedings. 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2003. (IROS 2003).*, volume 2, pages 1614–1619, December 2003. (Cited on page 73.)

[Shukla *et al.*, 2010] P. J. Shukla, D. S. Scherr, and J. W. Milsom. Robot-assisted surgery and health care costs. *New England Journal of Medicine*, 363(22):2174–2176, 2010. (Cited on page 10.)

[Simoes and Dimond, 1999] E. Simoes and K. Dimond. An evolutionary controller for autonomous multi-robot systems. In *Systems, Man, and Cybernetics, 1999. IEEE SMC '99 Conference Proceedings. 1999 IEEE International Conference on*, volume 6, pages 596–601 vol.6, 1999. (Cited on page 40.)

[Sims, 1994] K. Sims. Evolving 3d morphology and behavior by competition. In *Proceedings of Artificial Life IV*, pages 28–39. MIT Press, 1994. (Cited on page 28.)

[Smith, 2000] P. W. H. Smith. Controlling code growth in genetic programming. In R. John and R. Birkenhead, editors, *Advances in Soft Computing*, pages 166–171, De Montfort University, Leicester, UK, 2000. Physica-Verlag. (Cited on page 23.)

[Sober, 1992] E. Sober. The evolution of altruism: Correlation, cost, and benefit. *Biology and Philosophy*, 7(2), 1992. (Cited on pages 87 et 89.)

[Spector and Klein, 2006] L. Spector and J. Klein. Genetic stability and territorial structure facilitate the evolution of tag-mediated altruism. *Artificial Life*, 12, 2006. (Cited on page 90.)

[Spector *et al.*, 2004] L. Spector, J. Klein, and C. Perry. Tags and the evolution of cooperation in complex environments. In *Proceedings of the AAAI 2004 Symposium on Artificial Multiagent Learning*. AAAI Press, 2004. (Cited on page 90.)

[Sproewitz *et al.*, 2008] A. Sproewitz, R. Moeckel, J. Maye, and A. J. Ijspeert. Learning to move in modular robots using central pattern generators and online optimization. *Int. J. Rob. Res.*, 27(3-4):423–443, 2008. (Cited on page 11.)

[Spröwitz, 2010] A. Spröwitz. *Roombots: Design and Implementation of a Modular Robot for Reconfiguration and Locomotion*. PhD thesis, Lausanne, 2010. (Cited on page 11.)

[Stone and Veloso, 2000] P. Stone and M. Veloso. Multiagent systems: A survey from a machine learning perspective. *Auton. Robots*, 8(3):345–383, June 2000. (Cited on page 35.)

[Stone *et al.*, 2001] P. Stone, R. S. Sutton, and S. P. Singh. Reinforcement learning for 3 vs. 2 keepaway. In *RoboCup 2000: Robot Soccer World Cup IV*, pages 249–258, London, UK, UK, 2001. Springer-Verlag. (Cited on page 38.)

[Stone *et al.*, 2006] P. Stone, G. Kuhlmann, Matthew, and Y. Liu. Keepaway soccer: From machine learning testbed to benchmark. In I. Noda, A. Jacoff, A. Bredenfeld, and Y. Takahashi, editors, *RoboCup-2005: Robot Soccer World Cup IX*. Springer Verlag, Berlin, 2006. (Cited on page 36.)

[Stoy *et al.*, 2010a] K. Stoy, D. Brandt, and D. Christensen. *Self-reconfigurable robots: an introduction*. Intelligent robotics and autonomous agents. MIT Press, 2010. (Cited on page 28.)

[Stoy *et al.*, 2010b] K. Stoy, R. Nagpal, and W.-M. Shen, editors. *Multi-Robot Organisms: State of the Art*, 2010. (Cited on page 11.)

[Sutton and Barto, 1998] R. Sutton and A. Barto. *Reinforcement Learning: An Introduction*. Adaptive Computation and Machine Learning. MIT Press, 1998. (Cited on pages 17 et 18.)

[Sutton, 1996] R. S. Sutton. Generalization in reinforcement learning: Successful examples using sparse coarse coding. In *Advances in Neural Information Processing Systems 8*, volume 8, pages 1038–1044, 1996. (Cited on page 38.)

[Suzuki and Akiyama, 2007] S. Suzuki and E. Akiyama. Evolution of indirect reciprocity in groups of various sizes and comparison with direct reciprocity. *Journal of Theoretical Biology*, 245(3):539–552, 2007. (Cited on page 88.)

[Szer *et al.*, 2005] D. Szer, F. Charpillet, and S. Zilberstein. Maa*: A heuristic search algorithm for solving decentralized pomdps. In *Proceedings of the twenty-first conference on uncertainty in artificial intelligence*, pages 576–590, 2005. (Cited on page 37.)

