

*A computational ecosystem for
optimization: review and perspectives for
future research*

**Rafael Stubs Parpinelli & Heitor Silvério
Lopes**

Memetic Computing

ISSN 1865-9284

Volume 7

Number 1

Memetic Comp. (2015) 7:29-41

DOI 10.1007/s12293-014-0148-4



Your article is protected by copyright and all rights are held exclusively by Springer-Verlag Berlin Heidelberg. This e-offprint is for personal use only and shall not be self-archived in electronic repositories. If you wish to self-archive your article, please use the accepted manuscript version for posting on your own website. You may further deposit the accepted manuscript version in any repository, provided it is only made publicly available 12 months after official publication or later and provided acknowledgement is given to the original source of publication and a link is inserted to the published article on Springer's website. The link must be accompanied by the following text: "The final publication is available at link.springer.com".

A computational ecosystem for optimization: review and perspectives for future research

Rafael Stubs Parpinelli · Heitor Silvério Lopes

Received: 28 February 2014 / Accepted: 15 November 2014 / Published online: 7 December 2014
 © Springer-Verlag Berlin Heidelberg 2014

Abstract Nature exhibits extremely diverse, dynamic, robust, complex and fascinating phenomena and, since long ago, it has been a great source of inspiration for solving hard and complex problems in computer science. Hence, the search for plausible biologically inspired ideas, models and computational paradigms always drew the interest of computer scientists. It is worth mentioning that most bio-inspired algorithms only focuses 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. The ecosystem as a whole can be composed by species that respond to environmental and ecological stimuli. This work reviews the theoretical foundations and applications of a computational ecosystem for optimization, named ECO. Also, as some concepts and processes inherent to biological ecosystems have already been explored in the ECO approach, some related works are described. Finally, several future research directions are pointed.

Keywords Optimization · Cooperative search · Co-evolution · Ecosystems · Ecology · Computational ecosystem

1 Introduction

Problem solving methodologies involve two branches: exact methods and (meta-)heuristic methods. Metaheuristic approaches have proven to be efficient in solving hard and complex optimization problems, particularly where traditional methods fail. Bio-inspired algorithms are such metaheuristics that mimics/imitate the strategy of nature since many biological processes can be thought of as processes of optimization.

It is known that Nature is an endless source of inspiration for computational models and paradigms [5]. Bio-inspired computing in general have attracted great interest in almost every area of science, engineering, and industry over the last decades. In fact, the mid-2000s turned out to be a plentiful of bio-inspired algorithms.

The careful observation of the behavior of some living beings can give us insights on how to map their natural behavior into algorithmic routines, e.g., metaheuristic algorithms. Basically, these global optimization metaheuristics are composed by a selection (of the best) scheme and by a randomization scheme. The former hopefully guides the algorithm to converge to improved solutions (exploitation) and the later avoids both the loss of diversity and the algorithm to get trapped in local maxima (exploration). A good balance between exploitation and exploration may lead to the achievement of global optimality.

Some examples of natural processes that inspired computational methods are: the evolution of species, the multicellular development of organisms, the animal nervous system, the immunological system in vertebrates, the social behavior of insects, and the ecological relationships between populations. Evolutionary algorithms [13], cellular automata [15,21], artificial neural networks [29], artificial immune systems [10], and swarm algorithms [24,38,51] are some of

R. S. Parpinelli (✉)
 Graduate Program in Applied Computing, State University of Santa Catarina, Joinville, Brazil
 e-mail: rafael.parpinelli@udesc.br

H. S. Lopes
 Bioinformatics Laboratory, Federal Technological University of Paraná, Curitiba, Brazil
 e-mail: hslopes@utfpr.edu.br

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 [50].

It is worth mentioning that most bio-inspired algorithms only focuses 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 [12]. 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 [3, 18, 27, 30, 36, 47].

Taking into account the 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) [31]. 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 multiple environmental and ecological stimuli. Also, interactions and information exchanges between species can favor co-evolution. Co-evolution can be seen as a complementary form of evolution where two (or more) species reciprocally affect each other's evolution through symbiotic relationships [31].

An ecological framework for optimization, named ECO, was first presented in [39] and is discussed in this paper. Also, some other works [4, 37, 40–43] using the ECO framework are briefly analyzed.

The main discussion presented here is related to the cooperative use of populations of candidate solutions, co-evolving 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 into the optimization process, thus allowing the development of new biologically plausible hybrid systems [39].

This paper reviews and highlights the main achievements using an ecological framework for optimization and points promising research directions in the field of metaheuristics

concerning its application. Therefore, the paper is structured as follows. Section 2 presents some related works. Section 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. Section 4 shows several perspectives for future research. Finally, Sect. 5 presents final considerations.

2 Related works

The term Computational Ecosystem has been previously used in different contexts of applications. For example, in the context of computational ecology for simulating the behavior of real ecosystems [16, 23], and in the context of multi-agent systems for simulating the artificial life of virtual agents [1]. In this work, the context is focused on the use of computational ecosystems for optimization of complex problems.

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 paper.

In [49] was 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 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 resources allocation. This model proposes the use of Knowledge Discovery in Databases (KDD) strategies to aid the self-organization process of the system. The discovered knowledge can be used to predict the resource requirements and, thus, to optimize the allocation. However, in the work cited, no experiments were performed with the proposed model.

In both papers, [6, 7], 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 the movement through the connection, affecting

migration decisions. These probabilities are updated according to the success rate of migrating agents. An agent represents an 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 (represented by agents) in their respective habitats specifying a desired service or application. Next, a population is instantiated in the user's habitat in response to open requests and is fed by available agents in the habitat. The population then begins an evolution process of the agents with their respective semantic descriptions, in order to meet the user's request. In this model, a process based on a genetic algorithm performs a combinatorial search of the space of the possible software services available to the users. 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 texture synthesis problem in binary images [48]. Given a binary texture, the goal is to find the optimal set of parameters of a Markov Random Field (MRF) capable of generating the input texture. The texture whose parameters should be found is mapped into the environment. The parameters to be optimized are mapped as evolutionary features of the prey. Thus, each prey is born with a texture that camouflages itself in the environment. Every prey evolve at each iteration and have a life cycle such that 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 [28], 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.

