Université Libre de Bruxelles
Faculté des Sciences Appliquées
CoDE - Ingénierie de l'informatique et de la décision
IRIDIA - Institut de recherches interdisciplinaires
et de développements en intelligence artificielle

Incremental Social Learning in Swarm Intelligence Systems

Marco A. Montes de Oca Roldán

Promoteur:

Prof. Marco Dorigo

Directeur de Recherches du F.R.S.–FNRS
IRIDIA, CoDE, Université Libre de Bruxelles

Co-promoteur:

Dr. habil. Thomas Stützle

Chercheur Qualifié du F.R.S.–FNRS
IRIDIA, CoDE, Université Libre de Bruxelles

Thése présentée à la Faculté des Sciences Appliquées de l'Université Libre de Bruxelles en vue de l'obtention du titre de Docteur en Sciences de l'Ingenieur.

Année Académique 2010–2011

Abstract

A swarm intelligence system is a type of multiagent system with the following distinctive characteristics: (i) it is composed of a large number of agents, (ii) the agents that comprise the system are simple with respect to the complexity of the task the system is required to perform, (iii) its control relies on principles of decentralization and self-organization, and (iv) its constituent agents interact locally with one another and with their environment.

Interactions among agents, either direct or indirect through the environment in which they act, are fundamental for swarm intelligence to exist; however, there is a class of interactions, referred to as *interference*, that actually blocks or hinders the agents' goal-seeking behavior. For example, competition for space may reduce the mobility of robots in a swarm robotics system, or misleading information may spread through the system in a particle swarm optimization algorithm. One of the most visible effects of interference in a swarm intelligence system is the reduction of its efficiency. In other words, interference increases the time required by the system to reach a desired state. Thus, interference is a fundamental problem which negatively affects the viability of the swarm intelligence approach for solving important, practical problems.

We propose a framework called *incremental social learning* (ISL) as a solution to the aforementioned problem. It consists of two elements: (i) a growing population of agents, and (ii) a social learning mechanism. Initially, a system under the control of ISL consists of a small population of agents. These agents interact with one another and with their environment for some time before new agents are added to the system according to a predefined schedule. When a new agent is about to be added, it learns socially from a subset of the agents that have been part of the system for some time, and that, as a consequence, may have gathered useful information. The implementation of the social learning mechanism is application-dependent, but the goal is to transfer knowledge from a set of experienced agents that are already in the environment to the newly added agent. The process continues until one of the following criteria is met: (i) the maximum number of agents is reached, (ii) the assigned task is finished, or (iii) the system performs as desired. Starting with a small number of agents reduces interference because it reduces the number of interactions within the system, and thus, fast progress toward the desired state may be achieved. By learning socially, newly added agents acquire knowledge about their environment without incurring the costs of acquiring that knowledge individually. As a result, ISL can make a swarm intelligence system reach a desired state more rapidly.

We have successfully applied ISL to two very different swarm intelligence systems. We applied ISL to particle swarm optimization algorithms. The results of this study demonstrate that ISL substantially improves the performance of these kinds of algorithms. In fact, two of the resulting algorithms are competitive with state-of-the-art algorithms in the field. The second system to which we applied ISL exploits a collective decision-making mechanism based on an opinion formation model. This mechanism is also one of the original contributions presented in this dissertation. A swarm robotics system under the control of the proposed mechanism allows robots to choose from a set of two actions the action that is fastest to execute. In this case, when only a small proportion of the swarm is able to concurrently execute the alternative actions, ISL substantially improves the system's performance.

Acknowledgments

This dissertation represents the culmination of a process that started in late 2004, when I first contacted Marco Dorigo by email. From the second email on (he did not respond to the first email), I received his full support in practically everything I proposed to him. First, he supported my application for an Alβan scholarship, which was part of a European Union program of high-level scholarships for Latin America. His support was both academic and financial. Academically, he helped me define a project that would allow me to earn a Ph.D degree from the *Université Libre de Bruxelles*. Financially, Marco accepted to fund the remaining 33% of the costs of my initially 3-year long research project (for some reason, an Alβan scholarship did not fully fund the applicant). Eventually, I won the scholarship. While the application for the Alβan scholarship was being processed, we decided to apply for a CONACyT scholarship funded by the Mexican government. I remember being nervous about the fact that he had to sign documents in Spanish, which was a language he did not master. He did sign the documents nonetheless. I also won that scholarship. However, I declined it in order to accept Alβan scholarship No. E05D054889MX. That was the beginning of a journey that has had, and will continue to have, a tremendous impact on my professional and personal lives.

Here I am, five years and nine months after I arrived in Brussels, finally writing the last words of my dissertation. These last words have to be of gratitude because one never does anything truly by oneself. In other words, our actions are the result of past experiences that have been, to various extents, shaped by others (see the rest of this dissertation for a deeper discussion of this subject).

I want to thank Marco Dorigo, who played the role of my main advisor throughout these years. His academic and personal support were essential for the completion of my Ph.D. project. Also thanks to Marco, I was able to enjoy financial support from the ANTS and META-X projects, both of which were *Actions de Recherche Concertée* funded by the Scientific Research Directorate of the French Community of Belgium. I was also financially supported through the SWARMANOID project funded by the Future and Emerging Technologies programme (IST-FET) of the European Commission, under grant IST-022888, and by the VIRTUAL SWARMANOID project funded by the Fund for Scientific Research F.R.S.-FNRS of Belgium's French Community. I want to thank Thomas Stützle, my main co-advisor. Without Thomas, many of the ideas that are now presented in this dissertation would have been just ephemeral bits of inspiration that would have never seen the light of day. His constant support and encouragement were critical for the completion of my Ph.D. project. Thomas changed roles a number of times during these years. He started as my academic co-advisor. Then, he became my mentor. Now, he is also my friend. Mauro Birattari, my secondary co-advisor (just to give a name to his role), has always been critic of my work, and I thank him for that. Thanks to him, I learned that one should never be content with one's results. His deep knowledge about many fields allows him to take different perspectives from which to look at things. As a result, any work in which he is involved, is high-quality work. I wish I could have learned even more from him. I also want to thank Prof. Bernard Fortz, Dr. Rajan Filomeno Coelho, both from the *Université Libre de Bruxelles*, and Prof. Martin Middendorf from the *Universiät Leipzig*, for their comments and useful critique that helped me improve this dissertation.

My current and former fellow IRIDIAns deserve all my gratitude too. All of them,

Christos Ampatzis, Doğan Aydın, Prasanna Balaprakash, Hugues Bersini, Navneet Bhalla, Saifullah bin Hussin, Matteo Borrotti, Manuele Brambilla, Arne Brutschy, Alexandre Campo, Sara Ceschia, Marco Chiarandini, Anders Christensen, Alain Coletta, Muriel Decreton, Antal Decugnière, Giacomo Di Tollo, Jérémie Dubois-Lacoste, Stefan Eppe, Eliseo Ferrante, Gianpiero Francesca, Marco Frison, Matteo Gagliolo, António Gaspar-Cunha, Roderich Groß, Álvaro Gutiérrez-Martín, Takashi Kawakami, Stefanie Kritzinger, Manuel López-Ibáñez, Tom Lenaerts, Renaud Lenne, Tianjun Liao, Max Manfrin, Mattia Manfroni, Amin Mantrach, Bruno Marchal, Xavier Martínez-González, Franco Mascia, Nithin Mathews, Michael Maur, Prospero C. Naval, Jr, Shervin Nouyan, Rehan O'Grady, Sabrina Oliveira, Paola Pellegrini, Christophe Philemotte, Carlo Pinciroli, Giovanni Pini, Carlotta Piscopo, Jessica Rivero Espinosa, Andrea Roli, Fabio Rossi, Ruben Ruiz, Jodelson Sabino, Utku Salihoglu, Francesco Sambo, Francisco C. Santos, Alexander Scheidler, Krzysztof Socha, Paolo Spanevello, Alessandro Stranieri, Cristina Teixeira, Vito Trianni, Elio Tuci, Ali Emre Turgut, Colin Twomey, Thijs Urlings, David Weiss, and Zhi Yuan helped me in different ways throughout these years.

Five groups of people have been especially important for me during my stay in Brussels. The first group, composed of Doğan Aydın, Mauro Birattari, Barbara Di Camillo, Marco Dorigo, Andries Engelbrecht, Eliseo Ferrante, Manuel López-Ibáñez, Tianjun Liao, Nithin Mathews, Michael Maur, Sabrina Oliveira, Paola Pellegrini, Jorge Peña, Carlo Pinciroli, Francesco Sambo, Alexander Scheidler, Paolo Spanevello, Thomas Stützle, Ken Van den Enden, and Zhi Yuan contributed to my professional development through many fruitful collaborations that resulted in a number of publications during my Ph.D. Prasanna Balaprakash, Navneet Bhalla, Manuele Brambilla, Eliseo Ferrante, Max Manfrin, Nithin Mathews, Sabrina Oliveira, Carlo Pinciroli, Francesco Sambo, Francisco C. Santos, and Elio Tuci comprise the second group, which is composed of close friends. I thank them for being an extended family and giving me personal support during the time I lived in Brussels. The third group is composed of César A. Marín, Gisela Zamayoa, Lilia Pérez Romero, and Yumi (a little furry friend). They are my best Mexican friends in Europe. The fourth group is my beloved family. I thank my mom, my brother, and my sister-in-law for their unwavering love and support. The fifth group is a singleton whose only member is Caitlin J. deWilde, who has mostly been in the United States, patiently waiting for me. I thank her for all her support and encouragement, and above all, I thank her for her love during the past two years.

Marco A. Montes de Oca
June 28, 2011
Brussels, Belgium

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

The term *swarm intelligence* refers to the group-level intelligence that some groups of animals exhibit in nature (Bonabeau et al., 1999; Dorigo and Birattari, 2007; Garnier et al., 2007a). Famous examples of the swarm intelligence exhibited by some groups of animals are the ability of swarms of bees to choose the best site on which to build their nest (Seeley, 2010) or the ability of ant colonies to find the shortest path between their nest and a food source (Goss et al., 1989). A fundamental characteristic of a group exhibiting swarm intelligence is its ability to solve problems that the group's constituent members cannot solve individually. This fact has made scientists wonder whether it is possible to design problem-solving techniques or systems that use many, yet simple, constituent parts – referred to as *agents*[1]. A first wave of advances in swarm intelligence research led to the development of successful optimization techniques such as ant colony optimization (ACO) (Dorigo et al., 1991a,b; Dorigo, 1992; Dorigo et al., 1996; Dorigo and Di Caro, 1999; Bonabeau et al., 2000; Dorigo and Stützle, 2004; Dorigo, 2007) and particle swarm optimization (PSO) (Kennedy and Eberhart, 1995; Kennedy et al., 2001; Engelbrecht, 2005; Clerc, 2006; Poli et al., 2007; Dorigo et al., 2008). In this first wave of advances, swarm intelligence was also investigated in the context of multi-robot systems (Deneubourg et al., 1990b; Holland and Melhuish, 1999; Dorigo et al., 2004; Beni, 2005).