[Tabatabaee *et al.*, 2005] V. Tabatabaee, A. Tiwari, and J. K. Hollingsworth. Parallel parameter tuning for applications with performance variability. In *Proceedings of the 2005 ACM/IEEE conference on Supercomputing*, SC '05, page 57, Washington, DC, USA, 2005. IEEE Computer Society. (Cited on page 116.)

[Tang-Martinez, 2001] Z. Tang-Martinez. The mechanisms of kin discrimination and the evolution of kin recognition in vertebrates: a critical re-evaluation. *Behavioural Processes*, 53(1–2):21–40, 2001. (Cited on page 89.)

[Taylor *et al.*, 2008] R. H. Taylor, A. Menciassi, G. Fichtinger, and P. Dario. Medical robotics and computer-integrated surgery. In *Springer Handbook of Robotics*, Gale virtual reference library, pages 1199–1218. Springer, 2008. (Cited on page 10.)

[Tesla, 1898] N. Tesla. Method of and apparatus for controlling mechanism of moving vessels or vehicles, 11 1898. (Cited on page 6.)

[Theodoridis and Hu, 2006] T. Theodoridis and H. Hu. The fuzzy sarsa(lambda) learning approach applied to a strategic route learning robot behaviour. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 1767–1772, oct. 2006. (Cited on page 42.)

[Thrun *et al.*, 2005a] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. Intelligent Robotics and Autonomous Agents Series. Mit Press, 2005. (Cited on pages 16 et 154.)

[Thrun *et al.*, 2005b] S. Thrun, W. Burgard, and D. Fox. *Markov Decision Processes*, chapter 14. In *Intelligent Robotics and Autonomous Agents Series* [2005a], 2005. (Cited on page 16.)

[Thrun *et al.*, 2005c] S. Thrun, W. Burgard, and D. Fox. *Robot Motion*, chapter 5. In *Intelligent Robotics and Autonomous Agents Series* [2005a], 2005. (Cited on page 16.)

[Thrun, 2002] S. Thrun. Robotic mapping: A survey. In G. Lakemeyer and B. Nebel, editors, *Exploring Artificial Intelligence in the New Millenium*. Morgan Kaufmann, 2002. (Cited on page 16.)

[Tinbergen, 1964] N. Tinbergen. The evolution of signalling devices. In W. Etkin, editor, *Social Behavior and Organization Among Vertebrates*, pages 206–230. University of Chicago Press, Chicago, 1964. (Cited on page 118.)

[Tolley and Lipson, 2011] M. T. Tolley and H. Lipson. On-line assembly planning for stochastically reconfigurable systems. *Int. J. Rob. Res.*, 30(13):1566–1584, November 2011. (Cited on page 11.)

[Tomatis and Nourbakhsh, 2002] N. Tomatis and I. Nourbakhsh. Hybrid simultaneous localization and map building: closing the loop with multi-hypothesis tracking. In *Proc. 2002 IEEE Intl. Conf. on Robotics & Automation*, 2002. (Cited on page 16.)

[Touzet, 2004] C. F. Touzet. Distributed lazy q-learning for cooperative mobile robots. *International Journal of Advanced Robotic Systems*, 1(1):5–13, 2004. (Cited on pages 38 et 39.)

[Trianni *et al.*, 2008] V. Trianni, S. Nolfi, and M. Dorigo. Evolution, self-organization and swarm robotics. In C. Blum and D. Merkle, editors, *Swarm Intelligence*, Natural Computing Series, pages 163–191. Springer, 2008. (Cited on page 73.)

[Trivers, 1971] R. L. Trivers. The evolution of reciprocal altruism. *The Quarterly Review of Biology*, 46(1):35, 1971. (Cited on page 88.)

[Trueba *et al.*, 2011] P. Trueba, A. Prieto, P. Caamaño, F. Bellas, and R. J. Duro. Task-driven species in evolutionary robotic teams. In *Proceedings of the 4th international conference on Interplay between natural and artificial computation - Volume Part I*, IWINAC'11, pages 138–147, Berlin, Heidelberg, 2011. Springer-Verlag. (Cited on page 41.)

[Trueba *et al.*, 2012] P. Trueba, A. Prieto, F. Bellas, P. Caamaño, and R. J. Duro. Self-organization and specialization in multiagent systems through open-ended natural evolution. In C. D. Chio, A. Agapitos, S. Cagnoni, C. Cotta, F. F. de Vega, G. A. D. Caro, R. Drechsler, A. Ekárt, A. I. Esparcia-Alcázar, M. Farooq, W. B. Langdon, J. J. M. Guervós, M. Preuss, H. Richter, S. Silva, A. Simoes, G. Squillero, E. Tarantino, A. Tettamanzi, J. Togelius, N. Urquhart, S. Uyar, and G. N. Yannakakis, editors, *EvoApplications*, volume 7248 of *Lecture Notes in Computer Science*, pages 93–102. Springer, 2012. (Cited on page 41.)