The maintenance of diversity at all ecological levels is critical to the development and evolution of an ecosystem. The work of [32] presents several metrics to quantify the diversity of populations and species. The main idea is to generate foundations to develop methods for controlling the population diversity loss. However, in this work, ways to measure and maintain diversity in an ecosystemic context, considering multiple populations, are not addressed.

In [45] the Biogeography-based Optimization (BBO) was proposed. Biogeography deals with the aspects of living beings associated with its geographic and spatial distribution. This subject seeks to explain why the living organisms 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 [44] 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}). 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 multi-modal functions. The algorithm was applied to the optimization of a multi-modal 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 the previous works, it can be noticed 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. On the other hand, in the present work we discuss the application of an ecological framework for computational optimization inspired by biological ecosystems. It opens the possibility to develop new optimization systems with biologically plausible inspirations, and is presented next.

3 Computational ecosystem for optimization

A computational ecosystem for optimization is composed by candidate solutions (individuals) scattered in an environment that, itself, is the search space defined by an objective function $f(.)$ and its constraints which represents a hypersurface 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. A given set of candidate solutions define a population

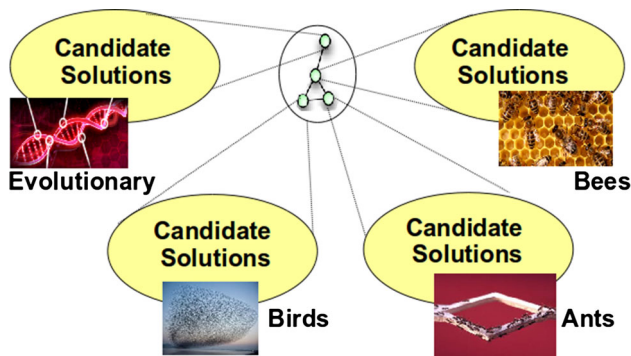


Fig. 1 Possible representation for the elements of a computational ecosystem. Four populations with different behaviors. Adapted from [39]

of the ecosystem [39]. The computational ecosystem can be composed of several populations that can interact to each other.

Figure 1 shows a possible representation for the elements of the proposed computational ecosystem. This figure shows four populations where each population behaves according to the mechanisms of intensification and diversification, tuned by the control parameters, specific to an optimization strategy. In this example, the behavior of individuals is driven by the survival of the fittest evolutionary mechanisms, the foraging strategies of bees, the foraging strategies of ants, and by the flocking behavior of birds. As Fig. 1 shows, the computational ecosystem can be composed of several search strategies. Concerning this diversity, the model can be classified as homogeneous or heterogeneous. 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. Hence, Fig. 1 exemplifies a heterogeneous model.

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 Ant Colony Optimization algorithm is not suitable for continuous problems as well as a canonical Particle Swarm Optimization algorithm (PSO) is not suitable for combinatorial problems. In other words, either canonical or not, all search strategies must be able to handle the problem features.

Another representation is shown in Fig. 2. The lower level of the figure illustrates an environment defined by a function $f(\cdot)$ which describes the hyper-surface of the search space. In this example, the hyper-surface has two dimensions only for visualization purposes.

In the intermediate level of Fig. 2, 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

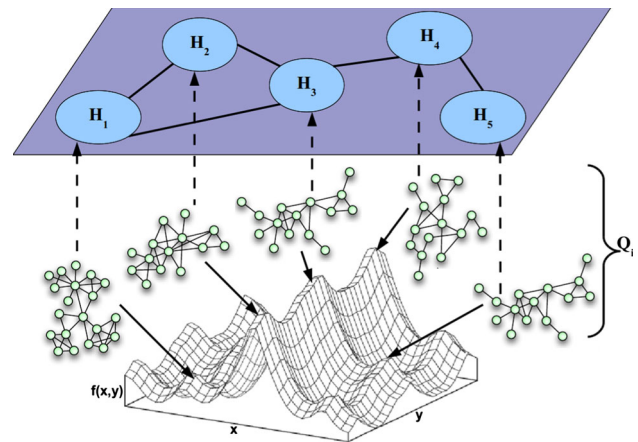


Fig. 2 Generical view of a computational ecosystem for optimization. *Lower level* problem-dependent search space that defines a hyper-surface. *Intermediate level* intra-habitats communication topologies where each small circle represents a population. *Upper level* five habitats connected through inter-habitats communication topology. Adapted from [39]

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 degree of variability of the organisms in the ecosystem, that is, the biodiversity, 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 Fig. 2.

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 [2,31].

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 Fig. 2 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 Fig. 2.

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 ecological relationships or ecological interactions define the ways in which individuals interact. In this way, symbiosis occurs and it can be defined by a relationship between two individuals where one individual directly affects the other individual [2].

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.

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

The maintenance of an assortment 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 (evolutionary or swarm intelligence algorithm, for instance), when considering the ecological context, the computational ecosystem provides 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 creating another diversity maintenance mechanism for the system.

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 [2]. 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.

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.

Figure 3 depicts the elements that compose the ECO framework and they will be further detailed in Sect. 4.

In order to better understand the basis of the proposal, in next section a conceptual illustration of a canonical ECO algorithm is presented.

3.1 Conceptual illustration

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 succes-

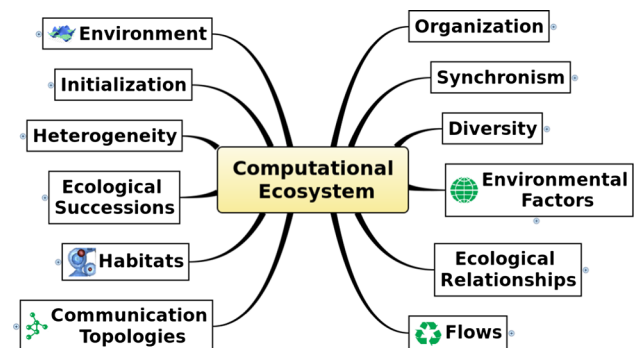


Fig. 3 The elements of a computational ecosystem

sions (*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 ρ used to define the habitats.

