

FEDERAL UNIVERSITY OF TECHNOLOGY – PARANÁ  
ELECTRICAL AND COMPUTER ENGINEERING GRADUATE  
PROGRAM

RAFAEL STUBS PARPINELLI

**AN ECOSYSTEMIC VIEW FOR DEVELOPING BIOLOGICALLY  
PLAUSIBLE OPTIMIZATION SYSTEMS**

THESIS

CURITIBA

2013

**RAFAEL STUBS PARPINELLI**

**AN ECOSYSTEMIC VIEW FOR DEVELOPING BIOLOGICALLY  
PLAUSIBLE OPTIMIZATION SYSTEMS**

Thesis submitted to the Electrical and Computer Engineering Graduate Program of Federal University of Technology – Paraná in partial fulfillment of the requirements for the degree of “Doctor of Philosophy” – Concentration Field: Computer Engineering.

Advisor: Heitor Silvério Lopes

**CURITIBA**

**2013**

---

Dados Internacionais de Catalogação na Publicação

---

P257 Parpinelli, Rafael Stubs  
An Ecosystemic View for Developing Biologically Plausible Optimization Systems/ Rafael Stubs Parpinelli. – 2013.  
128 f. : il. ; 30 cm

Guidance: Heitor Silvério Lopes.  
Thesis (Doctorate) – Federal University of Technology – Paraná. Electrical and Computer Engineering Graduate Program. Curitiba, 2013.  
Bibliography: p. 118-128.

1. Mathematical optimization. 2. Ecosystems. 3. Ecology. 4. Genetic algorithms. 5. Evolutionary computation. 6. Mathematical models. 7. Computer simulation. 8. Electric engineering – Thesis. I. Lopes, Heitor Silvério. II. Universidade Tecnológica Federal do Paraná. Programa de Pós-graduação em Engenharia Elétrica e Informática Industrial. III. Title.

CDD (22. ed.) 621.3

---

Biblioteca Central da UTFPR, Câmpus Curitiba

*Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial*

**Tese de Doutorado Nº. 83**

***“An Ecosystemic View for Developing Biologically  
Plausible Optimization Systems”***

por

**Rafael Stubs Parpinelli**

Esta tese foi apresentada como requisito parcial à obtenção do título de Doutor em CIÊNCIAS - Área de Concentração: Engenharia de Computação, pelo Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial – CPGEI, da Universidade Tecnológica Federal do Paraná – UTFPR, às 13:30h do dia 19 de fevereiro de 2013. O trabalho foi aprovado pela Banca Examinadora, composta pelos doutores:

---

Prof. Heitor Silvério Lopes, Dr.  
(Presidente - UTFPR)

---

Prof. Renato Tinós, Dr.  
(USP)

---

Prof. Wagner Rodrigo Weinert, Dr.  
(IFPR-Câmpus Paranaguá)

---

Prof. Leandro dos Santos Coelho, Dr.  
(PUC-PR)

---

Prof. Jean Marcelo Simão, Dr.  
(CPGEI/UTFPR)

Visto da Coordenação:

---

Prof. Ricardo Lüders, Dr.  
(Coordenador do CPGEI)

For my loving wife Elen, who believed in me even when I had doubts.

## **ACKNOWLEDGEMENTS**

I would like to express my warm thanks to:

- my supervisor, prof. Heitor Silvério Lopes, for his support and valuable discussions, sometimes exhausting brainstorming sessions, during my study and research on this thesis. His genuine love of science was a continual source of inspiration;
- FUMDES program and UDESC for the financial support;
- my wife Elen for her love, patience, and encouragement which helps me a lot;
- colleagues and friends who discussed with me for long hours on different topics concerned to this thesis;
- and my whole family for the constant support.

*“I am convinced that Natural Selection has been the main but not exclusive means of modification.”* Charles Darwin (1859)

## RESUMO

PARPINELLI, R. S.. AN ECOSYSTEMIC VIEW FOR DEVELOPING BIOLOGICALLY PLAUSIBLE OPTIMIZATION SYSTEMS. 128 f. Thesis – Electrical and Computer Engineering Graduate Program, Federal University of Technology – Paraná. Curitiba, 2013.

A busca por ideias, modelos e paradigmas computacionais biologicamente inspirados e plausíveis sempre atraiu o interesse de cientistas da computação, especialmente na área de Computação Natural. Além disso, o conceito de otimização pode ser abstraído de vários processos naturais como, por exemplo, na evolução das espécies, no comportamento de grupos sociais, na dinâmica do sistema imunológico, nas estratégias de busca por alimento e nas relações ecológicas entre populações de animais. Com o melhor de nosso conhecimento, os ecossistemas naturais e seus conceitos ainda não foram explorados computacionalmente no contexto de otimização de funções e, portanto, eles são abordados nesta tese. Este trabalho destaca as principais propriedades de ecossistemas naturais que podem ser importantes para a construção de ferramentas computacionais para resolver problemas complexos de otimização. Também, a modelagem computacional para tais funcionalidades são introduzidas. A principal discussão apresentada nesta tese refere-se ao uso cooperativo de populações de soluções candidatas, co-evoluindo em um contexto ecossistêmico. Com esta analogia, cada população comporta-se de acordo com uma estratégia de busca específica que é empregada na evolução das soluções candidatas. Além da possibilidade de utilizar diferentes estratégias de busca cooperativamente, esta analogia abre a possibilidade de inserção de conceitos ecológicos no processo de otimização, permitindo o desenvolvimento de novos sistemas de otimização biologicamente inspirados e plausíveis. O potencial de alguns conceitos ecológicos é apresentado em um algoritmo canônico ecologicamente inspirado, chamado ECO (Ecological-inspired Optimization algorithm). Alguns algoritmos baseados em população são utilizados para compor a abordagem proposta. Os problemas resolvidos nesta tese são várias funções contínuas de benchmark com um número alto de dimensões ( $D = 200$ ) e o problema de predição de estrutura de proteínas para o modelo 2D AB. Além disso, o uso de dinâmica populacional para auto-regular o tamanho das populações; o uso de modelos heterogêneos com diferentes estratégias de busca; e o uso de agrupamento hierárquico para ajustar dinamicamente a formação de habitats e probabilisticamente definir as topologias de comunicação são alguns estudos de caso investigados. Os resultados obtidos se mostraram promissores considerando a aplicação do ecossistema computacional. Finalmente, conclusões e várias ideias para pesquisas futuras são apresentadas.

**Palavras-chave:** otimização; busca cooperativa; coevolução; ecossistemas; ecologia



## ABSTRACT

PARPINELLI, R. S.. AN ECOSYSTEMIC VIEW FOR DEVELOPING BIOLOGICALLY PLAUSIBLE OPTIMIZATION SYSTEMS. 128 f. Thesis – Electrical and Computer Engineering Graduate Program, Federal University of Technology – Paraná. Curitiba, 2013.

The search for plausible biologically inspired ideas, models and computational paradigms always drew the interest of computer scientists, particularly those from the Natural Computing area. Also, the concept of optimization can be abstracted from several natural processes, for instance, in the evolution of species, in the behavior of social groups, in the dynamics of the immune system, in the food search strategies and in the ecological relationships of different animal populations. To the best of our knowledge, ecosystems and their concepts have not been explored computationally in the context of function optimization and, therefore, they are addressed in this thesis. This work highlights the main properties of ecosystems that can be important for building computational tools to solve complex problems. Also, it is introduced the computational modelling for such biologically plausible functionalities (e.g., habitats, ecological relationships, ecological succession, and another). The main discussion presented in this work relates to the cooperative use of populations of candidate solutions, coevolving in an ecological context. With this ecology-based analogy, each population can behave according to a specific search strategy, employed in the evolution of candidate solutions. In addition to the possibility of using different optimization strategies cooperatively, this analogy opens the possibility of inserting ecological concepts in the optimization process, thus allowing the development of new bio-plausible hybrid systems. The potentiality of some ecological concepts is also presented in a canonical Ecology-inspired Algorithm for Optimization, named ECO. Some population-based algorithms are used to compose the ecology-based approach. The problems solved in this thesis are several continuous benchmark functions with a high number of dimensions ( $D = 200$ ), and the protein structure prediction problem for the 2D AB model. Also, the use of population dynamics to self-regulate the size of populations during ecological successions; the use of heterogeneous models embedding different search strategies into the system; and the use of hierarchical clustering to dynamically adjust the habitats formation and probabilistically define the habitats communication are some case studies investigated. Results were promising concerning the application of the proposed computational ecosystem for optimization. Finally, concluding remarks and several ideas for future research are presented.

**Keywords:** optimization; cooperative search; co-evolution; ecosystems; ecology

## LIST OF FIGURES

FIGURE 1	– Taxonomy for cooperative search algorithms. (a) Considers the diversity of the algorithms employed and the implementation strategy, (b) Considers the type of decomposition of the search space (EL-ABD; KAMEL, 2005).	41
FIGURE 2	– Possible representation for the elements of a computational ecosystem. Three populations with different behaviors. ....	55
FIGURE 3	– Generical view of a computational ecosystem for optimization. Lower level: hyper-surface of the search space. Intermediate level: intra-habitats communication topologies where each small circle represents a population. Upper level: five habitats connected through inter-habitats communication topology. ....	56
FIGURE 4	– The elements of a computational ecosystem. ....	58
FIGURE 5	– Dendrogram generated using linkage information from Table 2. ....	62
FIGURE 6	– A change in the environmental factor intensity influences the growth and reproduction of a population. Adapted from (BEGON et al., 2006). ....	66
FIGURE 7	– 2D Schaffer function. ....	70
FIGURE 8	– Conceptual illustration of the proposed algorithm. ....	71
FIGURE 9	– Number of habitats at each ecological succession step. ....	74
FIGURE 10	– Extended illustrative map for the elements of a computational ecosystem. Highlighted are the features used in the canonical eco-inspired algorithm. ....	75
FIGURE 11	– <i>Pthreads</i> API schedules the populations to the processors. ....	77
FIGURE 12	– Generic representation of a hypothetic protein structure. ....	81
FIGURE 13	– Case study I: Difference between ABC and ECO <sub>ABC</sub> approaches. ....	85
FIGURE 14	– Case study I: Box plot for the results obtained ( $D = 200$ ). ....	87
FIGURE 15	– Case study I: Average number of habitats at each ecological succession. ..	88
FIGURE 16	– Case study II: Population dynamics according to the logistic model ( $a = 3.57$ ). ....	89
FIGURE 17	– Case study II: Box plot for the results obtained ( $D = 200$ ). ....	91
FIGURE 18	– Case study II: Bar graph of each benchmark function with $D = 200$ . The $x$ -axis represents the different approaches and the $y$ -axis shows the average best results. ....	92
FIGURE 19	– Case study III: Bar graph off each benchmark function with $D = 200$ . The $x$ -axis represents the different approaches and the $y$ -axis shows the average best results. ....	95
FIGURE 20	– Case study III: Box plot for the results obtained ( $D = 200$ ). ....	96
FIGURE 21	– Case study IV: Box plot for the results obtained ( $D = 200$ ). ....	101
FIGURE 22	– Case study IV: Average number of habitats at each ecological succession. ....	102
FIGURE 23	– Case study V: Box plot for the results obtained. ....	104
FIGURE 24	– Case study V: Best states found by ECO approaches listed in Table 16. ...	109

## LIST OF TABLES

TABLE 1	– Distance matrix for five items. ....	61
TABLE 2	– Single-link result for the data in Table 1. ....	62
TABLE 3	– Linearly scaled values for distance. ....	63
TABLE 4	– Mathematical Functions ....	78
TABLE 5	– Benchmark sequences for the 2D-AB <i>off-lattice</i> model ....	82
TABLE 6	– Case study I: Results obtained for the benchmark functions. ....	85
TABLE 7	– Case study I: Mean of the processing time over all runs (minutes). ....	86
TABLE 8	– Case study II: Results obtained for the benchmark functions. ....	90
TABLE 9	– Case study II: Mean of the processing time over all runs (minutes). ....	93
TABLE 10	– Case Study III: Results obtained for the benchmark functions. ....	95
TABLE 11	– Case study III: Mean of the processing time over all runs (minutes). ....	98
TABLE 12	– Case study IV: Results obtained for the benchmark functions. ....	100
TABLE 13	– Case study IV: Mean of the processing time over all runs (minutes). ....	102
TABLE 14	– Case study V: Quality of solutions using ECO approaches. ....	103
TABLE 15	– Case study V: Mean of the processing time over all runs (hours). ....	106
TABLE 16	– Case study V: Comparative of best solutions from different strategies with best ECO solutions ( $ECO_{Best}$ ). ....	107
TABLE 17	– Case study V: Comparative of overall best solutions found in literature ( $E^*$ ) with best ECO solutions ( $ECO_{Best}$ ). ....	107

## LIST OF ABBREVIATIONS

AB	AB off-lattice model
ABC	Artificial Bee Colony Algorithm
ACMC	Annealing Contour Monte Carlo
ACO	Ant Colony Optimization
API	Application Programming Interface
BBO	Biogeography-based Optimization
CAS	Complex Adaptive Systems
CPU	Central Processing Unit
CSA	Conformational Space Annealing
CSS	Cooperative Search Strategies
DE	Differential Evolution
DE-RI	Differential Evolution with Ring-Island
DEadp	Differential Evolution with self-adaptation
DNA	Deoxyribonucleic acid
EA	Evolutionary Algorithms
EC	Evolutionary Computation
ECO	Ecology-inspired Optimization
ELP <sub>+</sub>	Improved Energy Landscaping Paving
GA	Genetic Algorithms
GPU	Graphical Processing Units
HBS	Hybrid Bio-inspired Systems
HP	Hydrophobic-Polar
HSI	Habitat Suitability Index
ISODATA	Iterative Self-Organizing Data Analysis Technique
<i>k</i> -NN	Nearest Neighbor
KDD	Knowledge Discovery in Databases
MRF	Markov Random Field
PBHS	Population-Based Harmony Search
PDB	Protein Data Bank
PEA	Parallel Evolutionary Algorithms
PERM	Pruned-Enriched Rosenbluth Method
POSIX	Portable Operating System Interface
PSO	Particle Swarm Optimization
PSP	Protein Structure Prediction
QoS	Quality of Service
SI	Swarm Intelligence
SIV	Suitability Index Variable

## SUMMARY

<b>1 INTRODUCTION</b>	<b>23</b>
1.1 MOTIVATION	25
1.2 OBJECTIVES	26
1.3 DOCUMENT STRUCTURE	26
<b>2 CONCEPTUAL BACKGROUND</b>	<b>29</b>
2.1 META-HEURISTICS	29
2.1.1 Population-based Meta-Heuristics	31
2.2 EVOLUTIONARY COMPUTATION AND SWARM INTELLIGENCE	31
2.2.1 Artificial Bee Colony Optimization	33
2.2.2 Particle Swarm Optimization	33
2.2.3 Differential Evolution	36
2.2.4 Biogeography-based Optimization	36
2.2.5 EC and SI Parallel Models	38
2.3 COOPERATIVE SEARCH STRATEGIES	40
2.4 HYBRID BIO-INSPIRED SYSTEMS	42
2.4.1 Engineered HBS	42
2.4.2 Bio-plausible HBS	43
2.5 ECOLOGY AND ECOSYSTEMS	46
2.6 RELATED WORKS	50
<b>3 COMPUTATIONAL ECOSYSTEM FOR OPTIMIZATION</b>	<b>55</b>
3.1 DESCRIPTION	55
3.2 COMPUTATIONAL ISSUES	58
3.2.1 Environment Definition	58
3.2.2 System Initialization	58
3.2.3 System Heterogeneity	59
3.2.4 Ecological Succession	59
3.2.5 Definition of Habitats	60
3.2.6 Communication Topologies	64
3.2.7 System Flows	64
3.2.8 Ecological Relationships	65
3.2.9 Environmental Factors	65
3.2.10 Diversity Maintenance	66
3.2.11 System Synchronism	67
3.2.12 System Organization	67
3.3 A CANONICAL ECO-INSPIRED ALGORITHM	67
3.3.1 Conceptual Illustration	70
<b>4 EXPERIMENTS, RESULTS, AND ANALYSIS</b>	<b>77</b>
4.1 BENCHMARK FUNCTIONS	78
4.2 PROTEIN STRUCTURE PREDICTION PROBLEM	78
4.2.1 The AB off-lattice model	81
4.3 SETUP OF PARAMETERS AND STATISTICAL TESTS	82
4.4 CASE STUDY I: FIRST EXPERIMENTS	84

4.4.1 Results and Analysis .....	84
4.4.2 Considerations .....	87
4.5 CASE STUDY II: POPULATION DYNAMICS .....	88
4.5.1 Results and Analysis .....	90
4.5.2 Considerations .....	92
4.6 CASE STUDY III: HETEROGENEOUS MODEL .....	93
4.6.1 Results and Analysis .....	94
4.6.2 Considerations .....	98
4.7 CASE STUDY IV: HIERARCHICAL CLUSTERING FOR HABITATS FORMATION .....	98
4.7.1 Results and Analysis .....	99
4.7.2 Considerations .....	102
4.8 CASE STUDY V: PROTEIN FOLDING PROBLEM .....	102
4.8.1 Results and Analysis .....	103
4.8.2 Comparison with Other Approaches .....	106
4.8.3 Considerations .....	108
<b>5 CONCLUSIONS .....</b>	<b>111</b>
5.1 CONTRIBUTIONS .....	113
5.2 FUTURE RESEARCH .....	116
<b>REFERENCES .....</b>	<b>119</b>

## 1 INTRODUCTION

Nature has always been an endless source of inspiration for computational models and paradigms, particularly for the computer scientists of the area known as Natural Computing. In Natural Computing there are three main branches of research which combines computation with biology: simulation and emulation of biology by means of computation, computation with biological materials, and computation with inspiration from biology (DE CASTRO, 2007).

The first branch uses computational methods to help the understanding of biological processes and phenomena. With the development of advanced computational techniques, one now has unprecedented tools to support the study of biology (DE CASTRO, 2007). Generally, rather than using computational means to research biology, the second branch uses biological means to solve non-biological problems, such as DNA computing (WATADA, 2008). The last branch is the bio-inspired computation that takes inspiration from the Biology and develops systems to solve real-world problems (YANG, 2010). The present work is concentrated on bio-inspired systems.

Some examples of natural processes that inspired computational methods are: the evolution of species, the multicellular development of organisms, the animal nervous connections, the immunological system in vertebrates, the social behavior of insects, and the ecological relationships between populations. Evolutionary algorithms (ENGELBRECHT, 2007), cellular automata (GANGULY et al., 2003; KARI, 2005), artificial neural networks (MALIK, 2005), artificial immune systems (DASGUPTA et al., 2011), and swarm algorithms (PARPINELLI; LOPES, 2011b) are some of those bio-inspired methods. An important fact to be highlighted is that, despite the variety of available bio-inspired optimization strategies, it is generally difficult to determine a priori the best algorithm(s) to solve a given problem instance (WOLPERT; MACREADY, 1997).

Given limited time and resource, such bio-inspired algorithms are very effective in providing good quality solutions for several real problems. However, they tend to lose their efficiency when applied to large and complex problems. This undesirable feature that affects many of these algorithms is called “curse of dimensionality” (BELLMAN, 2003), which implies

that their performance deteriorates quickly as the dimensionality of the search space increases. The reasons for this phenomenon appear to be two-fold. First, the solution space of a problem often increases exponentially with the problem dimension. Second, also the characteristics of the problem may change with the dimensionality. An example is the Rosenbrock function that is unimodal for two dimensions but becomes multimodal for higher ones (SHANG; QIU, 2006). Because of such a worsening of the features of an optimization problem resulting from an increase in dimensionality, a previously successful optimization strategy may no longer be capable of finding a good solution. In both cases more efficient optimization strategies are required to explore all promising regions within a limited period of time.

It is worth mentioning that most bio-inspired algorithms only focus on and took inspiration from specific aspects of the natural phenomena. However, in nature, biological systems are interlinked to each other, e.g. biological ecosystems. Hence, hybrid bio-inspired systems arise and are developed with and defined by cooperative search concepts. Cooperative search strategies involve concepts of parallelism and hybridism where a set of potentially good algorithms for the optimization problem are executed in parallel, sharing information during the run (EL-ABD; KAMEL, 2005). These hybrid strategies are expected to provide more efficient and flexible approaches to solve complex problems that would be very difficult to solve with simple methods. Some related works have shown good results when using hybrid bio-inspired search strategies cooperatively (MASEGOSA et al., 2008; PARPINELLI et al., 2011; BENÍTEZ et al., 2012).

With such diversity of search strategies and the advantages of applying them cooperatively, it is possible to establish an analogy with the dynamics of biological ecosystems. An ecosystem can be considered as a set of species that interact and share information with each other in a given environment, and always search for an adapted and equilibrated state against disturbances that may suffer (i.e., homeostatic state) (MAY; MCLEAN, 2007). In this analogy with biological ecosystems each species can behave according to an optimization algorithm. The ecosystem as a whole can be composed by species that respond to environmental and ecological stimuli. To the best of our knowledge, ecosystems and their concepts such as habitats, ecological relationships, ecological succession, environmental factors, and others (BEGON et al., 2006; MAY; MCLEAN, 2007), have not been explored computationally as an ecological framework for function optimization and, therefore, they are addressed in this work. In (PARPINELLI; LOPES, 2011a) the authors illustrate the potentiality of such ecological concepts presenting a canonical ecology-inspired algorithm that will be later discussed in Section 3.3.

Therefore, the main discussion presented in this work relates to the cooperative use



of populations of candidate solutions, coevolving in an ecological context. With this ecology-based analogy, each population can behave according to a specific search strategy, employed in the evolution of candidate solutions. In addition to the possibility of using different optimization strategies cooperatively, this analogy opens the possibility of inserting ecological concepts in the optimization process, thus allowing the development of new bio-plausible hybrid systems.

## 1.1 MOTIVATION

The first motivation for the development of this thesis is to better understand the theories and concepts involved in the ecology of biological systems in order to promote the development of more plausible biologically inspired optimization algorithms. In this way, a computational algorithm or system can be classified as biologically plausible if it has some admissible and consistent association with the current knowledge of the biological process being explored. Hence, we believe that the concepts and processes involved in real ecosystems can be a source of inspiration with countless features and interesting possibilities for the development of computer systems, with emphasis on the self-organizational feature present in biological systems.

Problems that are non-deterministic, dynamic, nonlinear, with multiple objectives, with high-dimensionality, or with any combination of these features, usually have a high degree of complexity and are commonly typical of real world problems. Also, possible solutions to these problems can lead to substantial gains in the fields of economy, environment, and science. Thus, another motivation is to propose more robust computational models for solving problems with any of these features.

Concerning the “curse of dimensionality” inherent for many applications of population-based meta-heuristics to real world problems, one of the possible causes of it may be the under-exploitation of the potential offered by the search methods employed. It is believed that this is the case, for example, of the island model employed by evolutionary and swarm algorithms (Section 2.2.5). In most applications, the employed model uses static topologies with, at most, two search strategies (one for intensification and other for diversification). Hence, another motivation for the development of this work is to propose a cooperative ecology-inspired computational model that allows the aggregation of different meta-heuristics and the incorporation of ecological concepts in the context of optimization.

## 1.2 OBJECTIVES

The general objective of this work is to highlight the main properties of ecosystems that can be important for building computational tools to solve complex optimization problems, as well as to introduce computational descriptions for such plausible biologically inspired functionalities (e.g., habitats, ecological relationships, and ecological succession). In other words, the aim is to generate a solid theoretical foundation for the design of more plausible biologically inspired systems for optimization inspired by concepts and processes involved in real ecosystems.

The first specific objective is the conceptual definition of the ecosystemic environment where different meta-heuristics will act cooperatively. Shall be defined the biotic and abiotic components that will act on the environment, as well as the interaction rules between them.

The second specific objective is the determination of macro and microevolutionary strategies that will act in the individuals. In this item, in addition to the definition of natural selection routines, migration policies, among others, the key point is the definition of mechanisms for creating the habitats in the computational ecosystem.

The third specific objective is focused on the definition and experimentation of a new ecosystemic level of coevolution. In this item, intra and inter-habitats communications play the main role.

The fourth specific objective is to define the homeostatic mechanisms to maintain the ecosystem stable and enabling the continued life of the species.

The last specific goal to be achieved in this work is the application of the proposed conceptual model to complex optimization problems and compare the results obtained with the results already reported in the literature.

## 1.3 DOCUMENT STRUCTURE

The reminder of this work is structured as follows:

- Chapter 2 presents a review of the major topics required for a better theoretical understanding on the approached theme. The topics covered involve the conceptualization of heuristics and meta-heuristics, evolutionary computation and swarm intelligence, cooperative search strategies, hybrid bio-inspired systems, ecology and ecosystems, and a description of some related works;

- Chapter 3 describes the basic organization of a computational ecosystem and presents some fundamental ecological concepts that can be explored in the context of problem solving. Also, this section illustrates the potentiality of some ecological concepts presenting a canonical ecology-inspired algorithm for optimization;
- In Chapter 4 are presented the problems solved in this thesis, as well as how the experiments were conducted. Also, this section presents the case studies approached with its results, analysis and considerations;
- Chapter 5 concludes the document presenting the final considerations, contributions, and future research directions.



## 2 CONCEPTUAL BACKGROUND

In this section, a review of the key concepts necessary for a better theoretical understanding is presented.

### 2.1 META-HEURISTICS

A heuristic is an approximate method that seeks for near optimal solutions for certain classes of problems in a reasonable computational time. However, they do not have the compromise of guaranteeing the feasibility and optimality of the solutions found and, in some cases, it cannot even determine how close to the global optimum the solution found is. In contrast with an exact optimization method, a heuristic calculates the value of an objective function at points considered promising, identified along an iterative process. At the end of this process a solution to the problem is reported (REEVES, 1995).

The term meta-heuristics, first introduced by (GLOVER, 1986), can be seen as an iterative process for generating solutions using one or more embedded heuristics. In other words, meta-heuristics basically tries to combine basic heuristic methods in higher level frameworks aimed at efficiently and effectively exploring a search space. However, up to now there is no commonly accepted definition for the term meta-heuristic. Some definitions are:

“A meta-heuristic is formally defined as an iterative generation process which guides a subordinate heuristic by combining intelligently different concepts for exploring and exploiting the search space. Learning strategies are used to structure information in order to find efficiently near-optimal solutions” (OSMAN; LAPORTE, 1996).

“A meta-heuristic is a set of concepts that can be used to define heuristic methods that can be applied to a wide set of different problems. In other words, a meta-heuristic can be seen as a general algorithmic framework which can be applied to different optimization problems with relatively few modifications to make them adapted to a specific problem.”<sup>1</sup>.

---

<sup>1</sup>Website: <http://www.metaheuristics.net/> Visited in 2013.

Meta-heuristics are designed to tackle complex optimization problems where other optimization methods have failed to be either effective or efficient. The practical advantage of meta-heuristics lies in both their effectiveness and general applicability. For further reading, (GLOVER; KOCHENBERGER, 2003) provides a good introduction and general reference to many of the most popular meta-heuristics.

A meta-heuristic incorporates strategies to explore the space of solutions beyond local optima and usually these strategies are non-deterministic. Also, meta-heuristics are inspired by processes studied in different fields of science as, for example, the social sciences, physics, biology, etc. (DRÉO et al., 2006).

Some fundamental properties which characterize meta-heuristics are:

- Strategies that “guide” the search process;
- The goal is to efficiently explore the search space in order to find (near-)optimal solutions;
- The algorithms are approximate and usually non-deterministic methods;
- The basic concepts permit an abstract level description;
- They may incorporate mechanisms to avoid getting trapped in confined areas of the search space;
- Not problem-specific;
- They may make use of domain-specific knowledge in the form of heuristics that are controlled by the upper level strategy.

Each meta-heuristics has its own search strategies and these are directly related to the philosophy that inspires the creation of the meta-heuristics itself. These search strategies, intrinsic to each meta-heuristics, can be classified into diversification (*exploration*) strategies and intensification (*exploitation*) strategies (BLUM; ROLI, 2003). The term diversification refers to the global exploration of the solution space in the search for the most promising regions. On the other hand, the term intensification refers to the local exploitation of the solution space in the search for better solutions surrounding the promising regions.

The performance of a meta-heuristic is directly associated with an appropriate balance between the intensification and diversification procedures.

### 2.1.1 POPULATION-BASED META-HEURISTICS

Some meta-heuristics such as tabu search (GENDREAU, 2003) and simulated annealing (HENDERSON et al., 2003)(NOLTE; SCHRADER, 2000) perform the search by iteratively changing only one candidate solution at each iteration. However, this is not the case when applying population-based meta-heuristics, in which the main feature is the optimization process taking place in a set of candidate solutions at each iteration. This set of solutions can be called simply population, or else, swarm, school or hive, depending on the biological inspiration employed, and each corresponding candidate solution can be an individual, a particle, a bee or an ant. This is the case, for instance, of Evolutionary Algorithms (FOGEL, 2006), Particle Swarm Optimization (POLI et al., 2007), Ant Colony Optimization (DORIGO; STÜTZLE, 2004) and several other methods (PARPINELLI; LOPES, 2011b).

Algorithm 1 shows a general pseudocode of a population-based algorithm (BENÍTEZ et al., 2012). The main loop (between lines 3–7) represents the generational loop, and line 4 defines the mechanism or criterion for selecting the best solutions (i.e., survival of the fittest as in Evolutionary Computation (EC), or simply discarding the worst solutions). In line 5 of the algorithm, intensification (also known as exploitation) intends to search locally and more intensively around the selected solutions (i.e., a crossover procedure in Genetic Algorithms (GA), or a greedy search), while diversification (known as exploration) leads the algorithm to explore globally the search space (i.e., a mutation procedure in GA, or a large-scale randomization).

- 1: *Initialize* the population with random candidate solutions;
- 2: *Evaluate* each candidate solution;
- 3: **while** convergence criteria is not satisfied **do**
- 4:   Perform competitive *selection*;
- 5:   Apply *intensification* and *diversification* procedures;
- 6:   *Evaluate* the new pool of candidate solutions;
- 7: **end while**

**Algorithm 1:** General pseudocode of a population-based algorithm.

## 2.2 EVOLUTIONARY COMPUTATION AND SWARM INTELLIGENCE

The concept of optimization can be abstracted from several natural processes, for instance, in the evolution of species, in the behavior of social groups, in the dynamics of the immune system, in the food search strategies and in the ecological relationships of different animal populations. The two main families of optimization algorithms that constitute bio-

inspired approaches are the EC and the Swarm Intelligence (SI). Both EC and SI are subareas of Natural Computing.

One of the first sources of inspiration from nature for the development of bio-inspired computing systems is the theory of evolution of species. The EC involves concepts of natural selection, reproduction, genetic recombination (*crossover*), mutation and survival of the fittest to the environment. The environment is considered as a hypersurface of optimization, usually defined by one or more cost or profit functions. Individuals of a population are candidate solutions, and interact and compete with each other to produce descendants for the next generation. The best fitted individuals (situated in regions of the hypersurface with satisfactory values for the cost or profit function) are more likely to reproduce and therefore propagate their genetic material. The EC genetic material is like DNA of living beings encoded in a solution vector representing decision variables for a given problem being solved.

The EC algorithms that are based on these fundamental principles of biological evolution are called Evolutionary Algorithms (EA) (EIBEN; SMITH, 2003). The main and most used EA are Genetic Algorithms (GOLDBERG, 1989), Memetic Algorithms (MOSCATO, 2003), Genetic Programming (KOZA, 1992), Gene Expression Programming (FERREIRA, 2001), and Differential Evolution (STORN; PRICE, 1997). These EA approaches are applied in different domains (ENGELBRECHT, 2007).

The family of swarm-based algorithms are inspired by the behavior of some social living beings, such as ants, termites, birds, and fishes. Self-organization and decentralized control are remarkable features of swarm-based systems that, such as in nature, leads to an emergent behavior. Emergent behavior is a property that emerges through local interactions among system components and it is not possible to be achieved by any of the components of the system acting alone. An example of emergent behavior is observed in ant colonies when larvae and bodies are grouped by ants in a decentralized and self-organized manner. Another example of emergent behavior is the shortest path that ants are able to find between a food source and their nest (BONABEAU et al., 1999; GARNIER et al., 2007).

The main approaches that represent the SI area are Ant Colony Optimization (DORIGO; STÜTZLE, 2004), Particle Swarm Optimization (POLI et al., 2007), and Artificial Bee Colony Optimization (KARABOGA; AKAY, 2009). All these approaches are population-based and the components of the population (swarm) represent candidate solutions to a given problem. The components of the swarm interact directly or indirectly with each other in order to contribute synergistically to the emergent behavior. Many other swarm-based algorithms have emerged and have been successfully applied to a wide variety of problems (PARPINELLI;



LOPES, 2011b; CLERC, 2006).

In the following, some algorithms used in this thesis are described.

### 2.2.1 ARTIFICIAL BEE COLONY OPTIMIZATION

The Artificial Bee Colony Algorithm (ABC) was inspired by the foraging behavior of honey bees. ABC was first proposed by (KARABOGA, 2005) for solving multi-dimensional and multi-modal optimization problems. A recent work (KARABOGA; AKAY, 2009) compared the ABC algorithm performance against other population-based algorithms (Genetic Algorithm, Particle Swarm Optimization, Differential Evolution and Evolution Strategies) upon several benchmark functions. Results showed that the performance of the ABC was better than or similar to those of the other algorithms. Another relevant work concerning the ABC algorithm analysed the tuning of control parameters (AKAY; KARABOGA, 2009).

The ABC algorithm begins with  $n$  solutions (food sources) of dimension  $d$  that are modified by the artificial bees. In the same way as other evolutionary algorithms, each solution  $\vec{x}_i = [x_{i1}, x_{i2}, \dots, x_{id}]$  is evaluated by an objective function  $f(\vec{x}_i)$ ,  $i = 1, \dots, n$ . The bees aim at discovering places of food sources (that is, regions in the search space) with high amount of nectar (good objective function values, meaning good solutions for the problem). There are three types of bees: scout bees that randomly fly in the search space without guidance, employed bees that exploit the neighborhood of their locations selecting a random solution to be perturbed, and onlooker bees that use the objective function to select probabilistically a guiding solution to exploit its neighborhood. If the amount of nectar of a new source is higher than that of the previous one in their memory, they update the new position and forget the previous one (this is a greedy selection method). If a solution is not improved by a predetermined number of trials, controlled by the parameter *limit*, then the food source is abandoned by the corresponding employed bee and it becomes a scout bee. The ABC algorithm attempts to balance exploration and exploitation using the employed and onlooker bees to perform local search, and the scout bees to perform global search, respectively. The canonical ABC is shown in Algorithm 2, and further information about the ABC algorithm can be found in the repository<sup>2</sup>.