Most artificial swarm intelligence systems in existence today were inspired by natural swarms. For example, the foraging behavior of ants inspired the design of ACO (Dorigo and Stützle, 2004), and the flocking behavior of birds inspired the design of PSO (Kennedy and Eberhart, 1995). Likewise, in swarm robotics research it is possible to find complete research projects inspired by the way social insects, in particular ants, cooperate to solve problems (see e.g., Dorigo et al. (2004); Kernbach et al. (2008)). Despite the differences among these systems, their constituent agents share a common behavioral trait: they are usually *searching agents*, that is, they are agents that are continuously in search of a target state. What agents search for depends on the purpose of the system. For example, in ACO, the agents that form the swarm (called "colony" in the context of ACO) search for solutions to combinatorial optimization problems. In PSO, agents search for solutions to continuous optimization problems. In swarm robotics, the searching behavior of robots can be more elusive, but in many cases, it involves searching for a desired individual or collective state. For example, in the work of Turgut et al. (2008) or Trianni and Nolfi (2009), robots are continuously searching for a state that makes the swarm cohesive in space (flocking) or time (synchronization), respectively.

Swarm intelligence is the result of agents interacting with each other and with their environment. At the same time, however, sharing information and an environment with other agents produces negative interactions that we refer to as *interference*. This class of interactions blocks or hinders an agent's behavior. As a result of interference, the speed at which a swarm intelligence system reaches a desired state will be reduced. Importantly, interference will tend to increase with the size of the system as a result of the fact that

[1] Throughout this dissertation, we will use the word *agent* to generically refer to an entity, be it an animal or an artifact, such as a robot or a piece of software, capable of autonomous perception and action.

interference is a function of the number of interactions within a system. Thus, interference hinders the scalability of swarm intelligence systems.

Two examples will help us illustrate how interference reduces the performance of swarm intelligence systems. We first consider a PSO algorithm, in which a swarm of agents (called particles) exchange information with one another in order to bias their search toward the best points they find in the search space of a continuous optimization problem. Although cooperation is fundamental for the success of the algorithm, it is also a source of interference, especially during the first iterations of the algorithm. The mutual influence that particles exert on each other makes them move to regions that do not contain the optimal solution to the problem. If the swarm of particles is large, the number of objective function evaluations spent in this initial phase will also be large, and thus, the time needed by the swarm to start making progress toward good solutions will increase. As a second example, we consider a swarm robotics system in which robots have to search for a resource. Since the environment in which they move has finite dimensions, robots have to continuously avoid collisions with each other. If the swarm of robots is large, the space between robots may be such that robots spend most of their time and energy unproductively by avoiding collisions rather than completing their assigned tasks. The overall effect of interference in this example is also to slow down progress toward a desired state.

1.1 Objective

The main objective of the work presented in this dissertation is to reduce the effects of interference in swarm intelligence systems composed of multiple searching agents. Since interference manifests itself as an influence that slows down progress toward a desired state, reducing its effects helps a swarm intelligence system to reach a desired state more rapidly.

To meet the aforementioned objective, in this dissertation we introduce the *incremental social learning* (ISL) framework. This framework consists of two elements: (i) an initially small population of agents that grows over time, and (ii) a social learning process whereby new agents learn from more experienced ones. A small population of agents would reach a certain state more rapidly than a large population because of the reduced interference. However, it is possible that a small swarm cannot reach the desired state. For example, imagine a scenario in which too few robots cannot move a heavy object. We tackle this problem by adding agents to the swarm according to some predefined criterion. An agent that is added to the swarm learns from the agents that have been in the swarm for some time. This element of ISL is attractive because new agents acquire knowledge from more experienced ones without incurring the costs of acquiring that knowledge individually. Thus, ISL allows the new agents to save time that they can use to perform other tasks. After the inclusion of a new agent, the swarm needs to re-adapt to the new conditions; however, the agents that are part of the swarm do not need to start from scratch because some useful work would have already been completed.

1.2 Methodology

We considered two case studies of the application of the incremental social learning framework to swarm intelligence systems:

1. **Swarm intelligence for continuous optimization.** We considered PSO algorithms as a case study to measure the effectiveness of ISL. As a result, three PSO-based optimization algorithms are proposed. Two of these algorithms obtain results comparable with those obtained by other state-of-the-art continuous optimization algorithms. The development and analysis of these algorithms is presented in Chapter 4.

2. **Swarm intelligence for robotics.** As a second case study, we considered a swarm intelligence system in which robots perform a foraging task that involves collective

transport. In this task, robots need to choose one of two available paths to a storage room for transported objects. In this second case study, we first developed a collective decision-making mechanism that allows a swarm of robots to select the shortest path. Then, we instantiated the incremental social learning framework using the aforementioned decision-making mechanism as the searching algorithm used by the swarm. The collective decision-making mechanism and its combination with ISL are presented in Chapter 5.

In both case studies, the application of the incremental social learning framework resulted in a substantial improvement of the underlying system's performance. These successes should be taken as proof of concept. Our experiments are not formal proof that the approach will always produce positive results. However, some requirements that the underlying swarm intelligence system should satisfy in order to expect benefits from the application of ISL are proposed.

1.3 Contributions

In this dissertation, the following three contributions are presented:

1. **Incremental social learning framework**. This original framework aims to tackle interference in swarm intelligence systems. Since such systems are usually composed of a large number of interacting agents, interference can be a major problem because the effects of interference are stronger when a large population of agents is involved. The incremental social learning framework addresses this problem by making a swarm intelligence system start with a small population and by letting new agents learn from more experienced agents.

2. **High-performance PSO algorithms**. A number of high-performance PSO algorithms are proposed in this dissertation. Two of these algorithms are the result of the instantiation of the incremental social learning framework in the context of PSO algorithms. These algorithms are identified by the names IPSOLS and IPSOLS+. They are PSO algorithms with a growing population size in which individual and social learning are simulated through local search and biased initialization, respectively. The third algorithm, which is not based on the incremental social learning framework, is presented in Appendix A. This algorithm, called Frankenstein's PSO, is an integration of algorithmic components that were found to provide good performance in an extensive empirical evaluation of PSO algorithms.

3. **Self-organized collective decision-making mechanism for swarms of robots**. A self-organized collective-decision making mechanism with application to swarm robotics is proposed. Positive feedback and a consensus-building procedure are the key elements of this mechanism that allows a population of robots to select the fastest-to-execute action from a set of alternatives, thus improving the efficiency of the system. We apply the incremental social learning framework to this mechanism in order to make it more efficient in situations where a small fraction of the swarm can concurrently execute the available alternative actions.

1.4 Publications

A number of publications have been produced during the development of the research work presented in this dissertation. Many of these publications have been written in collaboration with colleagues under the supervision of Prof. Marco Dorigo and/or Dr. Thomas Stützle.

The publications associated with this dissertation are listed below. The majority of them deal with the incremental social learning framework and its applications. However,

we have also listed publications that laid the ground for the development of the incremental social learning framework.

1.4.1 International Journals

1. Montes de Oca, M. A., Ferrante, E., Scheidler, A., Pinciroli, C., Birattari, M., and Dorigo, M. (2010b). Majority-rule opinion dynamics with differential latency: A mechanism for self-organized collective decision-making. Technical Report TR/IRIDIA/2010-023, IRIDIA, Université Libre de Bruxelles, Brussels, Belgium. [**Revision submitted to Swarm Intelligence**]

2. Montes de Oca, M. A., Aydın, D., and Stützle, T. (2011a). An incremental particle swarm for large-scale optimization problems: An example of tuning-in-the-loop (re)design of optimization algorithms. *Soft Computing*. Forthcoming. DOI: 10.1007/s00500-010-0649-0

3. Montes de Oca, M. A., Stützle, T., Van den Enden, K., and Dorigo, M. (2011b). Incremental social learning in particle swarms. *IEEE Transactions on Systems, Man and Cybernetics - Part B: Cybernetics*, 41(2):368–384

4. Montes de Oca, M. A., Stützle, T., Birattari, M., and Dorigo, M. (2009c). Frankenstein's PSO: A composite particle swarm optimization algorithm. *IEEE Transactions on Evolutionary Computation*, 13(5):1120–1132

5. Dorigo, M., Montes de Oca, M. A., and Engelbrecht, A. P. (2008). Particle swarm optimization. *Scholarpedia*, 3(11):1486

1.4.2 International Conferences, Workshops and Symposia

1. Liao, T., Montes de Oca, M. A., Aydın, D., Stützle, T., and Dorigo, M. (2011). An incremental ant colony algorithm with local search for continuous optimization. In Krasnogor, N. et al., editors, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2011)*. ACM Press, New York. To appear. Preprint available at `http://iridia.ulb.ac.be/IridiaTrSeries/IridiaTr2011-005r002.pdf` [**Nominated for the best paper award in the Ant Colony Optimization and Swarm Intelligence track**]

2. Montes de Oca, M. A., Stützle, T., Birattari, M., and Dorigo, M. (2010c). Incremental social learning applied to a decentralized decision-making mechanism: Collective learning made faster. In Gupta, I., Hassas, S., and Rolia, J., editors, *Proceedings of the Fourth IEEE Conference on Self-Adaptive and Self-Organizing Systems (SASO 2010)*, pages 243–252. IEEE Computer Society Press, Los Alamitos, CA

3. Montes de Oca, M. A., Ferrante, E., Mathews, N., Birattari, M., and Dorigo, M. (2010a). Opinion dynamics for decentralized decision-making in a robot swarm. In Dorigo, M. et al., editors, *LNCS 6234. Proceedings of the Seventh International Conference on Swarm Intelligence (ANTS 2010)*, pages 251–262. Springer, Berlin, Germany [**Nominated for the best paper award**]

4. Yuan, Z., Montes de Oca, M. A., Stützle, T., and Birattari, M. (2010). Modern continuous optimization algorithms for tuning real and integer algorithm parameters. In Dorigo, M. et al., editors, *LNCS 6234. Proceedings of the Seventh International Conference on Swarm Intelligence (ANTS 2010)*, pages 204–215. Springer, Berlin, Germany