The generalized Schaffer function f was chosen to illustrate in details the behavior of the proposed algorithm [14]. Its definition is presented in Eq. 1 where $\mathbf{x}$ is a solution vector defined inside $[-100, 100]^n$ and the global optimum for $f(\mathbf{x})$ is 0, corresponding to the optimum solution $\mathbf{x}_{opt} = (x_1, x_2, \dots, x_n) = (0, 0, \dots, 0)$.

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

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 Fig. 4a, b.

The parameters used were: $NQ = 10$, $POP = 10$, *ECO-STEP* = 100, *EVO-STEP* = 100, *T-SIZE* = 5 and $\rho = 0.5$. 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 [20] 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.

First of all, all populations are randomly initiated. This initialization uses a normal distribution ($N(\mu, \sigma)$) with both average (μ) and standard deviation (σ) randomly chosen within the domain of each dimension of the problem. However, other distributions could be also explored.

Figure 5a shows the initial distribution of the individuals of all NQ populations: $Sp0, \dots, Sp9$. Also, each population has *POP* candidate solutions.

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 (*EVO-STEP*) was defined as 100 function evaluations.

After each evolutive period, the definition of habitats occurs. The habitats are the regions in which the populations are concentrated. In this illustration, the region of reference of a population is defined by its centroid and it is calculated by Eq. 2. Figure 5b illustrates the distribution of all centroids of all populations for the first ecological succession.

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

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 ρ . The adjacency matrix is generated according to ρ . Two populations are called adjacent among themselves if they are at a distance of at least ρ one of another. In Fig. 5b 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.