### 2.2.2 PARTICLE SWARM OPTIMIZATION

The Particle Swarm Optimization<sup>3</sup> (PSO) meta-heuristic was inspired by the coordinate movement of fish schools and bird flocks (KENNEDY; EBERHART, 2001), and

<sup>2</sup>ABC Repository: <http://mf.erciyes.edu.tr/abc/>

<sup>3</sup>PSO Repository: <http://www.particleswarm.info>

```

1: Set parameters:  $n$ ,  $limit$ 
2: Initialize the food sources  $\vec{x}_i$  randomly
3: Evaluate objective function  $f(\vec{x}_i)$  of the population
4:  $count_i = 0$ 
5: while stop condition not met do
6:   for  $i = 1$  to  $n/2$  do {Employed phase}
7:     Select  $k, j$  and  $r$  at random such that  $k \in \{1, 2, \dots, n\}, j \in \{1, 2, \dots, d\}$ ,
8:      $r \in [0, 1]$ 
9:      $\vec{v} = \vec{x}_{ij} + r \cdot (\vec{x}_{ij} - \vec{x}_{kj})$ 
10:    Evaluate solutions  $\vec{v}$  and  $\vec{x}_i$ 
11:    if  $f(\vec{v})$  is better than  $f(\vec{x}_i)$  then
12:      Greedy selection
13:    else
14:       $count_i = count_i + 1$ 
15:    end if
16:  end for
17:  for  $i = n/2 + 1$  to  $n$  do {Onlooker phase}
18:    Calculate selection probability
19:     $P(\vec{x}_k) = \frac{f(\vec{x}_k)}{\sum_{k=1}^n f(\vec{x}_k)}$ 
20:    Select a bee using the selection probability
21:    Produce a new solution  $\vec{v}$  from the selected bee
22:    Evaluate solutions  $\vec{v}$  and  $\vec{x}_i$ 
23:    if  $f(\vec{v})$  is better than  $f(\vec{x}_i)$  then
24:      Greedy selection
25:    else
26:       $count_i = count_i + 1$ 
27:    end if
28:  end for
29:  for  $i = 1$  to  $n$  do {Scout phase}
30:    if  $count_i > limit$  then
31:       $\vec{x}_i = \text{random}$ 
32:       $count_i = 0$ 
33:    end if
34:  end for
35:  Memorize the best solution achieved so far
36: end while
37: Postprocess results and visualization

```

**Algorithm 2:** Canonical ABC

has been applied to several optimization problems (see, for instance, (POLI et al., 2007; POLI, 2008)).

The PSO is a population-based meta-heuristic composed by a swarm of  $n$  particles. Each particle represents a potential solution to the problem to be solved. The position of a particle in the search space is determined by the solution it currently represents. Algorithm 3 shows the canonical PSO, as described by (KENNEDY; EBERHART, 2001). In this algorithm,

each solution  $\vec{x}_i = [x_{i1}, x_{i2}, \dots, x_{id}]$  of dimension  $d$  is evaluated by an objective function  $f(\vec{x}_i)$ ,  $i = 1, \dots, n$ . In PSO, particles “fly” through the hyperdimensional search space according to their velocity  $\vec{v}_i$ . Changes to the position of the particles within the search space are based on the socio-cognitive tendency of individuals to emulate the success of other individuals. Each individual of a population has its own life experience ( $\vec{p}_i$ ) and is able to evaluate the quality of its experience. They are social individuals and, so, they also have knowledge about the quality of their neighbors ( $\vec{g}$ ). These two sources of information corresponds to the cognitive component (individual learning) and social component (social learning), respectively. Hence, an individual decision is taken considering both the cognitive and the social components, thus leading the population to an emergent behavior of navigating coordinately through the search space.

The parameters  $\varphi_p$ , and  $\varphi_g$  determine the relative influence of the cognitive and social components, respectively, and both are often set to the same value so as to give each component (the cognition and social learning rates) the same decisional weight. The stochastic nature of PSO is evidenced by  $r_p$  and  $r_g$  that are numbers randomly generated in range  $[0, 1]$  each time the equation is computed.

- 1: Set parameters:  $n$ ,  $\varphi_p$ ,  $\varphi_g$
- 2: **for**  $i = 1$  **to**  $n$  **do**
- 3:     Initialize the positions  $\vec{x}_i$  and velocities  $\vec{v}_i$  randomly
- 4:     Evaluate objective function  $f(\vec{x}_i)$
- 5:     Initialize the particle’s best known position to its initial position:  $\vec{p}_i = \vec{x}_i$
- 6:     Evaluate  $f(\vec{x}_i)$
- 7:     **if**  $f(\vec{p}_i)$  is better than  $f(\vec{g})$  **then**
- 8:         Update the swarm’s best known position:  $\vec{g} = \vec{p}_i$
- 9:     **end if**
- 10: **end for**
- 11: **while** stop condition not met **do**
- 12:     **for**  $i = 1$  **to**  $n$  **do**
- 13:         Update particles’ velocity:  $\vec{v}_i = \vec{v}_i + \varphi_p * r_p * (\vec{p}_i - \vec{x}_i) + \varphi_g * r_g * (\vec{g} - \vec{x}_i)$
- 14:         Update particles’ position:  $\vec{x}_i = \vec{x}_i + \vec{v}_i$
- 15:         **if**  $f(\vec{x}_i)$  is better than  $f(\vec{p}_i)$  **then**
- 16:              $\vec{p}_i = \vec{x}_i$
- 17:             **if**  $f(\vec{p}_i)$  is better than  $f(\vec{g})$  **then**
- 18:                  $\vec{g} = \vec{p}_i$
- 19:             **end if**
- 20:         **end if**
- 21:     **end for**
- 22: **end while**
- 23: Postprocess results and visualization

**Algorithm 3:** Canonical PSO

### 2.2.3 DIFFERENTIAL EVOLUTION

The Differential Evolution (DE) (STORN; PRICE, 1997) is an optimisation method from the EC field and was proposed for solving optimization problems. DE is conceptually simple, easy to implement and has proven to be flexible and achieve good solutions for many interesting problems (PLAGIANAKOS et al., 2008; NERI; TIRRONEN, 2010). The DE is a population-based meta-heuristic composed by  $n$  solution vectors (candidate solutions). Each vector  $\vec{x}_i = [x_{i1}, x_{i2}, \dots, x_{id}]$  of dimension  $d$  represents a potential solution to the problem to be solved and each of them are evaluated by an objective function  $f(\vec{x}_i)$ ,  $i = 1, \dots, n$ . The basic idea of DE is the use of difference vectors for generating perturbations in a population  $Pop$  to form a new population  $NewPop$  of solution vectors. At each iteration, new vectors are generated by the combination of randomly chosen vectors weighted by a constant  $F$ . This operation, in the EC context, can be referred to as mutation. The outcoming vectors are then probabilistically mixed with another predetermined vector (i.e., the target vector) according to a parameter  $CR$ . This operation can be called recombination. This operation yields the so-called trial vector ( $\vec{y}$ ). The trial vector is accepted for the next iteration if and only if it reduces the value of the objective function. This operation can be referred to as a greedy selection.

DE has a specialized nomenclature that describes the configuration adopted. This takes the form of DE/ $x/y/z$ , where  $x$  represents the solution to be perturbed (such a *random* or *best*). The  $y$  indicates the number of difference vectors used in the perturbation of  $x$ , where a difference vector is the difference between two or more randomly selected, although distinct, members of the population. Finally,  $z$  indicates the recombination operator performed such as *bin* for binomial and *exp* for exponential. Many schemes for creation of candidate solutions are possible. Algorithm 4 shows the canonical DE with DE/*rand/1/bin* scheme, as described by (STORN; PRICE, 1997).

### 2.2.4 BIOGEOGRAPHY-BASED OPTIMIZATION

The Biogeography-based Optimization (BBO) is a population-based, biogeography inspired global optimization algorithm (SIMON, 2008). In BBO, each individual is considered as a “habitat” with a habitat suitability index (HSI), which is similar to the fitness of EAs, and used to measure the quality of the individual. A good solution is analogous to an island with a high HSI, and a poor solution indicates an island with a low HSI. High HSI solutions tend to share their features with low HSI solutions. Low HSI solutions accept a lot of new features from high HSI solutions.

A potential solution to the problem to be solved is represented by an individual  $\vec{x}_i =$

```

1: Set parameters:  $n, F, CR$ 
2: for  $i = 1$  to  $n$  do
3:   Initialize solution vector  $\vec{x}_i$  in  $Pop$  randomly
4:   Evaluate objective function  $f(\vec{x}_i)$ 
5: end for
6: Find the best solution vector  $S_{best}$  in  $Pop$ 
7: while stop condition not met do
8:    $NewPop = \emptyset$ 
9:   for  $i = 1$  to  $n$  do
10:    Select random indices  $r_1, r_2, r_3 \in n$  with  $r_1 \neq r_2 \neq r_3 \neq i$ 
11:    Select a dimension  $p \in d$  at random
12:    for  $j = 1$  to  $d$  do
13:      if ( ( $j$  is equal to  $p$ )  $\vee$  ( $rand() \leq CR$ ) ) where  $rand() \sim U[0,1]$  then
14:         $y_j = x_{r_3,j} + F \times (x_{r_1,j} - x_{r_2,j})$ 
15:      else
16:         $y_j = x_{ij}$ 
17:      end if
18:    end for
19:    Evaluate  $f(\vec{y})$ 
20:    if  $f(\vec{y})$  is better than  $f(\vec{x}_i)$  then
21:       $NewPop = \vec{y}$ 
22:    else
23:       $NewPop = \vec{x}_i$ 
24:    end if
25:  end for
26:   $Pop = NewPop$ 
27:  for  $i = 1$  to  $n$  do
28:    Evaluate objective function  $f(\vec{x}_i)$ 
29:  end for
30:  Find the best solution vector  $S_{best}$  in  $Pop$ 
31: end while
32: Postprocess results and visualization

```

**Algorithm 4:** Canonical DE with DE/*rand//bin* scheme

$[x_{i1}, x_{i2}, \dots, x_{id}]$  of dimension  $d$ . Each dimension in the solution vector is considered to be a suitability index variable (SIV). Also, each individual is evaluated by an objective function  $f(\vec{x}_i)$ ,  $i = 1, \dots, n$ . In BBO, each individual  $\vec{x}_i$  has its own immigration rate  $\lambda_i$  and emigration rate  $\mu_i$ . A good solution has high  $\mu$  and low  $\lambda$ , and the opposite for bad solutions. The immigration rate and the emigration rate are functions of the number of species in the habitat.

In Algorithm 5,  $n$  is the maximum number of habitats;  $E$  is the maximum possible emigration rate;  $I$  is the maximum possible immigration rate;  $k$  is the number of species of the  $k$ -th individual;  $S_{max}$  is the largest number of species a habitat can support;  $m_{max}$  is a user-defined mutation weight; and *elite* is the number of best individuals that will survive to next generation directly.

```

1: Set parameters:  $n, E, I, S_{max}, m_{max}, elite$ 
2: for  $i = 1$  to  $n$  do
3:   Initialize habitat  $\vec{x}_i$  randomly
4:   Evaluate objective function  $f(\vec{x}_i)$ 
5:   Initialize the species count probability of each habitat:  $P_i = \frac{1.0}{n}$ 
6: end for
7: while stop condition not met do
8:   Identify elite habitats
9:   Map HSI to the number of species by sorting the
   population from best to worst
10:  for  $i = 1$  to  $n$  do
11:    Calculate the immigration rate  $\lambda_i = I \left(1 - \frac{k}{n}\right)$ 
12:    Calculate the emigration rate  $\mu_i = \left(\frac{Ek}{n}\right)$ 
13:  end for
14:  for  $i = 1$  to  $n$  do {Habitat recombination}
15:    if ( $rand() \leq \lambda_i$ ) where  $rand() \sim U[0,1]$  then
16:      for  $j = 1$  to  $n$  do
17:        if ( $rand() \leq \mu_i$ ) where  $rand() \sim U[0,1]$  then
18:          Randomly select an SIV  $\sigma$  from  $\vec{x}_j$ 
19:          Replace a random SIV in  $\vec{x}_i$  with  $\sigma$ 
20:        end if
21:      end for
22:    end if
23:  end for
24:  for  $i = 1$  to  $n$  do
25:    Compute the time derivative  $P_i$  for each habitat
26:    Compute mutation rate for each habitat:  $m_i = m_{max} \left(\frac{1-P_i}{P_{max}}\right)$ 
   where  $P_{max} = argmax(P_i)$ 
27:  end for
28:  for  $i = 1$  to  $n$  do {Habitat mutation}
29:    for  $j = 1$  to  $d$  do
30:      if ( $rand() \leq m_i$ ) where  $rand() \sim U[0,1]$  then
31:        Replace  $\vec{x}_{ij}$  with a randomly generated SIV
32:      end if
33:    end for
34:    Evaluate objective function  $f(\vec{x}_i)$ 
35:  end for
36:  Keep elite habitats
37: end while
38: Postprocess results and visualization

```

**Algorithm 5:** Canonical BBO

### 2.2.5 EC AND SI PARALLEL MODELS

The fact that EA and SI algorithms manipulate a population of independent solutions make them suitable for the use of parallel computing architectures (MÜHLENBEIN et al., 1991;

PARPINELLI et al., 2011; SCHUTTE et al., 2003). The two main motivations for using parallel population-based algorithms are:

- To improve the performance (processing time) of the optimization process performing concurrent function evaluations of the candidate solutions;
- To improve the solution search process (quality of solution) overcoming difficulties that traditional population-based algorithms can face, for example, the diversity maintenance in order to prevent premature convergence.

The first to be parallelized were the EA and defined the Parallel Evolutionary Algorithms (PEA). The swarm-based parallel algorithms have been parallelized subsequently and use, mostly, parallel concepts applied in PEA.

A taxonomy classifying the various development strategies for PEA was proposed in the work of (NOWOSTAWSKI; POLI, 1999). The two main strategies are:

- Island model with multiple populations and migrations (coarse grained);
- Single population model with neighborhood diffusion (fine grained).

In the coarse grained island model the evolution occurs in multiple parallel subpopulations (islands), each one running an EA or SI algorithm. Each island evolves independently, and sporadic migrations of individuals may occur between subpopulations. The main parameters to be defined for the island model are (ALBA; TROYA, 1999; TOMASSINI, 1999, 2005):

- The number of subpopulations (islands);
- Homogeneity of the islands: Defines the evolutionary algorithms used and the parameters of each algorithm (number of individuals, *crossover* rate, mutation rate, etc). Islands with the same settings characterize a homogeneous model;
- Connection topology of the islands: ring, star, fully connected, and random;
- Static or dynamic connectivity: If the topology does not change during the optimization process, the connectivity is characterized as static. Otherwise it is characterized as dynamic;
- Mechanisms and policies of migration: To define how often the migration will occur, which individuals may migrate, what action to take when an island receives an individual, if the migration will be isolated, synchronous or asynchronous.

In the fine grained island model, each individual is associated with a processing unit (CPU or thread, for example) (ALBA; TROYA, 1999; TOMASSINI, 1999, 2005). A local neighborhood topology is then defined and it allows individuals to reproduce only with their neighbors. The exchange of information between individuals happens by overlapping neighborhoods and creates an implicit migration mechanism. All processing units run the same algorithm. Then, the algorithm selects individuals from the local neighborhood to reproduce, recombines these individuals, produces a descendant and decides when to replace the current individual by a descendant.

Several publications of PEA and parallel swarm-based algorithms can be found in various scientific search portals. An interesting reference is (ALBA; TOMASSINI, 2002) that reports some applications in different areas such as operations research, engineering, manufacturing, finance, design of electronic circuits, and telecommunications.

### 2.3 COOPERATIVE SEARCH STRATEGIES

For several algorithms to cooperate with each other to find the good solutions for a given problem, Cooperative Search Strategies (CSS) should be defined. The CSS use concepts of hybridism and parallelism.

The hybridism results from the combination of exact or approximation algorithms (or both) in order to improve the exploration and exploitation of the search space. The parallelism emerges as a necessary strategy for solving large-scale and high complexity problems (e.g., NP-complete problems) in an acceptable amount of time.

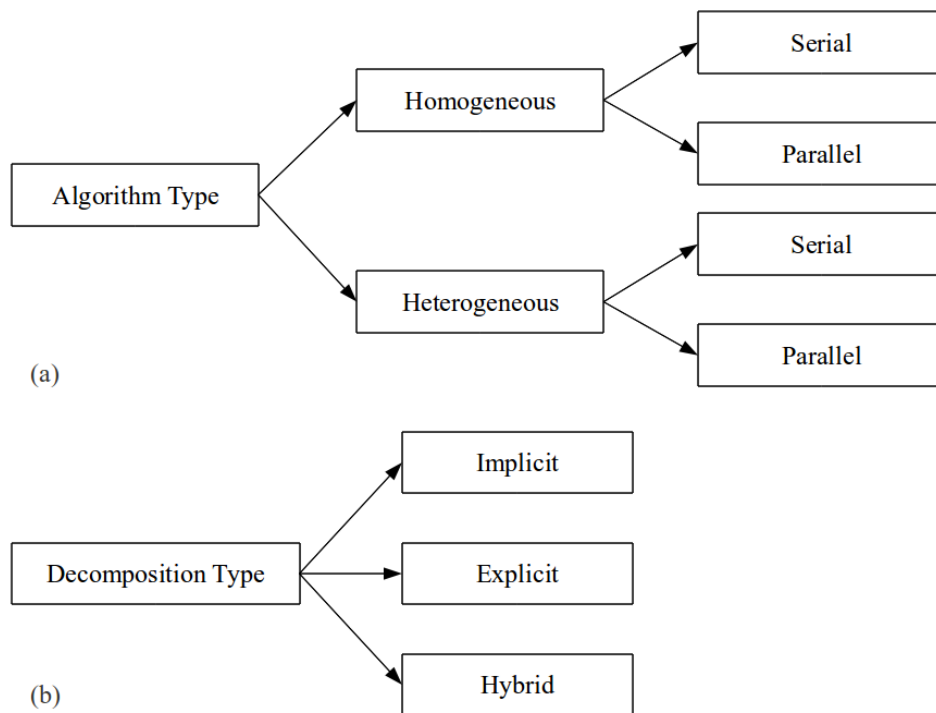
In (EL-ABD; KAMEL, 2005) two taxonomies have been proposed for the CSS. The first taxonomy (Figure 1a) considers the algorithms homogeneity and the implementation strategy used in the cooperative system, creating four categories:

- Homogeneous serial: different instances of the same algorithm are applied in a serial mode where each instance provides a partial solution to the problem. The partial solutions are used to generate a complete solution that is then evaluated and used in subsequent iterations;
- Homogeneous parallel: involves the parallel application of different instances of the same algorithm working independently. The search process uses information exchange between algorithmic instances (e.g., periodic migration of candidate solutions between algorithms);



- Heterogeneous serial: involves the serial application (*pipeline*) of different algorithms. The output of each algorithm feeds the input of the next algorithm, and so on;
- Heterogeneous parallel: same definition of homogeneous parallel but with the implementation of different algorithms.

It is worth mentioning here that implementation strategy is not synonymous of execution architecture. The latter defines the physical/hardware execution architecture of the system, which may be serial, parallel, or *tandem/interleaved*. That is, one can have a homogeneous parallel cooperative system (implementation strategy) running on a serial architecture (execution architecture).



**Figure 1: Taxonomy for cooperative search algorithms.** (a) Considers the diversity of the algorithms employed and the implementation strategy, (b) Considers the type of decomposition of the search space (EL-ABD; KAMEL, 2005).

The second taxonomy (Figure 1b) considers the way in which the search space is decomposed, creating three categories:

- Implicit decomposition: this category involves the use of different algorithms (or different instances of the same algorithm) running concurrently in different areas of the search

space (given the initialization conditions and parameters setup for each algorithm), each one running independently and sharing information with each other;

- Explicit decomposition: each algorithm optimizes a different subspace of the problem and provides a partial solution. At the end, all partial solutions are aggregated to form the complete solution;
- Hybrid approach: considers cooperative systems which employ both methods of decomposition.

## 2.4 HYBRID BIO-INSPIRED SYSTEMS

In addition to the typical bio-inspired systems, which mainly took inspiration from independent biological phenomena, there are a plenty of hybrid bio-inspired systems (HBS). According to the methodology used in the design of such HBS, they can be grouped into two different types: engineered HBS and bio-plausible HBS (LIU et al., 2008; PARPINELLI; LOPES, 2012e). In this section, it is described these two groups giving some examples of each one.

### 2.4.1 ENGINEERED HBS

The first group is the engineered hybrid bio-inspired systems. In these systems, the designers do not necessarily study biological systems, and there is only one purpose to combine more than one bio-inspired algorithm together that is to create a new algorithm. The most widely-seen implementation is to use an algorithm to serve another. For instance, using a swarm algorithm or an artificial immune algorithm to optimise parameters of an artificial neural network, such as in (KARABOGA et al., 2007; SOCHA; BLUM, 2007; CASTRO; VON ZUBEN, 2011; GHALAMBAZ et al., 2011). In this system, both swarm and immune algorithms are practically exchangeable, since they are both used for the same purpose that is optimization. In theory, almost all optimization algorithms can be potentially applicable to this task. Designers will choose one or another only because it performs better for a particular application. In (LUNG; DUMITRESCU, 2010) a hybrid approach is designed to deal with moving optima of optimization problems in dynamic environments. The algorithm uses three populations of individuals: two of them are evolved by a modified differential evolution algorithm and one is evolved by a particle swarm algorithm. The populations evolved by differential evolution are used to maintain the diversity of the search whilst the other one improves the precision of the search process.

Another engineered HBS implementations are some algorithms with different biological inspiration but with similar engineering functionalities. For instance, (KNIDEL et al., 2005) combined the ideas of self-organizing maps and immune systems (AiNet) together to be a new data clustering algorithm. Also, in (WHITE; YEN, 2005) the authors combined ideas of genetic algorithms and ant colony algorithms together to solve a traveling salesman problem. Another example is presented in (BISWAS et al., 2007) which couples synergistically a bacterial foraging algorithm (BFOA) with a particle swarm algorithm (PSO) to solve numerical benchmarks. The proposed algorithm performs local search through the chemotactic movement operation of BFOA whereas the global search is accomplished by a PSO.

In both cases, biological phenomena and inspiration are not as important as the combined techniques which, hopefully, may provide better performance.

#### 2.4.2 BIO-PLAUSIBLE HBS

Bio-plausible hybrid computational systems directly take inspiration from different aspects of biological phenomena. The main feature of these systems is the use of biological plausibility at some degree to hybridize bio-inspired algorithms. Designers of these systems generally aim to achieve plausible biologically inspired functionalities in non-biological contexts, such as the computational optimization of engineering problems.

The island model widely used in applications involving evolutionary computation and swarm intelligence techniques is an example of bio-plausible HBS (KALEGARI; LOPES, 2010; PARPINELLI et al., 2011). The plausible biologically inspired functionality is the coevolution of populations achieved by periodic migrations of individuals from one population to another. This scheme involves several self-contained algorithms (search strategies) performing a parallel search and, through information exchange, they cooperate to solve a difficult optimization problem. More often the island model perform better, such that each algorithm provides information to the others to help them (PARPINELLI et al., 2011).

In this model, population-based algorithms (a genetic algorithm, for example) evolve populations of individuals and individuals can migrate between them. This model is defined by several parameters: the topology that defines the connectivity between populations, the migration rate that controls the number of migrant individuals, the replacement strategy used, and a migration interval that defines how often migration occurs. Also, it is necessary to define if the model is homogeneous or heterogeneous concerning the strategies involved. In a homogeneous model all search strategies are the same whereas in a heterogeneous model, different strategies are used.

Recently, a parametric study done by (MASEGOSA et al., 2008) suggests that the cooperative use of heterogeneous search strategies give better results when compared to homogeneous models. The main reason to this improvement is the cooperative information exchange between search strategies employing different intensification and diversification procedures. Also, as in nature, populations can behave differently from each other. Hence, heterogeneous models are more bio-plausible than homogeneous models.

Most developments using the island model found in the literature apply the search strategies homogeneously (ALBA; TOMASSINI, 2002; MA et al., 2009; BORGULYA, 2010; JOVANOVIĆ et al., 2010; KALEGARI; LOPES, 2010; LUONG et al., 2010; VANNESCHI et al., 2010; PARPINELLI et al., 2011). However, in some applications the island model is heterogeneous.

In (BENÍTEZ et al., 2012) where the hybridization of an artificial bee colony algorithm (ABC) with a genetic algorithm (GA) for protein structure prediction is reported. In this work the model uses a hierarchical topology with four islands connected by an unidirectional ring with two ABC islands interleaved by two GA islands. At the upper level of the hybrid-hierarchical model, there are multiple-population coarse-grained islands that work independently. At the lower level, there are global single-population master-slaves that distribute the computational effort into several slaves. This combination aims at taking advantage of the benefits of both models and population-based meta-heuristics in a single approach.

In (CADENAS et al., 2007), a different search strategy is applied at each island to solve the knapsack problem. In this work only two islands are defined. One uses a genetic algorithm and the other uses an ant colony optimization algorithm. The island model is developed using a multi-agent architecture where each search strategy is an agent (island) that communicate with other agents through a blackboard. Also, a coordinating agent controls and modifies the behavior of the other agents. Machine learning routines are applied to add intelligence to the coordinator. Learning is achieved through a knowledge extraction process that obtain fuzzy rules. With these rules the coordinator can define the migration policy and change the parameters of a search strategy intelligently if it keeps on having bad results. In another work of the same authors (CADENAS et al., 2009), three search strategies were used with the island model: simulated annealing, genetic algorithm and tabu search.

Another example of bio-plausible HBS is presented in (KRINK; LOVBJERG, 2002). This work is inspired by the idea of life cycle stages found in nature. In biology, the term life cycle refers to the various phases an individual passes through from birth to maturity and

reproduction. This process often leads to drastic changes of the individual. Some life cycle changes are one-time events such as sexual maturity. Another changes are recurrent, such as mating seasons. During its life period, an individual can actively decide about its kind of life in response to its success in the environment. The model proposed by (KRINK; LOVBJERG, 2002) creates a self-adaptive search strategy in which each individual (representing a candidate solution) can decide whether it would prefer to belong to a population of a genetic algorithm, a particle swarm optimization algorithm, or become a stochastic hill climber.

Biological plausibility is also found in Membrane Computing or *P Systems* applications. Membrane Computing abstracts computational models inspired by the architecture and functioning of living cells, tissues and organs (PAUN, 2000; PAUN; PEREZ-JIMENEZ, 2006; TEUSCHER, 2007). *P Systems* involves two basic functions of cellular membranes: compartmentalization and filtering. The basic idea is to consider a distributed and parallel system, such as a cell, structured by means of an hierarchical arrangement of membranes bounding compartments. In a cell, for example, cellular membranes compartmentalize the nucleus, mitochondria, Golgi complex, and other cytoplasmic organelles. Each compartment is composed of various objects (chemical compounds) that interact between themselves according to previously established rules (chemical reactions). Besides, membranes keep other rules that control the input and exit of objects such as the removal of residues from a chemical reaction or the input of nutrients. The computational approaches involving the use of *P systems* and optimization algorithms are named membrane algorithms (NISHIDA, 2004).

In these approaches a number of candidate solutions (objects) for an optimization problem is set in different compartments in a membrane structure. To each compartment a search strategy (evolution rule) is assigned and evolves the candidate solutions present at each compartment. Periodically, a set of communication rules come into action to move objects between membranes. This process is iterated until a predefined number of steps is reached or no significant improvement in the best solution occurs (NISHIDA, 2006).

In (NISHIDA, 2007) a membrane algorithm is proposed to solve the traveling salesman problem. In this application, the inner membrane evolves according to a modified simulated annealing algorithm whereas in the remaining membranes a simplified genetic algorithm is used. Solutions are sent to the neighborhood regions by the communication rule. Usually, the best solution is sent to the inner region and the worst solution is sent to the outer region. In (NISHIDA et al., 2009) the same algorithm is applied to the job-shop scheduling problem.

Another example of bio-plausible HBS is presented in Section 3.3 where the potentiality of some ecological concepts in optimization tools design is shown.

In general, hybrid bio-inspired systems contain more than a single basic bio-inspired component. The differences between these approaches are mainly due to different perspectives of the designers. Some are more keen about the possible performance increment of combining different algorithms, others are more keen about interesting functionalities provided by the biological systems and phenomena, and others still are interested in both. Nevertheless, the objectives are the same: to create better computational systems for particular applications.

## 2.5 ECOLOGY AND ECOSYSTEMS

Ecology is the field of Biology that studies the interactions between all biological entities (biotic) with all non-living physico-chemical matter (abiotic) of the environment. Four ecological levels can be identified:

- Individual (organism) level that studies how individuals affect and are affected by the environment;
- Population level that refers to the growth of a population and the factors that influence its growth;
- Community level that studies the interactions between species and the interactions of these with the abiotic components of the environment;
- Ecosystem level that, in addition to the interactions between species and their interactions with the environment, attempts to the cycles of matter and energy which occur between the biotic and abiotic components of the system.

A system that includes biotic components in a given area, and takes into account the interactions among them and with the abiotic components of the environment, can be considered an ecological system or ecosystem. Furthermore, a flow of energy, matter or information can lead to a definition of a trophic structure (or food web<sup>4</sup>) between species (MAY; MCLEAN, 2007).

An ecosystem consists of populations of species where the actions of each member of the ecosystem can be advantageous, disadvantageous or neutral to the other members, creating a network of interactions. A species can be defined as a population of individuals/organisms having some reproductive isolation related to other populations. The place that an individual occupies in an ecosystem is its habitat which may include individuals of one or more species.

---

<sup>4</sup>A food web depicts feeding connections in an ecological community.

The role or function of a species and how they survive, given the environmental conditions, define their ecological niche (BEGON et al., 2006).

The ecosystem ontogeny (i.e., development) represents a particular form of evolution, and is called ecological succession to differentiate from the selective evolution that occurs internally in the parts of the ecosystem (BEGON et al., 2006). Hence, the ecological succession is the gradual process by which ecosystems change and develop over time. There are two main types of succession: primary and secondary. Primary succession is the series of changes which occur on an entirely new landscape which has never been colonized before. For example, a newly exposed rock face or sand dunes, or a newly formed lake.

Secondary succession is the series of changes which take place on a previously colonized, but disturbed or damaged habitat. For example, after felling trees in a woodland, land clearance or a fire. Also, the ecological succession can be classified as progressive or retrogressive. In a progressive succession, there is an increase in biomass, diversity, and structural complexity through time. In a retrogressive succession, the community becomes simplistic and there is a decrease in diversity and biomass over time. In this transformational process the ecosystem evolves, groups are created or destroyed (habitats), flows are modified and the system shapes itself through the process of self-organization.

In order to understand the evolution, distribution and diversity of species in an ecosystem, it is needed to know how individuals affect and are affected by the environment: what are the rates of growth and migration of a population, which factors influence its growth, what are the resources used by the species, and what are the intra and interspecies interactions.

Environmental factors can affect and be affected by the action of individuals. The change in one or more environmental factors can influence the behavior of a population. For instance, changes in temperature can influence the transition between growth and reproduction states of a given species. In this example, with favourable conditions to the reproduction, the increase in the number of individuals of a population may influence the change in temperature which, cyclically, can influence again in the state change of a species. Any condition beyond the limits of survival causes the death of individuals and can lead to the extinction of species.

The environmental factors promote the definition of ecological niches within habitats, since they influence in the life style of the species. For example, the temperature limits the growth and reproduction of several organisms, but different organisms respond to different limits of temperature. Hence, an ecological niche is defined by the action of  $n$  environmental factors. An ecological niche of a given species may be represented by a  $n$ -dimensional hyperspace of environmental factors.

Environmental factors may also influence the availability of resources needed for survival and/or development of species. All things (biotic and abiotic) consumed by an organism are considered resources for him. Some examples of resources are the solar radiation, water, individuals for mating, food, and even a suitable place to build its habitation. In each case, what is consumed becomes unavailable to other consumer and this leads the organisms to compete for resources. The more similar the ecological niches of two species are, the more intense is the competition between them. Thus, species that exploit the same resource in the habitat have their ecological niches superimposed.

In an ecosystem, the ecological relationships or biological interactions define the ways in which individuals interact. In this way, symbiosis can be defined by a relationship between two individuals where one individual directly affects the other individual (BEGON et al., 2006). This definition is controversial among researchers (DOUGLAS, 2010). Some believe symbiosis should only refer to persistent mutualisms, while others, including this author, believe it should apply to any types of persistent biological interactions. Also, using epistemology, symbiosis means “living together”. Hence, according to the type of dependency that individuals have with one another, these symbiotic relationships can be beneficial or malignant. If it is beneficial to one or both individuals, without detriment to the others, this relationship is called harmonic or positive. When there is injury or loss to one of its participants and advantage to the other, this type of relationship receives the name of inharmonious or negative. Both positive and negative relationships can occur between individuals of the same species (intraspecific relationships or homotypic) or between individuals of different species (interspecific relations or heterotypic). Examples of negative intraspecific relationships are cannibalism and competition. Examples of positive intraspecific relationships are the constitution of societies and colonies.

The positive interspecific relationships can be the mutualism, proto cooperation, inquilinism, and commensalism. In mutualism, both species take benefits and the association is mandatory for the survival of both species. In proto cooperation, although the two species involved be benefited, they can live independently. In inquilinism, only one of the participants is benefited, without causing any harm to the other. Finally, commensalism is a type of association between individuals where one of them takes advantage of the food that remains from the another. The living being that takes advantage of the food remaining is called commensal, while the living being that gives food is called host.