5. Montes de Oca, M. A., Ferrante, E., Mathews, N., Birattari, M., and Dorigo, M. (2009a). Optimal collective decision-making through social influence and different action execution times. In Curran, D. and O'Riordan, C., editors, *Proceedings of the*

Workshop on Organisation, Cooperation and Emergence in Social Learning Agents of the European Conference on Artificial Life (ECAL 2009). No formal proceedings published

6. Montes de Oca, M. A., Van den Enden, K., and Stützle, T. (2008). Incremental particle swarm-guided local search for continuous optimization. In Blesa, M. J. et al., editors, *LNCS 5296. Proceedings of the International Workshop on Hybrid Metaheuristics (HM 2008)*, pages 72–86. Springer, Berlin, Germany

7. Montes de Oca, M. A. and Stützle, T. (2008b). Towards incremental social learning in optimization and multiagent systems. In Rand, W. et al., editors, *Workshop on Evolutionary Computation and Multiagent Systems Simulation of the Genetic and Evolutionary Computation Conference (GECCO 2008)*, pages 1939–1944. ACM Press, New York

8. Montes de Oca, M. A. and Stützle, T. (2008a). Convergence behavior of the fully informed particle swarm optimization algorithm. In Keijzer, M. et al., editors, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2008)*, pages 71–78. ACM Press, New York [**Nominated for the best paper award in the Ant Colony Optimization, Swarm Intelligence, and Artificial Immune Systems track**]

9. Montes de Oca, M. A., Stützle, T., Birattari, M., and Dorigo, M. (2006a). A comparison of particle swarm optimization algorithms based on run-length distributions. In Dorigo, M. et al., editors, *LNCS 4150. Proceedings of the Fifth International Workshop on Ant Colony Optimization and Swarm Intelligence (ANTS 2006)*, pages 1–12. Springer, Berlin, Germany

10. Montes de Oca, M. A., Stützle, T., Birattari, M., and Dorigo, M. (2006b). On the performance analysis of particle swarm optimisers. *AISB Quarterly*, 124:6–7

1.5 Structure

This dissertation consists of six chapters and one appendix. In Chapter 2, we provide relevant background information for the rest of the dissertation. In Chapter 3, we present the rationale and the algorithmic structure of the incremental social learning framework as well as a discussion of related work. The application of ISL to PSO algorithms is described in Chapter 4. First, we present a simple incremental PSO algorithm, called IPSO. Then, we present two high-performing PSO algorithms, called IPSOLS and IPSOLS+, that are derived from it. In Chapter 5, we present the application of ISL to a swarm robotics system. First, we describe the actual swarm robotics system the framework is applied to. Then, we describe the application of ISL to this system. Finally, in Chapter 6, we present the main conclusions of the research work documented in this dissertation. Appendix A is devoted to the description of *Frankenstein's PSO* algorithm, which is the result of an extensive experimentation with several PSO algorithms. Results of those experiments inspired in part some features of the ISL framework.

Chapter 2

Background

In this chapter, we present some of the basic concepts of swarm intelligence and social learning, which are central to our work. In Section 2.1, we present the concept of swarm intelligence, and describe its principles and mechanisms. We also describe the most successful artificial swarm intelligence systems together with the natural phenomena that inspired their development. In Section 2.2, we present the concepts of individual and social learning, and describe the main mechanisms involved in social learning.

2.1 Swarm Intelligence

In nature, different kinds of animals tend to congregate in large numbers. For instance, European starlings can gather in thousands to form flocks (Carere et al., 2009), atlantic silversides form schools of hundreds of individuals (Partridge, 1982), and ants make colonies that range in size from a few dozen to millions of ants (Hölldobler and Wilson, 1990). When animals form these *swarms*, they are often able to solve problems that no single member could if it acted alone. From an external observer's point of view, it may appear as if the swarm possessed a certain level of intelligence that is well superior to that of any of its constituent members. This collective-level intelligence is called *swarm intelligence*.

The size and behavior of swarms have fascinated humans since antiquity. At times, swarms inspire fear. For example, it is written in the Bible that swarms of locusts plagued Egypt (Exodus:10.3–6). At other times, swarms inspire respect. An old Mesoamerican legend tells the story of how ants helped the gods feed all humans with cultivated maize (Nuttall, 1930). Both extremes of feelings, fear and awe, have motivated researchers to wonder whether it is possible to control swarms. On the one hand, controlling swarms would allow us to alleviate the effects of plagues, like those of locusts or termites (Buhl et al., 2006). On the other hand, controlling swarms would allow us to devise techniques that can be used to control man-made artifacts such as robots or software agents (Bonabeau et al., 1999). However, before we are able to control swarms, we need to understand their governing principles.

2.1.1 Principles and Mechanisms

Even though the characteristics of swarm-forming animals vary substantially, swarms exhibit behaviors that are in fact very similar. This similarity has pointed toward the existence of a set of general principles responsible for the emergence of swarm-level organization and intelligence (Buhl et al., 2006). The existence of these principles makes the design of artificial swarm intelligence systems possible. Thus, as a discipline, swarm intelligence has a twofold objective. First, it aims to understand the fundamental principles that are the responsible for the collective-level intelligence sometimes exhibited by large groups of animals. Second, it aims to define engineering methodologies for the design and construction

of large groups of man-made entities that collectively solve practical problems (Dorigo and Birattari, 2007).

Researchers have made progress in the study of swarm intelligence and a set of principles and mechanisms that make it possible have been identified. The principles and mechanisms that we will describe have been found to operate in many animal societies, but especially in social insects groups (Bonabeau et al., 1999; Garnier et al., 2007a; Beekman et al., 2008).

Decentralization

The behavior exhibited by a swarm is not dictated by any central authority. The unfortunate name given to the reproductive member of an ant colony or a bee hive (i.e., a "queen") gives the impression that the organization observed at the collective level is the result of a hierarchical command structure. However, it is now well known that such a structure does not exist (Bonabeau, 1998; Garnier et al., 2007a). In a swarm, no single agent supervises the actions of, or issues orders to, other members of the swarm. The perception and interaction scope of a swarm member is local. Thus, the swarm's organization is the result of local interactions, both among the swarm members and between the swarm members and the environment.

Stigmergy

The theory of stigmergy (from the Greek roots *stigma*, which means mark, sign, or puncture, and *ergon*, which means action, labor, or work) was proposed by Grassé (1959) in the context of task coordination and nest construction regulation in colonies of termites. Grassé defined stigmergy as "the stimulation of the workers by the very performances they have achieved" (Grassé, 1959) p. 79. In other words, stigmergy refers to the coordination process that arises when an agent performs an action as a consequence of stimuli that are the result of another agent's – or possibly the same agent's – actions.

Stigmergy is key to explain how termites and other social insects are able to build structures and produce collective-level patterns that are orders of magnitude larger than a single individual, all without a central authority or global blueprint. For example, the construction of soil arches in termite nests starts when a termite fortuitously places a soil pellet on top of other pellets. This bigger soil structure stimulates termites to keep placing pellets on top. A self-reinforcing process then follows: the larger the structure, the stronger the attraction termites feel toward that structure to deposit soil pellets. Eventually an arch is built if two pillars happen to be at an appropriate distance. Another prominent example of how stigmergy enables swarm intelligence to occur is the ability of ants of some species to find the shortest path between their nest and a food source. While moving, ants deposit on the ground chemical substances called *pheromones*. These pheromones modify the environment and trigger a change in the behavior of ants. In particular, ants become attracted to areas of the environment marked with pheromones. This pheromone laying and following behavior induces a positive feedback process whereby areas with high concentration of pheromones become more and more attractive as more ants follow them (Pasteels et al., 1987; Goss et al., 1989; Deneubourg et al., 1990a). As a result, if there are several paths to the same food source, the colony is more likely to select the shortest path because ants will traverse it faster, and thus, it will have a higher pheromone concentration than longer ones.

Self-Organization

The theory of self-organization has found applications in such diverse fields as economics, urbanism, physics, chemistry, and biology (Haken, 2008). For example, it has been used to explain chemical reactions, such as the Belousov-Zhabotinsky reaction (Zhabotinsky, 2007), and the organization of cities (Portugali, 2000). In biology, it has been used to explain the external patterns on the skin or on the protective shells of some animals (Camazine et al., 2001), the movement of vertebrates in crowds (Couzin and Krause, 2003), and, most

relevant for our discussion here, the behavior of social insects swarms (Bonabeau et al., 1997).

Self-organization is a term with different meanings in different contexts (Gershenson, 2007). In this dissertation, we adopt Camazine et al.'s definition:

Definition *Self-organization is a process in which [a] pattern at the global level of a system emerges solely from numerous interactions among the lower-level components of the system. Moreover, the rules specifying interactions among the system's components are executed using only local information, without reference to the global pattern.* (Camazine et al., 2001) p. 8.

With this definition, some forms of swarm intelligence can be considered to be the result of self-organization. For example, the ability of ant colonies to find the shortest path between their nest and a food source can be seen as a self-organization process. First, a pheromone trail that connects an ant colony nest to a food source is the pattern at the global level cited in Camazine et al.'s definition. Such a trail is the result of several ants reinforcing it every time they traverse it, that is, it is the result of multiple interactions among the system's components (the ants). Stigmergy is in this case the interaction mechanism. The pheromone-laying and pheromone-following behavior exhibited by ants serves as an interaction rule, which is triggered only when an ant perceives pheromones in its vicinity. Finally, the behavioral rules followed by ants do not make any reference to pheromone trails and do not encode desired goals such as finding shortest paths. The shortest path between an ant colony's nest and a food source is an emergent pattern.

Self-organization is itself the result of the interaction of several processes and elements. According to Camazine et al. (2001) and Moussaid et al. (2009), these processes and elements are the following:

1. Multiple direct or indirect interactions among the system's components. By definition, a self-organizing system is composed of a number of components whose behavior depends on the state of their immediate environment or on the information they possess. In such a setting, the system's components mutually influence each other because the behavior of one of them may affect the environment of, or the information perceived by, other components. If the system's components are able to communicate directly with each other, it is also possible to influence the behavior of these components via direct communication.

2. Presence of fluctuations. The components of a self-organizing system may be subject to external perturbations or may behave nondeterministically. As a result, there may be fluctuations in the system's state. For example, in the absence of pheromone trails, an ant chooses a walking direction at random, or an ant colony may suffer the sudden loss of several members due to the presence of predators or inclement weather conditions.