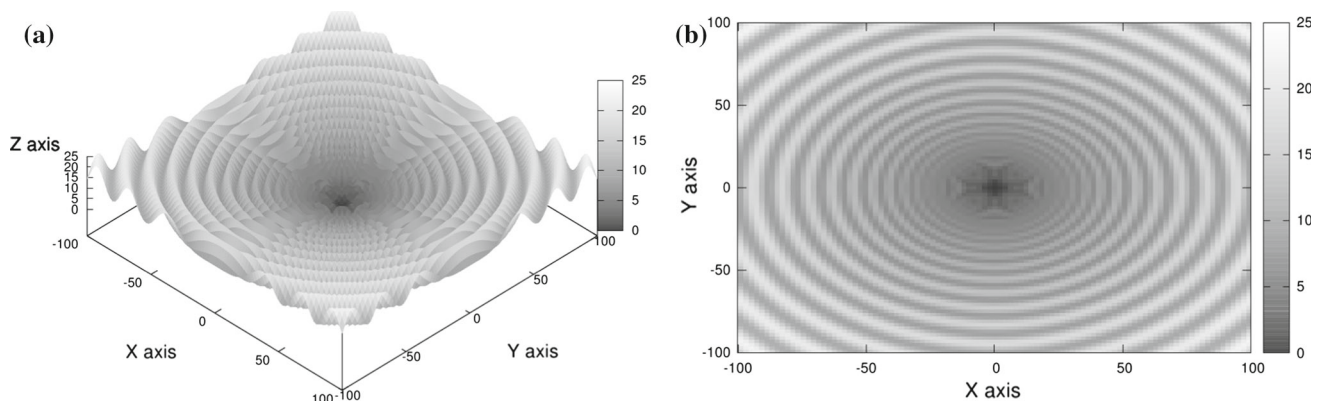


Fig. 4 2D Schaffer function. **a** Side view. **b** Upper view

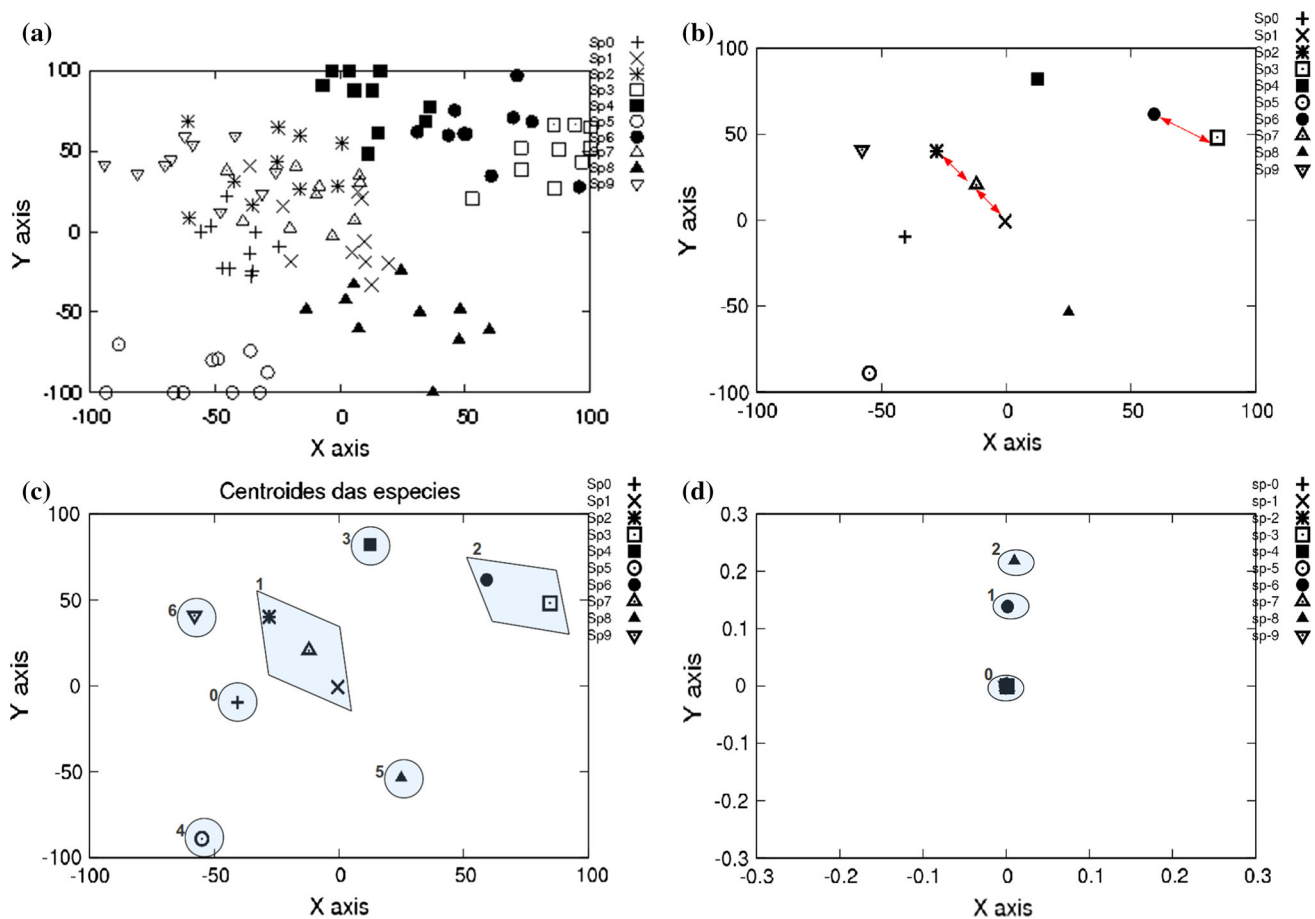


Fig. 5 Conceptual illustration of the proposed algorithm. **a** Initial distribution of all populations. **b** Centroids of all populations. **c** Initial distribution of habitats. **d** Final distribution of habitats

The habitats are generated using the adjacency matrix. Figure 5c illustrates the habitats found for the centroids distribution of Fig. 5b, 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 ecological relationship. Consider, for example, 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 ecological 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,

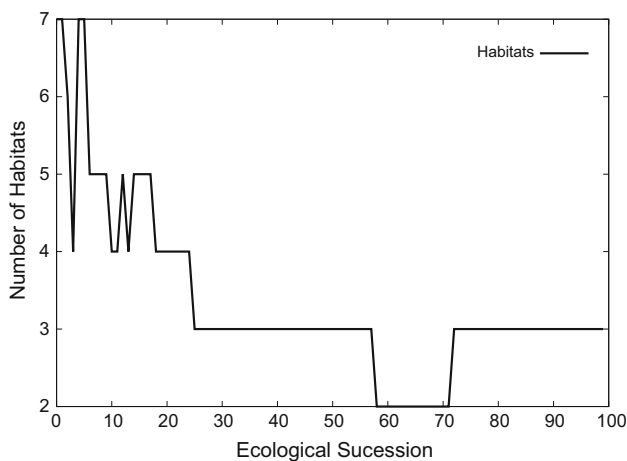


Fig. 6 Number of habitats at each ecological succession step

excluding the best individual. In this stage of the algorithm the ecological succession loop restarts. It is worth to mention that co-evolution is achieved by intra and inter-habitats communications.

Figure 5d 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 Fig. 5d 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 6 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, next section describes some applications of the ECO framework and points some future directions.

4 Features explored and perspectives for future research

Before discussing the functionalities of the ECO framework that were computationally explored, it is important to quote how the results were obtained and highlight the methodology for the statistical tests applied in the works reviewed.

Owing to the stochastic nature of the proposed ECO approach and other meta-heuristic algorithms employed, their performance cannot be evaluated by the result of a single run. Many trials with independent population initializations should be done to obtain an useful conclusion. Therefore, in the experiments reported below, the results were obtained with 30 trials. Also, statistical tests were conducted over the results obtained in order to better understand and support the conclusions. All statistical tests were run using the R Statistical Computing tool (see the R Project web site at <http://www.r-project.org/>). Some tests employed were: the

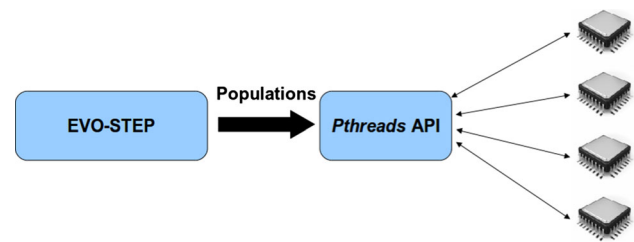


Fig. 7 Pthreads API schedules the populations to the processors

Shapiro–Wilk test to verify the null hypothesis of normality, the Kruskal–Wallis test to compare more than two algorithms, and the Wilcoxon rank-sum test to compare only two algorithms [9, 11, 34].

The ecology-based approach was developed using a parallel strategy to take advantage of the computational power available. The Portable Operating System Interface (POSIX) Threads Programming standards¹ were applied through the Pthreads Application Programming Interface (API). Fig. 7 shows how parallelism was achieved. At each evolutionary period (EVO-STEP) each population triggers a different thread and the Pthreads API schedules the populations to the available processors.

The first application of the ECO framework was done in [37] and a canonical ecology-based optimization algorithm was defined. The ABC algorithm [20] was used in a homogeneous model, i.e., all populations use this algorithm with the same control parameters to evolve their candidate solutions. The characterization of the computational ecosystem in this application, following the elements from Fig. 3, is:

- Environment: landscape defined by continuous unconstrained multi-modal benchmark functions, all static. The biotic components are the candidate solutions of each population. The abiotic component is the environment itself;
- Initialization: all populations are initialized using Normal distributions;
- Heterogeneity: concerning the heterogeneity of the model, in this application, all populations have the same evolutionary behavior. Thus, the system is classified as homogeneous;
- Ecological successions: primary ecological succession is explored;
- Habitats: are defined deterministically using the centroid information and a proximity threshold;
- Communication topologies: the intra-habitats communication topologies are defined deterministically and the inter-habitat communication topology is defined at random;

¹ Web site: <https://computing.llnl.gov/tutorials/pthreads/>.

