On the Use of Predation to Shape Evolutionary Computation

Felipe S. P. Andrade
RECOD Lab, Institute of Computing
University of Campinas
Campinas, Brazil
felipe.andrade@ic.unicamp.br

Claus Aranha
Faculty of EIS
University of Tsukuba
Tsukuba, Japan
caranha@cs.tsukuba.ac.jp

Ricardo da Silva Torres
Department of ICT and Natural Sciences
Norwegian University of Science and Technology
Ålesund, Norway
ricardo.torres@ntnu.no

***Abstract*—Classic Evolutionary Algorithms often use elitist approaches, such as fitness functions, to select individuals for new generations. In this work, we consider an alternative strategy to simulate the selection process that relies on exploiting ecological interactions between individuals instead of explicitly using a fitness based in the search progress. To demonstrate this strategy, we present an Artificial Life system which simulates an ecosystem where different species are different bio-inspired meta-heuristics, and the main ecological relationship is the *predation*. Specifically, individuals from a Particle Swarm Optimization (PSO), with movement rules defined by Genetic Programming, survive by predating on individuals from an Artificial Bee Colony (ABC) system that operates on traditional optimization rules. This ecology is investigated on optimization benchmarks, and we observed the development of interesting ecological dynamics between the two species.**

***Index Terms*—Artificial Life, Evolutionary Computation, Ecological Relationship**

I. Introduction

The Evolutionary Computation (EC) field is characterized by several algorithms that take inspiration from natural evolution. Representative EC algorithms include Genetic Algorithms (GA) [1], Genetic Programming (GP) [2], Evolution Strategies (ES) [3], and Differential Evolution (DE) [4]. General characteristics of these algorithms are the use of a population of individual solutions, the simulation of the evolutionary process through the selection of individuals through a fitness measure, and the use of variation operators based on genetics for the generation of new individuals, such as crossover and mutation.

Of these factors, the assessment and selection of individuals in the population can be said to be one of the key and most important characteristics of EC. The common procedure is to calculate a *fitness value* for individuals using a well defined fitness function, and then to sort and select individuals based on their fitness values, using several selection strategies that can be stochastic or deterministic. The fitness function and the selection strategy guide the search and the population of solutions through the solution search space.

This study was financed in part by Brazilian funding agencies CAPES and CNPq (grants #8881.135859/2016-01 and 141224/2016-9)

978-1-7281-2547-3/20/$31.00 ©2020 IEEE

Although the selection through fitness values is a key part of EC algorithms, it is also arguably a point where the evolutionary allegory breaks down. Evolution in nature does not have an explicit objective, so this process is closer to the process of human-directed selective breeding than evolution by natural selection. This can be important if we think of the grander question of Open-ended evolution (OEE), as the fitness function provides a clear end point for evolutionary algorithms. In this line of thought, Stanley and Lehman [5] have proposed *Novelty-Search*, a way to select solutions in EC algorithms based on their difference from existing solutions rather than an explicit fitness measure. Novelty search has since then inspired the creation of several different successful algorithms and variants.

We are interested in exploring a different direction for performing EC without directly relying on explicit fitness functions, aiming to explore the possibilities of the combination with OEE. Our idea is to create a simulation where the behavior of individuals is shaped by the interaction among themselves and their environment, as well as a more open-ended kind of evolution, limited mainly by the capacity of the individuals to obtain energy and reproduce in several ways.

We have recently proposed a framework for this simulation (EvoCycle), inspired by other uses of ecological concepts in EC, such as [6], [7]. This framework supports several species of individuals, which are based on bio-inspired meta-heuristics, but modified so that their behavior can evolve during the simulation, and that their survival and reproduction depend on a simulation of ecological interaction between individuals of the same and different species.

When simulating artificial ecosystems, it is possible to focus on several relationships between agents. One that is particularly interesting and popular is the predator-prey ecological relationship; probably because it is one of the most notable relationships and exerts a big role in selection pressure.

In this work, we study an specific implementation of EvoCycle that focuses on the predator-prey interaction between two different species (algorithms). The predator species is a variant of Particle Swarm Optimization (PSO) [8] where the movement equation of each particle is encoded as a Genetic Programming (GP) tree. This idea of evolving the movement equation of PSO using a GP has been explored before by Poli et al. [9]. However, while in that work each individual is an entire PSO population, and the GP is offline, in our PSOGP hybrid each individual has its own GP tree that controls its movement (genotype), and the evolution of the movement rules occurs in tanden with the PSO simulation. Other works approaching evolvable behaviour include Samarasinghe et al. [10], who presented a comparison of Grammatical Evolution (GE) and GP to evolve complex swarm behaviour in a boids simulation system, Hartman & Benes [11], who also used GE to perform the evolution of collective behaviour for robot swarms, and Karaboga & Basturk [12], who also used GP to generate self-organizing swarming rules.

The prey species in our simulation is the Artificial Bee Colony (ABC) [13] algorithm. The ABC operates very similarly to its original formulation, except that it accumulates energy based on the fitness value of the position were individuals (hives) are located. PSOGP individuals depend on this energy to live, reproduce, and evolve. So they need to feed on ABC individuals that are close enough (based on the euclidean distance in the search space). Therefore, in this simulation, ABC individuals are primary consumers, obtaining their energy directly from the environment (based on the fitness function), while PSOGP individuals are secondary consumers, that feed on successful ABC individuals.