Examples of negative interspecific relationships are the competition, amensalism, predatism, parasitism, and slavery. In competition, species compete each other for some type of resource. In amensalism, individuals of a population secrete or expel substances that inhibit or prevent the development of individuals of other species. In predatism, the predator captures



and kills another living being, the prey, in order to feed itself. In parasitism, an individual is the parasite that lives in or on the body of another individual and removes food. Finally, in slavery, a living being takes advantage of the activities, the work, or products produced by other living being.

From intraspecific relationships arise population level behaviors through the interactions between their individuals. From interspecific relationships arise the ecological communities. An ecological community is a group of species that occur in the same habitat and relate to each other in some way.

Ecosystems are examples of Complex Adaptive Systems (CAS) in such a way that macroscopic properties such as trophic structure, interaction networks, and flow patterns of energy or matter can emerge from local interactions between the components of the system (BROWNLEE, 2007). All these emergent behavior may influence subsequent interactions. In CAS, complex structures and interaction patterns may arise through the use of simple rules. The main elements of a CAS are:

- Diversity maintenance procedures;
- Local interaction rules between the members of the system;
- Autonomous selection process that acts upon the components of the system and leads to continuous adaptation, self-organization, and the emergence of complex behaviors.

The basic CAS properties in an ecosystemic context are (LEVIN, 1998):

- Aggregation: it regards on how the individuals are grouped into populations, the populations into species, and the species into ecological habitats. With aggregation, the emergence of patterns and hierarchical organizations are natural consequences of the self-organization of the system;
- Non-linearity: it refers to the possibilities for the development of the system;
- Diversity: it regards on the heterogeneity present in the ecosystem both at macro (with the presence of several species) and micro levels (with the genetic variability internal to each species). The diversity maintenance is of utmost importance for providing soft evolutionary gradients and ensure the homeostatic state in front of environmental adversities such as temperature, humidity, soil pH, water pH, salinity, presence of pollutants, etc. Diversity also provides resiliency of a species against possible risks of extinction, for instance;

- Flow: it refers to the flow of nutrients, energy, materials, and information. These flows define the interconnections between the parts of the system and transform isolated species in an integrated system, an ecosystem where the biotic and abiotic correlates between themselves.

From individuals to ecological communities, passing through populations, all involved biological entities require or move matter for its organization, energy for its activities, and information for its communication. In this sense, the biological ecosystems are defined.

Some concepts and processes inherent to biological ecosystems have already been used to develop computational systems for optimization. Following are briefly presented some works that, in some way, are related with the subject of this thesis.

## 2.6 RELATED WORKS

(WANG et al., 2007) proposed a model inspired by natural ecosystems to optimize resource management in a grid of computers. The model considers hardware and software resources, management policies, various applications, quality of service (QoS) and the users of the grid. The harmony of the computational ecosystem (homeostatic state) is given through the automatic management of computational resources and is verified by the quality of service achieved (QoS) in grid applications. In this computational ecosystem, competition for the available resources does exist between users and the evolution is achieved by optimizing the management process and the resource allocation. This model proposes the use of knowledge discovery in databases (KDD) strategies to aid the self-organization process of the system. In this way, the discovered knowledge can be used to predict the resource requirements and, thus, optimize the allocation. However, in the cited work, no experiments were performed with the proposed model.

In both, (BRISCOE; de WILDE, 2008, 2009) a Digital Ecosystem was proposed to optimize the use of software services available in a distributed network. The model uses concepts of multi-agent systems, distributed evolutionary computation, and ecology. In this Digital Ecosystem a decentralized point-to-point network forms a web of distributed agents that feed evolutionary algorithms located at each point of the network, called habitat. Each habitat represents a network user on an access point. Habitats, in turn, connect dynamically to each other in accordance to migratory paths, forming a network of habitats. Differently from the island model in evolutionary computation, each connection between habitats has a probability associated with movement through the connection, affecting migration decisions.

These probabilities are updated according to the success rate of migrating agents. An agent represents a user service and includes a semantic description of the business process involved. The dynamics of the proposed digital ecosystem occurs as follows: the users of the system formulate requests in their respective habitats specifying a desired service or application, where each request represents an agent. A population is then instantiated in the user's habitat in response to open requests and is fed by available agents in the habitat. The population then begins a process of evolution of the agents, with their respective semantic descriptions, in order to meet the user's request. In this model, a process based on genetic algorithm performs a combinatorial search in the space of possibilities of available software services. Finally, once executed an agent coming from the evolutionary process, it migrates to other habitats in order to meet other requests.

A predator-prey ecosystemic model served as inspiration to optimize the problem of synthesis of textures in binary images (VULLI; AGARWAL, 2008). Given a binary texture, the goal is to find the optimal set of parameters of the Markov Random Field (MRF) capable of generating the input texture. The texture whose parameters should be found is mapped to the environment. The parameters to be optimized are mapped as evolutionary features of prey. In this way, each prey are born with a texture that camouflages itself in the environment. Every prey evolve at each iteration and have a life cycle in which they are born, move, reproduce and die. A prey that cannot be seen by the predator is said to be fully adapted to the environment. The predator species is only capable of identifying the prey and kill them. In this work a logistic function was used to regulate the population dynamics in order to maintain the balance between predator and prey species. The predator-prey adaptation emerge from the interactions of individuals between themselves and with the environment.

In the work of (MA, 2011) experiments were carried out considering the population dynamics of natural ecosystems to self-adjust the population size of a genetic algorithm. The logistic function was used to modify the size of the population during the evolution process. The problem addressed was an extremely simple toy problem that consists of identifying blocks of prefixed sizes with value 1 in a binary chromosome. The greater the number of blocks identified, the higher the fitness. The results showed improvement in around 50% using the non-linear function with respect to the use of a genetic algorithm with fixed size population.

As mentioned in Section 2.5, the maintenance of diversity at all ecological levels is critical to the development and evolution of an ecosystem. The work of (MEIRELLES et al., 2010) presents several metrics to quantify the diversity of populations and species. The main idea is to generate foundations to develop methods to control the loss of population diversity. However, in this work, ways to measure and maintain diversity in an ecosystemic context,

considering multiple populations, are not addressed.

In (SIMON, 2008) the Biogeography-based Optimization (BBO) was proposed (Section 2.2.4). Biogeography deals with the aspects of living beings associated with its geographic and spatial distribution. This subject seeks to explain why the biota are in the place that they are and how this relates to their evolutionary past and its conservation. Biogeography is, therefore, very close to the ecology of populations and communities, and evolutionary biology. In BBO, each individual is considered as a “habitat” with a habitat suitability index (HSI), which is similar to the fitness of EAs, to measure the individual. The model uses the concepts of how a species migrates from one island to another and how species arise and are extinguished within the islands.

The work of (PASTI et al., 2010) formalizes what the authors called biogeographic computing. In the formalization of biogeographic computing, micro and macro-evolutionary operators are defined and the main features are the generation and maintenance of genetic diversity and automatic adjustment of the number of species and individuals. To define the species and habitats, metrics of dissimilarity (distance) are used. They are: dissimilarity between individuals ( $D_{II}$ ); dissimilarity between individuals and species ( $D_{IE}$ ); and dissimilarity between species ( $D_{EE}$ ). The metric  $D_{IE}$  is used for determining which individuals belong to which species and, when  $D_{IE}$  is larger than a given speciation threshold  $\sigma_S$ , a new species is created. The metric  $D_{EE}$  is used to determine which species belong to which habitat, respecting a given threshold  $\sigma_C$ .

Using this formalization, the work presents an evolutionary algorithm with dynamic control of the population size, covering concepts of speciation and gene flow to optimize multimodal functions. The initial number of species is equal to one and the number of individuals is equal to two, located close to each other, forcing them to belong to the same species. The speciation threshold is set to  $\sigma_S = 0.25$ . Reproduction and mutation operators are used to increase the number of individuals in each generation and to generate genetic diversity, respectively. Gene flow occurs randomly between species so that an individual of each species is selected and then these are combined to generate a third individual. Both individuals generated by gene flow and those generated by reproduction follow the speciation criteria defined by  $D_{IE}$ . Natural selection occurs deterministically excluding 20% of individuals of each species, keeping always the best individual. The algorithm was applied to the optimization of a multimodal continuous function. In a nutshell, in this algorithm, a population of candidate solutions (initially small) evolves in order to define geographically dispersed species in the search space being optimized.

In all previous works, one can notice that concepts and processes present in biological ecosystems are used in some way. However, none of these computational systems take into consideration the whole biological ecosystem as inspiration. Hence, in this thesis an ecological framework for computational optimization is presented. Inspired by biological ecosystems, the ecological framework opens the possibility to develop new plausible biologically inspired optimization systems, and is presented next.

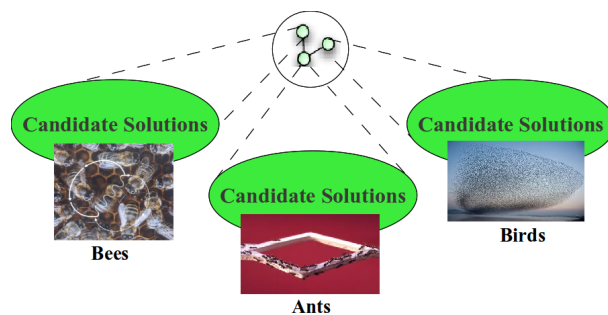


### 3 COMPUTATIONAL ECOSYSTEM FOR OPTIMIZATION

This chapter describes the organization of a computational ecosystem and presents some fundamental ecological concepts that can be explored in the context of problem solving (PARPINELLI; LOPES, 2012e). This section also illustrates the potentiality of some ecological concepts by presenting a canonical ecology-inspired algorithm for optimization.

#### 3.1 DESCRIPTION

A computational ecosystem for optimization is composed of candidate solutions (individuals) scattered in an environment that, itself, is the search space of the problem being solved. A given set of candidate solutions define a population of the ecosystem. The computational ecosystem can be composed of several populations that can interact to each other. Figure 2 shows a possible representation for the elements of the proposed computational ecosystem. This figure shows three populations where each population behaves according to the mechanisms of intensification and diversification, tuned by the control parameters, specific of an optimization strategy. In this example, the behavior of individuals is driven by the foraging strategies of bees, the foraging strategies of ants, and by the flocking behavior of birds, respectively.

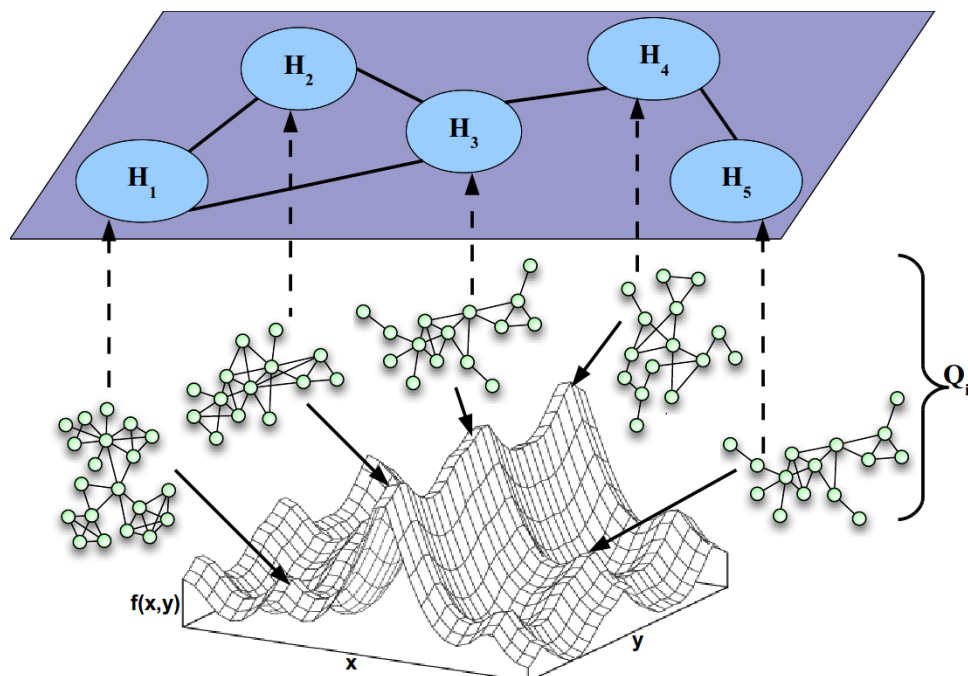


**Figure 2: Possible representation for the elements of a computational ecosystem. Three populations with different behaviors.**

Another representation is shown in Figure 3. The lower level of the figure illustrates an environment defined by a function  $f(\cdot)$  which describes a hyper-surface. In this example, the hyper-surface has two dimensions only for visualization purposes.

In the intermediate level of Figure 3, small circles represent populations  $Q_i$  with  $i = 1, \dots, NQ$ , where  $NQ$  is the total number of populations in the ecosystem. The  $NQ$  populations evolve and interact with each other and with the environment. Again, each population  $Q_i$  is composed of a set of candidate solutions, and behaves according to the rules of specific search strategies. The biodiversity of the ecosystem is represented by all biotic components, i.e., all individuals of all populations.

Once dispersed in the search space, populations of individuals established in the same region constitute an ecological habitat. Thus, a habitat is a group of populations that belongs to the same region in the search space. A hyper-surface may have several habitats  $H_j$  with  $j = 1, \dots, NH$ , where  $NH$  is the total number of habitats in the ecosystem. As well as in nature, the populations can move around through all the environment. Hence, the notation  $Q_i^j(t)$  means that population  $i$  belongs to the habitat  $j$  at time  $t$ . The ecosystem can be composed of several habitats that can also interact to each other, as shown in the upper level of Figure 3.



**Figure 3: Generical view of a computational ecosystem for optimization. Lower level: hyper-surface of the search space. Intermediate level: intra-habitats communication topologies where each small circle represents a population. Upper level: five habitats connected through inter-habitats communication topology.**



After defining the habitats, two categories of ecological communication topologies can be defined. Intra-habitats topology that occur between populations inside each habitat, and inter-habitats topology that occur between habitats (BEGON et al., 2006; MAY; MCLEAN, 2007).

Each habitat  $H_j(t)$ , with their respective populations, has an intra-habitat communication/interaction topological structure  $TC_j(t)$  that defines which populations, belonging to the habitat  $j$ , will be able to communicate/interact to each other at time  $t$ . The intermediate level of Figure 3 shows five intra-habitats communication topologies. Hence, populations located in the same habitat can interact among themselves according to their topology.

Besides the interconnection topology within each habitat, it is necessary to define the inter-habitats communication topology  $TH(t)$  at time  $t$ , as shown in the upper level of Figure 3.

Once defined the communication topologies  $TC_j(t)$  and  $TH(t)$ , flows of information, matter, and energy may arise within the ecosystem. With the appearance of flows it is necessary to define how populations can interact with each other, i.e., how each population will deal with the flow sent/received to/from other populations. The main forms of interaction are: gene flow, competition, cooperation (mutualism), amensalism, commensalism, parasitism and others (see Section 2.5).

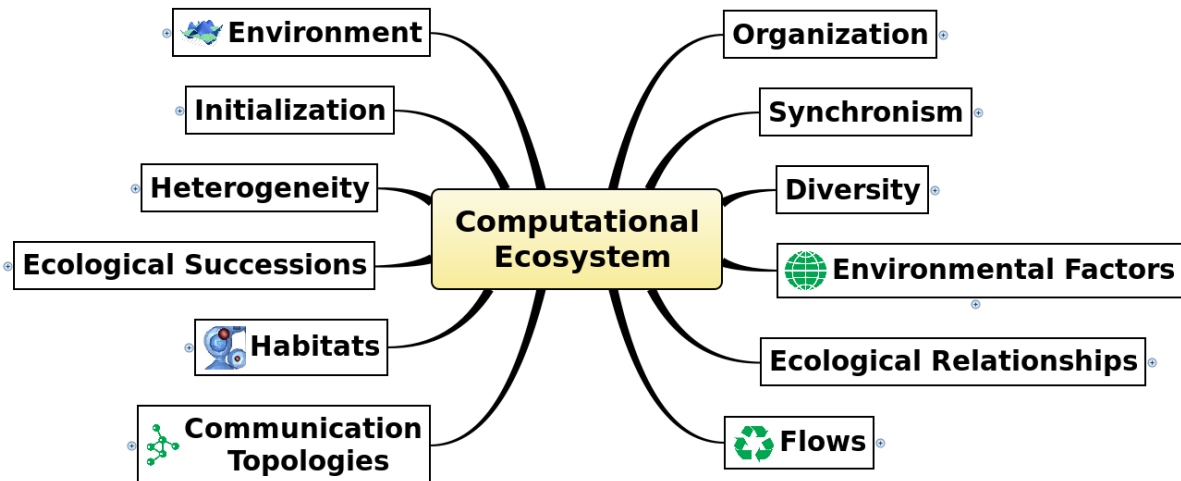
Within a computational ecosystem it must be defined which environmental factors may influence the development of the populations, e.g., temperature, humidity, and populational density. The environmental factors can influence the behavior of populations and the dynamics of the entire computational ecosystem, affecting diversity.

The maintenance of the diversity of populations and the diversity inside each population is fundamental for the evolution of a computational ecosystem. In addition to the mechanisms of intensification and diversification specific to each search strategy, when considering the ecological context, the computational ecosystem has a new level for information exchange between individuals. The intra-habitats relationships are responsible for intensifying the search and the inter-habitats relationships are responsible for diversifying the search. Thus, intra and inter-habitats relationships set a new level for diversity maintenance of the system.

Finally, the homeostatic state of a computational ecosystem can be seen as the stabilization of the ecological successions.

### 3.2 COMPUTATIONAL ISSUES

This section provides computational descriptions for the elements that can compound a computational ecosystem. Figure 4 depicts the elements that will be described.



**Figure 4: The elements of a computational ecosystem.**

#### 3.2.1 ENVIRONMENT DEFINITION

The environment is where the elements of a computational ecosystem will evolve. The landscape of the environment is defined by an objective function  $f(\cdot)$  and its constraints which represents a hyper-surface of the problem to be solved. The problem may have diverse characteristics: mono or multi-modal, constrained or unconstrained, continuous or discrete, static or dynamic, and others. The biotic components of a computational ecosystem are the candidate solutions that will co-evolve, and the abiotic components are the landscape itself and any other artifacts consumed or produced by populations (e.g., numerical information shared through local or global variables) or artifacts that limitate or influence the movement of populations through the environment (e.g., a search space decomposition and limits).

#### 3.2.2 SYSTEM INITIALIZATION

Once defined the environment of a computational ecosystem, the first issue to address is the random initialization of the candidate solutions for all populations. This can be done by using the same probability distribution for all populations or using different distributions for

each population. Some probability distributions widely used to sample random numbers are the Uniform (continuous and discrete domains), Normal (continuous domain), Binomial (discrete domain), and Poisson (discrete domain) distributions (OLOFSSON, 2005). Since in nature the populations are not equally scattered in an environment, a biased distribution, such as Normal or Poisson distributions, brings more biological plausibility to the computational ecosystem. For example, the initialization can use, for all populations, a normal distribution ( $N(\vec{\mu}, \sigma)$ ) with average  $\vec{\mu}$  randomly chosen within the domain of each dimension of the problem, and standard deviation  $\sigma$ , also randomly chosen inside the problem domain.

### 3.2.3 SYSTEM HETEROGENEITY

The computational ecosystem can be composed of several search strategies. Hence, it is necessary to define if the computational ecosystem is a homogeneous or a heterogeneous model concerning the strategies involved. In a homogeneous model all populations evolve in accordance to the same optimization strategy, configured with the same control parameters. Any change in strategies or parameters in at least one population characterizes a heterogeneous model. Recent literature has indicated that the use of heterogeneous search strategies working in a cooperative way can perform better than using single algorithms or homogeneous models (see, for instance, (BENÍTEZ et al., 2012), (MASEGOSA et al., 2008), (SUN et al., 2012), and (INTHACHOT; SUPRATID, 2007)). Also, a heterogeneous model is more biologically plausible than a homogeneous model, since, as in nature, populations can behave differently one each other.

A computational ecosystem can use any search strategy. However, something obvious to concern about the search strategies to employ is that they must be subject to the features of the problem being solved. For example, a canonical ACO algorithm is not suitable for continuous problems as well as a canonical PSO algorithm is not suitable for combinatorial problems. In other words, either canonical or not, all search strategies must be able to handle the problem features.

### 3.2.4 ECOLOGICAL SUCCESSION

The ecological succession is the directional change in the composition or structure of a community (a group of populations that occur in the same habitat) over time. The primary ecological succession starts with the initialization of the system and represents the first steps in search (colonization) of the environment. The primary ecological succession goes on until the system stabilization (convergence) or until the occurrence of a disturbance. In case of any

disturbance, the secondary ecological succession starts redirecting the search and regenerates the computational ecosystem. A disturbance can be performed, for example, by any routine of mass extinction or decimation in the community, or by the insertion of invasive/exotic species into a habitat (e.g., predators). Mass extinction is a commonly used strategy in population-based algorithms (see, for example, (KRINK; THOMSEN, 2001), and (LOPES; COELHO, 2005)). Progressive and retrogressive ecological successions (Section 2.5) can alternate accordingly to measures of biomass, diversity, and structural complexity. This alternation leads to an ecological level of balance between intensification and diversification, respectively.

Each ecological succession, either primary or secondary, or either progressive or retrogressive, is followed by the evolution of the biotic components of the system. For example, an ecological succession step can be characterized by an evolutive period performed by all populations, i.e., all populations evolve their solutions for a pre-established number of iterations.

### 3.2.5 DEFINITION OF HABITATS

Habitats are regions in the search space in which the populations are concentrated and this is one of the main concepts of a computational ecosystem. For example, in a multimodal hyper-surface, the surroundings of each peak can become a promising habitat for populations. Hence, in order to define the habitats, it is necessary first to identify the regions of reference of each population. A region of reference represents the point(s) or area(s) where a population is concentrated. A population can be concentrated in a single region or scattered in small groups in several regions. To establish the regions of reference of each population one can use the barycenter or centroid of the whole population or only the  $k$ -best individuals (with  $k = 1$ , the centroid is the position of the best individual), or any clustering algorithm (e.g., the nearest neighbor algorithm ( $k$ -NN),  $k$ -means, ISODATA, Jarvis-Patrick Clustering, or single-link/complete-link algorithm) to find different groups of individuals inside each population (JAIN et al., 1999).

Once found the region of reference of each population, the habitats can be defined. This can be done, for example, using a distance metric between regions of reference or using a clustering algorithm to find groups of regions of reference. It is worth to mention that the distance between populations is dependent on the definition and representation of each population. Examples of distance metrics that can be used are Euclidean distance, Manhattan distance, Hamming distance, and others (OLOFSSON, 2005). Section 4.4 describes a case study using the Euclidean distance as a metric to define the habitats.

For example, the use of a hierarchical clustering algorithm to define the habitats can be

considered. Hierarchical clustering refers to methods that produce a nested series of partitions (XU; WUNSCH, 2005). Single-link and complete-link algorithms are the most popular hierarchical clustering algorithms. These two algorithms differ in the way they characterize the similarity between a pair of clusters. In the single-link method, the distance between two clusters is the minimum of the distances between any two points (or patterns) in the different clusters. In the complete-link algorithm, the distance between two clusters is the maximum of all pairwise distances between any two points in the different clusters. In either case, two clusters are merged to form a larger cluster based on the minimum distance criteria. Section 4.7 describes a case study using the single-link algorithm.

A hierarchical algorithm yields a dendrogram representing the nested grouping of patterns and similarity levels at which grouping change (MURTAGH; CONTRERAS, 2012). Table 1 gives a distance matrix sample for five items (1 - 5). In the context of this thesis, each item represents the centroid of a given population and the distance matrix is computed using the Euclidean distance metric. The single-link algorithm uses the distance information from Table 1 and returns the linkage information needed to build a dendrogram (Table 2) in a matrix with three columns and  $NQ - 1$  rows, where  $NQ$  is the number of items (LEGENDRE; LEGENDRE, 1998).

In Table 2, each row identifies a node and represents a link between clusters. The first column identifies the nodes, and the two subsequent columns identify the clusters that have been linked. Negative items represent newly formed binary clusters. The fourth column contains the distance between these objects. The dendrogram of Figure 5 shows the series of merges that result from using the single-link technique. The height at which two clusters are merged in the dendrogram reflects the distance of the two clusters. The dendrogram can be broken into different levels to yield different clusterings of the data. For example, if a cut-off level at 3.0 is defined in the y-axis, three clusters are formed: one with items 1 and 2; other with items 4 and 5; and other with item 3.

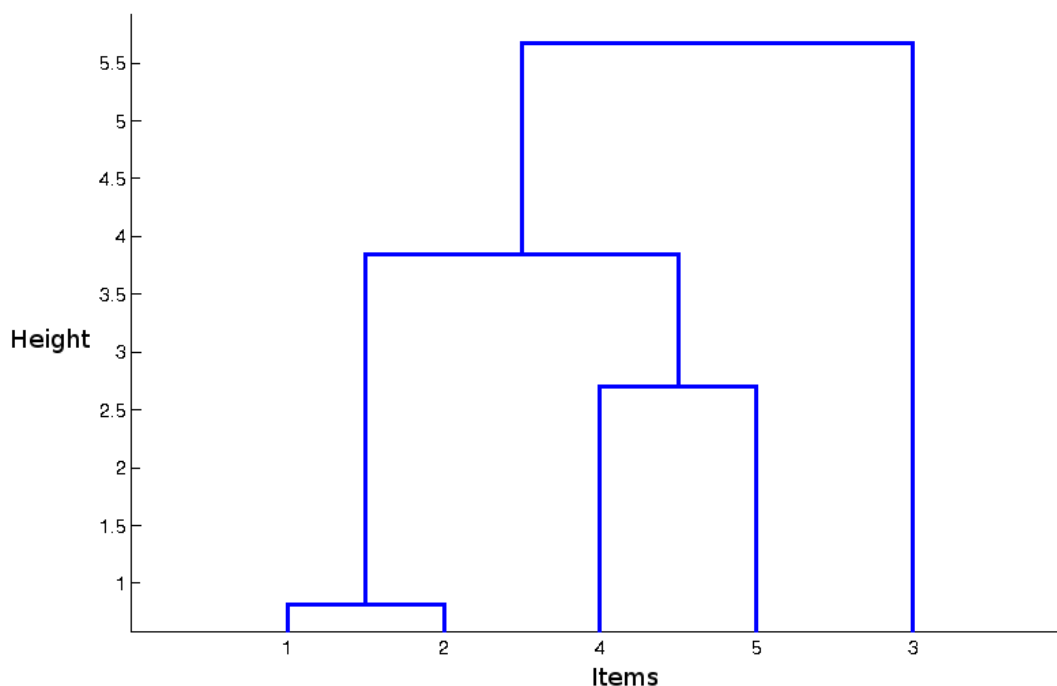
Items	1	2	3	4	5
1	0.0	0.5	4.3	3.8	4.8
2	0.5	0.0	4.7	3.3	4.4
3	4.3	4.7	0.0	6.2	6.6
4	3.8	3.3	6.2	0.0	1.1
5	4.8	4.4	6.6	1.1	0.0

**Table 1: Distance matrix for five items.**

A key concept of the proposed ecological system is the definition of habitats. With the use of a hierarchical clustering algorithm to setup the habitats, each cluster will represent

Node	Item <sub>left</sub>	Item <sub>right</sub>	Distance
1	1	2	0.8
2	4	5	2.7
3	-1	-2	3.8
4	3	-3	5.7

**Table 2: Single-link result for the data in Table 1.**



**Figure 5: Dendrogram generated using linkage information from Table 2.**

a habitat. Hence, the habitats are defined probabilistically taking into account the distance information returned by the single-link algorithm. This gives more biological plausibility to the system once, in nature, the habitats are not defined deterministically.

To create probabilistically the habitats it is possible to use the linkage information returned by the single-link algorithm (Table 2). The distance information can be used as probabilities to drive the formation of habitats in a top-down strategy (see Algorithm 6). It is a top-down strategy because it starts from the top of the dendrogram (farthest clusters) and goes down to the bottom of the dendrogram (closest clusters).

After some initializations, the first step of Algorithm 6 is to scale linearly the single-link distances in order to be able to work with this information as probabilities (line 6). The closed interval of  $[0.01, 0.99]$  was chosen in order to give one more biologically plausible feature to the system. Hence, concerning the lower bound, it means that as close as two populations are from each other, there is still 1% of chance of not grouping these two populations. There is

```

1:  $NH = 0$ ;
2:  $nodeCount = 0$ ;
3:  $curNode = NQ - 1$ ;
4:  $curHabitat = 0$ ;
5: Create  $H_{curHabitat}$  with no items;
6: Linearly scalonate the single-link distances;
7: while  $nodeCount < NQ - 1$  do
8:   if  $rand \geq Distance(curNode)$  then {Group items}
9:      $H_{curHabitat} = curNode.item_{left}$  and  $curNode.item_{right}$ ;
10:     $nodeCount = nodeCount + 1$ ;
11:   else {Separate items}
12:      $H_{curHabitat} = curNode.item_{nearest}$ ;
13:      $NH = NH + 1$ ;
14:     Create  $H_{NH}$  with no items;
15:      $H_{NH} = curNode.item_{farthest}$ ;
16:      $nodeCount = nodeCount + 1$ ;
17:   end if
18:   Update  $curHabitat$ ;
19:   Update  $curNode$ ;
20: end while
21: Return  $NH$ ;
22: Return  $H_j$  where  $j = 1, \dots, NH$ ;

```

**Algorithm 6:** Pseudo-code for probabilistic habitats formation.

a small chance to the closest populations not belong to the same habitat. Concerning the upper bound, it means that as far as two populations are from each other, there is still 1% of chance of grouping these two populations. There is a small chance to the farthest populations belong to the same habitat. Table 3 gives the linearly scaled values for the example of Table 2.

Node	Item <sub>left</sub>	Item <sub>right</sub>	Distance
1	1	2	0.01
2	4	5	0.39
3	-1	-2	0.62
4	3	-3	0.99

**Table 3: Linearly scaled values for distance.**

After that, the algorithm enters a loop until that all nodes are analysed (lines 7 to 20). The  $nodeCount$  variable counts the number of analysed nodes. Inside this loop a probabilistic conditional statement decides if the items will be grouped together or separated in two groups (line 8). Notice that the distance between items influence directly the probabilistic decision. The closer two items are from each other, the larger the chance to group these two items together. The opposite holds for the farthest items. If two items are decided to be grouped together, the current habitat ( $H_{curHabitat}$ ) receives the left and the right items from the node being analysed ( $curNode$ ) (lines 9 and 10). If two items are decided to be separated from each other, it is

necessary to decide which item stays and which item will belong to a new habitat. As a general rule, the closest item from the current group stays and the farthest item creates a new habitat (lines 12 to 16).

Next steps are to update the next habitat and the next node to be analysed. The *curHabitat* variable is updated to the absolute value of the first habitat with a negative item inside (newly formed binary clusters) (line 18). The *curNode* variable is updated to the absolute value of the first negative item inside  $H_{curHabitat}$  (line 19). Finally, the algorithm returns the number of habitats ( $NH$ ) and the habitats themselves ( $H_j$ ) (lines 21 and 22, respectively).

### 3.2.6 COMMUNICATION TOPOLOGIES

Two categories of communication topologies can be defined. Intra-habitats topologies that occur between populations inside each habitat and inter-habitats topologies that occur between habitats. Both topologies can be defined, for example, by using an adjacency matrix between the units (populations for intra-habitats topology and habitats for inter-habitats topology), probabilistically, or at random.

Consider, for example, that the habitats are already properly created. Hence, it is necessary to define the intra-habitats communication topologies for each habitat. This definition can be done deterministically using a proximity threshold or, aiming at improving the biological plausibility of the system, it can be probabilistically defined.

For a habitat with more than one population, intra-habitat communication can occur in such a way that each population inside the habitat chooses another population to perform communication. Here, the distance between populations influence directly the probabilistic decision. The closer two populations are from each other the higher is the chance of these two populations communicate. The opposite happens with farthest populations.

### 3.2.7 SYSTEM FLOWS

Information flows may occur, for example, by stigmergia (BONABEAU et al., 1999), where populations communicate indirectly through the environment. Flows of matter may occur by migration of individuals between populations or habitats. Energy flows can occur, for example, in the definition of the trophic structure between populations (Section 2.5). All informations can be recorded for further use in knowledge extraction procedures, as done in (CADENAS et al., 2009) (reviewed in Section 2.4.2). Machine learning strategies can be useful to self-tune and/or self-adapt the system. Also, flows can only occur if a communication



topology has been established between the units of the system.

### 3.2.8 ECOLOGICAL RELATIONSHIPS

The ecological relationships define the ways in which individuals and communities interact to each other (symbiotic relationships). Several ecological relationships can be found in a natural ecosystem (see Section 2.5). Once abstracted the biological essence of each ecological relationship, they are feasible to be modelled into a computational ecosystem.

For example, the cannibalism relationship (intra-species) can be modelled in such a way that the best fit individuals “eat” the worst fit individuals. The “eat” action can be defined by an energy value assigned to the best fit individual where the worst one is removed from the population. The same model can be used for the relationship of predatism (inter-species). The relationship of mating can also be modelled. Populations belonging to the same habitat can establish a reproductive link between their individuals, meshing the populations and favouring the coevolution and diversity of the involved populations. The mating relationship is a well known genetic operator in the Evolutionary Computation field in the form of crossover (GOLDBERG, 1989).

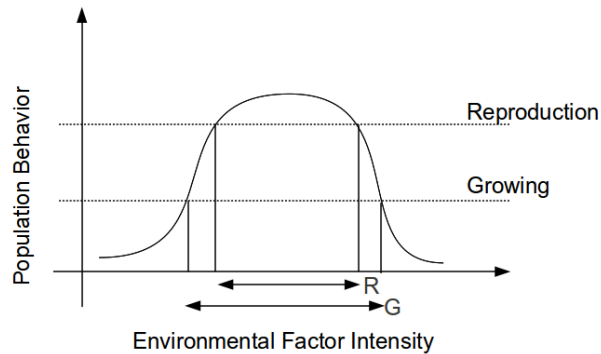
Another example is the migrations relationship widely used in the island model (KALEGARI; LOPES, 2010; PARPINELLI et al., 2011). In this relationship, individuals belonging to a given habitat migrate to other habitats aiming at identifying promising areas for survival and mating. In the relationship of amensalism, individuals of a population can “secrete” or “expel” numerical information in the environment that inhibit or prevent the development of individuals of other species, e.g., using artificial pheromone matrices (BONABEAU et al., 1999).