- Flows: flow of candidate solutions is explored;
- Ecological relationships: mating and migrations are explored;
- Environmental factors: it is not explored;
- Diversity: gene flow is explored
- Synchronism: the system is synchronous with periodical updates;
- Organization: structures and patterns are self-organized.

Although in its canonical version, according to the results reported in [37], the benefit of using the ecology-inspired approach can be best observed with problem instances of high dimensionality. This can be noticed in Figure 8 that summarizes the results and shows the differences between the results obtained by ABC and ECO_{ABC} approaches. The benchmark functions used are Schaffer, Rastrigin, Griewank, and Rosenbrock with 2, 5, 10, and 200 dimensions. The *x-axis* represents the dimensions and the *y-axis* represents the differences. For a better visualization, the *y-axis* is in logarithmic scale.

From Fig. 8 it is possible to observe that the difference between ABC and ECO_{ABC} increases proportionally to the number of dimensions for *Rastrigin* and *Rosenbrock* functions indicating that the more complex the problem, the more challenging it is 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 for both optimization approaches considering these degrees of complexity. For the *Schaffer* function, the difference increases from 2 to 5, and from 5 to 10 dimensions but it is almost constant from 10 to 200 dimensions. This fact suggests that it was more difficult for the algorithms to find good solutions for this function, regardless of its dimensionality. The overall analysis suggests that the ECO approach is more effective in solving more complex problems rather than populations evolving alone. Also, the ecological inter-

actions (intra and inter-habitats) favor the co-evolution of populations and better explores the diversity of solutions.

In [42] the use of population dynamics inside the ECO framework was explored. The population sizing is viewed not as a parameter but as a dynamic process that changes deterministically over time, and the logistic model was applied to control the size of populations [19,31]. The logistic map was chosen due to its simplicity and its rich dynamic behaviour as discrete-time demographic model. The one-parameter logistic map was applied to drive the population dynamics between ecological successions. Also, the logistic map parameter was set to $a = 3.57$ and this is called ‘route to chaos’ [19]. This choice was done based on the work of Ma [28] where experiments were performed with different values for the parameter a . The ABC algorithm was used in a homogeneous model. 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, the use of a population dynamics model inside the ECO framework considerably improved the results for the benchmark functions. From the previous application [37], the only modification in the characterization of the computational ecosystem is the use of primary and secondary ecological successions carried by the non-linear dynamics of the logistic model.

In another application [40] the authors explored the heterogeneity of the ECO framework using two different algorithms cooperatively: the previous ABC and also the PSO [8,22]. In the heterogeneous application, the ABC algorithm and the PSO algorithm are used in such a way that half 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. As result, the use of different search strategies 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 search space. From the first application [37], the only modification in the characterization of the computational ecosystem is the use of different evolutionary behaviors.

Next, in [41] a hierarchical clustering technique [25,33] was used as a biologically plausible strategy to probabilistically set the habitats of the computational ecosystem. The single-link hierarchical clustering algorithm was used to setup the habitats where each cluster represents a habitat. This approach suppressed the control parameter ρ that was used in previous versions as a proximity threshold. Also, differently from previous applications, the communication topologies (intra and inter-habitats) are probabilistically defined. In this

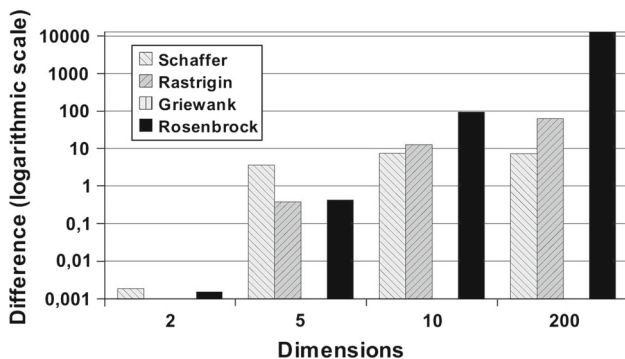


Fig. 8 Difference between ABC and ECO_{ABC} approaches

application, the ABC algorithm was used in a homogeneous model. According to the results reported, 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 occurs during the optimization process. From the first application [37], the two modifications in the characterization of the computational ecosystem are the use of a probabilistic strategy for habitats and communication topologies definition.

With the aim of applying the ECO approach to a complex problem from bioinformatics, [43] reported the application of ECO to the Protein Structure Prediction (PSP) problem. Concerning the AB off-lattice model in its 2D version the aim is to search for low energy conformations [26]. The AB off-lattice model is, possibly, the most studied non-lattice model to represent protein structures. In this model the protein sequences are composed by only two species of monomers: '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. The ECO framework was tested using six configurations and all configurations implement the strategy of defining the habitats with the hierarchical clustering algorithm previously mentioned. The first configuration, ECO_{ABC} , employs the ABC algorithm homogeneously. The second configuration, ECO_{PSO} , employs the PSO algorithm homogeneously. The third configuration, ECO_{DE} , employs the Differential Evolution (DE) [46] algorithm homogeneously. The fourth configuration, $ECO_{jDE/BBO}$, employs the BBO algorithm hybrid with DE [17], also homogeneously. The fifth, 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. Also, the overall best solutions are compared with other results found in literature. The authors reported that when using the heterogeneous model (ECO_{All}) the search process gets more robust than the other approaches (e.g., for 13, 21, and 34 amino acids-long sequences) 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. Results concerning the processing time encourages the use of massive parallel strategies inside the ECO framework when applied to such complex class of problem.