[Urieli *et al.*, 2011] D. Urieli, P. MacAlpine, S. Kalyanakrishnan, Y. Bentor, and P. Stone. On optimizing interdependent skills: a case study in simulated 3d humanoid robot soccer. In *AAMAS'11*, pages 769–776, 2011. (Cited on page 39.)

[Usui and Arita, 2003] Y. Usui and T. Arita. Situated and embodied evolution in collective evolutionary robotics. In *In Proc. of the 8th International Symposium on Artificial Life and Robotics*, pages 212–215, 2003. (Cited on pages 41 et 42.)

[Varela *et al.*, 1974] F. Varela, H. Maturana, and R. Uribe. Autopoiesis: The organization of living systems, its characterization and a model. *Biosystems*, 5(4):187–196, 1974. (Cited on page 42.)

[Vaughan, 2008] R. Vaughan. Massively multi-robot simulation in Stage. *Swarm Intelligence*, 2(2–4):189–208, 2008. (Cited on page 62.)

[Waibel *et al.*, 2009] M. Waibel, L. Keller, and D. Floreano. Genetic team composition and level of selection in the evolution of multi-agent systems. *IEEE Transactions on Evolutionary Computation*, 2009. (Cited on pages 12 et 89.)

[Walker *et al.*, 2003] J. Walker, S. Garrett, and M. Wilson. Evolving controllers for real robots: A survey of the literature. *Adaptive Behavior*, 11(3):179–203, 2003. (Cited on page 18.)

[Walter, 1950] G. W. Walter. An Imitation of Life. *Scientific American*, May:42–45, 1950. (Cited on page 6.)

[Watson *et al.*, 2002] R. Watson, S. Ficici, and J. Pollack. Embodied evolution: Distributing an evolutionary algorithm in a population of robots, 2002. (Cited on pages 40 et 41.)

[Watson, 1999] R. A. Watson. Embodied evolution: Embodying an evolutionary algorithm in a population of robots. In *Congress on Evolutionary Computation*, volume 1, pages 335–342, 1999. (Cited on pages 29 et 40.)

[Weel *et al.*, 2012] B. Weel, E. Haasdijk, and A. Eiben. The emergence of multi-cellular robot organisms through on-line on-board evolution. In C. Chio, A. Agapitos, S. Cagnoni, C. Cotta, F. Vega, G. Caro, R. Drechsler, A. Ekárt, A. Esparcia-Alcázar, M. Farooq, W. Langdon, J. Merelo-Guervós, M. Preuss, H. Richter, S. Silva, A. Simoes, G. Squillero, E. Tarantino, A. Tettamanzi, J. Togelius, N. Urquhart, A. Uyar, and G. Yannakakis, editors, *Applications of Evolutionary Computation*, volume 7248 of *Lecture Notes in Computer Science*, pages 124–134. Springer Berlin Heidelberg, 2012. (Cited on pages 32 et 41.)

[Weizenbaum, 1966] J. Weizenbaum. Eliza - a computer program for the study of natural language communication between man and machine. *Commun. ACM*, 9(1):36–45, January 1966. (Cited on page 7.)

[Weizenbaum, 1976] J. Weizenbaum. *Computer Power and Human Reason: From Judgment to Calculation*. W. H. Freeman & Co., New York, NY, USA, 1976. (Cited on page 7.)

[Weng *et al.*, 1997] J. J. Weng, N. Ahuja, and T. S. Huang. Learning recognition and segmentation using the cresceptron. *Int. J. Comput. Vision*, 25(2):109–143, November 1997. (Cited on page 18.)

[Weng *et al.*, 2001] J. Weng, J. McClelland, A. Pentland, O. Sporns, I. Stockman, M. Sur, and E. Thelen. Artificial intelligence: Autonomous mental development by robots and animals. *Science*, 291:599–600, 2001. (Cited on page 18.)

[West *et al.*, 2007] S. A. West, A. S. Griffin, and A. Gardner. Social semantics: altruism, cooperation, mutualism, strong reciprocity and group selection. *Journal of Evolutionary Biology*, 20:415–432, 2007. (Cited on pages 87, 88 et 90.)

[Wilcoxon, 1945] F. Wilcoxon. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83, 1945. (Cited on page 108.)