Ecological relationships can only occur if there are some kind of communications and flows.

### 3.2.9 ENVIRONMENTAL FACTORS

The behavior of populations can be biased by changes in environmental factors such as temperature and humidity. For instance, in Figure 6, a mathematical function represents the environmental factor of temperature. Changes in temperature can influence the transition between growth (G) and reproduction (R) states of a given species. The growth state can indicate that the population is increasing its mass through local search routines. The reproduction state represents the production of new individuals through asexual or sexual

reproduction. The change in the temperature may influence the number of individuals of a population which can influence in other transitions. Also, different populations can have different sensitivity to temperature and other environmental factors.



**Figure 6: A change in the environmental factor intensity influences the growth and reproduction of a population. Adapted from (BEGON et al., 2006).**

### 3.2.10 DIVERSITY MAINTENANCE

The biodiversity of a computational ecosystem is the result of the evolutionary process and the relationships between all its constituent elements. The diversity maintenance is necessary to the health of the system because it makes it more resilient to environmental adversities. Hence, it is desirable that diversity should be maintained during all successions of a computational ecosystem both at macro level, with different populations, and at micro level, with variability inside each population.

At micro level diversity, some Evolutionary Computation strategies can be used. For example, mutations, sharing, crowding, and genetic drift (GOLDBERG, 1989). The definition of environmental factors, the use of some ecological relationships such as mating, and others, can also enhance the micro level diversity. Concerning macro level diversity, some strategies can be the definition of mechanisms to control the complexity of intra-habitats and inter-habitats topologies (enhancing intra-habitats communication favours intensification and enhancing inter-habitats communication favours diversification). Also, the definition of mechanisms to control speciation and extinction of populations, the use of crowding and sharing in an ecological context, and the use of some ecological relationships such as migration, can be beneficial to macro level diversity.

### 3.2.11 SYSTEM SYNCHRONISM

Nature is asynchronous. Hence, a computational ecosystem is naturally well suited to asynchronous and parallel processing environments. However, it can also be simulated using serial and synchronous methods. For example, due to the optimization process inherent to each population, the distribution of its individuals can change with time. Consequently, the definition of habitats, and the consequent determination of which populations may be related to each other, becomes a dynamic process which can be updated continuously or periodically.

### 3.2.12 SYSTEM ORGANIZATION

In a computational ecosystem, the emergence of patterns and hierarchical structures are natural consequences of the self-organization of the system. However, a coordinator may be necessary to control the behavior of some routines, such as the influence of environment factors over each population or in the definition of ecological relationships. Here, meta-learning strategies can be useful to the system once all informations are available, such as the complexity and entropy of the system, and the established flows (information, matter, and energy).

Also, non-linear models can be applied to the system in order to regulate some variables. Non-linear models can be applied to adjust the number of individuals and the number of populations and to self-adjust the ecological succession interchange between progressive and retrogressive states. For example, a more biologically plausible survival selection mechanism can be achieved by the use of population dynamics where the logistic model can be applied to control the size of populations. The logistic map is often cited as an example of how chaotic behaviour can arise from very simple non-linear dynamical equations. Hence, it can be used as a discrete-time demographic model (KAPLAN; GLASS, 1995). Equation 1 presents the logistic map, where  $0 < a < 4$  is the logistic map parameter (also known as Lyapunov coefficient) and  $POP_i(t)$  represents the number of individuals in the population  $i$  at moment  $t$ . According to the initial conditions of  $POP_i$  at  $t = 0$  and the adjustment of the Lyapunov coefficient plenty of behaviors can appear ranging from periodic to chaotic oscillations.

$$POP_i(t+1) = a \times POP_i(t) \times (1 - POP_i(t)) \quad (1)$$

## 3.3 A CANONICAL ECO-INSPIRED ALGORITHM

In (PARPINELLI; LOPES, 2011a) a canonical ecology-inspired algorithm for optimization was first presented. In this application, called ECO, some ecological features

are used in order to bring a greater biological plausibility to an optimization tool and to insert these ecological concepts into the problem solving field. The ecological concepts addressed are the definition of habitats, two ecological relationships (mating and migration), and ecological successions. Along with these concepts, the definition of intra and inter-habitats communication topologies are inserted to compose the algorithm.

In ECO, each population behaves according to the intensification and diversification mechanisms and the control parameters specific to the Artificial Bee Colony Optimization (ABC) algorithm (KARABOGA; AKAY, 2009), in a homogeneous system. Hence, all populations use the ABC algorithm with the same control parameters to evolve their candidate solutions. Also, the ECO algorithm is synchronous concerning the evolution of the system.

Algorithm 7 shows the pseudo-code of the approach. First of all, at the instant  $t = 0$ , all individuals of all populations  $Q_i$ , with  $i = 1, \dots, NQ$ , are randomly initialized using a Normal distribution (line 2).  $NQ$  is the total number of populations.

The main loop represents the ecological successions (lines 3 to 12). The first step inside the main loop is the evolutive period for each population  $Q_i(t)$  (line 4). In this step each population explores the search area freely in accordance to its intensification and diversification criteria.

At the end of the evolutive period of all populations it is necessary to identify the region of reference for each population (line 5). The region of reference is used to define the habitats of the system. The metric chosen to define the region of reference is the centroid  $\vec{C}_i$ , with  $i = 1, \dots, NQ$ , calculated by Equation 2, where  $POP$  represents the number of individuals in the population and  $\vec{x}_k$  represents an individual (solution vector).

$$\vec{C} = \frac{\sum_{k=1}^{POP} \vec{x}_k}{POP} \quad (2)$$

Once found the centroids, the distances between populations are calculated and the habitats  $H_j(t)$  are defined using a minimum distance threshold  $\rho \in [0..1]$  (line 6). The distance between populations is dependent on the definition and representation of each population. In this application the Euclidean distance was used. At the end of this step all populations are associated with a habitat:  $Q_i^j(t)$ , population  $i$  belongs to the habitat  $j$  at the moment  $t$ .

With the  $NH$  habitats defined, it is necessary to determine the intra-habitats communication topologies  $CT_j(t)$  for each habitat  $H_j(t)$ , with  $j = 1, \dots, NH$  (line 7). The topologies are defined by an adjacency matrix between the populations belonging to each habitat. Once defined the intra-habitats topologies, the mating ecologic relationship between

- 1: Let  $i = 1, \dots, NQ$ ,  $j = 1, \dots, NH$  and  $t = 0$ ;
- 2: Initialize each population  $Q_i(t)$  with  $POP_i$  random candidate solutions;
- 3: **while** stop criteria not satisfied **do** {Ecological succession cycles}
- 4:   Perform evolutive period for each population  $Q_i(t)$ ;
- 5:   Identify the region of reference  $\vec{C}_i$  for each population  $Q_i(t)$ ;
- 6:   Using the  $\vec{C}_i$  values, define the  $NH$  habitats;
- 7:   For each habitat  $H_j(t)$  define the communication topology  $CT_j(t)$  between populations  $Q_i^j(t)$ ;
- 8:   For each topology  $CT_j(t)$ , perform interactions between populations  $Q_i^j(t)$ ;
- 9:   Define communication topology  $TH(t)$  between  $H_j(t)$  habitats;
- 10:   Perform interactions between  $H_j(t)$  habitats according to  $TH(t)$ ;
- 11:   Increase  $t$ ;
- 12: **end while**

**Algorithm 7:** Pseudo-code for ECO

adjacent populations occurs (line 8). In this relationship, one individual of each population is chosen, using the tournament strategy (BLICKLE, 2000), and a genetic exchange between two individuals is performed in order to generate a new individual. The new individual replaces an individual selected at random in its population, excluding the best one. This can be seen as a relationship of cannibalism.

Once the interactions between the populations of each habitat have been done, the  $TH(t)$  topology for interaction between habitats (line 9) is randomly defined. This inter-habitats topology  $TH(t)$  is used for the completion of the migrations ecologic relationship. In this relationship, for each habitat a random population belonging to it is chosen. The best individual of this population migrates to another habitat and, in the destination habitat, it replaces an individual randomly chosen, excluding the best one (line 10). This can be seen as a relationship of predatism.

The main loop continues until the ecological succession cycles reach the maximum predefined value.

The parameters of the canonical ECO algorithm are: number of populations ( $NQ$ ) that will be co-evolved, the initial population size ( $POP$ ), number of cycles for ecological successions ( $ECO-STEP$ ), the size of the evolutive period ( $EVO-STEP$ ) that represents the number of function evaluations in each  $ECO-STEP$ , the tournament size ( $T-SIZE$ ) used to choose solutions to perform intra and inter-habitat communications and the proximity threshold  $\rho$  used to define the habitats.

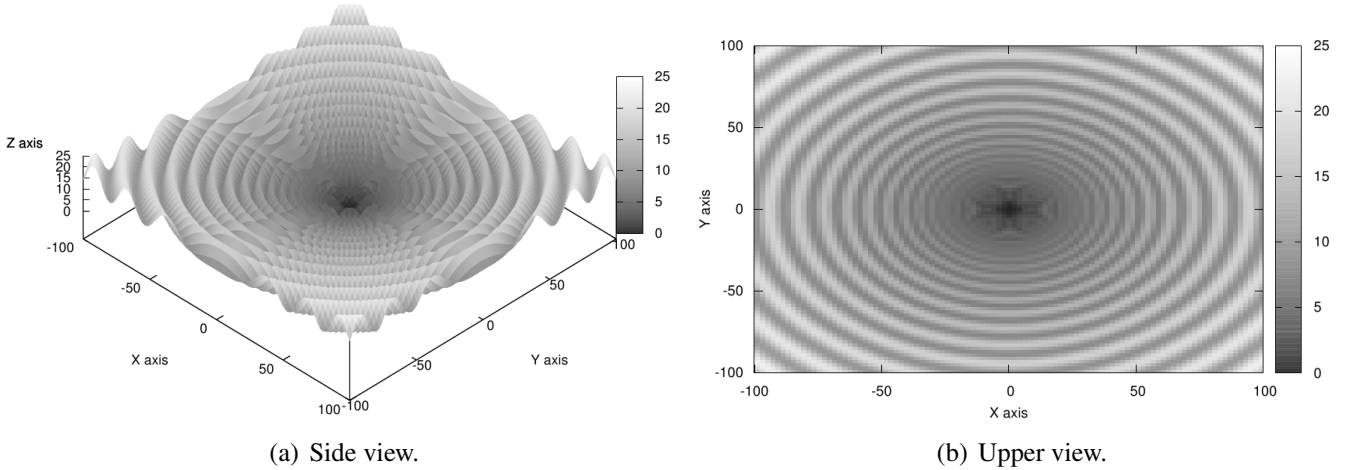
Following, a conceptual illustration of the canonical ECO algorithm is shown.

### 3.3.1 CONCEPTUAL ILLUSTRATION

The generalized Schaffer function was chosen to illustrate in details the behavior of the proposed algorithm (FLOUDAS; PARDALOS, 1990). Its definition is presented in Equation 3 where  $\vec{x}$  is a solution vector defined inside  $[-100, 100]$  and the global optimum for  $Schaffer(\vec{x})$  is 0, corresponding to the optimum solution  $\vec{x}_{opt} = (x_1, x_2, \dots, x_n) = (0, 0, \dots, 0)$ .

$$Schaffer(\vec{x}) = \sum_{i=1}^{n-1} \left( 0.5 + \frac{\sin^2 \left( \sqrt{x_{i+1}^2 + x_i^2} \right) - 0.5}{(0.001 (x_{i+1}^2 + x_i^2) + 1)^2} \right) \quad (3)$$

With the purpose of allowing a visual assessment of the behavior of the algorithm, the Schaffer function was defined here with only two dimensions. The surface plot for this function are shown in Figures 7(a) and 7(b).

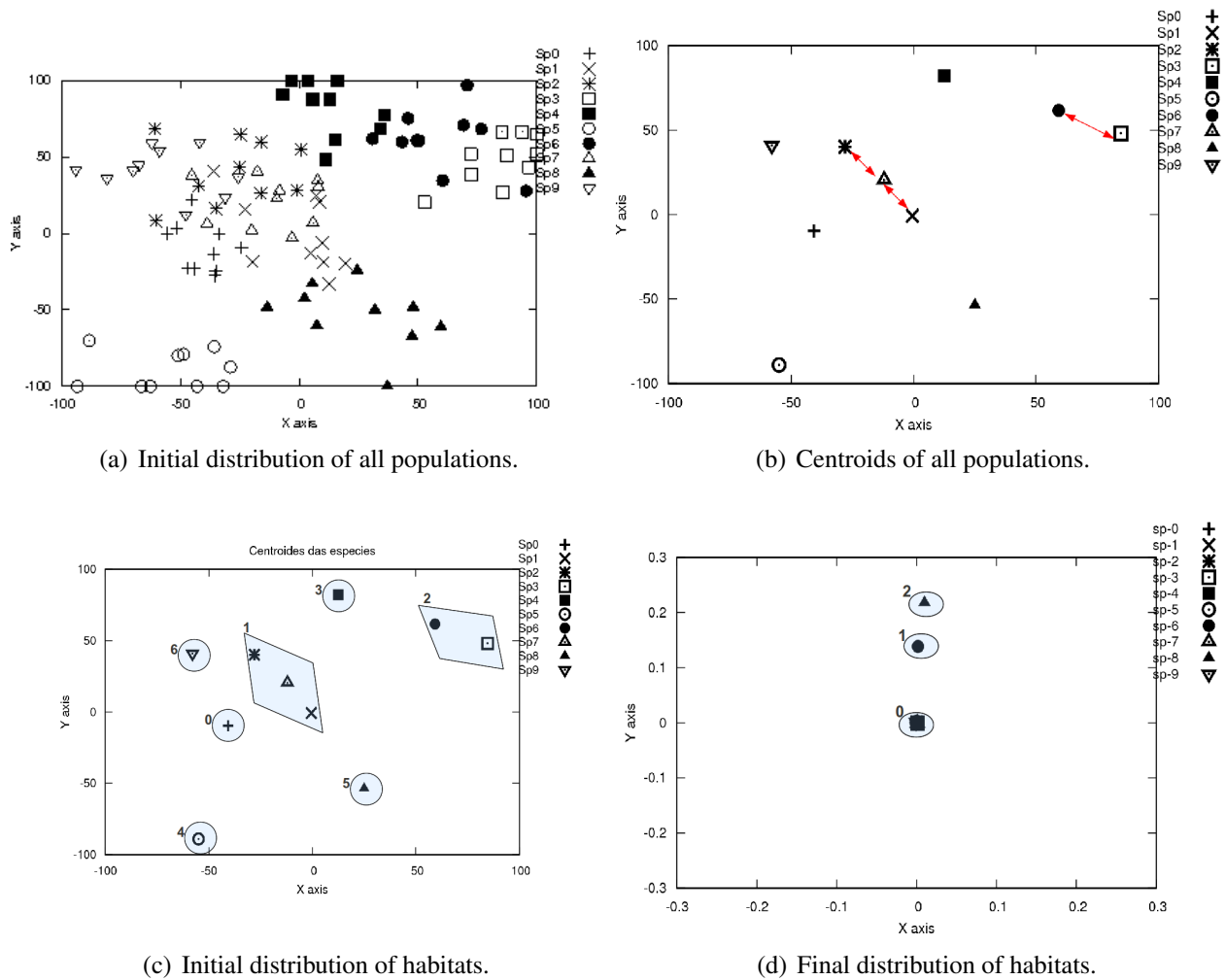


**Figure 7: 2D Schaffer function.**

The parameters used were:  $NQ = 10$ ,  $POP = 10$ ,  $ECO-STEP = 100$ ,  $EVO-STEP = 100$ ,  $T-SIZE = 5$  e  $\rho = 0.5$ . With this configuration, the total number of function evaluations is 10,000 for each population. A total of 100 evaluations for each population was done in each ecological succession. The parameters were chosen empirically.

In this illustration the Artificial Bee Colony Optimization (ABC) algorithm (KARABOGA; AKAY, 2009) was used in a homogeneous manner, i.e. all populations use this algorithm with the same adjustment of parameters to evolve their candidate solutions. For the ABC algorithm, besides the population size ( $POP$ ) parameter, another parameter is the  $limit = 100$  (Section 2.2.1).

First of all, all populations are randomly initiated. The initialization uses a normal distribution ( $N(\vec{\mu}, \sigma)$ ) with average  $\vec{\mu}$  (randomly chosen within the domain of each dimension



**Figure 8: Conceptual illustration of the proposed algorithm.**

of the problem) and standard deviation  $\sigma$  (also proportional to the problem domain).

Figure 8(a) shows the initial distribution of the individuals of all  $NQ$  populations:  $Sp0, \dots, Sp9$ .

Once initialized the populations, the algorithm enters the ecological succession loop that begins with the evolutive period. In this stage, all populations evolve their solutions for a pre-established number of function evaluations. In this example, the evolutive period was defined as 100 function evaluations.

After the evolutive period, begins the definition of habitats. The habitats are the regions in which the populations are concentrated. In this proposal, the region of reference of a population is defined by its centroid and it is calculated by Equation 2. Figure 8(b) illustrates the distribution of all centroids of all populations for the first ecological succession.

Once found the centroids for each population, the Euclidean distance between them is calculated and the habitats are defined in accordance to the minimum threshold  $\rho$ . The

adjacency matrix is generated according to  $\rho$ . Two populations are called adjacent among themselves if they are at a distance of at least  $\rho$  one of another. In Figure 8(b) all information of adjacency between centroids are shown in arrows:

- Sp0 is not adjacent to any population;
- Sp1 is adjacent to Sp7;
- Sp2 is adjacent to Sp7;
- Sp3 is adjacent to Sp6;
- Sp4 is not adjacent to any population;
- Sp5 is not adjacent to any population;
- Sp6 is adjacent to Sp3;
- Sp7 is adjacent to Sp1 and Sp2;
- Sp8 is not adjacent to any population; and
- Sp9 is not adjacent to any population.

The habitats are generated using the adjacency matrix. Figure 8(c) illustrates the habitats found for the centroids distribution of Figure 8(b), with  $\rho = 0.5$ . It is observed in this figure the existence of seven habitats:

- $H_0$  composed of Sp0;
- $H_1$  composed of Sp1, Sp2 and Sp7;
- $H_2$  composed of Sp3 and Sp6;
- $H_3$  composed of Sp4;
- $H_4$  composed of Sp5;
- $H_5$  composed of Sp8; and
- $H_6$  composed of Sp9.



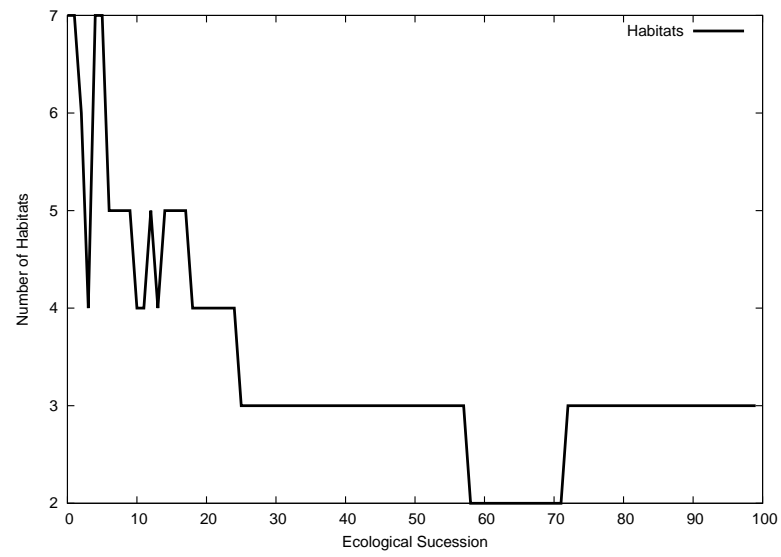
The adjacency matrix also defines the intra-habitat communication topology. For example, in habitat  $H_1$ , composed by populations Sp1, Sp2 and Sp7, the population Sp1 can establish a relationship with the population Sp7, the population Sp2 can establish a relationship with the population Sp7, and the population Sp7 can establish a relationship with populations Sp1 and Sp2. At this moment the habitats are well defined with their populations and communication topologies.

The next step is the communication between populations within each habitat. Populations that are adjacent between themselves interact by the mating ecologic relationship. Consider the habitat  $H_1$ . The population Sp7 selects an individual of its population and an adjacent population to establish a relationship. The adjacent population is chosen at random and, in this case, could be the population Sp1 or Sp2. In each population, the individuals chosen to carry out mating are selected by using the tournament selection of size 5. The new generated individual replaces an individual randomly chosen within the adjacent population. The populations Sp1 and Sp2 perform the same procedure. All habitats composed of more than one population carry out mating according to the topology defined by the adjacency matrix.

After the intra-habitats interactions it is necessary to define the inter-habitats communication topology. This topology is used to perform the great migrations ecologic relationship. In this relationship, for each habitat a random population is chosen at random. The best individual of the population chosen migrates to another random habitat and, in the destination habitat, it replaces an individual chosen at random, excluding the best individual. In this stage of the algorithm the ecological succession loop restarts. Figure 8(d) illustrates the habitats found in the last ecological succession step. They are:  $H_0$  composed of Sp0, Sp1, Sp2, Sp3, Sp4, Sp5, Sp7 and Sp9;  $H_1$  composed of Sp6; and  $H_2$  composed of Sp8. It is also possible to observe in Figure 8(d) that populations belonging to the habitat  $H_0$  converged towards the global optimum and other habitats have converged to regions close to the global optimum. Figure 9 shows the evolution of the number of habitats for each ecological succession step. It is observed that the system has evolved and converged to the formation of three habitats.

Once made the conceptual illustration, it is possible to highlight which features are used in the canonical eco-inspired algorithm in front of the whole ecological framework presented in Section 3.2. Figure 10 shows an extended illustrative map with all definitions that an computational ecosystem for optimization can have. Highlighted are the features used in the canonical eco-inspired algorithm.

Also, one can notice that there are plenty of other features that can be explored in the proposed ecological framework. In this way, next chapter presents some experiments in which



**Figure 9: Number of habitats at each ecological succession step.**

other features are explored inside the ecological framework.

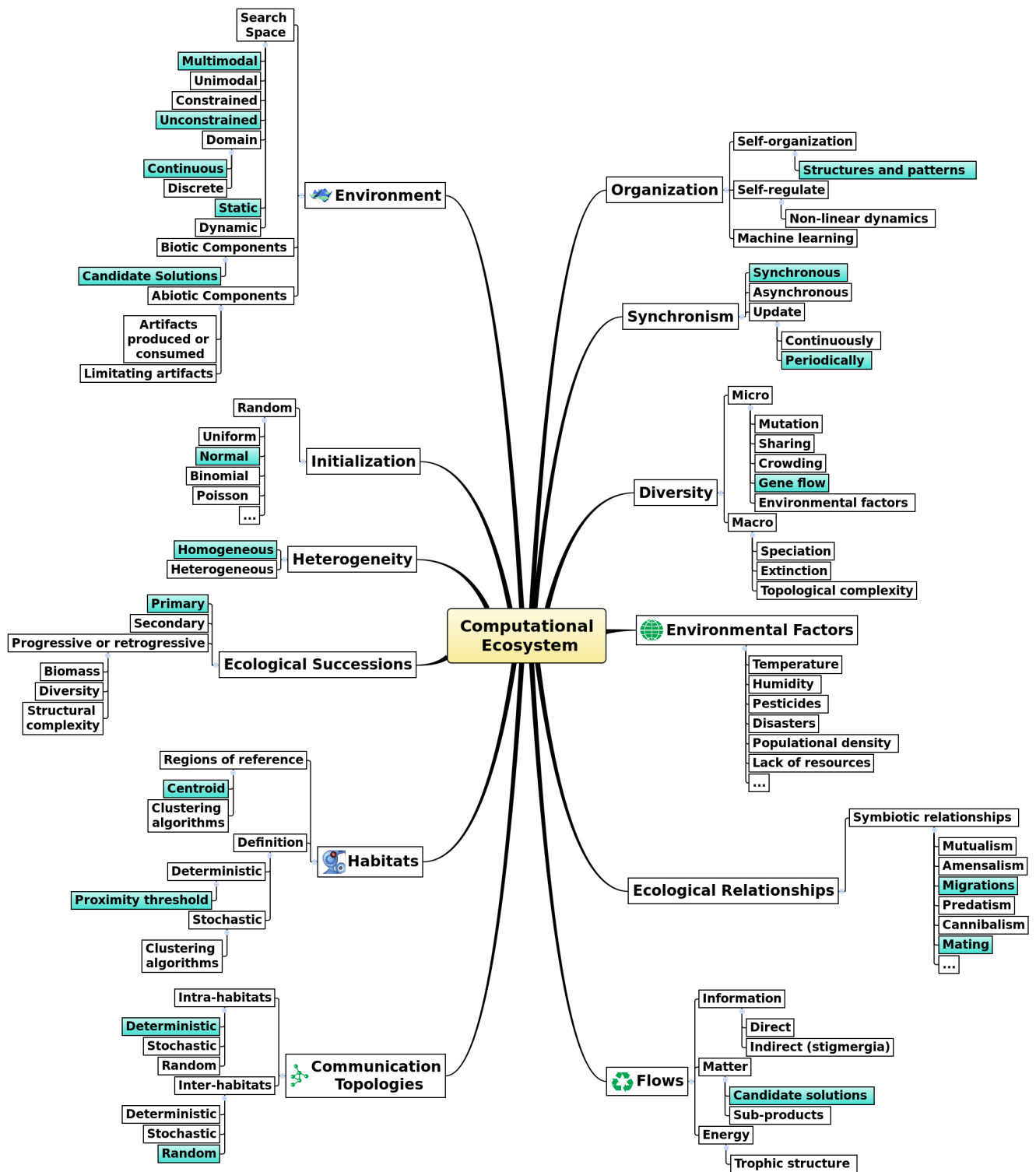


Figure 10: Extended illustrative map for the elements of a computational ecosystem. Highlighted are the features used in the canonical eco-inspired algorithm.



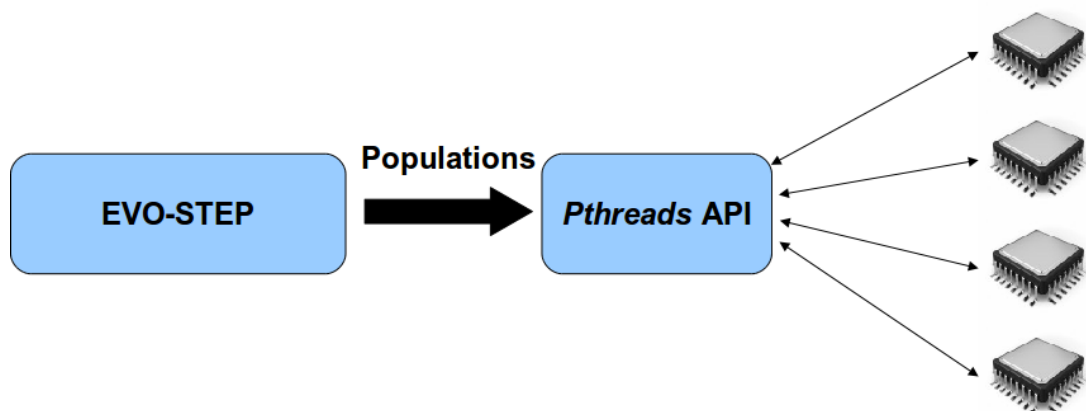
## 4 EXPERIMENTS, RESULTS, AND ANALYSIS

In this section a series of experiments that explores both the potentiality of the proposed approach and the insertion of new features is performed.

Owing to the stochastic nature of the proposed approach and other meta-heuristic algorithms, their performance cannot be evaluated by the result of a single run. Many trials with independent population initializations should be made to obtain an useful conclusion. Therefore, in this study the results are obtained in 30 trials.

All experiments reported were run in a computer with an Intel processor (Core2-Quad running at 2.8GHz) running Linux. The applications were developed using the C programming language.

Also, the ecology-based approaches were developed using a parallel strategy to take advantage of the computational power available. The POSIX Threads Programming standards<sup>5</sup> were applied through the *Pthreads* API. Figure 11 shows how parallelism is achieved. At each evolutionary period (EVO-STEP) each population triggers a different thread and the *Pthreads* API schedules the populations to the processors.



**Figure 11:** *Pthreads* API schedules the populations to the processors.

<sup>5</sup>Website: <https://computing.llnl.gov/tutorials/pthreads/> Visited in 2013.

In the following, the problems approached are described (i.e., mathematical benchmark functions and the protein folding problem), the setup of parameters, and some case studies applying the proposed eco-inspired approach.

#### 4.1 BENCHMARK FUNCTIONS

Some experiments were conducted using mathematical functions extensively used in the literature for testing optimization methods (DIGALAKIS; MARGARITIS, 2002)(KARABOGA; AKAY, 2009)(PIOTROWSKI et al., 2012). For all functions the goal is to minimize the objective function. Table 4 summarizes the information with respect to the used functions. As convention, when the global optimum of a function is  $f(\vec{x}) = 0$ , an evaluation of a candidate solution is considered to be zero if it is less than or equal to  $10^{-20}$ .

**Table 4: Mathematical Functions**

Function	Definition	Domain	Global Optimum
<i>Schaffer</i>	$\sum_{i=1}^{n-1} \left( 0.5 + \frac{\sin^2(\sqrt{x_{i+1}^2 + x_i^2}) - 0.5}{(0.001(x_{i+1}^2 + x_i^2) + 1)^2} \right)$	$-100 \leq x_i \leq 100$	$f_1(\vec{0}) = 0$
<i>Rastrigin</i>	$\sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i) + 10)$	$-5.12 \leq x_i \leq 5.12$	$f_2(\vec{0}) = 0$
<i>Griewank</i>	$\frac{1}{4000} (\sum_{i=1}^n x_i^2) - \left( \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) \right) + 1$	$-600 \leq x_i \leq 600$	$f_3(\vec{0}) = 0$
<i>Rosenbrock</i>	$\sum_{i=1}^{n-1} (100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2)$	$-30 \leq x_i \leq 30$	$f_4(\vec{1}) = 0$

The first function is known as generalized F6 Schaffer function that is strongly multimodal (FLOUDAS; PARDALOS, 1990). The second function is the Rastrigin function and it is also multimodal and based on the sphere function with the addition of a cosinoid modulation to produce a large number of local minima (MÜHLENBEIN et al., 1991). The third function is the Griewank function that, as the other, is strongly multimodal (GRIEWANK, 1981). Particularly the number of local minima for this function increases exponentially with the dimensionality, which makes it a very difficult optimization problem (CHO et al., 2008). The fourth function is the Rosenbrock function, which has the particularity of having a search hyper-surface in the form of a long and narrow parabolic valley, where the global optimum is located (ROSENBROCK, 1960).

#### 4.2 PROTEIN STRUCTURE PREDICTION PROBLEM

Proteins are the basic structures of all living beings (HUNTER, 1993). They are composed of a chain of amino acids that are linked together by means of peptide bonds. Each amino acid is characterized by a central carbon atom (also called as alpha carbon –  $C\alpha$ ) to

which are attached a hydrogen atom, a carboxyl group (COOH), an amino group (NH<sub>2</sub>), and a side-chain. It is known that the side-chain defines the physical and chemical properties of the amino acid (COOPER, 2000). In turn, peptide bonds are formed from the condensation of two amino acids, when the carboxyl group of an amino acid reacts with the amino group of the other. This process is also called as dehydration because it releases a molecule of water.

Several amino acids exist in nature, but only 20 are proteinogenic. They can be classified into two classes, according to their affinity to water: Hydrophilic (or Polar) and Hydrophobic. According to this behavior, one can conclude that the hydrophobicity of the side chain is one of the main processes that governs the process of forming protein structures (LODISH et al., 2000).

From the chemical point of view, proteins are structurally complex and functionally sophisticated molecules (ALBERTS et al., 2002). The structural organization of proteins is commonly described into four levels of complexity: primary, secondary, tertiary and quaternary structures. It is important to know that the upper levels cover the properties of lower ones. The primary structure refers to the linear sequence of amino acids, the secondary represents local conformations of some part of a three-dimensional structure. The tertiary structure represents the conformation of the whole polypeptide chain, i.e. the three-dimensional arrangement of the amino acids. Finally, regular associations of three-dimensional structures constitute quaternary structures.

*Protein folding* is the process by which a polypeptide chain is transformed into a compact structure that performs some biological function. These functions include control and regulation of essential chemical processes for the living organisms. Under physiological conditions, the most stable three-dimensional structure is called the native conformation and actually allows a protein to perform its function.

Failure to fold into the intended three-dimensional conformation usually leads to proteins with different properties that simply become inactive. In the worst case, such misfolded (incorrectly folded) proteins can be harmful to the organism. For instance, several diseases such as Alzheimer's disease, cystic fibrosis and some types of cancer, are believed to result from the accumulation of misfolded proteins.

It is known that better understanding the protein folding process can result in important medical advancements and development of new drugs. Thanks to the several genome sequencing designs being conducted in the world, a large number of new proteins have been discovered. However, only a small amount of such proteins have its 3-dimensional structure known. For instance, in October/2012, the UniProtKB/TrEMBL repository of protein sequences

has currently more than 27 million records<sup>6</sup>, and the Protein Data Bank – PDB (BERMAN et al., 2000) has the structure of only 86,172 proteins<sup>7</sup>. This fact is due to the cost and difficulty in unveiling the structure of proteins.

Computer science has an important role here, proposing models for studying the Protein Structure Prediction (PSP) problem (LOPES, 2008). Nowadays, the simulation of computational models that take into account all the atoms of a protein is frequently unfeasible, even with the most powerful computational resources. Consequently, several simplified models that abstract the protein structure have been proposed.