In [4] a heterogeneous parallel version of the ecology-inspired approach (pECO) was applied to another application of the PSP problem, now concerning the 3D AB off-lattice model [35]. In pECO, a parallel master-slave architecture is employed in order to allow the accelerate the processing time of the computational ecosystem. The processing

load is divided into several processors (slaves), under the coordination of a master processor. Each processor (master or slaves) is responsible for initializing the population, and performing the evolutive period of an population independently. The master processor is responsible for defining the communication topologies between populations and habitats. As in the previous work, four different algorithms were used: ABC, PSO, DE, and jDE/BBO. In such heterogeneous model of the computational ecosystem, 1/4 of the populations behave according to specific rules of one of those algorithms. All speed-ups obtained in the results were higher than one indicating that the parallelization of the algorithm decreases the overall computational cost. Also, the computed efficiency suggests that the processors are not being fully used all the time. In fact, speed-up and efficiency are a direct consequence of the balance between the processing load of the slaves and the communication load between master and slaves. Concerning the energy of conformations, pECO approach obtained the best conformation for the 13 amino-acid long sequence and competitive results for the other sequences when compared with the ground solutions found in literature.

Once a brief review of the applications using the ECO framework is presented, it is possible to highlight which features of the entire ecological framework were used in those applications. Figure 9 shows an extended illustrative map with all definitions that a full computational ecosystem for optimization can have. Highlighted are the features used in at least one of the previously mentioned applications.

It is possible to note that there are plenty of other features that can still be explored in the proposed ecological framework. Some few examples are:

- The environment can be explored with the insertion of abiotic components biasing the behavior of populations. Also, other problem domains can be explored (e.g., discrete domains) as well as dynamic instances;
- By using some source of feedback from the optimization process during its course, the habitats formation can be better distributed, as well as the intra and inter-habitats communication topologies, can be better defined;
- Flows of information (e.g., stigmergic communication) and energy (e.g., definition of trophic structure) can be explored;
- Other ecological relationships can be applied;
- Environmental factors can be added to bias the development of populations;
- Strategies and metrics for maintaining the diversity of solutions both at micro and macro levels can be applied;
- Single solution optimization algorithms can be added, e.g., Variable Neighbor Search and Simulated Annealing;
- The whole ECO framework can be explored asynchronously;

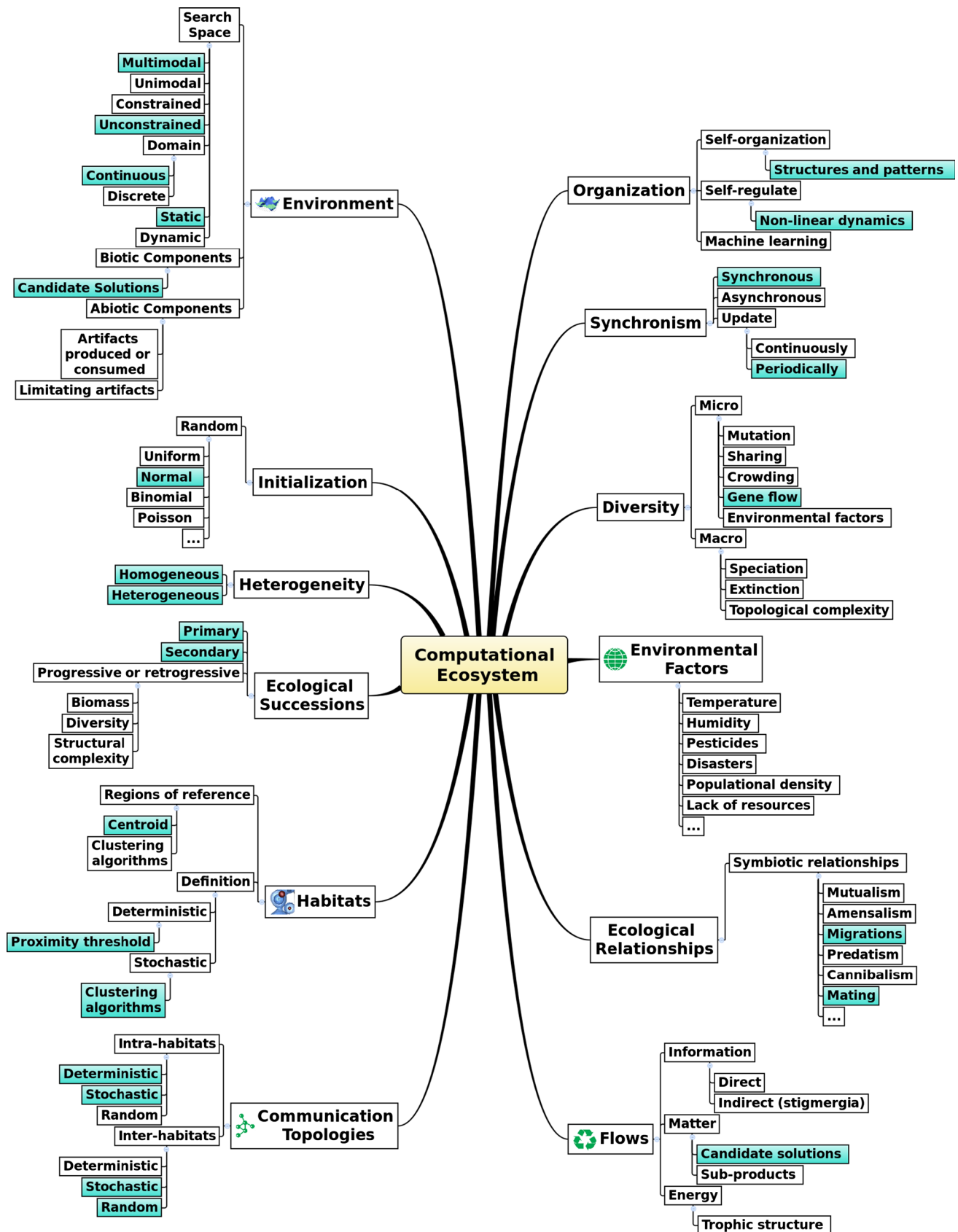


Fig. 9 Extended illustrative map for the elements of a computational ecosystem. *Highlighted* are the features used in at least one previously mentioned application

- Decentralized paradigms can be used to develop the ECO framework (e.g., P-systems).