[Wilkinson, 2000] S. Wilkinson. "gastrobots"—benefits and challenges of microbial fuel cells in food-powered robot applications. *Autonomous Robots*, 9:99–111, 2000. (Cited on page 13.)

[Williams *et al.*, 2008] P. Williams, R. Beer, and M. Gasser. Evolving referential communication in embodied dynamical agents. In *ALife XI: Proceedings of the Eleventh International Conference on Artificial Life*, pages 702–709, 2008. (Cited on page 118.)

[Wilson *et al.*, 1992] D. Wilson, G. Pollock, and L. Dugatkin. Can altruism evolve in purely viscous populations? *Evolutionary Ecology*, 6:331–341, 1992. (Cited on page 90.)

[Wilson, 1980] D. S. Wilson. *The Natural Selection of Populations and Communities.* Benjamin/Cummings, Menlo Park, 1980. (Cited on page 90.)

[Wilson, 1987] J. B. Wilson. Group selection in plant populations. *TAG Theoretical and Applied Genetics*, 74:493–502, 1987. (Cited on page 90.)

[Wilson, 1990] S. W. Wilson. The animat path to ai. In *Proceedings of the first international conference on simulation of adaptive behavior on From animals to animats*, pages 15–21, Cambridge, MA, USA, 1990. MIT Press. (Cited on page 17.)

[Winfield and Holland, 2000] A. F. T. Winfield and O. E. Holland. The application of wireless local area network technology to the control of mobile robots. *Microprocessors and Microsystems*, 23(10):597–607, 2000. (Cited on page 61.)

[Wischmann *et al.*, 2007] S. Wischmann, K. Stamm, and F. Wörgötter. Embodied evolution and learning: the neglected timing of maturation. In *Proceedings of the 9th European conference on Advances in artificial life*, ECAL'07, pages 284–293, Berlin, Heidelberg, 2007. Springer-Verlag. (Cited on page 41.)

[Wolff and Nordin, 2001] K. Wolff and P. Nordin. Evolution of efficient gait with autonomous biped robot using visual feedback. pages 482–489, San Francisco, California, USA, 9-11 July 2001. (Cited on page 27.)

[Wolpert and Macready, 1997] D. Wolpert and W. Macready. No free lunch theorems for optimization. *Evolutionary Computation, IEEE Transactions on*, 1(1):67–82, apr 1997. (Cited on page 15.)

[Wolpert and Tumer, 2001] D. H. Wolpert and K. Tumer. Optimal payoff functions for members of collectives. *Advances in Complex Systems*, 4(2/3):265–279, 2001. (Cited on page 85.)

[Wynne-Edwards, 1986] V. C. Wynne-Edwards. *Evolution through group selection.* Blackwell Scientific, Oxford, 1986. (Cited on page 89.)

[Yaeger, 1994] L. Yaeger. Computational genetics, physiology, metabolism, neural systems, learning, vision, and behavior or polyworld: Life in a new context, 1994. (Cited on page 43.)

[Yim *et al.*, 2003] M. Yim, K. Roufas, D. Duff, Y. Zhang, C. Eldershaw, and S. Homans. Modular reconfigurable robots in space applications. *Autonomous Robots*, 14:225–237, 2003. (Cited on page 11.)

[Yim *et al.*, 2007] M. Yim, W.-M. Shen, B. Salemi, D. Rus, M. Moll, H. Lipson, E. Klavins, and G. Chirikjian. Modular self-reconfigurable robot systems. *Robotics Automation Magazine, IEEE*, 14(1):43–52, march 2007. (Cited on page 28.)

[Zefran, 1994] M. Zefran. Review of the literature on time-optimal control of robotic manipulators. Technical report, University of Pennsylvania, School of Engineering and Applied Science, Department of Computer and Information Science, 1994. (Cited on page 15.)

[Zufferey *et al.*, 2002] J.-C. Zufferey, D. Floreano, M. van Leeuwen, and T. Merenda. Evolving vision-based flying robots. In *Proceedings of the Second International Workshop on Biologically Motivated Computer Vision*, BMCV '02, pages 592–600, London, UK, UK, 2002. Springer-Verlag. (Cited on page 27.)

[Zykov *et al.*, 2004] V. Zykov, J. Bongard, and H. Lipson. Evolving dynamic gaits on a physical robot. In *Proceedings of Genetic and Evolutionary Computation Conference (GECCO'04)*, 2004. (Cited on page 25.)

[Zykov *et al.*, 2008] V. Zykov, P. Williams, N. Lassabe, and H. Lipson. Molecubes extended: diversifying capabilities of opensource modular robotics. In *Proceedings of the IEEE Intelligent Robots and Systems 2008, Self-Reconfigurable Robotics Workshop*, 2008. (Cited on page 11.)