3. Positive feedback. Fluctuations, random or not, are often reinforced in self-organizing systems. The way termites construct pillars with soil pellets or the reinforcement of pheromone trails by ants are examples of positive feedback processes. Positive feedback is responsible for the appearance of new structures (e.g., pillars or pheromone trails) that in turn modify the behavior of the system.

4. Negative feedback. The self-reinforcing process brought about by positive feedback loops must be limited. It is impossible, for example, that the concentration of pheromones in an ant trail grows to infinity. In self-organizing systems this task is performed by a so-called negative feedback process. Negative feedback encompasses all limiting environmental factors and a system's internal regulation processes. In the

ant trails example, negative feedback includes pheromone evaporation, food depletion and satiation.

5. Bifurcations and multiple stable states. Self-organizing systems often show abrupt changes in their behavior without an abrupt change in the value of a control parameter. For example, the density of insects is often a parameter that affects the behavior of a swarm. Below a certain threshold, no swarm behavior is observed, whereas above it, a swarm behavior suddenly appears (Buhl et al., 2006). A self-organizing system will reach a stable state which depends on the initial conditions. Since self-organization is often triggered by random fluctuations, the stable state of a system may be just one of several available states.

Currently, there is a growing interest in developing methodologies for the design and control of self-organizing systems (see, for example, Gershenson (2007); Di Marzo Serugendo et al. (2004); Bruecker et al. (2005, 2006, 2007)). The knowledge gained in the process will certainly affect our ability to design and control swarm intelligence systems.

Other Mechanisms

Self-organization can account for many swarm intelligence behaviors, but they may also be the result of other mechanisms, either alone or in combination with a self-organizing process (Camazine et al., 2001; Johnson, 2009). Some of these mechanisms are leadership, blueprints, recipes, templates, or threshold-based responses (Bonabeau, 1998; Camazine et al., 2001; Garnier et al., 2007a). Leadership may play a role when some individuals are more experienced than others or simply when there are better informed individuals. This mechanism, as we will discuss in Chapter 3, is important in the framework proposed in this dissertation. Leadership plays an important role in large groups of moving animals as suggested by recent studies (Couzin et al., 2005). Blueprints are usually associated with the process of constructing a structure. They are representations of the desired structure; however, they do not specify how such a structure should be built. There is an ongoing debate as to whether blueprints are actually used by building animals; however, it is definitely possible to imagine man-made swarm intelligence systems in which agents use such a mechanism. Recipes are step-by-step directions to carry out a task. The execution of a recipe often ignores feedback from the execution process. This aspect of recipes is fundamental in order to distinguish them from stigmergic task execution, in which the execution of an action modifies the environment providing feedback to the acting animal or agent. A template is a kind of "preexisting pattern" in the environment that elicits a specific response from the members of a swarm, normally to actually build over them. For example, termites build a chamber around the body of the queen which produces a pheromone gradient that serves as a template (Bonabeau et al., 1998). Finally, in a threshold-based mechanism, an action is performed as a response to the strength of a stimulus. Threshold-based models have been used in the context of social insects to explain division of labor (Theraulaz et al., 1998), the mechanism whereby insects split responsibilities, as well as to explain collective phenomena in humans (Granovetter, 1978).

2.1.2 Artificial Swarm Intelligence Systems

The design and construction of artificial swarm intelligence systems have been heavily inspired by the behavior of natural swarms. The first efforts toward the development of artificial swarm intelligence systems began in the 1990s with pioneering works in robotics, data mining, and optimization. In fact, these domains are still the application areas of most artificial swarm intelligence systems (Dorigo and Birattari, 2007).

In the remainder of this section, we describe some of the most successful swarm intelligence systems devised to date.

Ant Colony Optimization

The ants' pheromone trail laying and trail following behavior described in Section 2.1.1 inspired the development of ant colony optimization (ACO) (Dorigo et al., 1991a,b; Dorigo, 1992; Dorigo et al., 1996; Dorigo and Di Caro, 1999; Bonabeau et al., 2000; Dorigo and Stützle, 2004; Dorigo, 2007). Some aspects of the real behavior of ants that allows them to find shortest paths in nature are simulated in ACO algorithms in order to tackle optimization problems. In nature, real ants form pheromone trails; in ACO, artificial ants construct candidate solutions to the problem instance under consideration. Solution construction is a stochastic process biased by artificial pheromone trails and possibly by available heuristic information based on the input data of the instance being solved. Pheromones are simulated as numerical information associated with appropriately defined solution components. A positive feedback process implemented by iterative modifications of the artificial pheromone trails is key for all ACO algorithms. In ACO algorithms, pheromone trails can be thought of as a function of the ants' search experience. The goal of positive feedback is to bias the colony towards the most promising solutions.

The ACO metaheuristic (Dorigo and Di Caro, 1999; Dorigo et al., 1999) is an algorithmic framework that allows the implementation of the aforementioned ideas for the approximate solution of optimization problems. Such a framework needs to be instantiated into an algorithm in order to tackle a specific problem. The framework is flexible enough to accommodate specialized problem-solving techniques.

ACO is commonly used to solve combinatorial optimization problems. A formal definition of a combinatorial optimization problem is given next.

Definition *A combinatorial optimization problem is modeled by the tuple (S, f, Ω), where:*

- *S is the set of candidate solutions defined over a finite set of discrete decision variables $\mathbb{X}$. S is referred to as the search space of the problem being tackled;*
- *$f : S \to \mathbb{R}$ is an objective function to be minimized;*[1]
- *Ω is a (possibly empty) set of constraints among the decision variables.*

A decision variable $X_i \in \mathbb{X}$, with $i = 1, \ldots, n$, is said to be instantiated when a value v_i^j that belongs to its domain $D_i = \left\{v_i^1, \ldots, v_i^{|D_i|}\right\}$ is assigned to it. A solution $s \in S$ is called feasible *if each decision variable has been instantiated satisfying all constraints in the set Ω. Solving the optimization problem requires finding a solution s^* such that $f(s^*) \leq f(s)\ \forall s \in S$, while satisfying all constraints in Ω.*

Three high-level procedures compose ACO (see Algorithm 1):

- **ConstructSolutions**. This procedure implements the artificial ants' incremental construction of candidate solutions.

  In ACO, an instantiated decision variable $X_i \leftarrow v_i^j$ is called a solution component $c_{ij} \in C$, where C denotes the set of solution components. A pheromone trail value τ_{ij} is associated with each component $c_{ij} \in C$.

  A solution construction starts from an initially empty partial solution s^p. At each construction step, it is extended by appending to it a feasible solution component from the set of feasible neighbors $N(s^p) \subseteq C$ that satisfies the constraints in Ω. The choice of a solution component is guided by a stochastic decision policy, which is biased by both the pheromone trail and the heuristic values associated with c_{ij}. The exact rules for the probabilistic choice of solution components vary across different ACO variants. The rule proposed in the Ant System algorithm (Dorigo et al., 1996) is the best known rule:

[1] Note that minimizing the value of an objective function f is the same as maximizing the value of $-f$; hence, every optimization problem can be described as a minimization problem.

Algorithm 1 Basic structure of an ant colony optimization algorithm

repeat
  ConstructSolutions
  DaemonActions /* Optional */
  UpdatePheromones
until Stopping criterion is satisfied

$$p_{c_{ij}|s^p} = \frac{[\tau_{ij}]^\alpha \cdot [\eta_{ij}]^\beta}{\sum_{c_{il} \in N(s^p)} [\tau_{il}]^\alpha \cdot [\eta_{il}]^\beta}\,, \tag{2.1}$$

where τ_{ij} and η_{ij} are, respectively, the pheromone value and the heuristic value associated with the component c_{ij}. The parameters $\alpha > 0$ and $\beta > 0$ determine the relative importance of pheromone versus heuristic information.

- **DeamonActions**. This procedure, although optional, is important when state-of-the-art results are sought (Dorigo and Stützle, 2004). It allows for the execution of problem-specific operations, such as the use of local search procedures, or of centralized actions that cannot be performed by artificial ants. It is usually executed before the update of pheromone values so that ants bias their search toward high quality solutions.

- **UpdatePheromones**. This procedure updates the pheromone trail values associated with the solution components in the set C. The modification of the pheromone trail values is composed of two stages: (i) *pheromone evaporation*, which decreases the pheromone values of all components by a constant factor ρ (called *evaporation rate*) in order to avoid premature convergence, and (ii) *pheromone deposit*, which increases the pheromone trail values associated with components of a set of promising solutions S_{upd}. The general form of the pheromone update rule is as follows:

$$\tau_{ij} \leftarrow (1-\rho) \cdot \tau_{ij} + \rho \cdot \sum_{s \in S_{upd} | c_{ij} \in s} F(s)\,, \tag{2.2}$$

where $\rho \in (0, 1]$ is the evaporation rate, and $F : S \rightarrow \mathbb{R}^+$ is a function such that $f(s) < f(s') \Rightarrow F(s) \geq F(s')$, $\forall\ s \neq s' \in S$. $F(\cdot)$ is called the fitness function. Different definitions for the set S_{upd} exist. Two common choices are $S_{upd} = s_{bsf}$, and $S_{upd} = s_{ib}$, where s_{bsf} is the best-so-far solution, that is, the best solution found since the start of the algorithm, and s_{ib} is the best solution of the current iteration. The specific implementation of the pheromone update mechanism differs across ACO variants (Dorigo et al., 1991a,b, 1996; Dorigo and Gambardella, 1997; Gambardella and Dorigo, 1996; Stützle and Hoos, 2000).

Many ACO algorithms have been proposed. Some of them aim to solve specific problems, and others have a more general purpose. In Table 2.1, we list some of the most representative ACO algorithms proposed to date.

Particle Swarm Optimization

Particle swarm optimization (PSO) (Kennedy and Eberhart, 1995; Eberhart and Kennedy, 1995; Kennedy et al., 2001; Engelbrecht, 2005; Clerc, 2006; Poli et al., 2007; Dorigo et al., 2008) is a population-based stochastic optimization technique primarily used to tackle continuous optimization problems. A continuous optimization problem is defined as follows:

Table 2.1: Representative ACO works.