5 Conclusion

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 inter-linked to each other, e.g., biological ecosystems. This work presented a formal computational model relating the cooperative use of populations of candidate solutions, co-evolving in an ecological context. This is the basis of a computational ecosystem for difficult optimization problems.

The ECO framework is an approach that takes inspiration from real ecosystems. Firstly, this paper described all components that can compose a computational ecosystem. Then, we present several works focusing on different characteristics of the framework. The aim of such experiments were to validate the relevance and importance of the ecologically inspired approach in the field of optimization. Hence, the main goal of the presented work was to point out which features were explored and to propose research lines for future works.

The main features of the ECO framework is that it opens the possibility to integrate different meta-heuristics, the topological structure for communication between populations is dynamically defined using informations from the populations during the optimization process, there are two levels of communications, namely intra-habitats and inter-habitats communication, that favors co-evolution. Also, through intra and inter-habitats communications different kinds of ecological relationships can be modeled in order to better explore the space of solutions. The source code of the ECO framework can be found in the following link: <http://udesc.academia.edu/RafaelStubsParpinelli>.

Based in all works previously done using the ECO framework it is possible to conclude that it is better suited for highly constrained multi-modal problems, such as the PSP problem. Certainly, for uni-modal problems with few or no constraints other classical methods can be more effective.

Although some features were experimented inside the ECO framework, many other computational issues can be approached to bring even more biological plausibility to the system (e.g., speciation, other ecological relationships, environmental factors, etc). It is very important to highlight that adding new biologically plausible features to the ECO framework does not guarantee improvement of the optimization process. Only extensive large-scale experiments can determine their usefulness.

Besides adding new features to the ECO framework, a research direction for future developments is the understand-

ing of the relationships between the parameters of ECO. Also, in order to increase the computational capabilities of the framework, massive parallel architectures such as general-purpose graphical processing units (GP-GPUs) can be naturally explored due the intrinsic parallelism and asynchronism present in ecological systems.

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

References

1. Antunes R, Leymarie F (2013) Real-time behavioral animation of humanoid non-player characters with a computational ecosystem. In: Aylett R, Krenn B, Pelachaud C, Shimodaira H (eds) Intelligent virtual agents, Lecture notes in Computer Science, vol 8108. Springer, Berlin, pp 382–395
2. Begon M, Townsend CR, Harper JL (2006) Ecology: from individuals to ecosystems, 4th edn. Blackwell Publishing, Oxford
3. Benítez CMV, Parpinelli RS, Lopes HS (2012) Parallelism, hybridism and coevolution in a multi-level ABC-GA approach for the protein structure prediction problem. *Concurr Comput Pract Exp* 24(6):635–646
4. Benítez CMV, Parpinelli RS, Lopes HS (2013) A heterogeneous parallel ecologically-inspired approach applied to the 3D-AB off-lattice protein structure prediction problem. In: Proceedings of the 1st BRICS countries congress (BRICS-CCI) and 11th brazilian congress (CBIC) on computational intelligence
5. Binitha S, Sathya SS (2012) A survey of bio inspired optimization algorithms. *Int J Soft Comput Eng* 2(2):137–151
6. Briscoe G, de Wilde P (2008) Digital ecosystems: optimisation by a distributed intelligence. Proceedings of IEEE international conference on digital ecosystems and technologies. IEEE Press, Phitsanulok, pp 192–197
7. Briscoe G, de Wilde P (2009) Computing of applied digital ecosystems. Proceedings of the international conference on management of emergent digital ecosystems. ACM Press, Lyon, pp 28–35
8. Clerc M (2006) Particle swarm optimization. ISTE Press, London
9. Dalggaard P (2008) Introductory statistics with R, 2nd edn. Springer, NY
10. Dasgupta D, Yu S, Nio F (2011) Review article: recent advances in artificial immune systems: models and applications. *Appl Soft Comput* 11(2):1574–1587
11. Derrac J, García S, Molina D, Herrera F (2011) A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm Evol Comput* 1(1):3–18
12. El-Abd M, Kamel M (2005) A taxonomy of cooperative search algorithms. Proceedings of the second international conference on hybrid metaheuristics, Lecture notes in computer science, vol 3636. Springer, Berlin, pp 32–41
13. Engelbrecht AP (2007) Computational intelligence: an introduction, 2nd edn. John Wiley & Sons, Chichester
14. Floudas CA, Pardalos PM (1990) A collection of test problems for constrained global optimization problems, vol 455. Lecture notes in computer ScienceSpringer, Berlin
15. Ganguly N, Sikdar BK, Deutsch A, Canright G, Chaudhuri PP (2003) A survey on cellular automata. Center of High-Performance Computing, Dresden University of Technology, Dresden, Tech. rep.