Basically, there are two types of representation of polypeptides, the analytical and the discrete. The analytical representation describes all the information about the atoms that compose the proteins. On the other hand, the discrete representation describes a protein in a reduced level of details. Although such discrete models are not realistic, they use some biochemical properties of amino acids, and its simulation can show some interesting characteristics of real proteins. They also allow an extensively exploration of the conformational space and can be generators of hypotheses that cannot be obtained by other approaches, but that may be reproducible experimentally or through refined simulations (DILL, 1999). This is an important motivation for developing computational methods for predicting the structure of proteins. The simplest computational model for the PSP problem is known as Hydrophobic-Polar (HP) model, both in two (2D-HP) and three (3D-HP) dimensions (DILL et al., 1995).

From the chemical point of view, the most realistic method is called molecular dynamics (HARDIN et al., 2002). The main idea of this approach is to simulate atom movements according to the rules of classical mechanics. On the other hand, computational approaches for searching a solution for the PSP using its simplest model (the HP model) were proved to be *NP*-complete (ATKINS; HART, 1999; BERGER; LEIGHTON, 1998; CRESCENZI et al., 1998). Therefore, this fact has motivated the development of several meta-heuristics to deal with the problem. Many other PSP models of abstraction there exists. Some of them are the the three-dimensional HP Side-Chain model (3DHP-SC) (BENÍTEZ; LOPES, 2010), and the 2D and 3D-AB off-lattice models (STILLINGER; HEAD-GORDON, 1995; KALEGARI; LOPES, 2010).

---

<sup>6</sup>See <http://www.ebi.ac.uk/uniprot/> for updated information

<sup>7</sup>See <http://www.pdb.org> for updated information

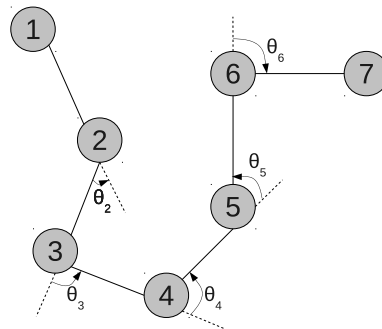


#### 4.2.1 THE AB OFF-LATTICE MODEL

The AB off-lattice model was one of the first non-lattice model to represent protein structures. In this model the protein sequences are composed of only two species of monomers ( $\xi$ ): ‘A’ for hydrophobic amino acids and ‘B’ for hydrophilic (or polar) amino acids. Although it is a very simplified representation of a real protein structure, this model is useful to verify some of the properties of proteins in the real world.

Monomers have an unit length of distance between them, in such a way that a monomer is connected to the next one in the chain through a bond that forms an angle relative to its predecessor.

In the AB model, a protein composed of  $N$ -monomers needs  $N - 2$  angles to be represented. These angles are defined in the range  $[-\pi, \pi]$ . Figure 12 shows an example of a hypothetical protein with seven amino acids.



**Figure 12: Generic representation of a hypothetical protein structure.**

The model defines the energy values for the monomers: ‘A’ has energy 1 and ‘B’ has energy -1. Considering two generic monomers  $i$  and  $j$ , and the types  $\xi_i$  and  $\xi_j$ , respectively, the interaction between the monomers leads to different values of potential energy ( $C$ ). Positive values represent attraction and negative, repulsion: AA bonds have energy 1 (the monomers AA tend to attract each other strongly), BB bonds have energy 1/2 (they tend to attract each other weakly) and AB or BA bonds have energy -1/2 (they have a weak repulsion). The energy  $E$  of the structure of a protein with  $n$  monomers ( $n$ -mers) is given by Equation 4:

$$E(\vec{\theta}, \vec{\xi}) = \sum_{i=1}^{n-1} V_1(\theta_i) + \sum_{i=1}^{n-2} \sum_{j=i+2}^n V_2(d_{ij}, \xi_i, \xi_j) \quad (4)$$

Equation 4 postulates two types of intermolecular potential energies, terms  $V_1$  and  $V_2$ . The former represents the backbone potentials. It is defined by Equation 5 and depends only on the angle between monomers. The latter, defined by Equation 6, represents the potential energy present in the non-bonded interactions and it is known as the Lennard-Jones potential.

$$V_1(\theta_i) = \frac{1}{4} \times (1 - \cos(\theta_i)) \quad (5)$$

$$V_2(d_{ij}, \xi_i, \xi_j) = 4 \times (d_{ij}^{-12} - C(\xi_i, \xi_j) \times d_{ij}^{-6}) \quad (6)$$

where

$$C(\xi_i, \xi_j) = \frac{1}{8 \times (1 + \xi_i + \xi_j + 5 \times \xi_i \times \xi_j)} \quad (7)$$

Equation 7 is the potential energy due to the interaction between monomers  $i$  and  $j$ , and  $d_{ij}$  is the distance between these monomers in the chain, such that  $i < j$ .

In the experiments reported in Section 4.8, a total of 5 synthetic protein sequences were used. These sequences have been previously used by other researchers (HSU et al., 2003; KALEGARI; LOPES, 2010). In Table 5,  $N$  is the number of monomers of each sequence (13, 21, 34 and 55 amino acids-long sequences) and it is followed by the sequence itself.

**Table 5: Benchmark sequences for the 2D-AB off-lattice model**

$N$	Sequence
13	ABBABBABABBAB
21	BABABBABABBABABBAB
34	ABBABBABABBABABBABABBABABBABABBAB
55	BABABBABABBABABBABABBABABBABABBABABBAB BABABBABABBABABBAB

### 4.3 SETUP OF PARAMETERS AND STATISTICAL TESTS

In all experiments the parameters of the algorithms were defined empirically. A better understanding of parameters relationship between each other and the definition of default values is quoted as an important future research to address.

For all experiments using benchmark functions the initial population size was set to  $POP = 10$ . For the number of dimensions ( $D$ ) equal to 2, 5, and 10, the parameters used were  $NQ = 100$ ,  $ECO-STEP = 100$ ,  $EVO-STEP = 100$ ,  $T-SIZE = 5$  and  $\rho = 0.5$ . With this configuration, the total number of function evaluations was 10,000 for each population. For  $D = 200$ , given the high number of dimensions, some parameters were empirically redefined:  $NQ = 200$ ,  $ECO-STEP = 500$ ,  $EVO-STEP = 200$ . With this adjustment of parameters, for 200 dimensions, the total number of function evaluations was 100,000 evaluations for each population.

For the experiments carried out with the PSP 2D-AB, the parameters used were  $POP = 40$ ,  $NQ = 200$ ,  $ECO-STEP = 6250$ ,  $EVO-STEP = 800$ , and  $T-SIZE = 5$ . In this problem, the number of dimensions ( $D$ ) corresponds to bonding angles. Thus,  $D$  is 11, 19, 32, and 53 for the

sequences of size 13, 21, 34, and 55, respectively. With this configuration, the total number of function evaluations was 5,000,000 for each population.

Default parameters recommended in the literature were used in the algorithms employed. *POP* is a common parameter between all algorithms and is adjusted as previously mentioned. For ABC algorithm, there is only one control parameter, *limit* = 100 (KARABOGA; AKAY, 2009). For PSO algorithm, besides *POP*, the parameters were set to standard values<sup>8</sup>: inertia weight  $W = 0.721$ ; cognitive and social components  $\varphi_p = \varphi_g = 1.193$ , respectively. For DE algorithm, the parameters are  $F = 0.9$  and  $CR = 1.0$ . And for jDE/BBO the parameters used are  $I = E = 1.0$ ,  $CR = 0.9$ ,  $F = 0.5$ , and  $S_{max} = POP$  (GONG et al., 2010).

Statistical tests were also conducted to better understand the results. All tests were run using the R Statistical Computing tool<sup>9</sup>. In all case studies the Shapiro-Wilk test was applied to verify the null hypothesis that the results obtained came from a normally distributed population. For all cases this null hypothesis was rejected with a level of significance of 0.1%. Once made this verification some non-parametric statistical tests can be applied if necessary (OLOFSSON, 2005; DALGAARD, 2008; DERRAC et al., 2011).

Before performing any non-parametric statistical test, the box-plot information is first analysed. This analysis helps to visually decide if it is necessary to perform or not any statistical test.

When comparing more than two algorithms the Kruskal-Wallis test is applied and the  $p$ -value<sub>K</sub> is analysed. The null hypothesis of this test is that the results from which the samples originate are equivalent. When the Kruskal-Wallis test leads to significant results (rejection of the null hypothesis) then at least one of the algorithms is different from the other algorithms with some statistical significance. The test does not identify where the differences occur or how many differences actually occur.

On the other hand, when comparing only two algorithms, the Wilcoxon rank-sum statistical test is applied and the  $p$ -value<sub>W</sub> is analysed. A choice must be made in the case there are several algorithms to test and it is necessary to choose only two of them to compare. This choice is made depending on the goal of the case being studied and according to the box-plot analysis of each case. Again, the null hypothesis of this test is that the results from which the samples originate are equivalent. When the Wilcoxon rank-sum test leads to significant results (rejection of the null hypothesis) it is possible to conclude that one algorithm is different from the other with some statistical significance. Also, as consequence, it is possible to conclude with

---

<sup>8</sup>Standard PSO (SPSO-07): <http://www.particleswarm.info/Programs.html>

<sup>9</sup>R Project web site: [www.r-project.org/](http://www.r-project.org/)

some statistical significance if one algorithm is better than or equivalent to another algorithm.

Another non-parametric test that can be applied is the Friedman rank test that determine whether the algorithms being compared have been selected from populations having equal medians. As lower the ranking, the better the algorithm.

#### 4.4 CASE STUDY I: FIRST EXPERIMENTS

This particular case study presents the use of the ecological concepts of habitats, ecological relationships and ecological successions to optimize some benchmark functions (PARPINELLI; LOPES, 2011a) (PARPINELLI; LOPES, 2011c). The experiments were conducted using the benchmark functions listed in Table 4.

As presented in Section 3.3.1, in this case study, the Artificial Bee Colony Optimization (ABC) algorithm was also applied in a homogeneous manner, i.e. all populations use this algorithm with the same adjustment of parameters to evolve their candidate solutions.

The eco-inspired algorithm was tested using two configurations. The first configuration employs the ABC algorithm and implements the Algorithm 7 as described in Section 3.3, with the definitions of habitats, topologies and ecological relations of mating and migrations ( $ECO_{ABC}$ ). In the second configuration the ability to create habitats is disabled and, consequently, topologies and interactions are not defined. This second configuration simulates the evolution of the populations completely isolated. Thus, the aim is to verify how the proposed ecological-based approach ( $ECO_{ABC}$ ) performs when compared with its stand-alone algorithm (ABC).

##### 4.4.1 RESULTS AND ANALYSIS

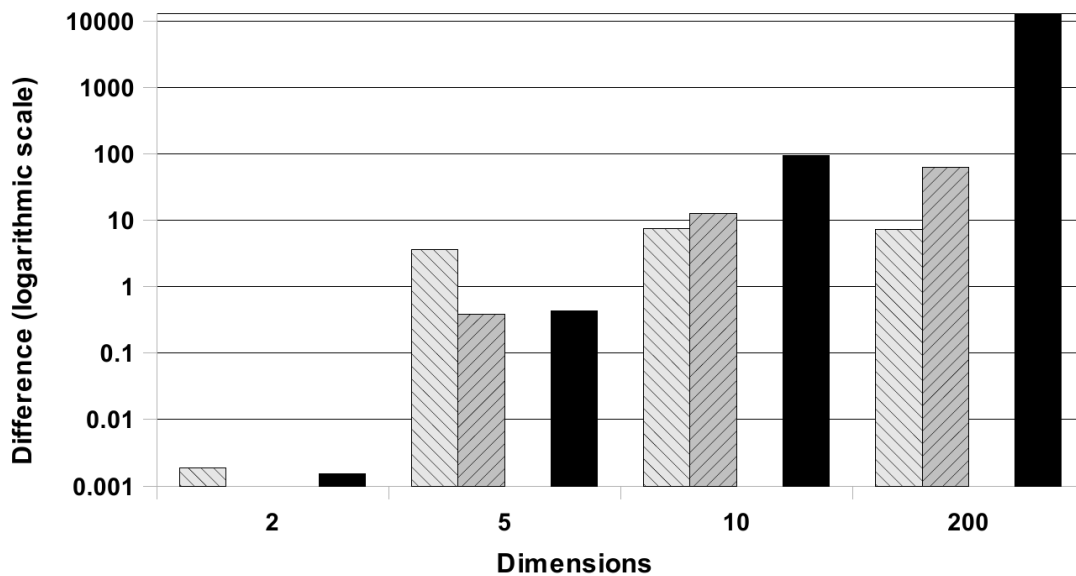
In order to evaluate the behavior of the proposed approach, the benchmark functions of Table 4 were tested using different degrees of complexity (i.e, 2, 5, 10 and 200 dimensions). Table 6 shows the averaged results. For each function, the first line shows the number of dimensions. The second line presents the configuration of the algorithm, with ( $ECO_{ABC}$ ) and without (ABC) the use of ecological concepts, respectively. The ABC usage is equivalent to run the algorithm without co-evolution, where the populations evolve without any exchange of information. The third line (*Global Best*) shows the average and standard deviation of the best result obtained by all populations in all runs.

Analysing Table 6 it is possible to verify that the proposed  $ECO_{ABC}$  algorithm obtained better results than the ABC algorithms for all functions. Also, the gain of  $ECO_{ABC}$  over ABC

**Table 6: Case study I: Results obtained for the benchmark functions.**

<i>Schaffer</i>	$D = 2$		$D = 5$		$D = 10$		$D = 200$	
Model	$ECO_{ABC}$	ABC	$ECO_{ABC}$	ABC	$ECO_{ABC}$	ABC	$ECO_{ABC}$	ABC
Global Best	$0.0000 \pm 0.0$	$0.0186 \pm 0.1$	$0.1341 \pm 0.0$	$3.7867 \pm 1.5$	$1.1344 \pm 0.2$	$4.6569 \pm 0.8$	$20.2792 \pm 0.4$	$27.5936 \pm 0.7$
<i>Rastrigin</i>	$D = 2$		$D = 5$		$D = 10$		$D = 200$	
Model	$ECO_{ABC}$	ABC	$ECO_{ABC}$	ABC	$ECO_{ABC}$	ABC	$ECO_{ABC}$	ABC
Global Best	$0.0000 \pm 0.0$	$0.0000 \pm 0.0$	$0.0000 \pm 0.0$	$0.3832 \pm 0.6$	$0.0000 \pm 0.0$	$10^{-11} \pm 0.0$	$10^{-05} \pm 0.0$	$62.1453 \pm 9.6$
<i>Griewank</i>	$D = 2$		$D = 5$		$D = 10$		$D = 200$	
Model	$ECO_{ABC}$	ABC	$ECO_{ABC}$	ABC	$ECO_{ABC}$	ABC	$ECO_{ABC}$	ABC
Global Best	$0.0000 \pm 0.0$	$10^{-19} \pm 0.0$	$10^{-19} \pm 0.0$	$10^{-12} \pm 0.0$	$10^{-13} \pm 0.0$	$10^{-06} \pm 0.0$	$10^{-11} \pm 0.0$	$10^{-07} \pm 0.0$
<i>Rosenbrock</i>	$D = 2$		$D = 5$		$D = 10$		$D = 200$	
Model	$ECO_{ABC}$	ABC	$ECO_{ABC}$	ABC	$ECO_{ABC}$	ABC	$ECO_{ABC}$	ABC
Global Best	$0.0000 \pm 0.0$	$0.0015 \pm 0.0$	$0.0019 \pm 0.0$	$0.4379 \pm 0.2$	$0.0086 \pm 0.0$	$0.0098 \pm 0.0$	$137.8643 \pm 42.0$	$13036.1 \pm 4193.4$

can be best observed with the increase of dimensionality of each function. Figure 13 shows the difference between the results obtained by ABC and  $ECO_{ABC}$  approaches for each dimension of each benchmark function from Table 6. The x-axis represents the dimensions and the y-axis represents the differences. For a better visualization, the y-axis is in logarithmic scale.

**Figure 13: Case study I: Difference between ABC and  $ECO_{ABC}$  approaches.**

From Figure 13 it is possible to verify that the difference between ABC and  $ECO_{ABC}$  increases proportionally to the number of dimensions for *Rastrigin* and *Rosenbrock* functions indicating that higher complexities are more challenging to be solved. For the *Griewank* function the difference is around zero for all dimensions and the results obtained are around the global optimum for both algorithms (i.e, the algorithms are in their limit, very close to the global optimum). This indicates that the function is not challenging enough to both optimization approaches considering these degrees of complexity. Analysing the *Schaffer* function, the difference is accentuated from 2 to 5, and from 5 to 10 dimensions but it is almost constant from 10 to 200 dimensions. This indicates that the algorithms have certain difficulty to solve this

function both with few and several dimensions. Hence, in order to deal with more challenging instances, from this point forward, only 200 dimensions will be considered.

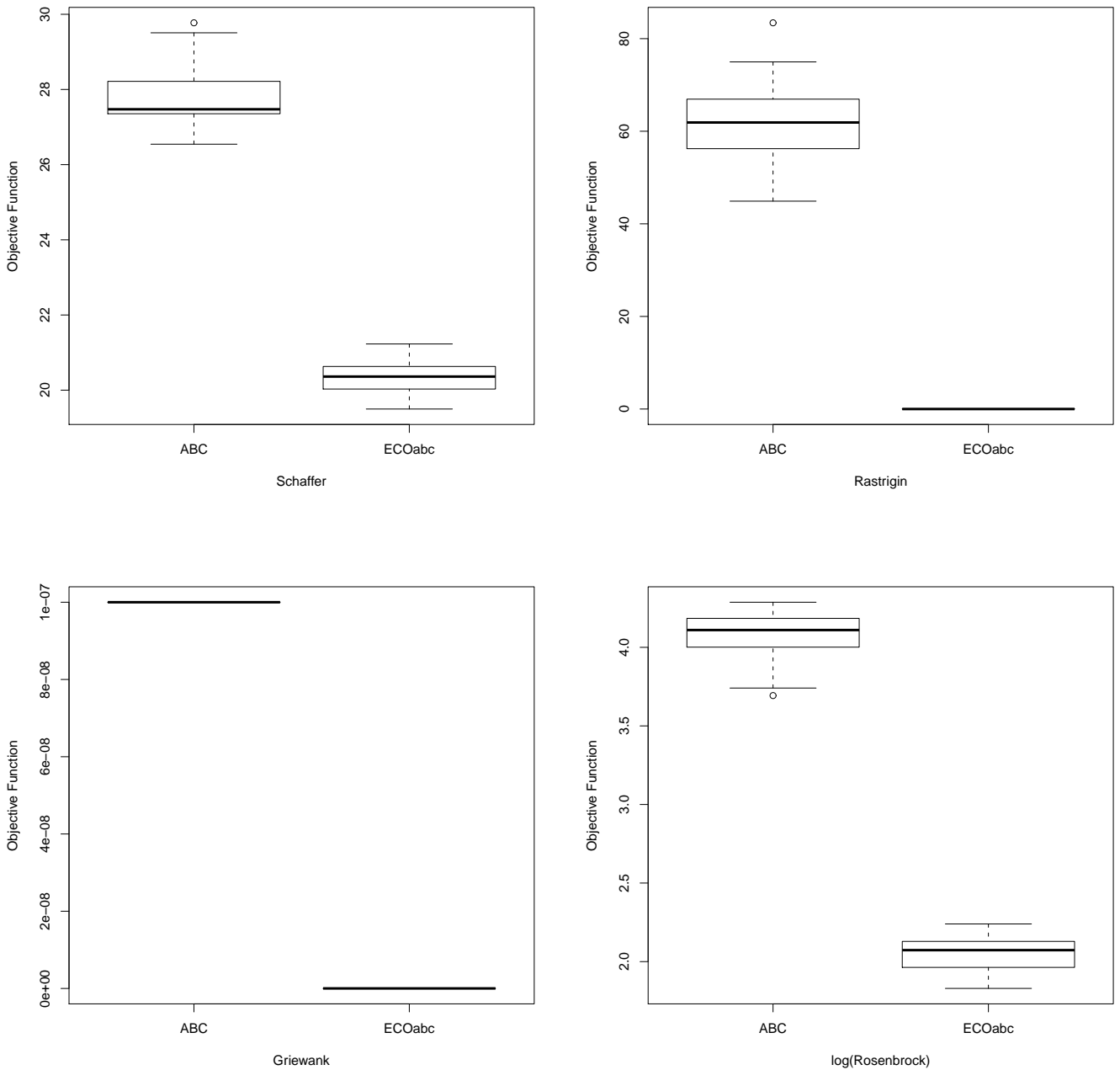
Analysing Table 6 for  $D = 200$  and using the box-plot shown in Figure 14 one can verify that the results obtained by ABC and  $\text{ECO}_{ABC}$  approaches are statistically different one from another (i.e., there is no intersection between their ranges). Hence, it is possible to conclude that the proposed  $\text{ECO}_{ABC}$  algorithm obtained significantly better results than the algorithm running without the concepts of habitat and interactions between populations (ABC), for all functions. The better performance of  $\text{ECO}_{ABC}$  is possibly due to the ecological interactions (intra and inter-habitats) that favour the co-evolution of populations. Also, the proposed approach better explores the diversity of solutions.

Table 7 shows the mean of the elapsed time in minutes by each configuration for each function, followed by the respective standard deviations. It is possible to notice that the  $\text{ECO}_{ABC}$  slightly increased the processing time, around 0.1680 minutes on average, due the addition of new ecosystemic routines for information exchange. The processing time increasing is not so significant when compared with the improvements achieved by  $\text{ECO}_{ABC}$  in the overall results obtained.

**Table 7: Case study I: Mean of the processing time over all runs (minutes).**

	$D = 200$	
	ABC	$\text{ECO}_{ABC}$
<i>Schaffer</i>	$0.91 \pm 0.01$	$1.14 \pm 0.01$
<i>Rastrigin</i>	$1.14 \pm 0.01$	$1.27 \pm 0.01$
<i>Griewank</i>	$1.08 \pm 0.01$	$1.29 \pm 0.01$
<i>Rosenbrock</i>	$1.21 \pm 0.01$	$1.31 \pm 0.01$

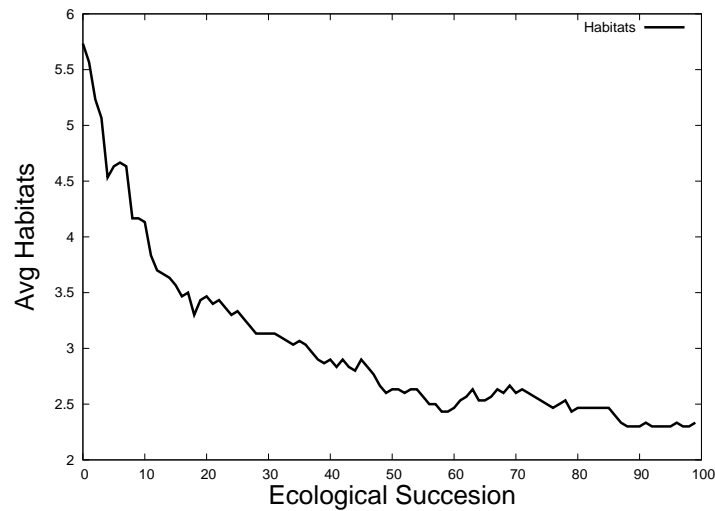
Figure 15 shows the evolution of the average number of habitats for each ecological succession step for 30 executions of the  $\text{ECO}_{ABC}$  algorithm. The observed behavior is the same for all functions, and Figure 15 shows the evolution for the *Schaffer* function with  $D = 10$ . It is observed that, at the beginning of the optimization process, with the populations widely dispersed in the search space, there is a large number of habitats. As the optimization process moves through the ecological successions, the populations tend to move through the search space converging to specific regions. As shown in Figure 15, the number of habitats decreases with the ecological succession cycles, indicating that the populations tend to converge to points close to each other.



**Figure 14: Case study I: Box plot for the results obtained ( $D = 200$ ).**

#### 4.4.2 CONSIDERATIONS

According to the results, the benefit of using the ecology-inspired approach can be best observed with problem instances with higher dimensionality. This indicates that the ecology-inspired approach can be more effective in solving more complex problems rather than populations evolving alone. Also, the ecological interactions (intra and inter-habitats) favour the co-evolution of populations and better explores the diversity of solutions.



**Figure 15: Case study I: Average number of habitats at each ecological succession.**

#### 4.5 CASE STUDY II: POPULATION DYNAMICS

In nature, populations are always dynamic in such a way that the size of populations oscillate across their habitats over time. However, in most Evolutionary Computation applications, the population size is constant and does not change during the search (EIBEN; SMITH, 2003). Current practice of manual setting of the population size in evolutionary computation is experience-based, but not robust. Hence, this case study aims to explore the population sizing not as a parameter but as a dynamic process that changes deterministically over time (PARPINELLI; LOPES, 2012c).

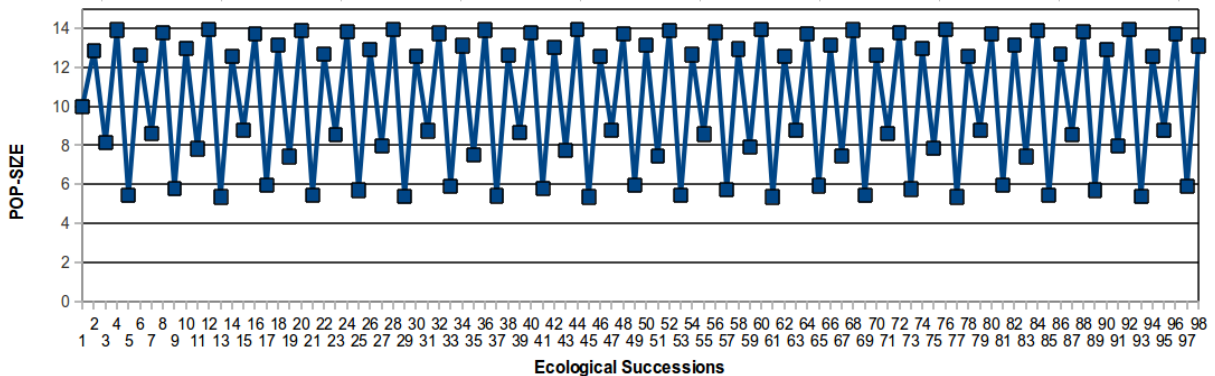
The Artificial Bee Colony Optimization (ABC) algorithm (KARABOGA; AKAY, 2009) was used in a homogeneous model, i.e. all populations use this algorithm with the same control parameters.

This case study explores a more biologically plausible survival selection mechanism through the use of population dynamics where the logistic model is applied to control the size of populations (KAPLAN; GLASS, 1995)(MAY; MCLEAN, 2007). The logistic map was chosen due to its simplicity and its rich dynamic behaviour as discrete-time demographic model. The aim is to apply population dynamics in the stand-alone algorithm (ABC) and in the proposed ECO approach (Algorithm 7) in order to verify if it is beneficial or not.

The one-parameter logistic map was applied to drive the population dynamics between ecological successions (Section 3.2.12). The logistic map (Equation 1) was applied to adjust the population size dynamically. The logistic map parameter was set to  $a = 3.57$  and this is called ‘route to chaos’ (KAPLAN; GLASS, 1995). This choice was done based on the work



of Ma (MA, 2011) where experiments were performed with different values for the parameter  $a$ . Figure 16 shows the resizing projection for 100 ecological successions with  $POP_i = 10$  as initial condition for all populations. In this simulation, 5 is the minimum and 14 is the maximum number of candidate solutions achieved.



**Figure 16: Case study II: Population dynamics according to the logistic model ( $a = 3.57$ ).**

Three populational scenarios can occur between ecological successions (line 11 in Algorithm 7). The first is when there are no changes in the size of populations from time  $t$  to  $t + 1$ . In this case the evolution proceeds as usual. The second is when there is increment in the size of populations from  $t$  to  $t + 1$ . In this case, new solutions are randomly generated using the current centroid as reference. The third scenario is when there is decrement in the size of populations from  $t$  to  $t + 1$ . In this case, the population is ranked and the worst solutions are discarded. Hence, the second scenario favors exploration and the third scenario favours exploitation. In addition to the exploration and exploitation routines provided by the evolution of populations and by the ecological interactions (inter and intra-habitats), the use of population dynamics creates a new biologically plausible mechanism to diversify the search.

The ecological-inspired framework (ECO) was tested using four configurations. The first configuration implements the Algorithm 7 as described in Section 3.3, with the definitions of habitats, topologies and ecological relations. The second configuration complements the first one by adding population dynamics. The third configuration disables the ability to create habitats and, consequently, topologies and interactions are not defined. This configuration simulates the evolution completely isolated populations, and they evolve without exchanging information. The fourth configuration complements the third one by adding population dynamics. For each configuration, the algorithm was run 30 times.

#### 4.5.1 RESULTS AND ANALYSIS

The experiments were conducted using the benchmark functions shown in Table 4. Each of these functions was tested with 200 dimensions.

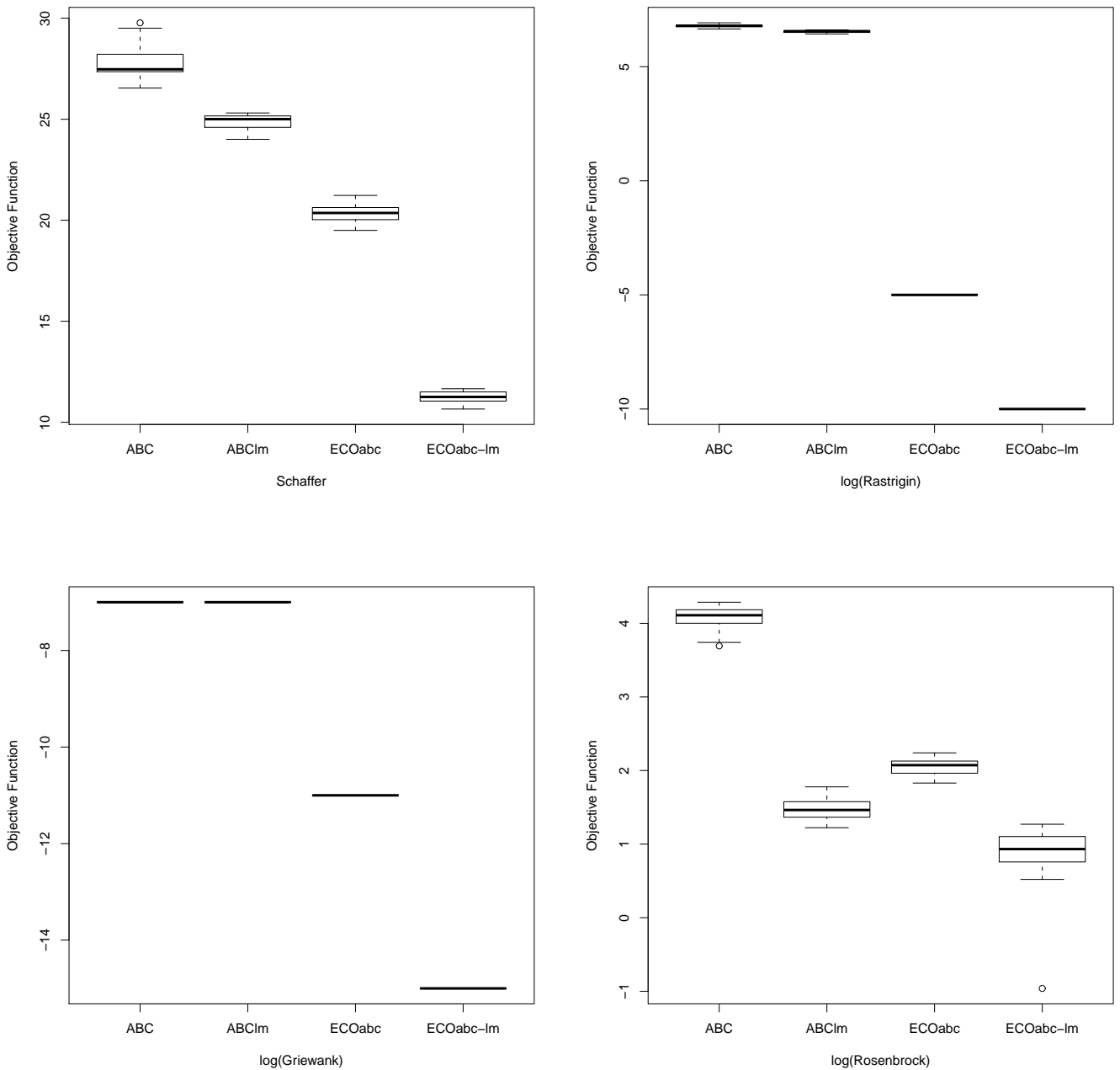
Table 8 shows the averaged results obtained for the benchmark functions. The results obtained by each configuration of the algorithms are presented in columns 2 to 5. Column 2 shows the results obtained by the ABC algorithm running alone, without co-evolution. Column 3 shows the results obtained by the ABC algorithm running with the logistic model for modelling the population dynamics ( $ABC_{LM}$ ). Column 4 shows the results obtained by the ABC algorithm using the ecological-inspired approach ( $ECO_{ABC}$ ). Finally, column 5 shows the results obtained by the ABC algorithm using the ecological-inspired approach running with the logistic model for the population dynamics ( $ECO_{ABC-LM}$ ). Each cell of the table shows the average and standard deviation of the best result obtained by all populations in all runs (*Global Best*).

**Table 8: Case study II: Results obtained for the benchmark functions.**

	$D = 200$			
	ABC	$ABC_{LM}$	$ECO_{ABC}$	$ECO_{ABC-LM}$
<i>Schaffer</i>	$27.5936 \pm 0.73$	$24.7426 \pm 0.5$	$20.2792 \pm 0.40$	$11.2742 \pm 0.4$
<i>Rastrigin</i>	$62.1453 \pm 9.6$	$34.0388 \pm 4.3$	$10^{-05} \pm 0.0$	$10^{-10} \pm 0.0$
<i>Griewank</i>	$10^{-7} \pm 0.0$	$10^{-7} \pm 0.0$	$10^{-11} \pm 0.0$	$10^{-15} \pm 0.0$
<i>Rosenbrock</i>	$13036.1 \pm 4193.4$	$35.1444 \pm 11.6$	$137.86 \pm 42.0$	$9.2568 \pm 4.1$

Lets use the box-plot shown in Figure 17 and Table 8 to analyse the results. Analysing the ABC and  $ABC_{LM}$  configurations one can observe that the use of population dynamics improved the results in most cases (for *Schaffer*, *Rastrigin*, and *Rosenbrock* functions) and remained the same in only one function (*Griewank* function). Also, using the box-plot visualization, it is possible to conclude that the results, for those functions where  $ABC_{LM}$  is better than ABC, are statistically significant (i.e., there is no intersection between their ranges).