ACO algorithm	*Main references*
Ant System (AS)	(Dorigo et al., 1991b; Dorigo, 1992; Dorigo et al., 1996)
Elitist AS	(Dorigo et al., 1991b; Dorigo, 1992; Dorigo et al., 1996)
Ant-Q	(Gambardella and Dorigo, 1995)
Ant Colony System (ACS)	(Dorigo and Gambardella, 1997; Gambardella and Dorigo, 1996)
$\mathcal{MAX}$–$\mathcal{MIN}$ Ant System ($\mathcal{MM}$AS)	(Stützle and Hoos, 1996, 1997, 2000)
Rank-based AS	(Bullnheimer et al., 1999)
ANTS	(Maniezzo, 1998, 1999)
Best-worst AS	(Cordón et al., 2002, 2000)
Population-based ACO	(Guntsch and Middendorf, 2002)
Beam-ACO	(Blum, 2004, 2005)

Definition *Given a set* $\Theta \subseteq \mathbb{R}^n$ *and an objective function* $f : \Theta \to \mathbb{R}$*, the continuous optimization problem consists in finding at least one member of the set*

$$\Theta^* = \arg\min_{\boldsymbol{\theta} \in \Theta} f(\boldsymbol{\theta}) = \{\boldsymbol{\theta}^* \in \Theta \colon f(\boldsymbol{\theta}^*) \leq f(\boldsymbol{\theta}), \quad \forall \boldsymbol{\theta} \in \Theta\}\,.$$

The set Θ *is referred to as the feasible solution space or as the search space of function* f*. If* $\Theta = \mathbb{R}^n$*, then the problem is called an unconstrained continuous optimization problem. Otherwise, the problem is called a constrained continuous optimization problem.*

PSO has roots in computer graphics, social psychology, and natural swarm intelligence. Within the computer graphics field, the first antecedents of PSO can be traced back to the work of Reeves (1983), who proposed particle systems to model objects that are dynamic and cannot be easily represented by polygons or surfaces. Examples of such objects are fire, smoke, water and clouds. In these systems, particles are independent of each other and their movements are governed by a set of rules. A few years later, Reynolds (1987) used a particle system to simulate the collective behavior of a flock of birds. In a similar kind of simulation, Heppner and Grenander (1990) included a roost that was attractive to the simulated birds. Reynolds's and Heppner and Grenander's models inspired the set of rules that were later used in the original PSO algorithm (Kennedy and Eberhart, 1995). According to Kennedy (2006), social psychology research, in particular the theory of social impact (Latané, 1981; Nowak et al., 1990), was another source of inspiration in the development of the first particle swarm optimization algorithm (see Chapter 3 for more information).

PSO is a *direct search* method, which means that it works only with ordinal relations between objective function values and does not use the actual values to model, directly or indirectly, higher order properties of the objective function. In a PSO algorithm, simple agents, called *particles*, move in the solution space of an n-dimensional objective function f (see definition above). There are three vectors associated with a particle i at time step t: its position vector $\mathbf{x}_i^t$, which represents a candidate solution, its velocity vector $\mathbf{v}_i^t$, representing the particle's search direction, and its *personal best* vector $\mathbf{pb}_i^t$, which denotes the particle's best position attained by particle i since the beginning of the algorithm's execution.

The rules that determine the particles' movement are the core of any PSO algorithm. These rules determine from which other particles a certain particle i should get information, and how that information should be exploited. The set of particles from which particle i may obtain information is referred to as particle i's *neighborhood* and is denoted by $\mathcal{N}_i$. However, particle i's *informers*, denoted by $\mathcal{I}_i$ with $\mathcal{I}_i \subseteq \mathcal{N}_i$, are the particles from which

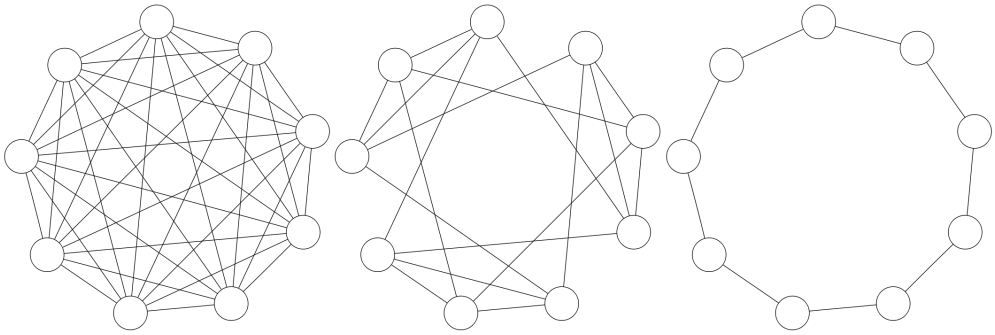

Figure 2.1: Example population topologies. The leftmost picture depicts a fully connected topology, that is, $\mathcal{N}_i$ is composed of all the particles in the swarm (self-links are not drawn for simplicity). The picture in the center depicts a so-called von Neumann topology, in which $|\mathcal{N}_i| = 4\ \forall i$. The rightmost picture depicts a ring topology in which each particle is neighbor to two other particles.

it actually obtains information. The sets $\mathcal{N}_i$ can be visualized as a graph called *population topology* (see Figure 2.1). The *model of influence* defines the mechanism to form $\mathcal{I}_i$ from $\mathcal{N}_i$. Finally, a particle's velocity-update rule determines how to compute the particle's next position using information from its informers.

In the standard PSO algorithm (Bratton and Kennedy, 2007), for example, the aforementioned factors are instantiated as follows: (i) fully-connected graphs or rings (respectively known as *gbest* and *lbest* models in PSO parlance) as population topologies, (ii) a *best-of-neighborhood* model of influence such that only the best particle in the neighborhood and the particle itself are taken as informers, and (iii) an update rule for the jth component of the ith particle's velocity and position vectors given by

$$v_{i,j}^{t+1} = wv_{i,j}^{t} + \varphi_1 U_1 \left(pb_{i,j}^{t} - x_{i,j}^{t}\right) + \varphi_2 U_2 \left(lb_{i,j}^{t} - x_{i,j}^{t}\right) , \tag{2.3}$$

and

$$x_{i,j}^{t+1} = x_{i,j}^{t} + v_{i,j}^{t+1} , \tag{2.4}$$

where w is a parameter called *inertia weight* (Shi and Eberhart, 1998a), φ_1 and φ_2 are parameters called *acceleration coefficients*, U_1 and U_2 are uniformly distributed pseudo-random numbers in the range $[0, 1)$ that are generated for each particle for each coordinate at each iteration. A particle's velocity in each coordinate j is usually constrained within the range $[-v_{\max}, v_{\max}]$. Finally, the vector $\boldsymbol{lb}_i^t$ is the best solution in particle i's neighborhood $\mathcal{N}_i$, that is:

$$\boldsymbol{lb}_i^t = \arg\min_{j \in \mathcal{N}_i} f(\boldsymbol{pb}_j^t) . \tag{2.5}$$

The basic structure of a PSO algorithm is shown in Algorithm 2. In the procedure *InitializeSwarm*, a certain number of particles are created and placed uniformly at random in the problem's search space. Each particle's velocity is initialized to zero or a small random value (Dorigo et al., 2008). In this procedure, the population topology is also initialized. In the procedure *EvaluateSwarm*, each particle's position is evaluated using the problem's objective function. If a particle finds a position that is better than its personal best solution, it updates its memory. Otherwise, it remains unchanged. In the procedure *UpdatePositions*, all particles are moved using Eqs. 2.3 and 2.4. The procedures *EvaluateSwarm* and *UpdatePositions* are executed iteratively until the stopping criterion is satisfied.

Different settings for the population topology, the model of influence, or the velocity-update rule give rise to different PSO algorithms. Two-dimensional lattices, small-world networks or random graphs are among the possible choices for replacing the standard fully-connected or ring graphs as population topologies (Kennedy, 1999; Kennedy and

Algorithm 2 Basic structure of a particle swarm optimization algorithm

```
InitializeSwarm
repeat
  EvaluateSwarm
  UpdatePositions
until Stopping criterion is satisfied
```

Table 2.2: Representative PSO works

Investigated Aspect	*Main references*
Acceleration Coefficients	Kennedy (1997); Ratnaweera et al. (2004); Chatterjee et al. (2007); Chaturvedi et al. (2009)
Inertia Weight	Shi and Eberhart (1998a,b, 1999, 2001); Eberhart and Shi (2001); Zheng et al. (2003a,b); Chatterjee and Siarry (2006)
Model of Influence	Mendes et al. (2004); Jordan et al. (2008); Montes de Oca and Stützle (2008a)
Population Size	van den Bergh and Engelbrecht (2001); Lanzarini et al. (2008); Coelho and de Oliveira (2008); Chen and Zhao (2009)
Population Topology	Kennedy (1999); Suganthan (1999); Janson and Middendorf (2003, 2005); Mohais et al. (2005); Kennedy and Mendes (2006)
Theoretical Aspects	Ozcan and Mohan (1999); Clerc and Kennedy (2002); Trelea (2003); Kadirkamanathan et al. (2006); Poli (2007, 2009); Fernández Martínez and García Gonzalo (2009); Ghosh et al. (2011)
Velocity-Update Rule	Kennedy (2003); Blackwell and Branke (2006); Mendes and Kennedy (2007); dos Santos Coelho (2008)

Mendes, 2002). Likewise, alternatives to the *best-of-neighborhood* model of influence can be implemented. The most salient example is the *fully-informed* model, in which a particle is informed by all of its neighbors (Mendes et al., 2004; Mendes, 2004). In Table 2.2 we list a number of works in which one or more of the three aforementioned factors are investigated.

Swarm Robotics

Robotics has been pivotal in the development of the swarm intelligence field. In fact, it was in a robotics paper that the term swarm intelligence was first used (Beni and Wang, 1993; Beni, 2005). Swarm intelligence applied to the multi-robot domain is called *swarm robotics* (Dorigo and Şahin, 2004; Şahin, 2005; Bayindir and Şahin, 2007). It is sometimes defined as "the study of how [a] large number of relatively simple physically embodied agents can be designed such that a desired collective behavior emerges from the local interactions among agents and between the agents and the environment." (Şahin, 2005) (p. 12). This definition is very similar to that of the engineering branch of the swarm intelligence field (Dorigo and Birattari, 2007). The particularity of swarm robotics is the embodiment of robots. In one of the first works in the field, Deneubourg et al. (1990b) used the term "ant-like" to describe the robots they used in one of the first experiments in the history of the swarm robotics field. At the same time, Deneubourg et al. reinforced the link of the field with one of its major sources of inspiration: social insects. Deneubourg et al. also showed that swarm robotics could be used as a scientific tool to test hypotheses about the mechanisms involved in swarm organization in animals— cf. Webb (2000). For

this and other reasons, swarm robotics research, unlike ACO and PSO research, does not focus solely on applications.