16. Glance N, Hogg T, Huberman BA (1991) Computational ecosystems in a changing environment. *Int J Mod Phys C* 2(3):735–753
17. Gong W, Cai Z, Ling CX (2010) DE/BBO: a hybrid differential evolution with biogeography-based optimization for global numerical optimization. *Soft Comput* 15(4):645–665
18. González J, Masegosa A, García I (2011) A cooperative strategy for solving dynamic optimization problems. *Memet Comput* 3(1):3–14
19. Kaplan D, Glass L (1995) *Understanding nonlinear dynamics*. Springer, New York
20. Karaboga D, Akay B (2009) A comparative study of artificial bee colony algorithm. *Appl Math Comput* 214(1):108–132
21. Kari J (2005) Theory of cellular automata: a survey. *Theor Comput Sci* 334(1):3–33
22. Kennedy J, Eberhart R (2001) *Swarm intelligence*. Morgan Kaufmann, San Francisco
23. Kephart JO, Hogg T, Huberman BA (1989) Dynamics of computational ecosystems. *Phys Rev A* 40:404–421
24. Krause J, Cordeiro J, Parpinelli RS, Lopes HS (2013) A survey on swarm algorithms applied to discrete problems. In: Yang XS, Cui Z, Xiao R, Gandomi AH, Karamanoglu M (eds) *Swarm intelligence and bio-Inspired computation: theory and applications*. Elsevier Science, Amsterdam, pp 169–192
25. Legendre P, Legendre L (1998) *Numerical ecology*. Elsevier, Amsterdam
26. Lopes H (2008) Evolutionary algorithms for the protein folding problem: a review and current trends. In: Smolinski T, Milanova M, Hassanien AE (eds) *Computational intelligence in biomedicine and bioinformatics*, vol 1. Springer, Berlin, pp 297–315
27. Lung R, Chira C, Dumitrescu D (2012) Guest editorial: special issue on nature inspired cooperative strategies for optimization. *Memet Comput* 4(4):247–248
28. Ma ZS (2011) Ecological theatre for evolutionary computing play: some insights from population ecology and evolutionary ecology. *Int J Bio-Inspired Comput* 3(1):31–55
29. Malik N (2005) Artificial neural networks and their applications. In: *Conference on unearthing technological developments and their transfer to serving masses*. ITM, Mathura
30. Masegosa AD, Pelta DA, García del Amo I, Verdegay JL (2008) On the performance of homogeneous and heterogeneous cooperative search strategies. In: Krasnogor N, Melin-Batista B, Moreno-Prez JA, Moreno-Vega JM, Pelta DA (eds) *International workshop on nature inspired cooperative strategies for optimization, studies in computational intelligence*. Springer, Berlin, pp 287–300
31. May RMC, McLean AR (2007) *Theoretical ecology: principles and applications*. Oxford University Press, Oxford
32. Meirelles M, Jr., G, Fausto A, Araújo R (2010) Diversity-based adaptive evolutionary algorithms. Korosec P (ed) *New achievements in evolutionary computation*. InTech, Rijeka, pp 1–16
33. Murtagh F, Contreras P (2012) Algorithms for hierarchical clustering: an overview. *Data Min Knowl Discov* 2(1):86–97
34. Olofsson P (2005) *Probability, statistics, and stochastic processes*. Wiley-Interscience, New York
35. Parpinelli RS, Benítez CMV, Cordeiro J, Lopes HS (2014) Performance analysis of swarm intelligence algorithms for the 3D-AB off-lattice protein folding problem. *J Mult-Valued Logi Soft Comput* 22:267–286
36. Parpinelli RS, Benítez CMV, Lopes HS (2011) Parallel approaches for the artificial bee colony algorithm. In: Panigrahi BK, Shi Y, Lim M (eds) *Handbook of swarm intelligence: concepts, principles and applications*. Series: adaptation, learning, and optimization. Springer, Berlin, pp 329–346
37. Parpinelli RS, Lopes HS (2011) An eco-inspired evolutionary algorithm applied to numerical optimization. *Proceedings of the third world congress on nature and biologically inspired computing*. IEEE Press, Piscataway, pp 473–478
38. Parpinelli RS, Lopes HS (2011) New inspirations in swarm intelligence: a survey. *Int J Bio-Inspired Comput* 3(1):1–16
39. Parpinelli RS, Lopes HS (2012) Biological plausibility in optimization: an ecosystemic view. *Int J Bio-Inspired Comput* 4:345–358
40. Parpinelli RS, Lopes HS (2012) An ecology-based heterogeneous approach for cooperative search. In: Barros L, Finger M, Pozo A (eds) *Proceedings of the Brazilian Symposium on Artificial Intelligence (SBIA), Lecture notes in computer Science*, vol 7589. Springer, Berlin, pp 212–221
41. Parpinelli RS, Lopes HS (2012) A hierarchical clustering strategy to improve the biological plausibility of an ecology-based evolutionary algorithm. In: PAVN J. et al. (ed) *Ibero-American Conference on Artificial Intelligence (IBERAMIA), Lecture notes in computer Science/Lecture notes in artificial intelligence*, vol 7637. Springer, Berlin, pp 310–319
42. Parpinelli RS, Lopes HS (2012) Population resizing using nonlinear dynamics in an ecology-based approach. In: Yin H, Costa JAF, Barreto GDA (eds) *Proceedings of the intelligent data engineering and automated learning*, *Lecture notes in computer science*, vol 7435. Springer, Berlin, pp 27–34
43. Parpinelli RS, Lopes HS (2013) An ecology-based evolutionary algorithm applied to the 2D-AB off-lattice protein structure prediction problem. In: *Proceedings of the 2nd Brazilian Conference on Intelligent Systems*. Conference Publishing Services
44. Pasti R, de Castro LN, von Zuben FJ (2010) Computação biogeográfica: Formalização e proposição de uma meta-heurística espaço-temporal. *Anais do XVIII Congresso Brasileiro de Automática* pp 5065–5072
45. Simon D (2008) Biogeography-based optimization. *IEEE Trans Evol Comput* 12(6):702–713
46. Storn R, Price K (1997) Differential evolution a simple and efficient heuristic for global optimization over continuous spaces. *J Glob Optim* 11(4):341–359
47. Tonda A, Lutton E, Squillero G (2012) A benchmark for cooperative coevolution. *Memet Comput* 4(4):263–277
48. Vulli SS, Agarwal S (2008) Individual-based artificial ecosystems for design and optimization. In: Keijzer M (ed) *Proceedings of the conference on genetic and evolutionary computation*. ACM, Atlanta, pp 273–280
49. Wang Y, Wang T, Tang L, Zhong D (2007) Ecosystem model based grid resource optimization management. *Int J Comput Sci Netw Secur* 7(2):61–66
50. Wolpert D, Macready W (1997) No free lunch theorems for optimization. *IEEE Trans Evol Comput* 1(1):67–82
51. Yang XS, He X (2013) Bat algorithm: literature review and applications. *Int J Bio-Inspired Comput* 5(3):141–149