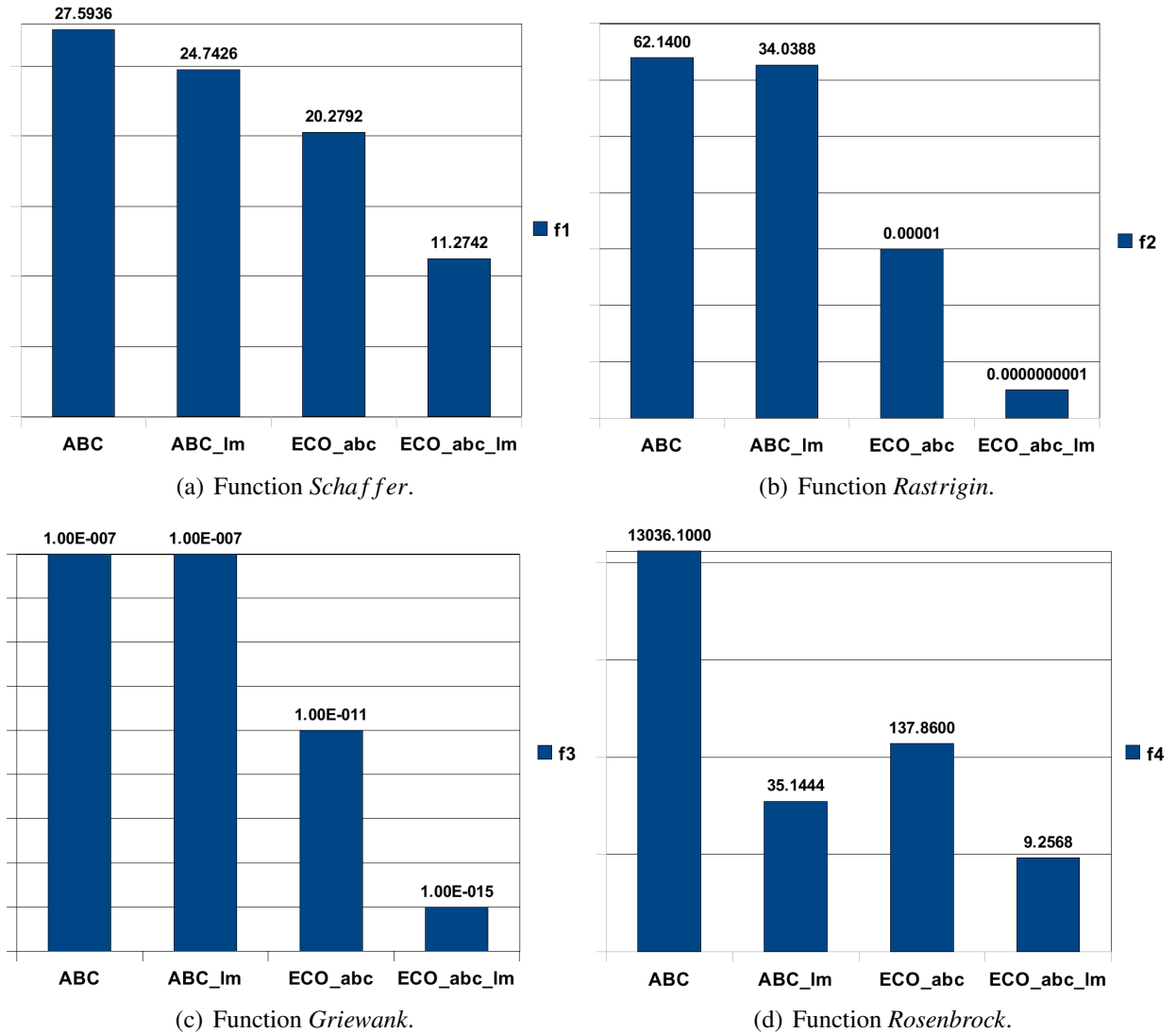
Analysing the results for the ecological-inspired approach with population dynamics,  $ECO_{ABC-LM}$ , it is possible to observe that the results were significantly better than the ecological-inspired approach without population dynamics ( $ECO_{ABC}$ ) for all functions. Again, using the box-plot visualization, one can verify that the results are statistically significant (i.e., there is no intersection between their ranges). This gain is mainly due to the use of a more natural survival selection mechanism afforded by population dynamics. For all functions, the  $ECO_{ABC-LM}$  approach is statistically different from the others (see Figure 17). Hence, it is possible to conclude that the proposed  $ECO_{ABC-LM}$  algorithm obtained significantly better results for all functions.



**Figure 17: Case study II: Box plot for the results obtained ( $D = 200$ ).**

In Figure 18 it is possible to visually verify the results, where the  $x$ -axis shows the different approaches and the  $y$ -axis represents the *Global Best* values for each approach. The  $y$ -axis of Figures 18(b), 18(c), and 18(d) are in logarithmic scale.

Applying the Friedman rank test to the results shown in Table 8, the following rank is obtained: ECO<sub>ABC-LM</sub> with 1; ECO<sub>ABC</sub> with 2.25; ABC<sub>LM</sub> with 2.86; and ABC with 3.86. The lower the ranking, the better the algorithm. Therefore, it is clear that the ECO<sub>ABC-LM</sub> algorithm works very well in these functions because it has the lower rank which reinforces the use of



**Figure 18: Case study II: Bar graph of each benchmark function with  $D = 200$ . The  $x$ -axis represents the different approaches and the  $y$ -axis shows the average best results.**

populations dynamics to adjust the size of populations.

Table 9 shows the mean of the elapsed time in minutes by each configuration for each function, followed by the respective standard deviations. One can notice that the use of populations dynamics slightly increased the processing time, around 0.1041 minutes on average, for  $ABC_{LM}$  and  $ECO_{ABC-LM}$  compared with  $ABC$  and  $ECO_{ABC}$ , respectively. From the results, it is possible to conclude that the processing time increasing is not so significant when compared with the improvements achieved by  $ECO_{ABC-LM}$  in the overall results obtained.

#### 4.5.2 CONSIDERATIONS

This case study adds to the ECO algorithm the ecological concept of population dynamics to set up the population sizes. The population dynamics model applied was the

**Table 9: Case study II: Mean of the processing time over all runs (minutes).**

	$D = 200$			
	ABC	$ABC_{LM}$	$ECO_{ABC}$	$ECO_{ABC-LM}$
<i>Schaffer</i>	$0.91 \pm 0.01$	$0.99 \pm 0.01$	$1.14 \pm 0.01$	$1.21 \pm 0.01$
<i>Rastrigin</i>	$1.14 \pm 0.01$	$1.31 \pm 0.01$	$1.27 \pm 0.01$	$1.36 \pm 0.01$
<i>Griewank</i>	$1.08 \pm 0.01$	$1.26 \pm 0.01$	$1.29 \pm 0.01$	$1.39 \pm 0.01$
<i>Rosenbrock</i>	$1.21 \pm 0.01$	$1.24 \pm 0.01$	$1.31 \pm 0.01$	$1.41 \pm 0.01$

logistic map due to its simplicity and its rich dynamic behaviour as discrete-time demographic model.

The addition of this feature brought a higher biological plausibility to the proposed algorithm, opposed to most bio-inspired algorithms that take inspiration only from one biological phenomenon.

In addition to the exploration and exploitation routines provided by the evolution of populations and by the ecological interactions (inter and intra-habitats), the use of population dynamics creates a new biologically plausible mechanism to diversify the search. Also, due this new feature to diversify candidate solutions, the use of a population dynamics model inside the ECO framework considerably improved the results for these benchmark functions.

#### 4.6 CASE STUDY III: HETEROGENEOUS MODEL

This case study explores the heterogeneity of the ecology-inspired approach using two different algorithms cooperatively: the ABC (KARABOGA; AKAY, 2009) and the PSO (CLERC, 2006).

The ecological-inspired algorithm (ECO) was tested using three models. The first model implements the Algorithm 7 as described in Section 3.3, with the definitions of habitats, topologies and ecological relations. Using this model it is intended to compare the results obtained by the homogeneous application of ECO against the results obtained by the heterogeneous application of ECO. In the heterogeneous application, the ABC algorithm and the PSO algorithm are used in such a way that half of the number of populations ( $NQ/2$ ) is managed by the ABC algorithm, and the other half by the PSO algorithm to evolve their candidate solutions (PARPINELLI; LOPES, 2012a).

In the second model, the ability to create habitats is disabled and, consequently, topologies and interactions are not defined. This second model simulates the evolution completely isolated populations, and they evolve without exchanging information.

The third model disables the ability to create habitats, but explores co-evolution using multi-populations in a ring topology (i.e., the well known island model in EC). In this model, migration policy and migration frequency occurs exactly as in the first configuration. Thus, another aim of this case study is to compare the results obtained by the homogeneous and heterogeneous application of the island model against the results obtained by the homogeneous and heterogeneous application of the ECO approach, respectively.

From these three models, eight configurations are derived using both ABC and PSO and are described in next section. The experiments were conducted using the benchmark functions shown in Table 4. Each of these functions was tested with 200 dimensions and, for each configuration, the algorithm was run 30 times.

#### 4.6.1 RESULTS AND ANALYSIS

Table 10 shows the averaged results obtained for the benchmark functions by each configuration of the algorithms (column 2 to 5 in both upper and lower parts of the table). In the upper part, column 2 shows the results obtained by the ABC algorithm running alone, without co-evolution. Column 3 shows the results obtained by the ABC algorithm running with a ring topology ( $ABC_{ring}$ ). Column 4 shows the results obtained by the ABC algorithm using the ecological-inspired approach homogeneously ( $ECO_{ABC}$ ). Column 5 shows the results obtained by the PSO algorithm running alone, without co-evolution. In the lower part, column 2 shows the results obtained by the PSO algorithm running with a ring topology ( $PSO_{ring}$ ). Column 3 shows the results obtained by the PSO algorithm using the ecological-inspired approach homogeneously ( $ECO_{PSO}$ ). Column 4 shows the results obtained by the ABC and PSO algorithms alternated within a ring topology ( $PSO-ABC_{ring}$ ). In this configuration, half population is controlled by the ABC algorithm and the other half is controlled by the PSO algorithm. Finally, column 5 of the lower part shows the results obtained by the ABC and PSO algorithms using the ecological-inspired approach heterogeneously ( $ECO_{PSO-ABC}$ ). Again, half of the number of populations ( $NQ/2$ ) is controlled by the ABC algorithm and the other half by the PSO. Each cell of the table shows the average and standard deviation of the best result obtained by all populations in all runs (*Global Best*).

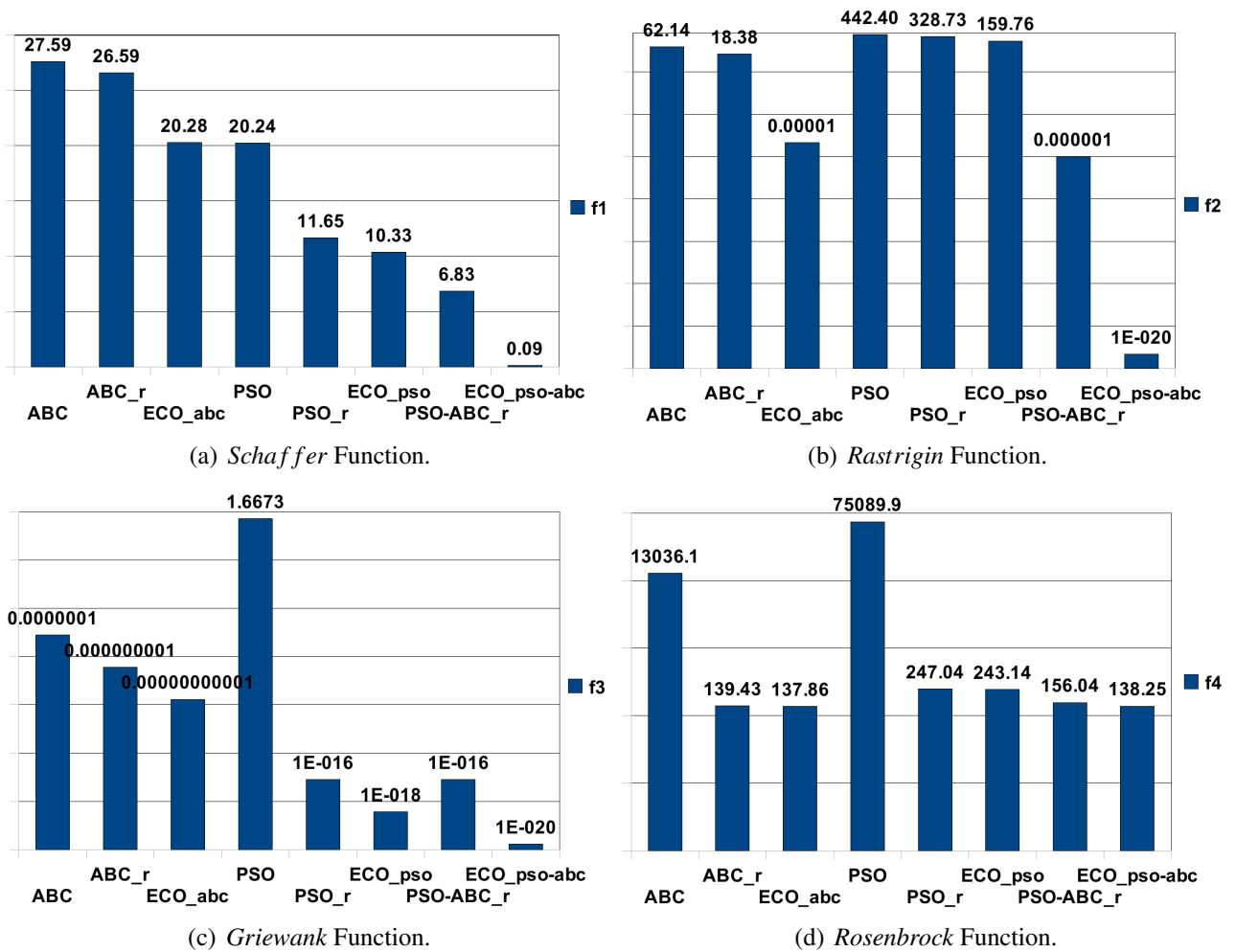
In Figure 19 the results can be visualized, where the *x-axis* shows the different approaches and the *y-axis* represents the *Global Best* values for each approach. The *y-axis* of Figures 19(b), 19(c), and 19(d) are in logarithmic scale.

Let's first analyse the results obtained by the homogeneous application of ECO ( $ECO_{ABC}$  and  $ECO_{PSO}$ ) against the results obtained by the heterogeneous application of ECO

**Table 10: Case Study III: Results obtained for the benchmark functions.**

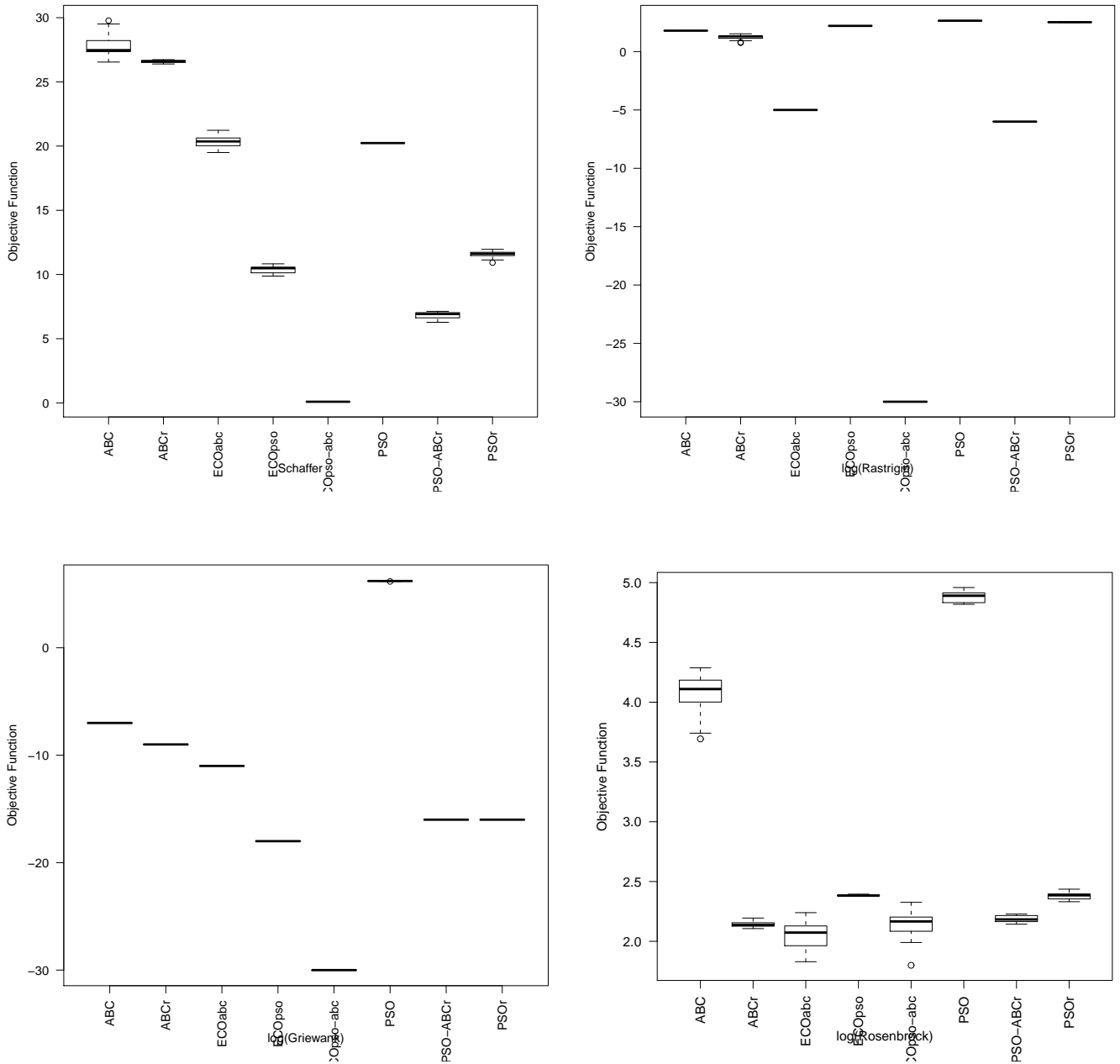
	$D = 200$			
	ABC	$ABC_{Ring}$	$ECO_{ABC}$	PSO
<i>Schaffer</i>	$27.5936 \pm 0.7$	$26.5936 \pm 0.1$	$20.2792 \pm 0.4$	$20.2355 \pm 0.0$
<i>Rastrigin</i>	$62.1453 \pm 9.6$	$18.3877 \pm 6.2$	$10^{-05} \pm 0.0$	$442.4012 \pm 8.7$
<i>Griewank</i>	$10^{-07} \pm 0.0$	$10^{-09} \pm 0.0$	$10^{-11} \pm 0.0$	$1.6673 \pm 0.1$
<i>Rosenbrock</i>	$13036.1 \pm 4193.4$	$139.4325 \pm 7.4$	$137.86 \pm 42.0$	$75089.9 \pm 9999.6$
	$PSO_{Ring}$	$ECO_{PSO}$	$PSO-ABC_{Ring}$	$ECO_{PSO-ABC}$
<i>Schaffer</i>	$11.6461 \pm 0.3$	$10.3316 \pm 0.3$	$6.8273 \pm 0.3$	$0.0906 \pm 0.0$
<i>Rastrigin</i>	$328.7323 \pm 12.1$	$159.7621 \pm 0.5$	$10^{-06} \pm 0.0$	$0.0000 \pm 0.0$
<i>Griewank</i>	$10^{-16} \pm 0.$	$10^{-18} \pm 0.0$	$10^{-16} \pm 0.0$	$0.0000 \pm 0.0$
<i>Rosenbrock</i>	$247.041 \pm 13.1$	$243.14 \pm 2.6$	$156.0453 \pm 16.4$	$138.25 \pm 37.9$

( $ECO_{PSO-ABC}$ ). Using the box-plot shown in Figure 20, the results from Table 10 and the bar graphs of Figure 19 it is possible to verify that for three functions, namely, *Schaffer*, *Rastrigin*



**Figure 19: Case study III: Bar graph off each benchmark function with  $D = 200$ . The  $x$ -axis represents the different approaches and the  $y$ -axis shows the average best results.**

and *Griewank*, the heterogeneous application of ECO obtained significantly better results over the homogeneous applications.



**Figure 20: Case study III: Box plot for the results obtained ( $D = 200$ ).**

For the *Rosenbrock* function it is possible to notice that some statistical inference is needed due the proximity of the results obtained. Applying the Kruskal-Wallis test over  $ECO_{ABC}$ ,  $ECO_{PSO}$  and  $ECO_{PSO-ABC}$ , a  $p\text{-value}_K$  of  $10^{-16}$  is returned indicating that, with a significance level of 5%, the null hypothesis for this test is rejected. This means that at least one of these algorithms is significantly different from the others. From the box-plot one can see that this difference is provided by the  $ECO_{PSO}$ . Remains now to check if the results obtained by



$ECO_{ABC}$  and  $ECO_{PSO-ABC}$  are statistically different. Applying the Wilcoxon rank-sum test, a  $p$ -value<sub>W</sub> of 0.6827 is returned indicating that, with a significance level of 5%, the null hypothesis for this test is accepted. This means that the results obtained by  $ECO_{ABC}$  and  $ECO_{PSO-ABC}$  are statistically equivalent. Overall, the heterogeneous approach of ECO obtained better results than the homogeneous approach of ECO mainly due to the use of different search strategies in the optimization process. Also, both homogeneous and heterogeneous ECO approaches obtained better results than the ABC and PSO algorithms running stand-alone.

Now the results obtained by the homogeneous island approaches,  $ABC_{Ring}$  and  $PSO_{Ring}$ , and by the homogeneous ECO approaches,  $ECO_{ABC}$  and  $ECO_{PSO}$ , will be compared. Comparing  $ABC_{Ring}$  with  $ECO_{ABC}$ , the results obtained by  $ECO_{ABC}$  are clearly better in functions *Schaffer*, *Rastrigin* and *Griewank*. For the *Rosenbrock* function the obtained results are statistically the same with a  $p$ -value<sub>W</sub> of 0.4610 at a significance level of 5%. Comparing  $PSO_{Ring}$  and  $ECO_{PSO}$ , the results obtained by  $ECO_{PSO}$  are clearly better in functions *Rastrigin* and *Griewank*. For *Schaffer* function, a statistical test is required due the proximity of the results and, with a  $p$ -value<sub>W</sub> of  $10^{-08}$ , it is possible to infer that the results obtained by these two approaches are statistically different considering a significance level of 5%. For the *Rosenbrock* function the obtained results are statistically equivalent with a  $p$ -value<sub>W</sub> of 0.3669, considering a significance level of 5%. Overall, the homogeneous ECO approaches performed better than the homogeneous island approaches in three functions (*Schaffer*, *Rastrigin* and *Griewank*) and performed equivalently in one function (*Rosenbrock*). This gain is possibly due to the use of ecological interactions (intra and inter-habitats) that favour the co-evolution of populations.

Analysing the box-plots for the heterogeneous application of the ECO approach,  $ECO_{PSO-ABC}$ , it is possible to observe that the results were significantly better than the heterogeneous island model ( $PSO-ABC_{Ring}$ ) in three functions (*Schaffer*, *Rastrigin* and *Griewank*). Besides the ecological interactions, this gain is possibly due to the use of different search strategies in the optimization process. For the *Rosenbrock* function the obtained results are statistically equivalent with a  $p$ -value<sub>W</sub> of 0.1026, considering a significance level of 5%. Moreover, if the four average best algorithms for the *Rosenbrock* function ( $ABC_{Ring}$ ,  $ECO_{ABC}$ ,  $PSO-ABC_{Ring}$  and  $ECO_{PSO-ABC}$ ) is considered, a  $p$ -value<sub>K</sub> of 0.4497 is obtained, concluding that they are statistically equivalent. Overall, the  $ECO_{PSO-ABC}$  performed better than  $PSO-ABC_{Ring}$ .

Applying the Friedman rank test to Table 10, the following rank is obtained:  $ECO_{PSO-ABC}$  with 1.25;  $PSO-ABC_{Ring}$  with 2.875;  $ECO_{ABC}$  with 3.75;  $ECO_{PSO}$  with 4;  $ABC_{Ring}$  with 5;  $PSO_{Ring}$  with 5.125; ABC with 6.75; and PSO with 7.25. It is clear that the  $ECO_{PSO-ABC}$  algorithm works very well because it has the lower rank, which confirms the

robustness of the heterogeneous model compared with the other approaches.

Table 11 shows the mean of the elapsed time in minutes by each configuration for each function, followed by the respective standard deviations. Important to remember that all approaches ran with the same number of function evaluations. It is possible to notice that the homogeneous and the heterogeneous application of the ECO approaches achieved practically the same processing times when compared with the homogeneous and heterogeneous application of the *Ring* approaches, respectively.

**Table 11: Case study III: Mean of the processing time over all runs (minutes).**

	$D = 200$			
	ABC	ABC <sub>Ring</sub>	ECO <sub>ABC</sub>	PSO
<i>Schaffer</i>	0.9112±0.0078	1.0945±0.0078	1.1388±0.0078	89.4500±0.0137
<i>Rastrigin</i>	1.1388±0.0078	1.2722±0.0078	1.2667±0.0137	91.1445±0.0157
<i>Griewank</i>	1.0778±0.0078	1.2500±0.0000	1.2945±0.0078	92.0500±0.2947
<i>Rosenbrock</i>	1.2112±0.0078	1.2167±0.0000	1.3112±0.0078	90.2278±0.0157
	PSO <sub>Ring</sub>	ECO <sub>PSO</sub>	PSO-ABC <sub>Ring</sub>	ECO <sub>PSO-ABC</sub>
<i>Schaffer</i>	93.3555±0.0393	93.1222±0.1595	45.3333±0.4552	45.6945±0.0208
<i>Rastrigin</i>	91.1222±0.0208	91.7945±0.0078	48.6445±0.2495	43.9555±0.0550
<i>Griewank</i>	93.6055±0.0437	93.3500±0.0408	45.3945±0.1335	44.7388±0.0342
<i>Rosenbrock</i>	94.6500±0.0000	96.2888±0.0078	42.7612±0.0208	43.0722±0.0078

#### 4.6.2 CONSIDERATIONS

In this case study the Artificial Bee Colony Optimization algorithm and the Particle Swarm Optimization algorithm were used to build an heterogeneous/hybrid ecological-inspired algorithm.

The use of different search strategies during the optimization process inside the ECO framework obtained better results than the homogeneous application of ECO. The main reason for this improvement is the fact that the heterogeneous model uses different intensification and diversification procedures. This creates different dynamics and evolutive behaviors in the search for promising regions in the space of solutions.

Also, the use of a new level of coevolution through the use of ecological interactions (intra and inter-habitats) is beneficial in the search for best solutions.

#### 4.7 CASE STUDY IV: HIERARCHICAL CLUSTERING FOR HABITATS FORMATION

It is well known that, in nature, populations are dynamic in space and time. This means that the formation of habitats changes over time and its formation is not deterministic.

This case study explores the use of a hierarchical clustering technique (MURTAGH; CONTRERAS, 2012)(LEGENDRE; LEGENDRE, 1998) as a biologically plausible strategy to set probabilistically the habitats of the computational ecosystem (Section 3.2.5, Algorithm 6) (PARPINELLI; LOPES, 2012b).

The aim is to compare the results obtained by the ECO algorithm using hierarchical clustering with the results obtained by the algorithm running without this functionality.

A key concept of the proposed ECO system is the definition of habitats (line 6 in Algorithm 7). In Section 4.4 (Case study I) the definition of habitats is performed deterministically by the use of a user defined proximity threshold  $\rho$ . In this development (ECO-C), a hierarchical clustering algorithm was used to setup the habitats where each cluster represents a habitat. Also, this approach suppress the control parameter  $\rho$ .

Differently from the work done in (PARPINELLI; LOPES, 2011a) (described in Section 4.4), in this study the definition of the intra-habitats communication topology does not use any proximity threshold. Again, aiming at improving the biological plausibility of the system, here, a communication topology that is probabilistically defined was used, as described in Section 3.2.6.

All the non-mentioned procedures of Algorithm 7 remain the same as described in Section 3.3.

These experiments were conducted using the benchmark functions shown in Table 4. Each of these functions was tested with 200 dimensions.

Again, in all experiments, the Artificial Bee Colony Optimization (ABC) algorithm (KARABOGA; AKAY, 2009) was used in a homogeneous model.

#### 4.7.1 RESULTS AND ANALYSIS

Table 12 shows results obtained for the benchmark functions and are presented in columns 2 to 3. The ecological-inspired framework was tested using two configurations. The first configuration implements Algorithm 7 as described in Section 3.3, with the definitions of habitats using the proximity threshold  $\rho$  (ECO<sub>ABC</sub>, second column of Table 12). The second configuration implements Algorithm 7 and enables the ability to probabilistically create habitats using the single-link clustering information upon the proposed Algorithm 6 (ECO-C<sub>ABC</sub>, third column of Table 12). For each configuration, the algorithm was run 30 times. Each cell of this table shows the average and standard deviation of the best result obtained by all populations in all runs (*Global Best*).

**Table 12: Case study IV: Results obtained for the benchmark functions.**

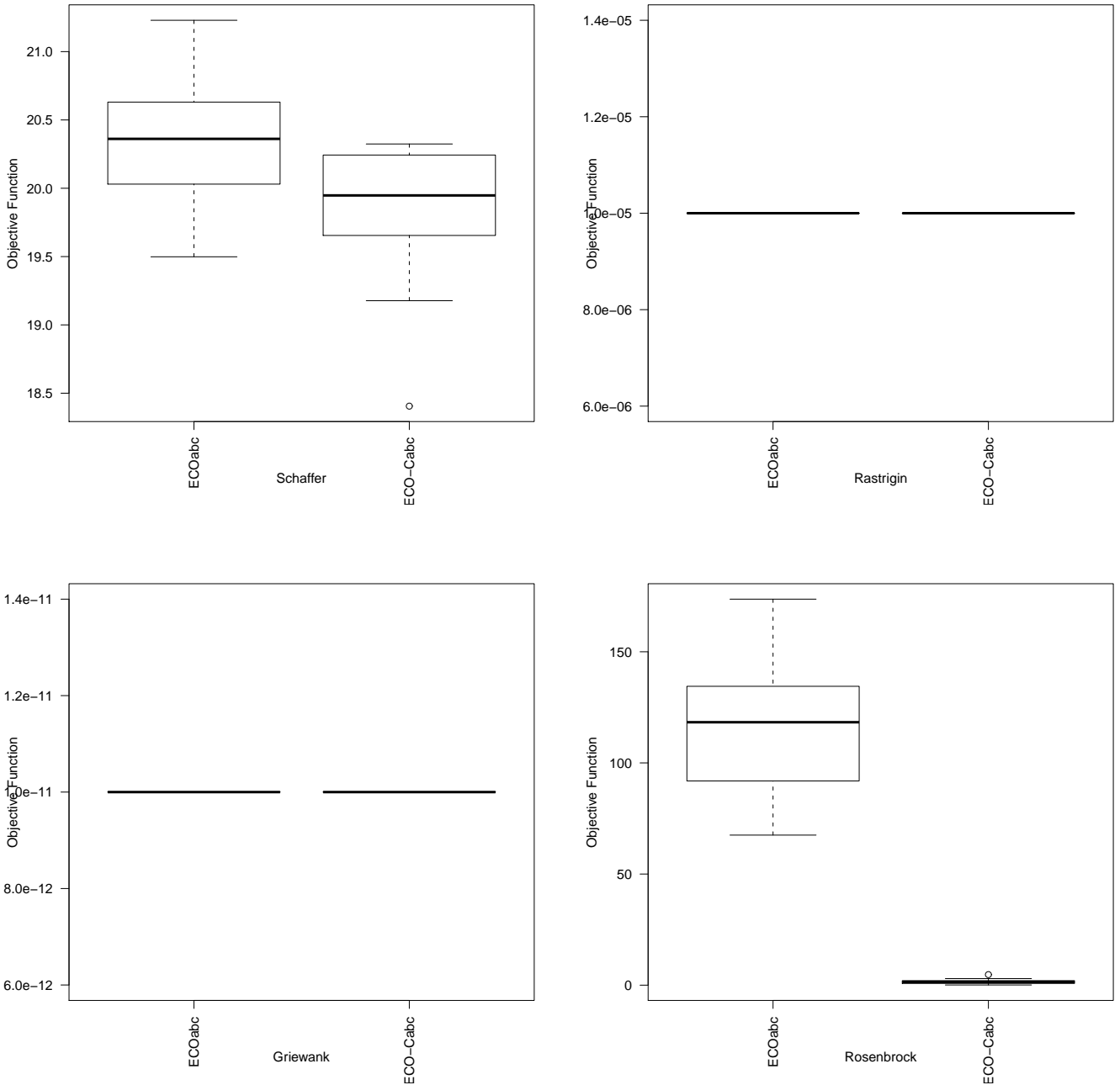
	$D = 200$	
	$ECO_{ABC}$	$ECO-C_{ABC}$
<i>Schaffer</i>	$20.2792 \pm 0.4$	$19.8027 \pm 0.5$
<i>Rastrigin</i>	$10^{-05} \pm 0.0$	$10^{-05} \pm 0.0$
<i>Griewank</i>	$10^{-11} \pm 0.0$	$10^{-11} \pm 0.0$
<i>Rosenbrock</i>	$137.86 \pm 42.0$	$1.8778 \pm 1.7$

Using the box-plot shown in Figure 21 and the results from Table 12 it is possible to verify that for two functions, namely, *Rastrigin* and *Griewank*,  $ECO-C_{ABC}$  obtained the same results of  $ECO_{ABC}$ . For *Schaffer* function, a statistical test is required due the proximity of the results and, with a  $p$ -value<sub>W</sub> of 0.0175, can be inferred that the results obtained by these two approaches are statistically equivalent considering a significance level of 5%. For the *Rosenbrock* function the obtained results are significantly different.