In swarm robotics, some mechanisms involved in robot control and the benchmark tasks robots solve, have been inspired by studies of real swarm-forming animals. For example, Deneubourg et al. (1990b), Holland and Melhuish (1999), Wilson et al. (2004), and Melhuish et al. (2006) studied swarms of robots performing spatial sorting inspired by the brood sorting behavior of ants; Theraulaz and Bonabeau (1995) and Grushin and Reggia (2008) studied structure building mechanisms inspired by wasps and other social insects; Kube and Bonabeau (2000) and Groß and Dorigo (2008) reproduced with robots the cooperative transport abilities of ants; and Mondada et al. (2004) and O'Grady et al. (2010b) draw inspiration from social insect assemblages (Anderson et al., 2002) to devise control algorithms that allow swarms of robots to perform collective tasks.

Research in swarm robotics is not only focused on tasks that can be solved collectively by robots. There are also practical problems that need to be tackled in a swarm robotics system. For example, robots that are part of a swarm may need to know when one of their peers stops working properly, or they may need to know how many robots compose the swarm. Some of these problems have been tackled using nature-inspired as well as purely engineered approaches. For instance, Christensen et al. (2009) proposed a distributed mechanism for robot fault detection within a swarm that was inspired by models of firefly synchronization. Using a similar approach, Brambilla et al. (2009) built on the work of Holland et al. (1999) to design a mechanism that allows individual robots to reliably estimate the size of the group that they belong to. Energy supply within a swarm is another practical problem that needs to be dealt with. Batteries have a limited capacity, thus, robots have a limited lifetime. If the robots lifetime is short, a swarm of robots is of little practical use. To tackle this problem, some researchers, for example Witkowski (2007), Melhuish and Kubo (2007), and Schloler and Ngo (2008) have proposed energy sharing mechanisms inspired by trophallaxis, that is, the direct exchange of food between animals (Hölldobler and Wilson, 1990). By sharing charge with one another, some robots can continuously operate while other robots get their batteries recharged.

One application area for which swarm robotics is particularly appealing is the construction of two- and three-dimensional structures (Stewart and Russell, 2006; Werfel and Nagpal, 2008; Mellinger et al., 2010). In this application area, most of the basic collective behaviors inspired by animals can be integrated into a single complex task. For example, robots need to aggregate, find construction materials, sort them, transport them from one place to another (most likely, cooperatively), and finally, coordinate their actions in order to actually build the desired structure.

Other Swarm Intelligence Systems

ACO, PSO, and swarm robotics have undoubtedly been the most popular swarm intelligence systems to date. However, other systems exist and deserve being mentioned.

A family of swarm intelligence systems is used to perform data clustering. The goal of any clustering algorithm is to partition a set of data or objects into clusters (groups, subsets, classes) so that elements belonging to the same cluster are as similar as possible and elements that belong to different clusters are as dissimilar as possible (Höppner et al., 1999). Some of these swarm intelligence systems for data clustering focus on optimization, and thus, use ACO, or PSO to tackle the problem (Martens et al., 2011). Other systems, however, are inspired by the brood sorting behavior of some ant species. These systems are called ant-based clustering algorithms (Lumer and Faieta, 1994; Handl et al., 2005; Handl and Meyer, 2007).

Ant-based clustering algorithms are related to experiments in swarm robotics. Deneubourg et al. (1990b) made robots execute the following rules: pick up an object if it is relatively isolated, and put down an object if there are other objects around. As a result, the robots created "heaps" of objects in the environment. Lumer and Faieta (1994) implemented in software a similar system in which agents move over a toroidal square grid on which there

are objects representing data items. Agents pick up an object with high probability if it is not surrounded by other similar objects. By the same token, agents put down objects on any free location surrounded by similar objects to the one they are carrying. As a result, groups of similar data items are created. In other words, the algorithm performs data clustering. A number of improvements of the basic technique have followed (see the work of Handl and Meyer (2007) for one of the latest surveys of the topic).

A family of swarm intelligence algorithms, inspired by the behavior of bees, is attracting the attention of researchers in the field (see the work of Karaboga and Akay (2009) for a recent review). One of the algorithms that belong to this category is called Bee Colony Optimization (BCO) (Teodorović, 2009). This algorithm is typically used to tackle combinatorial optimization problems. BCO consists of two procedures that are executed iteratively. In the first procedure, artificial bees build partial candidate solutions. In the second procedure, the artificial bees "meet" in order to recruit other bees to search in the area in proximity to the best found partial solutions. These two procedures roughly mimic the behavior of scout bees looking for rich food sources and of the waggle dance of bees, which is aimed at recruiting other bees from the nest. Another bee-inspired algorithm, the Artificial Bee Colony (ABC) algorithm (Karaboga and Basturk, 2007), is used for tackling continuous optimization problems. In ABC, the position of the bees represent candidate solutions to the problem. The algorithm works through the interaction of three kinds of artificial bees. Bees can be play three roles. They can be "employed", "onlookers", or "scouts." An employed bee exploits a promising region. In other words, the bee carries out a sort of local search. Onlooker bees search around promising regions based on their quality. Onlooker bees can compare the quality of different regions in the search space, thus they perform a more global search than employed bees. Finally, scout bees perform random search, which enables them to discover new promising regions in the search space.

2.2 Social Learning

Social and individual learning are terms that are often used vaguely, meaning different things in different contexts. For the purpose of this dissertation, it is therefore important to clearly define the meaning of these two concepts and their relationship.

Individual (or asocial) learning is the process whereby an agent benefits from experience to become better adapted to its environment (Rescorla, 1988). The exact meaning of "experience" and "adaptation" depends on the context in which the term "learning" is used. In any case, learning implies a change in an agent's behavior from the moment in which it interacts with its environment, or gains "experience", and the moment in which its level of "adaptation" to its environment is measured or observed. In Chapters 4 and 5, we will explicitly define these terms in the context of the two case studies presented in this dissertation.

From a machine learning perspective, learning is finding an association between inputs and some output. Inputs can have many forms, from abstract data, to actual information gathered through electronic sensors. An agent's output can be, for example, actions that change the agent's environment, or an abstract concept, such as a category identifier. The association between inputs and output changes during the lifetime of the learning agent. This association represents the agent's "experience" discussed in the previous paragraph. The purpose of associating inputs with outputs is to maximize some performance measure. A better score using a performance measure means that the agent is "better adapted" to its environment. There are roughly three categories of learning problems (Birattari, 2009): supervised, reinforcement, and unsupervised. In supervised learning (Aha et al., 1991), a *supervisor* provides examples of the desired input-output associations. In this case, a learning agent tries to minimize the differences between its own responses and the desired ones. Reinforcement learning (Kaebling et al., 1996) is based on rewards given to a learning agent when it performs actions that lead to a certain environment state. In this case, a learning agent tries to maximize the collected rewards. Unsupervised learning (Jain et al.,

1999) does not require any examples or rewards. In this case, a learning agent tries to identify input patterns that trigger similar outputs or responses.

There are more definitions of social learning than of individual learning. Fortunately, Heyes (1994) provides a definition onto which one can map many working definitions existing in the literature:

Definition *The term 'social learning' refers to learning that is influenced by observation of, or interaction with, another animal (typically a conspecific) or its products [...]. The complementary set is commonly known as 'individual learning'.* (Heyes, 1994) p. 207.

Heyes's definition is general enough to encompass the definitions of Biederman et al. (1993) who refer to social learning as learning from the observation of others' behavior, and Caldwell and Millen (2009) who use the term social learning as learning from the interaction with others. Other authors prefer to use Heyes's full definition (Brown and Laland, 2003; Caldwell and Millen, 2009; Rendell et al., 2010b,a, 2011).

Social learning in animals has been studied since the 19th century (Galef Jr., 1990). In humans, social learning started to be seriously studied around the 1970s with the work of Bandura (1977) and other psychologists. Similarly to other theories of behavior, social learning in humans and animals has been studied from a mechanistic as well as from a functional point of view. Ethologists and psychologists take a mechanistic perspective in order to determine the mechanisms and strategies that animals use to learn from others. Biologists and scientists from other disciplines, including economics, study social learning from a functional perspective in order to answer the question of why and under which circumstances social learning is useful.

2.2.1 Social Learning Mechanisms and Strategies

Social learning mechanisms (how an agent may learn from others) and strategies (when and from whom should an agent learn socially) are the subject matter of the mechanistic approach to the study of social learning. In the following paragraphs, we will briefly define some of the most commonly studied social learning mechanisms and strategies.

Mechanisms

Imitation, emulation, enhancement, conditioning, facilitation and mimicking are social learning mechanisms. They are not learning phenomena themselves, but they may lead to learning (Heyes et al., 2000). Imitation and emulation involve copying. When an observer imitates, it copies the actions of a demonstrator with the goal of reproducing the actions' effects; when an observer emulates, it uses its own actions to reproduce the results produced by a demonstrator's actions (Heyes, 1994; Caldwell and Millen, 2009; Cakmak et al., 2010). Imitation has been traditionally assumed to be the main mechanism through which animals learn socially (Galef Jr., 1990). However, imitation is a relatively complex process that implies that the copying animal is able to take the perspective of the demonstrating animal. Thus, to explain social learning in animals that are considered to have limited cognitive abilities, such as insects, simpler mechanisms have been sought. One such mechanism is called social enhancement (Franz and Matthews, 2010). Some authors distinguish between two forms of social enhancement: stimulus enhancement and local enhancement. Stimulus enhancement occurs when an agent calls the attention of another one to a particular object, increasing the likelihood that the observer interacts with that object (or with objects with similar physical features) in the future, regardless of the objects' location (Heyes, 1994; Bonnie and de Waal, 2007; Franz and Matthews, 2010). Local enhancement occurs when an agent is attracted to the location where a certain behavior was observed (Galef Jr., 1990; Heyes, 1994; Franz and Matthews, 2010). Social enhancement makes some features of the environment more salient than others. As a result, the observer may save time and effort exploring the environment in order to find interesting objects or locations. Social enhancement imposes lower cognitive capabilities

Table 2.3: Examples of "When" and "From whom" components of a social learning strategy.

When	From whom
When established behavior is unproductive	From majority
When asocial learning is costly	From successful individuals
When uncertain	From good social learners
When dissatisfied	From related individuals (kin)
When environment is stable	From familiar individuals
	From older individuals

on animals than imitation or emulation do. Conditioning in a social context means that an animal learns an association between two stimuli as a result of observing the reaction of a demonstrator to a stimulus (Heyes, 1994). Social facilitation occurs when an animal manifests a behavior more (or less) strongly in the presence of another passive animal of the same species (Zajonc, 1965; Guérin, 1993; Heyes et al., 2000). Social facilitation is considered a social learning mechanism because the influence of another animal may increase or decrease the responsiveness of the observer to its environment, and thus, may change the observer's learning ability. Mimicking is similar to imitation in that the observer copies the actions of a demonstrator. However, when mimicking, the observer is not trying to get the same results as the demonstrator; it simply performs the actions without regard to the actions' goals (Tomasello, 2004). Mimicking could be seen as a socially mediated action exploration mechanism.

Strategies

Functional studies of social learning (see Section 2.2.2) suggest that agents should not learn socially all the time. Instead, these studies conclude that agents should selectively choose between individual and social learning depending on the characteristics of their environment. The strategy used by an agent to decide when and from whom to learn is called a *social learning strategy* (Laland, 2004; Galef Jr., 2009).

Social learning strategies have been studied mostly theoretically within a functional framework to determine which ones are more likely to offer advantages under predefined circumstances (Laland, 2004). Examples of social learning strategies can be built from the components listed in Table 2.3, which lists some plausible "when" and "from whom" components of a social learning strategy. This list was proposed by Laland (2004) and later adapted by Galef Jr. (2009).

In experiments with animals, some scientists have reported what probably is the execution of certain social learning strategies. For example, a copy-when-uncertain social strategy could explain the behavior of Norway rats in an experiment designed by Galef Jr. (1996) in which Norway rats had to choose between two completely novel foods. In such an uncertain situation, the rats preferred the foods that had been consumed by other rats (detected through breath odor) instead of trying any of them with equal probability, which would have been the case if they had been learning individually.

The study of social learning strategies is still in its infancy, but some important efforts are being made in order to discover strategies robust to different environmental conditions. For example, Rendell et al. (2010a) organized a computer-based tournament aimed at discovering effective social learning strategies under a wide range of environmental conditions. In total, 104 strategies were submitted and the final outcome of the tournament has given researchers useful insight into what makes a social learning strategy successful. The strategy that won the tournament favored social learning almost all the time. The reason, Rendell et al. conclude, is that since agents frequently demonstrated the highest-payoff behavior, social learners could observe and copy only promising behaviors. In effect, through the demonstration of good behaviors, agents were filtering out mediocre behaviors that could not be spread through social learning.

2.2.2 Functional Value of Social Learning

The functional approach to the study of social learning aims at understanding the conditions under which social learning evolved and what the adaptive value of social learning is. This approach is of interest to scientists of many disciplines. For example, biologists wonder how variable can an environment be so that social learning evolves (Wakano et al., 2004). Sociologists consider that social learning is at the root of culture but since social learning is a product of evolution, they wonder how cultural and genetic evolution interact (Cavalli-Sforza and Feldman, 1983; Boyd and Richerson, 1985; Flinn, 1997). Economists wonder what is the effect of social learning in the decisions economic agents make and its consequences for the population as a whole (Ellison and Fudenberg, 1995; Chamley, 2004). Computer scientists and engineers are interested in exploiting social learning in the design and use of software and robots (Thomaz, 2006; Nehaniv and Dautenhahn, 2007; Cakmak et al., 2010). We belong to this last class of researchers. As it will be discussed in more detail in Chapter 3, the work presented in this dissertation takes a functional approach toward the application of social learning ideas.

The adaptive value of social learning has been studied mainly through mathematical and computational models. Almost all models assume that social learning is a convenient way to acquire adaptive behavior because it allows the social learning agent to save time and energy that it would otherwise spend learning individually (Laland, 2004). There are also other advantages associated with social learning, such as reducing the risk of exposure to predators or lowering the chances of getting poisoned as a result of trying unknown foods (Galef Jr., 2009). Consequently, it would be reasonable to assume that a population composed of social learning agents would have a higher average fitness than a population composed of only individual learning agents. As it turns out, this reasoning is flawed as shown by Rogers (1988). He demonstrated that social learning agents have an advantage only when individual learning agents are present. This insight motivates research on social learning strategies as we saw above.

A family of social learning models is aimed at investigating the degree to which an environment can change so that social learning is useful (Bergman and Feldman, 1995; Wakano et al., 2004; Laland and Kendal, 2003; Galef Jr., 2009). These models study the relative advantage that reliance on social and individual learning as well as genetically encoded behavior offers to an agent in the presence of a changing environment. As a result of years of theoretical work, it is now well established that when the environment does not change, or when it changes too frequently, a genetically encoded behavior prevails. In the first case, it is assumed that there is a cost associated to learning. Thus, a genetically encoded behavior provides everything an agent needs at a lower cost. In the second case, there is no possibility of learning and thus, again for economic reasons, a genetically encoded behavior prevails. At high rates of change that still allow for some predictability of the environment, individual learning lets an agent have up-to-date information whereas social learning can potentially be harmful since outdated information can pass from one agent to another. At intermediate rates of change social learning flourishes more than individual learning because it is a cheaper way of obtaining adaptive information. Note that social learning models and their implications are subject to change because their predictions have been subjected to limited empirical tests (Laland and Kendal, 2003). As recently shown by (Rendell et al., 2010a), a population of agents might still rely on social learning even in a frequently changing environment simply because demonstrators will tend to adapt their own behavior to the new circumstances and thus, they can still pass useful information to others.

Other models have been devised in order to study the spread of behavior through social learning (Laland and Kendal, 2003; Cavalli-Sforza and Feldman, 1981). The goal of these models is to find a "signature" of social learning in the curves that represent the proportion of individuals in a population adopting a particular behavior. Unfortunately, these models do not consider simple explanations that could account for the adoption patterns observed (Laland and Kendal, 2003). Finally, there are models aimed at under-

standing whether culture (the cumulative effect of social learning) and natural evolution interact (Feldman and Laland, 1996; Laland and Kendal, 2003). The basic assumption here is that an animal's genotype may determine what it learns, and that learned behavior affects, in turn, the selection pressure on that genotype.

Chapter 3

Incremental Social Learning

In this chapter, we present the incremental social learning (ISL) framework. First, we describe the problem of interference in multiagent systems. Then, we explain how interference is addressed by the ISL framework and present the framework's algorithmic structure. We finish with a brief discussion of related work. Work specifically related to each instantiation of the ISL framework in our case studies is discussed in Chapters 4 and 5.

3.1 Interference

There are different kinds of interactions among agents in multiagent systems. Depending on the effect of such interactions, they can be labeled as "positive", "negative", or "neutral" interactions (Gershenson, 2007). Positive interactions facilitate the accomplishment of an assigned task. For example, in a collective transport task, robots form teams in order to transport objects that are too difficult for a single robot to move (Kube and Bonabeau, 2000; Tuci et al., 2006). Negative interactions, also called *interference*[1] (Matarić, 1997), *friction* (Gershenson, 2007), or *repulsive and competitive interactions* (Helbing and Vicsek, 1999), are those that block or hinder the functioning of the system's constituent agents. As a result, interference decreases the performance of a multiagent system. For instance, in an ant-based clustering algorithm (see Section 2.1.2) agents can undo the actions of other agents, which increases the time needed by the algorithm to find a satisfactory final clustering. A neutral interaction does not affect the system's dynamics in such a way that it benefits or harms progress toward the completion of an assigned task. Deciding whether an interaction is positive, negative, or neutral depends on the time scale used to measure the interaction's effects. For example, an interaction that involves two robots performing a collision avoidance behavior can be labeled as a negative interaction in the short term because time is spent unproductively. However, if the time horizon of the task the robots are performing is significantly longer than the time frame of a collision avoidance maneuver, then the overall effect of such an interaction may be negligible. In this case, such interaction can be labeled as neutral.

Interference is one of the main challenges to overcome during the design and operation of systems composed of many agents (Gershenson, 2007). For example, Kennedy and Eberhart, the designers of the first PSO algorithm, pondered different candidate particle interaction rules before proposing the rules that we now know (see Eqs. 2.3 and 2.4). Their ultimate goal was to design rules that promoted positive interactions between particles. In the final design, particles cooperate, that is, they engage in positive interactions, by exchanging information with one another about the best solution to an optimization problem that each particle finds during its lifetime. At the same time, however, such an exchange of information can "distract" particles and make them search in regions of a problem's search space that seem promising but that in fact do not contain the optimal solution

[1] In this dissertation, we use the term interference to refer to the set of negative interactions that occur within multiagent systems, including swarm intelligence systems.

that the particles are looking for. The net effect of such interactions is that particles may spend objective function evaluations unproductively. This effect intensifies as the size of the particle swarm increases.

Directly measuring interference is difficult. First, one can determine whether the effects of an interaction, or a set of interactions, are beneficial or not only after the task has been performed. Second, as we mentioned before, an interaction may be positive, negative or neutral, depending on the time scale used to measure its effect. In this dissertation, we advocate for qualifying interactions based on their effects in the long term. We do so because it is only at a time scale similar to the time a system needs to perform a task that labeling interactions is relevant for practical purposes. Third, the nature of the interactions themselves poses a challenge. In some systems, agents interact directly on a one-to-one or one-to-some basis, such as in PSO algorithms. In other systems, such as ACO algorithms, agents interact indirectly through the environment and there may be extended periods of time between the moment an agent acts and the moment another agent is affected by those actions. With these restrictions, interference can only be measured indirectly through observation of the system's performance. Despite these difficulties, two measures can be used to indirectly gauge interference: (i) the time needed by the system to reach a desired or target state, or (ii) the amount of work performed in a certain amount of time. If one compares two systems, we expect the system with higher interference to make progress toward a desired state more slowly than the system with lower interference. As a result, if one let two systems run for the same amount of time, the system with larger interference would perform less work than the system with lower interference.

There are two properties of systems composed of many agents that are in direct relation with interference:

1. Interference increases with the number of agents in the system. This effect is the result of the increased number of interactions within the system. The larger the number of agents that comprise the system, the higher the probability of a negative interaction occurring.