Overall, it is possible to observe that the results obtained by the ecological-inspired approach with probabilistic habitat definition,  $ECO-C_{ABC}$ , were equivalent or better for all functions when compared with the ecological-inspired approach without the use of clustering strategy ( $ECO_{ABC}$ ). This analysis indicates that the behavior of the ecological algorithm does not change when using the proposed hierarchical clustering strategy to probabilistically setup the habitats and communication topology. It is worth mentioning that with this strategy a critical parameter ( $\rho$ ) is suppressed. Also, one can notice that the results obtained by  $ECO-C_{ABC}$  for *Rosenbrock* function was much better than the  $ECO_{ABC}$  approach. This result indicates that the value for the  $\rho$  parameter present in  $ECO_{ABC}$  was not the best choice and should be optimized. With the application of the hierarchical clustering this problem is clearly solved.

Figure 22 shows the evolution of the average number of habitats for each ecological succession step for 30 executions of the ECO-C algorithm. As the observed behavior is the same for all functions, Figure 22 shows the evolution for the *Schaffer* function. It can be observed that, at the beginning of the optimization process, with the populations widely dispersed in the search space, there is a greater number of habitats. To the extent that the optimization process moves through the ecological successions, the populations tend to move through the search space converging to specific regions. As shown in Figure 22, the number of habitats decreases with the ecological succession cycles, indicating that the populations tend to converge to points close to each other.

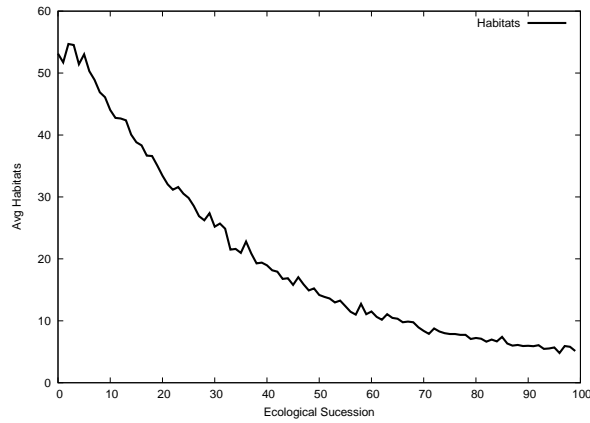
Important to notice in Figure 22 is the presence of a transient state in earlier ecological successions (successions one to seven). The transient state shows the self-adaptation process carried out by the hierarchical clustering strategy for formation of habitats. The self-adaptation



**Figure 21: Case study IV: Box plot for the results obtained ( $D = 200$ ).**

process persists through all ecological successions and, as demonstrated in the results obtained, it directs the search to promising regions.

Table 13 shows the mean of the elapsed time in minutes by each configuration for each function, followed by the respective standard deviations. It is possible to notice that the ECO approach using hierarchical clustering achieved practically the same processing times when compared with its application without using this feature. This encourages even more the use of this strategy inside the ECO framework.



**Figure 22: Case study IV: Average number of habitats at each ecological succession.**

**Table 13: Case study IV: Mean of the processing time over all runs (minutes).**

	$D = 200$	
	$ECO_{ABC}$	$ECO-C_{ABC}$
<i>Schaffer</i>	$1.14 \pm 0.01$	$1.13 \pm 0.01$
<i>Rastrigin</i>	$1.27 \pm 0.01$	$1.27 \pm 0.00$
<i>Griewank</i>	$1.29 \pm 0.01$	$1.31 \pm 0.01$
<i>Rosenbrock</i>	$1.31 \pm 0.01$	$1.27 \pm 0.01$

#### 4.7.2 CONSIDERATIONS

In this case study a hierarchical clustering strategy to probabilistically setup the distribution of populations into habitats was used. A more biologically plausible definition of habitats is achieved by using probabilistically the distance information returned by the single-link clustering algorithm.

The addition of this feature (the probabilistic definition of habitats) brought a higher biological plausibility to the proposed algorithm, opposed to most bio-inspired algorithms that take inspiration only from one biological phenomenon.

Also, besides suppressing the proximity threshold  $\rho$ , the use of a probabilistic strategy for habitats definition allowed the system to self-adapt in the search for best regions in the space of solutions. Moreover, this self-adaptation is made during the optimization process.

#### 4.8 CASE STUDY V: PROTEIN FOLDING PROBLEM

Aiming at applying the ecology-based approach to a real world complex problem, the purpose of this case study is to search low energy conformations for the PSP AB off-lattice model (Section 4.2).

The experiments were conducted using the benchmark sequences shown in Section 4.2, Table 5.

In order to enrich the discussion, the ecological-inspired algorithm (ECO) was tested using six configurations. All configurations implement the Algorithm 7 as described in Section 3.3, with the definitions of habitats using hierarchical clustering, as presented in Case study IV. The first configuration,  $ECO_{ABC}$ , employs the ABC algorithm (Section 2.2.1) homogeneously. The second configuration,  $ECO_{PSO}$ , employs the PSO algorithm (Section 2.2.2) homogeneously. The third configuration,  $ECO_{DE}$ , employs the DE algorithm (Section 2.2.3) homogeneously. The fourth configuration,  $ECO_{jDE/BBO}$ , employs the BBO algorithm (Section 2.2.4) hybrid with DE (GONG et al., 2010), also homogeneously. The fifth configuration,  $ECO_{All}$ , employs a heterogeneous approach combining all four algorithms in with 1/4 of the populations behaves according to one of these strategies. The sixth configuration,  $ECO_{All-LM}$ , acts heterogeneously as the fifth configuration and adds the population resizing feature. For each configuration, the algorithm was run 30 times.

Also, the overall best solutions are compared with other results found in literature.

#### 4.8.1 RESULTS AND ANALYSIS

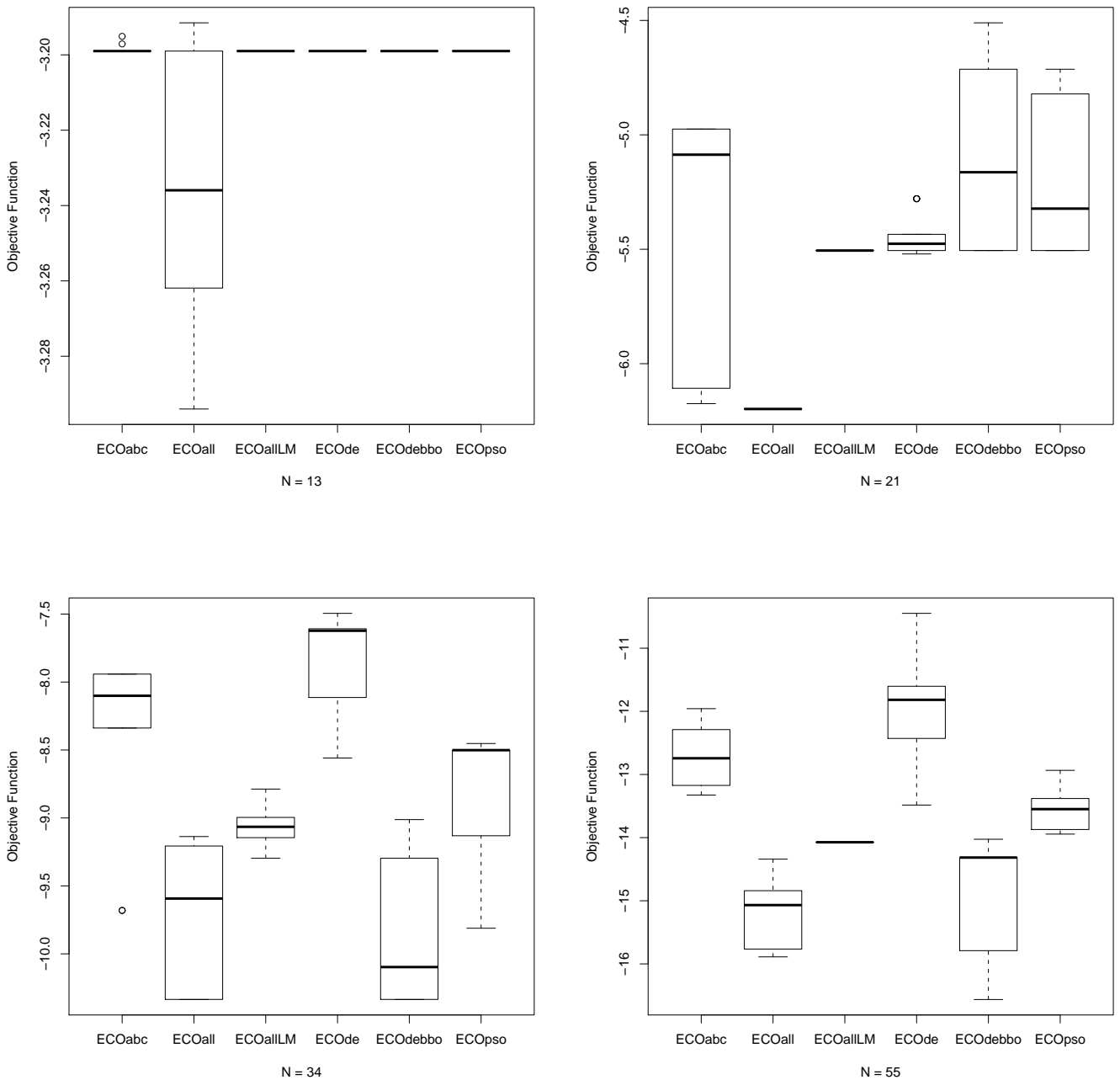
Table 14 shows the results obtained for the benchmark sequences using the ecological approaches. In this table, the first column identifies the sizes  $N$  of amino acids sequences and the remaining columns show the average and standard deviation obtained by each configuration followed by its best result. In bold the best result obtained for each sequence are shown.

**Table 14: Case study V: Quality of solutions using ECO approaches.**

$ECO_{ABC}$			$ECO_{PSO}$		$ECO_{DE}$	
N	Avg	Best	Avg	Best	Avg	Best
13	-3.1987±0.0010	-3.1990	-3.1990±0.0	-3.1990	-3.1990±0.0	-3.1990
21	-5.3743 ±0.5065	-6.1747	-5.1850 ±0.3569	-5.5056	-5.4402±0.0932	-5.5205
34	-8.2718 ±0.5404	-9.6805	-8.7419±0.4610	-9.8114	-7.8561±0.3671	-8.5590
55	-12.7603±0.5019	-13.3262	-13.5588±0.3304	-13.9440	-11.9394±0.8278	-13.4844
$ECO_{jDE/BBO}$			$ECO_{All}$		$ECO_{All-LM}$	
N	Avg	Best	Avg	Best	Avg	Best
13	-3.1990±0.0	-3.1990	-3.2352±0.0356	<b>-3.2940</b>	-3.1990±0.0	-3.1990
21	-5.1049±0.4205	-5.5056	-6.1980±0.0	<b>-6.1980</b>	-5.5056±0.0	-5.5056
34	-9.8464±0.4861	<b>-10.3360</b>	-9.7185±0.5121	<b>-10.3360</b>	-9.0593±0.1633	-9.2965
55	-14.9310±0.9884	<b>-16.5641</b>	-15.1982±0.5089	-15.8887	-14.0732±0.0	-14.0732

With a close look in Table 14 and in the box-plot shown in Figure 23, considering first the 13 amino acids sequence, a  $p$ -value $_K$  of  $10^{-10}$  is obtained when comparing all approaches

indicating that, with a significance level of 5%, one of them is statistically different from the others. From the box-plot, it is possible to notice that this difference is provided by the  $ECO_{All}$  approach. For the 21 amino acids sequence,  $ECO_{All}$  achieved the best results with statistical relevance (see box-plot information).



**Figure 23: Case study V: Box plot for the results obtained.**

Also, analysing the box-plot it is possible to notice that some statistical inference is needed for the two larger sequences. For both sequences it is necessary to verify the relevance of the results between  $ECO_{jDE/BBO}$  and  $ECO_{All}$ . For 34 and 55 amino acids sequences, both



$ECO_{jDE/BBO}$  and  $ECO_{All}$  achieved statistically equivalent results with a  $p\text{-value}_W$  of 0.2671 and 0.6236, respectively, at a significance level of 5%.

When using the heterogeneous model ( $ECO_{All}$ ) for sequences 13, 21, and 34, the optimization process took advantage of using different intensification and diversification strategies provided by different search algorithms. However, although  $ECO_{All}$  has had the best average result for the largest sequence of 55 amino acids, the  $ECO_{jDE/BBO}$  achieved the best result. Overall, the best results are obtained by  $ECO_{jDE/BBO}$  and  $ECO_{All}$  approaches.

Although the results of using different search strategies have already shown to be promising, with this observation was realized that the use of different algorithms can be better explored if using some source of feedback from the optimization process during its course. The main concern is to use this heuristic information to better distribute the habitats formation and to better define the intra and inter habitats communication topologies. We have a strong insight that this can be done using the information contained in the hierarchical clustering procedure. We believe that biasing the dendrogram weights it should be possible to achieve different probabilities for habitats formation and communication topologies definition during the optimization process. This analysis is quoted as future research.

Concerning the results obtained by  $ECO_{All-LM}$  approach it is possible to verify that, despite its competitive results when compared with  $ECO_{ABC}$ ,  $ECO_{PSO}$ , and  $ECO_{DE}$ , the use of a population resizing procedure stifled the benefits of using different search strategies. This can be noticed when comparing the results obtained between  $ECO_{All-LM}$  and  $ECO_{All}$ . For all sequences,  $ECO_{All-LM}$  obtained worst results. A possible research direction would be to apply the resizing procedure not to all populations but only to those that achieve some criteria (e.g., stagnation for a predefined period or loss of diversity).

Applying the Friedman rank test to Table 14, the following rank is obtained:  $ECO_{All}$  with 1.2500;  $ECO_{All-LM}$  with 2.6250;  $ECO_{jDE/BBO}$  with 2.8750;  $ECO_{PSO}$  with 3.8750;  $ECO_{DE}$  with 4.3750; and  $ECO_{ABC}$  with 5.0000. As lower the ranking better the algorithm is, it is possible to conclude that the  $ECO_{All}$  approach works very well in these sequences highlighting the beneficial effect of using the heterogeneous model.

Table 15 shows the mean of the elapsed time in hours by each configuration for each sequence, followed by the respective standard deviations. The processing time employed by the  $ECO_{All}$  approach is around the average elapsed time employed by the other strategies. Also, the times shown in this table encourages the use of massive parallel strategies inside the ECO framework (e.g., using cluster of computers or GPU).

**Table 15: Case study V: Mean of the processing time over all runs (hours).**

$N$	$ECO_{ABC}$	$ECO_{PSO}$	$ECO_{DE}$	$ECO_{jDE/BBO}$	$ECO_{All}$	$ECO_{All-LM}$
13	$1.45 \pm 0.00$	$3.42 \pm 0.01$	$2.00 \pm 0.01$	$5.36 \pm 0.02$	$2.40 \pm 0.00$	$2.49 \pm 0.00$
21	$3.82 \pm 0.00$	$5.34 \pm 0.02$	$4.22 \pm 0.03$	$6.83 \pm 0.05$	$4.40 \pm 0.00$	$4.62 \pm 0.01$
34	$10.15 \pm 0.01$	$11.59 \pm 0.02$	$10.73 \pm 0.05$	$15.01 \pm 0.09$	$10.74 \pm 0.08$	$11.09 \pm 0.00$
55	$26.95 \pm 0.02$	$29.23 \pm 0.02$	$27.82 \pm 0.02$	$32.09 \pm 0.18$	$27.80 \pm 0.23$	$28.81 \pm 0.01$

#### 4.8.2 COMPARISON WITH OTHER APPROACHES

Table 16 shows the lowest energies obtained by the best ECO approaches along with the lowest energies obtained by other works using different methods. In this table,  $E_{PERM}$  is a pruned-enriched Rosenbluth method – PERM (HSU et al., 2003);  $E_{min}$  is the minimum energy obtained by the same method with subsequent conjugate gradient minimisation (HSU et al., 2003);  $E_{ground}$  is the putative ground state energy obtained by Stillinger and Head-Gordon using a Monte Carlo method hybridised with Newtonian conjugate gradient minimisation (STILLINGER; HEAD-GORDON, 1995); PBHS is the best result obtained by a population-based Harmony Search algorithm (SCALABRIN et al., 2013); DE-RI is the best result obtained by a Differential Evolutionary algorithm with a ring-island configuration (KALEGARI; LOPES, 2010); DEadp is the lowest energy obtained using a Differential Evolution algorithm with self-adaptation of the  $F$  parameter and with other improvements (KALEGARI, 2010); ACMC is the lowest energy obtained using an Annealing Contour Monte Carlo Method (LIANG, 2004); CSA shows the energy using a Conformational Space Annealing approach (KIM et al., 2005);  $ELP_+$  is the lowest energy obtained using the improved energy landscaping paving method (LIU et al., 2009). In bold the best results found in literature are shown.

One of the most challenging tasks when comparing different algorithms is to perform a fair comparison between them. To the best of our knowledge, a good base for comparison is to compare the results obtained by the algorithms using the same computational effort (e.g., processing time or number of function evaluations). Here the function evaluations criteria was used. Thus, it is worth remembering that for all ECO approaches the number of function evaluations was set to five millions (Section 4.3). That is the same number of function evaluations used for PBHS and DE-RI algorithms. For DEadp, the number of function evaluations was set to 35 milions that is a lot more than what was applied in the experiments. For the other methods no other information about the computational effort employed to obtain the results was found. Also, for all ECO approaches, all algorithms employed are in its canonical versions.

From Table 16 it can be seen that the results obtained by the proposed approach are

**Table 16: Case study V: Comparative of best solutions from different strategies with best ECO solutions ( $ECO_{Best}$ ).**

N	$E_{PERM}$	$E_{min}$	$E_{ground}$	PBHS	DE-RI	DEadp	ACMC	CSA	ELP <sub>+</sub>	$ECO_{Best}$
13	-3.2167	-3.2939	-3.2235	-3.28	-3.2924	-3.1999	<b>-3.2941</b>	<b>-3.2941</b>	<b>-3.2941</b>	-3.2940
21	-5.7501	-6.1976	-5.2881	-5.96	-6.1979	<b>-6.1980</b>	-6,1979	<b>-6.1980</b>	<b>-6.1980</b>	<b>-6.1980</b>
34	-9.2195	-10.7001	-8.9749	-8.33	-9.6838	-10.5565	<b>-10.8060</b>	<b>-10.8060</b>	-10.7453	-10.3360
55	-14.9050	-18.5154	-14.4089	-11.51	-14.6847	-17.3133	-18.7407	-18.9110	<b>-18.9301</b>	-16.5641

better than those of the  $E_{PERM}$ ,  $E_{ground}$ , PBHS, and DE-RI for all four sequences, with the energy difference increasing gradually for longer chains. For sequences with length 13 and 21, the results obtained by the eco-approach were slightly better than that of  $E_{min}$  and competitive with the ACMC, CSA, and ELP<sub>+</sub> results. For other cases, 34 and 55 length sequences, however, the proposed approach cannot reach the energy yielded by  $E_{min}$ , ACMC, CSA and ELP<sub>+</sub>.

Applying the Friedman rank test to the overall best solutions shown in Table 16, the following rank is obtained: CSA with 1.6250; ELP<sub>+</sub> with 1.7500; ACMC with 2.8750;  $ECO_{Best}$  with 4.3750;  $E_{min}$  with 5.0000; DEadp with 5.3750; DE-RI with 6.6250;  $E_{PERM}$  with 8.2500; PBHS with 8.7500; and  $E_{ground}$  with 9.0000. As lower the ranking better the algorithm, the results suggest that the CSA approach is the best one and gives the overall best solutions for all sequences.

The closest approach to the proposed method is the DE-RI, which employs a Differential Evolutionary algorithm with a ring-island configuration (sixth column of Table 16). From the results and from the Friedman rank test, it can be noticed that the eco-approach works very well when compared with the DE-RI approach.

Table 17 shows a comparison of the best solutions found in literature ( $E^*$ ) with the best ECO solutions ( $ECO_{Best}$ ). The fourth column ( $Diff_{ECO_{Best} \times E^*}$ ) shows the percentual difference between  $E^*$  and  $ECO_{Best}$ . For the first sequence the difference observed is minimal, almost null. For the sequence of 21 amino acids the difference is null. For the two large sequences the differences are more accentuated, with 4.3494% and 12.4986%, respectively.

**Table 17: Case study V: Comparative of overall best solutions found in literature ( $E^*$ ) with best ECO solutions ( $ECO_{Best}$ ).**

N	$E^*$	$ECO_{Best}$	$Diff_{ECO_{Best} \times E^*}$
13	-3.2941	-3.2940	0.0018%
21	-6.1980	-6.1980	0.0000%
34	-10.8060	-10.3360	4.3494%
55	-18.9301	-16.5641	12.4986%

To evaluate visually the quality of the foldings produced by the ECO approaches,

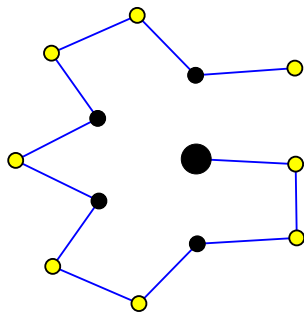
the best results shown in Table 16 were used to draw the planar form of the sequence (conformation). A program in MATLAB was developed to convert the string of angles into  $(x, y)$  coordinates and plot the structure. The larger dot represents the start of the sequence, black dots represent 'A' monomers and the yellow dots represent 'B' monomers. Recall that the energy of the folding is a function of the proximity of monomers, especially the 'A' monomers. Therefore, compact structures tend to have lower energy levels than those structures more dispersed. Figure 24 shows the best foldings obtained with the ECO implementations for sequences with 13, 21, 34, and 55 monomers.

From Figure 24 it is possible to see that the hydrophobic A monomers tend to form a hydrophobic core in the 13 amino acids sequence or clusters of typically 3-5 monomers in other sequences. This can be explained by the fact that hydrophobic monomers are always flanked by the hydrophilic monomers along the sequence. This shows that the AB off-lattice model reflects the native characters of the real proteins in two-dimensions but it still is not perfect.

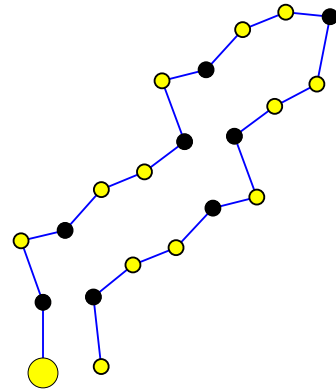
#### 4.8.3 CONSIDERATIONS

This case study applied the ecology-based approach in the PSP AB off-lattice model. Six configurations of the ECO approach were configured and, when using the heterogeneous model ( $ECO_{All}$ ) the search process gets more robust than the other approaches (e.g., for sequences 13, 21, and 34) possibly due the use of different intensification and diversification strategies provided by different search algorithms. Also,  $ECO_{All}$  has had the best average result for the large sequence of 55 amino acids.

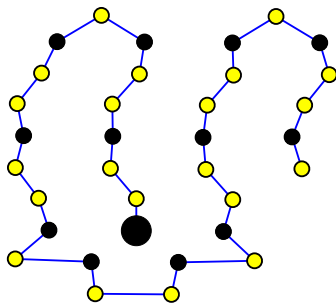
It is possible to highlight that the use of different algorithms can be better explored if using some source of feedback from the optimization process during its course. This analysis is pointed as future research.



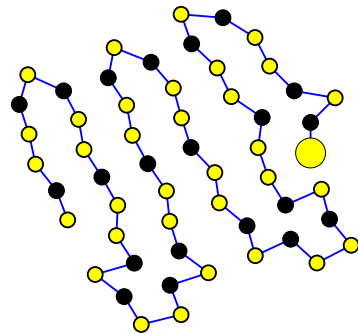
(a) Best folding for  $N = 13$  ( $\text{ECO}_{All}$ ).



(b) Best folding for  $N = 21$  ( $\text{ECO}_{All}$ ).



(c) Best folding for  $N = 34$  ( $\text{ECO}_{jDE/BBO}$  e  $\text{ECO}_{All}$ ).



(d) Best folding for  $N = 55$  ( $\text{ECO}_{jDE/BBO}$ ).

**Figure 24: Case study V: Best states found by ECO approaches listed in Table 16.**



## 5 CONCLUSIONS

Typical bio-inspired systems are influenced by different aspects of biological phenomena. Also, most of them focus only on and take inspiration from isolated aspects of such phenomena. However, in nature, biological systems are interlinked to each other, e.g., biological ecosystems.

Also, the “curse-of-dimensionality” affecting bio-inspired algorithms leads to the necessity of developing more robust and efficient search strategies to better explore promising regions in a search space. Hence, hybrid bio-inspired systems (HBS) takes place and are both developed and defined by cooperative search concepts. In this thesis a classification of the HBS was proposed: bio-plausible HBS and engineered HBS. The main difference between these two classes is the presence or not of some degree of biological plausibility.

This thesis presents a computational model relating the cooperative use of populations of candidate solutions, coevolving in an ecological context. With this ecology-based analogy, each population can behave according to a specific search strategy, employed in the evolution of candidate solutions. In addition to the possibility of using different optimization strategies cooperatively, this analogy opens the possibility of inserting ecological concepts in the optimization process, thus allowing the development of new bio-plausible HBS. In this way, the basis of a computational ecosystem for optimization and a canonical ecology-inspired framework (ECO) are presented.

Some population-based algorithms, namely, ABC, PSO, DE, jDE/BBO, are used to compose the ecology-based approach. The problems solved in this thesis are some continuous benchmark functions with a high number of dimensions and the protein structure prediction problem for the 2D AB model.

In addition to the mechanisms of intensification and diversification specific to each search strategy, the proposed computational ecosystem provides a new ecological level for information exchange. This is carried through the use of intra and inter-habitats relationships, respectively. From the results, it is possible to observe that the use of this new level of

coevolution through ecological interactions is beneficial in the search for best solutions. Moreover, coevolution only occurs if some sort of symbiosis is present in the optimization process. In fact, in the presence of symbiosis, natural selection produces coevolution. Thus, the proposed new level of ecological interactions favours coevolution.

A hierarchical clustering strategy to setup probabilistically the distribution of populations into habitats was also applied. A more biologically plausible definition of habitats is achieved by using probabilistically the distance information returned by the single-link clustering algorithm. Besides suppressing the proximity threshold  $\rho$ , the use of a probabilistic strategy for habitats definition allowed the system to self-adapt during the search in the space of solutions.

Also, the use of different search strategies during the optimization process inside the ECO framework (heterogeneous application) obtained better results than the homogeneous application of ECO. Although the heterogeneous approach have shown to be promising, it is believed that the use of different algorithms can be even better explored if using some source of heuristic information to better distribute the habitats formation and to better define the communication topologies.

The use of population dynamics to self-regulate the size of populations during ecological successions is also investigated. In addition to the exploration and exploitation routines provided by the evolution of populations and by the ecological interactions (inter and intra-habitats), the use of population dynamics creates a new biologically plausible mechanism to diversify the search. Although the use of population dynamics inside the ECO framework improved the results in most cases, we realize that its use should be viewed with caution when applied to real world complex problems, as discussed in Case study V.

Overall, the results obtained are promising concerning the application of the proposed computational ecosystem for optimization problems. However, due the ‘no free lunch’ theorem, it is not possible to generalize this analysis to all optimization problems (WOLPERT; MACREADY, 1997).

Although some features were experimented inside the ECO framework, many other computational issues (Section 3.2) can be approached to bring more biological plausibility to the system (e.g., speciation, other ecological relationships, environmental factors, ...). Important to highlight that adding new biologically plausible features to the ECO framework can be or cannot be useful to the optimization process. Only large-scale experiments can determine their usefulness.

As its main drawback, one can mention its high number of user-defined parameters



(number of populations ( $NQ$ ), population size ( $POP$ ), number of cycles for ecological successions ( $ECO-STEP$ ), the size of the evolutive period ( $EVO-STEP$ ), and the tournament size ( $T-SIZE$ )) in addition to the parameters of each search strategy employed.

Finally, the description of a computational ecosystem presented in this thesis does not accomplish the whole complexity of a real ecosystem but shows some potential directions to develop new bio-plausible hybrid systems.

## 5.1 CONTRIBUTIONS

During the development of this thesis, several subjects were investigated and some of them have generated scientific publications leading to some relevant contributions. In each study, the results obtained (both experimental and conceptual) helped to define with greater certainty the paths to be chosen and also helped to give a more theoretical and technical background for future developments.

With the study done in (PARPINELLI; LOPES, 2011b) and in (KRAUSE et al., 2013) was possible to verify a growing number of new optimization algorithms, in particular in the field of Swarm Intelligence for both continuous and discrete domains, respectively. Also discussing new algorithms and applications, in (PARPINELLI; LOPES, 2012d) a book was edited concerning some recent advances in Swarm Intelligence, specially related to new swarm-based optimization methods and hybrid algorithms for several applications.

With such variety of optimization algorithms, each one applying its own algorithmic particularity in search the space of solutions to a given problem, we noticed the possibility of joining them in a cooperative way. At this point the research began drawing a parallel with the dynamics of biological ecosystems, where each population behaves according to a specific meta-heuristic and the ecosystem as a whole would be composed of populations responding to environmental and ecological stimuli. This is the key inspiration for this thesis and it is described in Section 3.2 and in the work done in (PARPINELLI; LOPES, 2012e). Thus, a solid theoretical foundation for the design of more plausible biologically inspired systems for optimization inspired by concepts and processes involved in real ecosystems were presented. Also, hybrid bio-inspired systems (HBS) were categorized into bio-plausible HBS and engineered HBS.

In the work of (OLIVEIRA et al., 2011) a new algorithm for continuous optimization inspired by the bioluminescent behavior of fireflies was proposed with the same algorithmic components of the Algorithm 1 shown in Section 2.1.1. The proposed algorithm,

*Bioluminescent Swarm Optimization algorithm* (BSO), was applied to four continuous benchmark functions. The results obtained were compared with the results obtained by a standard PSO. The main feature of the proposed BSO algorithm is the ability to explore the search space smoothly, avoiding getting stuck in local points. This characteristic reflects in the results by showing superiority compared to the results obtained by the standard PSO. We did not use this algorithm in the experiments of this thesis due technical problems concerning the BSO source code but we intend to use it in future experiments.

Still working with the proposal of algorithms, in the work of (TEODORO et al., 2010) improvements to the Bacterial Foraging Optimization (BFO) algorithm were proposed. By combining features of *swarming* (S) and *self-adaptation* (A), this work presents results for different versions of the BFO algorithm, namely: BFO-S, BFO-A e BFO-SA. From the results, it was found that, in general, the variant BFO-A presents solutions of better quality than the others. Again, we did not use this algorithm in the experiments of this thesis due to the same reasons of BSO.

Continuing the previous work, a paper comparing the performance of three swarm intelligence algorithms for the optimization of hard engineering problems was presented in (PARPINELLI et al., 2012). The algorithms tested were BFO, PSO, and ABC. The algorithms were evaluated using two criteria: quality of solutions and the number of function evaluations. The results show that PSO presented the best balance between these two criteria. For the optimization problems approached in this work, we also concluded that the explosion procedure resulted in no significant improvements.

In another work comparing different algorithms, the performance of four swarm intelligence algorithms were evaluated for the optimization of the PSP (PARPINELLI, R. S.; BENÍTEZ, C. M. V.; CORDEIRO, J.; LOPES, H. S., 2013). In this work, we tested the standard versions of the following algorithms: PSO, ABC, Gravitational Search Algorithm (GSA), and the Bat Algorithm (BA). The algorithms were evaluated using two criteria: quality of solutions and the processing time. The results show that the PSO algorithm presented the overall best balance between these two criteria. Also, both PSO and GSA displayed potential to evolve even better solutions, if more iterations were given.

As this thesis deals with the formalization of a new cooperative search approach, a technical background was done considering some parallelization technologies. Although this subject was not experimentally approached in the ECO framework, the aim of such deepening is to explore the intrinsic parallelism and asynchronism present in biological systems in future researches. Thus, in the work of (SCALABRIN et al., 2010) a parallel approach to the *Harmony*

*Search* algorithm (HS) was proposed using the CUDA parallel programming architecture (*Compute Unified Device Architecture*) in a Graphic Processing Unit (GPU). The HS algorithm was modified in order to implement it in the parallel architecture. The results show that the running time of the HS algorithm using GPU is significantly less when compared with CPU execution time, both with the same quality of solutions. It was observed that the influence of the number of variables on the running time is less significant on GPU than CPU. It was also observed that the higher the complexity, the greater the speed-up provided by the use of GPU.

In another work, (SCALABRIN et al., 2013), a new evolutionary algorithm based on the standard Harmony Search strategy, called population-based Harmony Search (PBHS) was presented. Also, this work provided a parallelization method for the proposed PBHS on GPU, allowing multiple function evaluations at the same time. Results have shown that the quality of solutions and speed-ups achieved by the PBHS are significantly better than the HS.

In (PARPINELLI et al., 2011) parallelization strategies were investigated for the *Artificial Bee Colony Algorithm* (ABC). In addition to the sequential versions of the algorithm with and without the addition of local search, three other parallel models were compared: master-slave approach that divides the processing load across multiple processors; multi-hive approach which promotes seasonal migrations between independent populations; and hierarchical approach that hybridizes the two previous models. All parallel models were developed using the MPICH2 library<sup>10</sup>. The results indicated that the local search routine improves the quality of solutions and, due to the coevolutionary effect, the multi-hive and hierarchical approaches obtained better results with lower computational effort (function evaluations).