2. Interference tends to decrease over time. At one extreme of the spectrum, one can find a system in which interactions between agents are completely random or not purposeful. In such a case, it is expected that agents cannot coordinate and thus, cannot perform useful work. Thus, we expect interference to remain at a constant level over time. At the other extreme of the spectrum, one finds well-behaved systems consisting of a number of agents whose interaction rules are designed in order to make agents coordinate with each other. Initially, we expect a high-level of interference because agents would not have enough knowledge about their current environment. However, over time, the behavioral rules of these agents would exploit any gained knowledge in order to make progress toward the completion of the assigned task. Thus, we expect that in cases like these, interference decreases over time, because the other alternatives would be a random behavior or a pathological system in which interference increases.

By making use of these two properties, it is possible to control, to a certain extent, the levels of interference in a multiagent system. The incremental social learning framework, which will be described next, is based on this observation.

3.2 The Incremental Social Learning Framework

Our goal with the incremental social learning (ISL) framework is to reduce the effects of interference in swarm intelligence systems. ISL is a framework because it offers a conceptual structure that does not prescribe a specific implementation of the ideas on which it relies. Each instantiation of the framework will benefit from knowledge about the specific application domain, and therefore, specific properties of the framework should be analyzed in an application-dependent context.

Algorithm 3 Incremental social learning framework

Input: Agent addition criteria, stopping criteria
1: /* Initialization */
2: $t \leftarrow 0$
3: Initialize environment $\mathbf{E}^t$
4: Initialize population of agents $\mathbf{X}^t$
5:
6: /* Main loop */
7: **while** Stopping criteria not met **do**
8: **if** Agent addition criteria is not met **then**
9: default($\mathbf{X}^t, \mathbf{E}^t$) /* Default system */
10: **else**
11: Create new agent a_{new}
12: slearn($a_{new}, \mathbf{X}^t$) /* Social learning */
13: $\mathbf{X}^{t+1} \leftarrow \mathbf{X}^t \cup \{a_{new}\}$
14: **end if**
15: $\mathbf{E}^{t+1} \leftarrow$ update($\mathbf{E}^t$) /* Update environment */
16: $t \leftarrow t + 1$
17: **end while**

The ISL framework consists of two elements that manipulate and exploit the two properties mentioned in Section 3.1. The first element of the framework directly affects the interference levels within a system by manipulating the number of interactions among the system's constituent agents. Such a control is achieved by varying the number of agents in the system. The strategy for controlling the size of the agent population exploits the second property, that is, that interference tends to decrease over time. The system starts with a small population that grows at a rate determined by agent addition criteria specified by the user. Two phenomena with opposite effects occur while the system is under the control of the ISL framework. On the one hand, interference increases as a result of adding new agents to the swarm (first property described in Section 3.1). On the other hand, interference decreases because the system operates while the population grows (second property described in Section 3.1).

The second element of the framework is social learning. This element is present before a new agent freely interacts with its peers. Social learning is used so that the new agent does not produce extra interference due to its lack of knowledge about the environment. Leadership, a swarm intelligence mechanism (see Chapter 2), is present in the framework in the process of selecting a subset of agents from which the new agent learns. The best strategy to select such a set depends on the specific application. However, even in the case in which a random agent is chosen as a "model" to learn from, knowledge transfer occurs because the selected agent will have more experience than the new agent that is about to be added. As stated in Chapter 2, we take a functional approach to the use of social learning concepts. We do not pay attention to the mechanisms used by the agents to learn from each other. Instead, we are interested in the effects that social learning has on the agents and on the system.

The two elements that compose ISL are executed iteratively as shown in Algorithm 3.

In a typical implementation of the ISL framework, an initial population of agents is created and initialized (line 4). The size of the initial population depends on the specific application domain. In any case, the size of this initial population should be small in order to reduce interference to the lowest level possible. A loop structure allows the interspersed execution of the underlying system and the creation and initialization of new agents (line 7). This loop is executed until some user-specified stopping criteria are met. Stopping criteria can be specific to the application or related to the ISL framework. For example, the framework may stop when the task assigned to the swarm intelligence system is completed

or when a maximum number of agents are reached. While executing the main loop, agent addition criteria, which are also supplied by the user, are repeatedly evaluated (line 8). The criteria can range from a predefined schedule to conditions based on statistics of the system's progress. If the agent addition criteria are not met, the set of agents work normally, that is, the underlying swarm intelligence system is executed. In line 9, such an event is denoted by a call to the procedure default($\mathbf{X}^t, \mathbf{E}^t$). If the agent addition criteria are satisfied, a new agent is created (line 11). In contrast to a default initialization such as the one in line 4, this new agent is initialized with information extracted from a subset of the currently active population (line 12). Such an initialization is denoted by a call to the procedure slearn($a_{new}, \mathbf{X}^t$). This procedure is responsible for the selection of the agents from which the new agent will learn, and for the actual implementation of the social learning mechanism. Once the new agent is properly initialized, it becomes part of the system (line 13). In line 15, we explicitly update the environment. However, in a real implementation, the environment may be continuously updated as a result of the system's operation.

In most swarm intelligence systems, the population of agents is large and homogeneous, that is, it is composed of agents that follow exactly the same behavioral rules. Thus, any knowledge acquired by an agent is likely to be useful for another one. The social learning mechanism used in an instantiation of the ISL framework should allow the transfer of knowledge from one agent to the other. In some cases, it is possible to have access to the full state of the agent that serves as a "model" to be imitated, and thus, the social learning mechanism is simple. In other cases, access to the model agent's state may be limited and a more sophisticated mechanism is required. In most cases, the result of the social learning mechanism will not be simply a copy of the model agent's state, but a biased initialization toward it. Copying is not always a good idea because what may work very well for an agent in a system composed of n agents may not work well in a system of $n + 1$ agents.

3.3 Related Work

The ISL framework and many works in the field of multiagent systems (Wooldridge, 2009) share a common goal: interference reduction. The means used by these works and the ISL framework to achieve this goal differ. In traditional multiagent systems, interference is a problem that has been tackled indirectly through the careful design of interaction protocols that consider all the possible events that the agents can possibly experience (Shoham and Tennenholtz, 1995; Gmytrasiewicz and Durfee, 2000). Examples of protocols designed in such a way are the following: Contract Net (Smith, 1980), coalition formation algorithms (Shehory and Kraus, 1998), or the protocols used for negotiation in agent-mediated electronic commerce applications (He et al., 2003). Tackling interference has required a significant effort on the part of the multiagent systems community. These efforts could be grouped into categories such as methodologies, standards, or communication protocols. Early on in the development of the field of multiagent systems, researchers recognized that for analyzing and designing multiagent systems, new methodologies were required. Well-known methodologies that are the result of work in this direction are MaSE (Deloach et al., 2001) and the Gaia methodology (Zambonelli et al., 2003). Through these methodologies, interactions between agents are identified and carefully designed. Standards have been proposed to allow interoperability of agents developed by different parties. The best known organization dedicated to establish specifications for multiagent systems is the Foundation for Intelligent Physical Agents (FIPA)[2] (O'Brien and Nicol, 1998). A sign that interactions are one of the main issues in the design of multiagent systems is that the core FIPA specification is the one related to agent communication. Methodologies and standards call for a common communication language between the agents that comprise a system. As a result, some agent languages have been proposed. For example, languages such as KQML (Finin et al., 1994), or FIPA-ACL (IEEE Foundation for Intelligent Physical Agents, 2011) have

[2] `http://www.fipa.org`

explicit specifications that let agents exchange knowledge with each other.

A complete review of the literature in the field of multiagent systems that deals with interference, either directly or indirectly, is out of the scope of this dissertation. However, we can say that there are strong differences between practically all previous works in the field of multiagent systems and the ISL framework. First, the size of the systems that can be designed with a traditional approach is limited to just a few and very sophisticated agents. Moreover, when taking a traditional approach, one is necessarily assuming that the number of agents is constant over time. This assumption is needed because with traditional approaches, each agent plays a specific role in the system, and adding or removing an agent would require the designer to re-program all or at least some of the agents that comprise the system.

In contrast, in the ISL framework we assume that the agents are very similar, if not identical, to each other. As a result, since each agent does not play a specific role, it is possible to assume that the number of agents can change over time and that the total number of agents can be very large. Thus, even though the framework may work for small systems, we are proposing the framework to be primarily used with systems composed of a large number of agents. Hence, we expect the ISL framework to have a larger impact on the design and operation of swarm intelligence systems than on the design and operation of small multiagent systems.

The other body of literature that is related to the ISL framework is the one in which social learning or related concepts are used in the context of multiagent systems and swarm intelligence systems. Two main categories of works can be distinguished: (i) those that study social learning using a multiagent system as a tool, and (ii) those that exploit social learning as a tool for developing better performing systems. The ISL framework belongs to this second category of works. Until recently, the first category was the most active of the two. Simulations of social systems in computers began in the 1950s (Conte et al., 1998) and have continued gaining popularity. This increased popularity is evidenced by the fact that there are now scholarly journals, such as the Journal of Artificial Societies and Social Simulation (JASS)[3], devoted to the topic. Areas of interest in this category range from the study of the usefulness of social learning under different environmental conditions (Annunziato and Pierucci, 2003; Noble and Franks, 2003; van der Post and Hogeweg, 2004; Priesterjahn and Eberling, 2008) to the evolution of language and culture (Divina and Vogt, 2006; Vogt, 2006). The second category of works has being attracting the attention of a growing community. Social learning as a mechanism to improve the performance of systems composed of many agents has been investigated in the context of robotics (Matarić, 1997; Pini and Tuci, 2008; Cakmak et al., 2010), multiagent systems (Kopp and Graeser, 2006; García-Pardo et al., 2010), and neural computation (Jang and Cho, 2002).

In the swarm intelligence field, social learning concepts have been associated with PSO algorithms almost since they were first proposed. Kennedy (2006) explains how the development of the first PSO algorithm was heavily influenced by Latané's social impact theory (Latané, 1981). This theory argues that an individual changes its psychological state to a degree that is a function of the strength, immediacy, and the number of other individuals. In the context of PSO algorithms, this theory was another source of inspiration for the rules that govern the movement of particles. Although swarm intelligence is based on the idea that the actions of one agent can affect the behavior of another agent, for instance, via stigmergy (see Section 2.1), social learning has been overlooked by researchers in the field. We hope that this dissertation makes social learning research more visible to the swarm intelligence community, and that the community of scientists studying social learning in animals becomes aware of the potential of swarm intelligence as a hypothesis and application testing field. We hope that the mutual exchange of ideas will serve to enrich both fields.

In the next two chapters, we will describe the case studies designed to test the effectiveness of the ISL framework. Previous work specifically related to the instantiation of the ISL framework in the context of each case study is presented in the corresponding chapter.

[3] `http://jasss.soc.surrey.ac.uk`