Another work exploring the hierarchical model of parallel computing is presented in (BENÍTEZ et al., 2012). In this work, the hierarchical model was used in an architecture that employs different meta-heuristics in different islands of the parallel model. The goal is to verify the coevolutionary effect when using search strategies with different mechanisms of intensification and diversification. The meta-heuristics employed in this work are the Genetic Algorithms and the Artificial Bee Colony (ABC) algorithm. The results were compared with those obtained by (BENÍTEZ; LOPES, 2010) that applied to the same problem a hierarchical model with ring topology composed only by the ABC algorithm. The results showed that a combination of approaches with different search strategies, in the hierarchical model, outperformed the results obtained by the approach consisting only by the ABC algorithm.

In two other publications, the canonical ecology-inspired algorithm for optimization

---

<sup>10</sup>MPICH2: [www.mcs.anl.gov/research/projects/mpich2/](http://www.mcs.anl.gov/research/projects/mpich2/)

presented in Section 3.3 was first applied (PARPINELLI; LOPES, 2011a)(PARPINELLI; LOPES, 2011c). The description of these works are presented in Case study I (Section 4.4). The main contribution of these works is the insertion of a new ecological level to balance between intensification and diversification (intra and inter-habitats relationships).

In (PARPINELLI; LOPES, 2012c), Case study II (Section 4.5), the population sizing not as a parameter but as a dynamic process that changes deterministically over time was explored.

In another work, (PARPINELLI; LOPES, 2012a), the heterogeneity of the ecology-inspired approach was explored using two different algorithms cooperatively: the ABC and the PSO. The description of this work is presented in the Case study III (Section 4.6).

In (PARPINELLI; LOPES, 2012b) the use of a hierarchical clustering technique was explored as a biologically plausible strategy to probabilistically set the habitats of the computational ecosystem. Discussions of this work are presented in Case study IV (Section 4.7).

## 5.2 FUTURE RESEARCH

There are several research directions for future developments. Some of them are:

- Understand the relationship between parameters of ECO and define default values;
- Take into account strategies to self-adapt the parameters, aiming at reducing the amount of user defined parameters. The use of Cultural Algorithms and Learning Automata are possible directions;
- Diversify the evolutive behaviors of the computational ecosystem inserting other algorithms (e.g., Tabu Search, Genetic Algorithm, Simulated Annealing, Variable Neighbour Search, Hill Climbing, Covariance Matrix Adaptation Evolution Strategy, ...);
- Use feedback from the optimization process during its course to better distribute the habitats formation and to better define the intra and inter habitats communication topologies. This could be achieved using the heuristic information from the hierarchical clustering procedure or other KDD strategies to aid the self-organization process of the system;
- In order to increase the computational capabilities of the computational ecosystem for optimization, parallel architectures such as clusters of computers and graphical processing

units can be naturally explored due the intrinsic parallelism and asynchronism present in ecological systems;

- Increase the biological plausibility of the system adding other ecological concepts into the ECO framework;
- Apply the proposed ECO framework in other real optimization problems in both continuous and discrete domains;
- Consider the application in dynamic problems and in multi-objective problems;
- Apply other population dynamics models such as the Lotka-Volterra predator-prey model;
- Define strategies and metrics for maintaining the diversity of solutions both at micro and macro levels concerning the ecosystemic context. In this item mechanisms to maintain the gradual emergence of new traits and behaviors will be defined in order to keep emerging continually smooth evolutionary gradients;
- Adapt the system to handle individual level dynamics and relationships.



## REFERENCES

- AKAY, B.; KARABOGA, D. Parameter tuning for the artificial bee colony algorithm. In: NGUYEN, N. T.; KOWALCZYK, R.; CHEN, S.-M. (Ed.). **Proceedings of the 1st International Conference on Computational Collective Intelligence - Semantic Web, Social Networks & Multiagent Systems**. Wrocław, Poland: Springer, 2009. (Lecture Notes in Computer Science, v. 5796), p. 608–619.
- ALBA, E.; TOMASSINI, M. Parallelism and evolutionary algorithms. **IEEE Transactions on Evolutionary Computation**, v. 6, n. 5, p. 443–462, 2002.
- ALBA, E.; TROYA, J. M. A survey of parallel distributed genetic algorithms. **Complexity**, v. 4, n. 4, p. 31–52, 1999.
- ALBERTS, B. et al. **Molecular Biology of The Cell**. 4. ed. New York, USA: Garland Science, 2002.
- ATKINS, J.; HART, W. On the intractability of protein folding with a finite alphabet. **Algorithmica**, v. 25, n. 2-3, p. 279–294, 1999.
- BEGON, M.; TOWNSEND, C. R.; HARPER, J. L. **Ecology: from individuals to ecosystems**. 4th. ed. Oxford, UK: Blackwell Publishing, 2006.
- BELLMAN, R. E. **Dynamic Programming**. 1st. ed. Princeton, NY: Princeton University Press, 2003.
- BENÍTEZ, C.; LOPES, H. Hierarchical parallel genetic algorithm applied to the three-dimensional HP side-chain protein folding problem. In: **Proceedings of the IEEE International Conference on Systems, Man and Cybernetics**. Washington, USA: IEEE Computer Society, 2010. p. 2669–2676.
- BENÍTEZ, C.; LOPES, H. Protein structure prediction with the 3D-HP side-chain model using a master-slave parallel genetic algorithm. **Journal of the Brazilian Computer Society**, v. 16, n. 1, p. 69–78, 2010.
- BENÍTEZ, C. M. V.; PARPINELLI, R. S.; LOPES, H. S. Parallelism, hybridism and coevolution in a multi-level ABC-GA approach for the protein structure prediction problem. **Concurrency and Computation: Practice and Experience**, v. 24, n. 6, p. 635–646, 2012.
- BERGER, B.; LEIGHTON, F. Protein folding in the hydrophobic-hydrophilic (HP) model is NP-complete. **Journal of Computational Biology**, v. 5, n. 1, p. 27–40, 1998.
- BERMAN, H. et al. The protein data bank. **Nucleic Acids Research**, v. 28, n. 1, p. 235–242, 2000.
- BISWAS, A. et al. Synergy of PSO and bacterial foraging optimization - a comparative study on numerical benchmarks. In: CORCHADO, E.; CORCHADO, J. M.; ABRAHAM, A. (Ed.). **Innovations in Hybrid Intelligent Systems**. Berlin, Heidelberg: Springer Verlag, 2007, (Advances in Soft Computing, v. 44). p. 255–263.

- BLICKLE, T. Tournament selection. In: BACK, T.; FOGEL, D.; MICHALEWICZ, Z. (Ed.). **Evolutionary Computation 1: Basic Algorithms and Operators**. Bristol, UK: Institute of Physics Publishing, 2000. p. 181–186.
- BLUM, C.; ROLI, A. Metaheuristics in combinatorial optimization: overview and conceptual comparison. **ACM Computing Surveys**, v. 35, n. 3, p. 268–308, 2003.
- BONABEAU, E.; DORIGO, M.; THERAULAZ, G. **Swarm intelligence: from natural to artificial systems**. Oxford, UK: Oxford University Press, 1999.
- BORGULYA, I. An island model for the no-wait flow shop scheduling problem. In: SCHAEFER, R. et al. (Ed.). **Proceedings of the 11th International Conference on Parallel Problem Solving from Nature: Part II**. Kraków, Poland: Springer-Verlag, 2010. (Lecture Notes in Computer Science, v. 6239), p. 280–289.
- BRISCOE, G.; de WILDE, P. Digital ecosystems: Optimisation by a distributed intelligence. In: **Proceedings of IEEE International Conference on Digital Ecosystems and Technologies**. Phitsanulok, Thailand: IEEE Press, 2008. p. 192–197.
- BRISCOE, G.; de WILDE, P. Computing of applied digital ecosystems. In: **Proceedings of the International Conference on Management of Emergent Digital EcoSystems**. Lyon, France: ACM Press, 2009. p. 28–35.
- BROWNLEE, J. **Complex Adaptive Systems**. Melbourne, Australia: Swinburne University of Technology Press, 2007.
- CADENAS, J. M.; GARRIDO, M. C.; MUÑOZ, E. A hybrid system of nature inspired metaheuristics. In: KRASNOGOR, N. et al. (Ed.). **Nature Inspired Cooperative Strategies for Optimization**. Berlin Heidelberg: Springer, 2007, (Studies in Computational Intelligence, v. 129). p. 95–104.
- CADENAS, J. M.; GARRIDO, M. C.; MUÑOZ, E. Using machine learning in a cooperative hybrid parallel strategy of metaheuristics. **Information Sciences**, v. 179, n. 19, p. 3255–3267, 2009.
- CASTRO, P. A. D.; VON ZUBEN, F. J. Training multilayer perceptrons with a Gaussian artificial immune system. **Proceedings of IEEE Congress on Evolutionary Computation - CEC**, IEEE Press, New Orleans, LA, USA, p. 1250 – 1257, 2011.
- CHO, H.; OLIVERA, F.; GUIKEMA, S. A derivation of the number of minima of the Griewank function. **Applied Mathematics and Computation**, v. 204, n. 2, p. 694–701, 2008.
- CLERC, M. **Particle Swarm Optimization**. London, UK: ISTE Press, 2006.
- COOPER, G. **The Cell: A Molecular Approach**. Sunderland, UK: Sinauer Associates, 2000.
- CRESCENZI, P. et al. On the complexity of protein folding. **Journal of Computational Biology**, v. 5, n. 3, p. 423–446, 1998.
- DALGAARD, P. **Introductory statistics with R**. 2. ed. New York, NY, USA: Springer, 2008.
- DASGUPTA, D.; YU, S.; NIÑO, F. Review article: Recent advances in artificial immune systems: Models and applications. **Applied Soft Computing**, v. 11, n. 2, p. 1574–1587, 2011.



- DE CASTRO, L. N. Fundamentals of natural computing: an overview. **Physics of Life Reviews**, v. 4, n. 1, p. 1–36, 2007.
- DERRAC, J. et al. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. **Swarm and Evolutionary Computation**, v. 1, n. 1, p. 3–18, 2011.
- DIGALAKIS, J. G.; MARGARITIS, K. G. An experimental study of benchmarking functions for evolutionary algorithms. **International Journal of Computer Mathematics**, v. 79, n. 4, p. 403–416, 2002.
- DILL, K. Polymer principles and protein folding. **Protein Science**, v. 8, n. 6, p. 1166–1180, 1999.
- DILL, K. et al. Principles of protein folding - a perspective from simple exact models. **Protein Science**, v. 4, n. 4, p. 561–602, 1995.
- DORIGO, M.; STÜTZLE, T. **Ant Colony Optimization**. Cambridge, MA, USA: MIT Press, 2004.
- DOUGLAS, A. E. **The Symbiotic Habit**. Princeton, New Jersey: Princeton University Press, 2010.
- DRÉO, J. et al. **Metaheuristics for Hard Optimization: Methods and Case Studies**. Berlin, Heidelberg: Springer, 2006.
- EIBEN, A.; SMITH, J. **Introduction to Evolutionary Computing**. London, UK: Springer-Verlag, 2003.
- EL-ABD, M.; KAMEL, M. A taxonomy of cooperative search algorithms. In: **Proceedings of the Second international conference on Hybrid Metaheuristics**. Berlin, Heidelberg: Springer-Verlag, 2005, (Lecture Notes in Computer Science, v. 3636). p. 32–41.
- ENGELBRECHT, A. P. **Computational Intelligence: An Introduction**. 2nd. ed. Chichester, UK: John Wiley & Sons, 2007.
- FERREIRA, C. Gene expression programming: a new adaptive algorithm for solving problems. **Complex Systems**, v. 13, n. 2, p. 87–129, 2001.
- FLOUDAS, C. A.; PARDALOS, P. M. **A collection of test problems for constrained global optimization problems**. Berlin, Heidelberg: Springer, 1990. (Lecture Notes in Computer Science, v. 455).
- FOGEL, D. B. **Evolutionary Computation: Toward a New Philosophy of Machine Intelligence**. 3rd. ed. Piscataway, NJ: IEEE Press, 2006.
- GANGULY, N. et al. **A Survey on Cellular Automata**. Dresden, Germany, 2003.
- GARNIER, S.; GAUTRAIS, J.; THERAULAZ, G. The biological principles of swarm intelligence. **Swarm Intelligence**, v. 1, n. 1, p. 3–31, 2007.
- GENDREAU, M. An introduction to tabu search. In: GLOVER, F. W.; KOCHENBERGER, G. A. (Ed.). **Handbook of Metaheuristics: International Series in Operations Research and Management Science**. Norwell, USA: Kluwer, 2003. p. 37–54.

GHALAMBAZ, M. et al. A hybrid neural network and gravitational search algorithm (HNNGSA) method to solve well known Wessinger's equation. **World Academy of Science, Engineering and Technology**, WASET, Las Cruces, USA, v. 49, p. 803–807, 2011.

GLOVER, F.; KOCHENBERGER, G. **Handbook of Metaheuristics**. Hingham, MA, USA: Kluwer Academic Publishers, 2003. (International Series in Operations Research & Management Science).

GLOVER, F. W. Future paths for integer programming and links to artificial intelligence. **Computers and Operations Research**, v. 13, n. 5, p. 533–549, 1986.

GOLDBERG, D. E. **Genetic Algorithms in Search, Optimization, and Machine Learning**. Reading, MA, USA: Addison-Wesley, 1989.

GONG, W.; CAI, Z.; LING, C. X. DE/BBO: a hybrid differential evolution with biogeography-based optimization for global numerical optimization. **Soft Computing**, v. 15, n. 4, p. 645–665, 2010.

GRIEWANK, A. Generalized descent for global optimization. **Journal of Optimization Theory and Applications**, v. 34, n. 1, p. 11–39, 1981.

HARDIN, C.; POGORELOV, T.; LUTHEY-SCHULTEN, Z. *Ab-initio* protein structure prediction. **Current Opinion in Structural Biology**, v. 12, n. 2, p. 176–181, 2002.

HENDERSON, D.; JACOBSON, S. H.; JOHNSON, A. W. The theory and practice of simulated annealing. In: GLOVER, F. W.; KOCHENBERGER, G. A. (Ed.). **Handbook of Metaheuristics: International Series in Operations Research and Management Science**. Norwell, USA: Kluwer, 2003. p. 287–319.

HSU, H.; MEHRA, V.; GRASSBERGER, P. Structure optimization in an off-lattice protein model. **Physical Review E**, v. 68, n. 2, p. 1–4, 2003.

HUNTER, L. **Artificial Intelligence and Molecular Biology**. 1. ed. Boston, USA: AAAI Press, 1993.

INTHACHOT, M.; SUPRATID, S. A multi-subpopulation particle swarm optimization: A hybrid intelligent computing for function optimization. **Proceedings of International Conference on Natural Computation**, IEEE Computer Society, Los Alamitos, CA, USA, v. 5, p. 679–684, 2007.

JAIN, A. K.; MURTY, M. N.; FLYNN, P. J. Data clustering: a review. **ACM Computing Surveys**, ACM, New York, NY, USA, v. 31, n. 3, p. 264–323, 1999.

JOVANOVIĆ, R.; TUBA, M.; SIMIAN, D. Comparison of different topologies for island-based multi-colony ant algorithms for the minimum weight vertex cover problem. **WSEAS Transactions on Computers**, v. 9, n. 1, p. 83–92, 2010.

KALEGARI, D. H. **Algoritmo de evolução diferencial paralelo aplicado ao problema da predição da estrutura de proteínas utilizando o modelo AB em 2D e 3D**. Universidade Tecnológica Federal do Paraná: CPGEI, 2010.

KALEGARI, D. H.; LOPES, H. S. A differential evolution approach for protein structure optimisation using a 2D off-lattice model. **International Journal of Bio-Inspired Computation**, v. 2, n. 3/4, p. 242–250, 2010.

- KAPLAN, D.; GLASS, L. **Understanding Nonlinear Dynamics**. New York, USA: Springer-Verlag, 1995.
- KARABOGA, D. **An idea based on honey bee swarm for numerical optimization**. Kayseri, Turkey: Erciyes University, Engineering Faculty, Computer Engineering Department, 2005.
- KARABOGA, D.; AKAY, B. A comparative study of artificial bee colony algorithm. **Applied Mathematics and Computation**, v. 214, n. 1, p. 108–132, 2009.
- KARABOGA, D.; AKAY, B.; OZTURK, C. Artificial bee colony (ABC) optimization algorithm for training feed-forward neural networks. In: TORRA, V.; NARUKAWA, Y.; YOSHIDA, Y. (Ed.). **Proceedings of the 4th International Conference on Modeling Decisions for Artificial Intelligence**. Berlin, Heidelberg: Springer-Verlag, 2007. (MDAI '07, 1), p. 318–329.
- KARI, J. Theory of cellular automata: A survey. **Theoretical Computer Science**, v. 334, n. 1-3, p. 3–33, 2005.
- KENNEDY, J.; EBERHART, R. **Swarm Intelligence**. San Francisco, CA, USA: Morgan Kaufmann, 2001.
- KIM, S.-Y.; LEE, S. B.; LEE, J. Structure optimization by conformational space annealing in an off-lattice protein model. **Physical Review E**, v. 72, n. 1, p. 011916, 2005.
- KNIDEL, H.; CASTRO, L. N. de; VON ZUBEN, F. J. Data clustering with a neuro-immune network. In: WANG, L.; CHEN, K.; ONG, Y.-S. (Ed.). **Proceedings of International Conference on Natural Computation**. Berlin, Heidelberg: Springer, 2005. (Lecture Notes in Computer Science, v. 3610), p. 1279–1288.
- KOZA, J. R. **Genetic Programming**. Cambridge, MA, USA: MIT Press, 1992.
- KRAUSE, J. et al. A survey on swarm algorithms applied to discrete problems. In: YANG, X.-S. et al. (Ed.). **Swarm Intelligence and Bio-inspired Computation**. New York, NY, USA: Elsevier, 2013. To appear.
- KRINK, T.; LOVBJERG, M. The lifecycle model: combining particle swarm optimisation, genetic algorithms and hillclimbers. In: GUERVÓS, J. J. M. et al. (Ed.). **Parallel Problem Solving From Nature**. Berlin, Heidelberg: Springer, 2002. (Lecture Notes in Computer Science, v. 2439), p. 621–630.
- KRINK, T.; THOMSEN, R. Self-organized criticality and mass extinction in evolutionary algorithms. In: **Proceedings of the 2001 Congress on Evolutionary Computation**. Piscataway, NJ, USA: IEEE Press, 2001. p. 1155–1161.
- LEGENDRE, P.; LEGENDRE, L. **Numerical ecology**. Amsterdam, The Netherlands: Elsevier, 1998.
- LEVIN, S. A. Ecosystems and the biosphere as complex adaptive systems. **Ecosystems**, v. 1, n. 5, p. 431–436, 1998.
- LIANG, F. Annealing contour monte carlo algorithm for structure optimization in an off-lattice protein model. **Journal of Chemical Physics**, American Institute of Physics, Pennsylvania, USA, v. 120, n. 14, p. 6756–63, 2004.

- LIU, J. et al. Structure optimization of the two-dimensional off-lattice hydrophobic-hydrophilic model. **Journal of Biological Physics**, Springer, Netherlands, v. 35, n. 3, p. 245–253, 2009.
- LIU, Y.; TIMMIS, J.; CLARKE, T. A neuro-immune inspired robust real time visual tracking system. In: **Proceedings of the Seventh International Conference on Artificial Immune Systems**. Berlin, Heidelberg: Springer-Verlag, 2008. (ICARIS '08), p. 188–199.
- LODISH, H. et al. **Molecular Cell Biology**. 4. ed. New York, NY, USA: Freeman, 2000.
- LOPES, H. Evolutionary algorithms for the protein folding problem: A review and current trends. In: SMOLINSKI, T.; MILANOVA, M.; HASSANIEN, A.-E. (Ed.). **Computational Intelligence in Biomedicine and Bioinformatics**. Berlin, Heidelberg: Springer-Verlag, 2008. v. 1, n. 1, p. 297–315.
- LOPES, H. S.; COELHO, L. S. . Particle swarm optimization with fast local search for the blind traveling salesman problem. In: NEDJAH, N. et al. (Ed.). **Proceedings of the Fifth International Conference on Hybrid Intelligent Systems**. Los Alamitos, CA, USA: IEEE Computer Society, 2005. p. 245–250.
- LUNG, R. I.; DUMITRESCU, D. Evolutionary swarm cooperative optimization in dynamic environments. **Natural Computing**, v. 9, n. 1, p. 83–94, 2010.
- LUONG, T. V.; MELAB, N.; TALBI, E.-G. GPU-based island model for evolutionary algorithms. In: BRANKE, J. et al. (Ed.). **Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation**. New York, NY, USA: ACM, 2010. p. 1089–1096.
- MA, Y. et al. Improvement on parallel AQPSO using the best position. In: **Proceedings of the 2009 Second International Workshop on Knowledge Discovery and Data Mining**. Washington, DC, USA: IEEE Computer Society, 2009. p. 825–828.
- MA, Z. S. Ecological theatre for evolutionary computing play: some insights from population ecology and evolutionary ecology. **International Journal of Bio-Inspired Computation**, v. 3, n. 1, p. 31–55, 2011.
- MALIK, N. Artificial neural networks and their applications. In: **Conference on Unearthing Technological Developments and Their Transfer to Serving Masses**. Mathura, India: ITM, 2005.
- MASEGOSA, A. D. et al. On the performance of homogeneous and heterogeneous cooperative search strategies. In: KRASNOGOR, N. et al. (Ed.). **International Workshop on Nature Inspired Cooperative Strategies for Optimization**. Berlin, Heidelberg: Springer, 2008. (Studies in Computational Intelligence, v. 236), p. 287–300.
- MAY, R. M. C.; MCLEAN, A. R. **Theoretical Ecology: Principles and Applications**. Oxford, UK: Oxford University Press, 2007.
- MEIRELLES, M. et al. Diversity-based adaptive evolutionary algorithms. In: KOROSEC, P. (Ed.). **New Achievements in Evolutionary Computation**. Rijeka, Croatia: InTech, 2010. p. 1–16.
- MOSCATO, P. A gentle introduction to memetic algorithms. In: **Handbook of Metaheuristics**. Hingham, MA, USA: Kluwer Academic Publishers, 2003. p. 105–144.

- MURTAGH, F.; CONTRERAS, P. Algorithms for hierarchical clustering: an overview. **Data Mining and Knowledge Discovery**, v. 2, n. 1, p. 86–97, 2012.
- MÜHLENBEIN, H.; SCHOMISCH, D.; BORN, J. The parallel genetic algorithm as function optimizer. **Parallel Computing**, v. 17, n. 6-7, p. 619–632, 1991.
- NERI, F.; TIRRONEN, V. Recent advances in differential evolution: a survey and experimental analysis. **Artificial Intelligence Review**, Kluwer Academic Publishers, Norwell, MA, USA, v. 33, n. 1-2, p. 61–106, 2010.
- NISHIDA, T. Y. An approximate algorithm for NP-complete optimization problems exploiting P systems. In: **Proceedings of the Brainstorming Workshop on Uncertainty in Membrane Computing**. Palma de Mallorca, Spain: UIB Press, 2004. p. 185–192.
- NISHIDA, T. Y. Membrane algorithms. In: **Proceedings of International Workshop on Membrane Computing**. Heidelberg, Germany: Springer-Verlag, 2006. (Lecture Notes in Computer Science, v. 3850), p. 55–66.
- NISHIDA, T. Y. Membrane algorithm with Brownian subalgorithm and genetic subalgorithm. **International Journal of Foundations of Computer Science**, v. 18, n. 6, p. 1353–1360, 2007.
- NISHIDA, T. Y.; SHIOTANI, T.; TAKAHASHI, Y. Membrane algorithm solving job-shop scheduling problems. In: FRISCO, P.; CORNE, D. W.; PAUN, G. (Ed.). **Proceedings of International Workshop on Membrane Computing, Lecture Notes in Computer Science**. New York, NY, USA: Elsevier, 2009. p. 363–370.
- NOLTE, A.; SCHRADER, R. A note on the finite time behaviour of simulated annealing. **Mathematics of Operations Research**, v. 25, n. 3, p. 476–484, 2000.
- NOWOSTAWSKI, M.; POLI, R. Parallel genetic algorithm taxonomy. In: JAIN, L. C. (Ed.). **Proceedings of the International Conference on Knowledge-Based Intelligent Information Engineering Systems**. Piscataway, NJ, USA: IEEE Press, 1999. p. 88–92.
- OLIVEIRA, D. R.; PARPINELLI, R. S.; LOPES, H. S. Bioluminescent swarm optimization algorithm. In: KITA, E. (Ed.). **Evolutionary Algorithms**. Rijeka, Croatia: InTech, 2011. p. 69–84.
- OLOFSSON, P. **Probability, Statistics, and Stochastic Processes**. New York, NY: Wiley-Interscience, 2005.
- OSMAN, I.; LAPORTE, G. Metaheuristics: A bibliography. **Annals of Operations Research**, v. 63, n. 5, p. 511–623, 1996.
- PARPINELLI, R. S.; BENÍTEZ, C. M. V.; LOPES, H. S. Parallel approaches for the artificial bee colony algorithm. In: PANIGRANI, B. K.; SHI, Y.; LIM, M. (Ed.). **Handbook of Swarm Intelligence: Concepts, Principles and Applications. Series: Adaptation, Learning, and Optimization**. Berlin, Heidelberg: Springer, 2011. p. 329–346.
- PARPINELLI, R. S.; LOPES, H. S. An eco-inspired evolutionary algorithm applied to numerical optimization. In: **Proceedings of the Third World Congress on Nature and Biologically Inspired Computing**. Piscataway, NJ, USA: IEEE Press, 2011a. p. 473–478.

PARPINELLI, R. S.; LOPES, H. S. New inspirations in swarm intelligence: a survey. **International Journal of Bio-Inspired Computation**, v. 3, n. 1, p. 1–16, 2011b.

PARPINELLI, R. S.; LOPES, H. S. Um algoritmo evolutivo eco-inspirado aplicado a otimização numérica. In: **Anais do X Congresso Brasileiro de Inteligência Computacional**. Fortaleza, Brazil: SBC, 2011c. p. 1–8.

PARPINELLI, R. S.; LOPES, H. S. An ecology-based heterogeneous approach for cooperative search. In: BARROS, L.; FINGER, M.; POZO, A. (Ed.). **Proceedings of the Brazilian Symposium on Artificial Intelligence (SBIA)**. Berlin, Heidelberg: Springer, 2012a. (Lecture Notes in Computer Science, 1), p. 212–221.

PARPINELLI, R. S.; LOPES, H. S. A hierarchical clustering strategy to improve the biological plausibility of an ecology-based evolutionary algorithm. In: PAVÓN, J. et al. (Ed.). **Ibero-American Conference on Artificial Intelligence (IBERAMIA)**. Berlin, Heidelberg: Springer-Verlag, 2012b. (Lecture Notes in Computer Science/Lecture Notes in Artificial Intelligence, 1), p. 310–319.

PARPINELLI, R. S.; LOPES, H. S. Population resizing using nonlinear dynamics in an ecology-based approach. In: YIN, H.; COSTA, J. A. F.; BARRETO, G. D. A. (Ed.). **Proceedings of the Intelligent Data Engineering and Automated Learning**. Berlin, Heidelberg: Springer, 2012c. (Lecture Notes in Computer Science, 1), p. 27–34.

PARPINELLI, R. S.; LOPES, H. S. **Theory and New Applications of Swarm Intelligence**. Rijeka, Croatia: InTech, 2012d.

PARPINELLI, R. S.; LOPES, H. S. Biological plausibility in optimization: An ecosystemic view. **International Journal of Bio-Inspired Computation**, v. 4, p. 345–358, 2012e.

PARPINELLI, R. S.; TEODORO, F. R.; LOPES, H. S. A comparison of swarm intelligence algorithms for structural engineering optimization. **International Journal for Numerical Methods in Engineering**, v. 91, n. 6, p. 666–684, 2012.

PARPINELLI, R. S.; BENÍTEZ, C. M. V.; CORDEIRO, J.; LOPES, H. S. Performance analysis of swarm intelligence algorithms for the 3D-AB off-lattice protein folding problem. **International Journal of Computational and Theoretical Nanoscience**, 2013. To appear.

PASTI, R.; de CASTRO, L. N.; von ZUBEN, F. J. Computação biogeográfica: Formalização e proposição de uma meta-heurística espaço-temporal. **Anais do XVIII Congresso Brasileiro de Automática**, p. 5065–5072, 2010.

PAUN, G. Computing with membranes. **Journal of Computer and System Sciences**, v. 61, n. 1, p. 108–143, 2000.

PAUN, G.; PEREZ-JIMENEZ, M. J. Membrane computing: brief introduction, recent results and applications. **Biosystems**, v. 85, n. 1, p. 11–22, 2006.

PIOTROWSKI, A. P.; NAPIORKOWSKI, J. J.; KICZKO, A. Differential evolution algorithm with separated groups for multi-dimensional optimization problems. **European Journal of Operational Research**, v. 216, n. 2, p. 33–46, 2012.

- PLAGIANAKOS, V.; TASOULIS, D.; VRAHATIS, M. A review of major application areas of differential evolution. In: CHAKRABORTY, U. (Ed.). **Advances in Differential Evolution**. Berlin Heidelberg: Springer, 2008, (Studies in Computational Intelligence, v. 143). p. 197–238.
- POLI, R. Analysis of the publications on the applications of particle swarm optimisation. **Journal of Artificial Evolution and Applications**, v. 1, n. 4, p. 1–10, 2008.
- POLI, R.; KENNEDY, J.; BLACKWELL, T. Particle swarm optimization: an overview. **Swarm Intelligence**, v. 1, n. 1, p. 33–57, 2007.
- REEVES, C. R. **Modern Heuristic Techniques For Combinatorial Problems**. London, UK: McGraw-Hill, 1995.
- ROSENBROCK, H. An automatic method for finding the greatest or least value of a function. **The Computer Journal**, v. 3, n. 3, p. 175–184, 1960.
- SCALABRIN, M. H. et al. Population-based harmony search using GPU applied to protein structure prediction. **International Journal of Computational Science and Engineering**, 2013. To appear.
- SCALABRIN, M. H.; PARPINELLI, R. S.; LOPES, H. S. Paralelização do algoritmo harmony search utilizando unidade de processamento gráfico. **Proceedings of XXIX Ibero Latin American Conference on Mecânica Computacional**, n. 72, p. 7109–7121, 2010.
- SCHUTTE, J. F. et al. Parallel global optimization with the particle swarm algorithm. **Journal of Numerical Methods in Engineering**, v. 61, p. 2296–2315, 2003.
- SHANG, Y.-W.; QIU, Y.-H. A note on the extended Rosenbrock function. **Evolutionary Computation**, v. 14, n. 1, p. 119–126, 2006.
- SIMON, D. Biogeography-based optimization. **IEEE Transactions on Evolutionary Computation**, v. 12, n. 6, p. 702–713, 2008.
- SOCHA, K.; BLUM, C. An ant colony optimization algorithm for continuous optimization: application to feed-forward neural network training. **Neural Computing and Applications**, v. 16, n. 3, p. 235–247, 2007.
- STILLINGER, F.; HEAD-GORDON, T. Collective aspects of protein folding illustrated by a toy model. **Physical Review E**, v. 52, n. 3, p. 2872–2877, 1995.
- STORN, R.; PRICE, K. Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. **Journal of Global Optimization**, v. 11, n. 4, p. 341–359, dez. 1997.
- SUN, L. et al. A cooperative particle swarm optimizer with statistical variable interdependence learning. **Information Sciences**, v. 186, n. 1, p. 20–39, 2012.
- TEODORO, F. R.; PARPINELLI, R. S.; LOPES, H. S. Otimização por colônia de bactérias aplicada a problemas de engenharia. **Proceedings of XXIX Ibero Latin American Conference on Mecânica Computacional**, n. 98, p. 9615–9627, 2010.
- TEUSCHER, C. From membranes to systems: self-configuration and self-replication in membrane systems. **Biosystems**, v. 87, n. 2-3, p. 101 – 110, 2007.

TOMASSINI, M. Parallel and distributed evolutionary algorithms: A review. In: MIETTINEN, K. et al. (Ed.). **Evolutionary Algorithms in Engineering and Computer Science**. Chichester, UK: John Wiley & Sons, 1999.

TOMASSINI, M. **Spatially Structured Evolutionary Algorithms: Artificial Evolution in Space and Time**. Berlin, Heidelberg: Springer, 2005.

VANNESCHI, L.; CODECASA, D.; MAURI, G. A study of parallel and distributed particle swarm optimization methods. In: **Proceedings of the 2nd Workshop on Bio-inspired Algorithms for Distributed Systems**. New York, NY, USA: ACM, 2010. p. 9–16.

VULLI, S. S.; AGARWAL, S. Individual-based artificial ecosystems for design and optimization. In: KEIJZER, M. (Ed.). **Proceedings of the Conference on Genetic and Evolutionary Computation**. Atlanta, USA: ACM, 2008. p. 273–280.

WANG, Y. et al. Ecosystem model based grid resource optimization management. **International Journal of Computer Science and Network Security**, v. 7, n. 2, p. 61–66, 2007.

WATADA, J. DNA computing and its application. In: FULCHER, J.; JAIN, L. C. (Ed.). **Computational Intelligence: A Compendium**. Berlin, Heidelberg: Springer, 2008, (Studies in Computational Intelligence, v. 115). p. 1065–1089.

WHITE, C. M.; YEN, G. G. A hybrid evolutionary algorithm for traveling salesman problem. **Proceedings of the IEEE Congress on Evolutionary Computation - Stillwater, OK, USA**, IEEE Press, Piscataway, NJ, USA, v. 2, p. 1473–1478, 2005.

WOLPERT, D.; MACREADY, W. No free lunch theorems for optimization. **IEEE Transactions on Evolutionary Computation**, v. 1, n. 1, p. 67–82, 1997.

XU, R.; WUNSCH, D. Survey of clustering algorithms. **IEEE Transactions on Neural Networks**, v. 16, n. 3, p. 645–678, 2005.

YANG, X. S. **Nature-inspired Metaheuristic Algorithms**. 2nd. ed. Frome, UK: Luniver Press, 2010.