We observed the simulation of this implementation of EvoCycle on landscapes based on continuous functions available in the BBOB benchmark.[1] The simulation shows that PSO individuals evolve to effectively search, approach, and keep close to successful ABC individuals. As the ABC individuals progress their search for areas with high fitness values, the PSO swarm follows them, behaving like a kind of local search to the global search of the ABC, even if that was not their original intent. In this way, we see that the predator-prey dynamic shapes the collective behavior of individuals over the generations, and motivates the exploration of more complex ecological relations.

II. Related Work

This work is mainly concerned with the simulation of artificial ecosystems in the context of the study of artificial life. An initial inspiration is the Echo system [14], which was implemented to validate the theory of complex adaptive systems [15]. Echo System is defined as complex since it is composed of agents that interact and produce emergent patterns, with the agents and interactions evolving through time.

Gras et al. [16] proposed an agent-based predator-prey ecosystem (Ecosim) using Fuzzy Cognitive Maps (FCM) to model the behaviour of the agents. With a distance perception mechanism, the agents present the behaviour of evasion and search for food. Their main results were the differentiation of species through the simulation. Alfonseca and Gil [17] evolved an ecosystem of mathematical expressions using GE. As their system evolves new expressions, there is no spatial information associated with the individuals of the ecosystem. Individuals are evaluated based on fitness functions and predator-prey population dynamics are produced.

A 3d ecosystem model was proposed by Ito et al. [18]. In this system the agents have physical attributes that play an important role on movement and reflect on the agent's fitness. GA is used to evolve the characteristics of each population, separately from the interaction among the agents. Classic patterns of predator-prey could be observed, and they noted a correlation between body volume and the presence or absence of predators. Adami et al. [19] proposed a predator-prey ecosystem in a two dimensional grid. Each agent has its field of vision modeled by a Markov network, corresponding to its genetic code and evolved using a GA in separate steps of the population interactions. Obtained results demonstrate that, once the preys learn to swarm, predators benefit from a more focused view. But when this happens, preys start presenting a spreading behaviour, making the predator adopt wider vision.

Sunehag et al. [20] developed a predator-prey ecosystem using reinforcement learning agent models to move in 2D and 3D spaces. One of their main contributions is to develop a generic simulation with 3 trophic levels. In their findings, we could see population dynamics and the emergence of swarming behaviour, as well as different interactions patterns between the species.

Trying to design a system where agents of a swarm could achieve open ended evolution (OEE), Witkowski and Ikegami [21] proposed a system using boids with behaviours established by neural networks (NN). The NN configuration evolves through time using GA. Their main findings were associated with the fact that in order to obtain OEE, the decentralization aspect is very important, as well as the communication between agents.

Compared to these works, a key characteristic of the EvoCycle framework is that the evolution process happens concurrently with the simulation (as opposed to the evolution of populations at the end of each simulation). Also, by using swarm intelligence meta-heuristics as the basis for the different species, the individuals of our simulation are simple and easy to implement and modify, while at the same time showing interesting degrees of similarity and different characteristics.

III. Background

This section introduces the EvoCycle framework and used base algorithms in the framework instantiation.

A. *EvoCycle*

Figure 1 presents a conceptual view of the EvoCycle framework. Its goal is to exploit the interaction of individuals, organized according to a particular collective behaviour, which is defined in terms of genetic material.

The bottom layer is the genetic layer. In this level, there are structures used in Evolutionary Algorithms. Instead of representations of a solution, those structures encode (genotype) different search strategies that will be used in the middle layer (arrow 1). The middle layer is composed of swarm intelligence

[1]Accessible at http://coco.gforge.inria.fr/doku.php (as of August 2020).

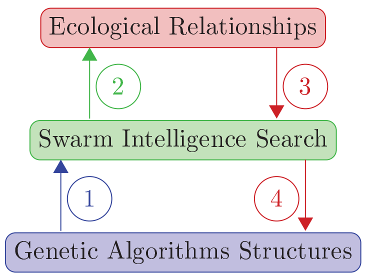


Fig. 1: Conceptual view of the EvoCycle framework.

algorithms. Each agent follows a search rule (phenotype) encoded as an evolutionary algorithm genetic code (bottom layer). The difference from the traditional approach is that each individual may have a different configuration of search strategy even inside the same population. From the search results (arrow 2), the third layer defines different ways for individuals to interact with each other and with the environment, based on ecological relationships (arrow 3). Besides that, the system in a whole works in an artificial life simulation fashion, in the sense that individuals live in an environment with different capacities of providing resources, i.e., individuals are born, age, seek energy sources, reproduce, and eventually die. Those relationships will determine which individuals remain on the population, and, as a consequence, define which genetic codes are preserved in the population as well (arrow 4). After the execution of this cycle over generations (iterations), populations are expected to converge in such a way they end up holding specific characteristics that might be of interest for their preservation.

Figure 2 illustrates the EvoCycle evolution process with more details. We will use the following concepts to characterize our framework: let P be the set of different populations; I be the set of individuals (agents); R be the set of relationships; and S be the representation of the space from the environment.

The general framework for the EvoCycle is divided into three parts. The main flow is represented in blue. The first step is to initialize all the populations $p \in P$ of search agents i_p that are designed as hybrids between Swarm Intelligent and Evolutionary Algorithms. Once the initialization of the agents is finished, filling all populations, the artificial life attributes are set. First, a counter T, the number of iterations, is initialized. T can be seen as a proxy of time. Then, the age ($i_p.Age$) and the energy ($i_p.Energy$) attributes of each individual of the populations are set. The age encodes for how many iterations an individual has been living, while energy is a proxy of how "healthy" it is. Initially, $i_p.Age = 0$ and $i_p.Energy = 0$.

Populations are implemented though the use of swarm intelligence algorithms. The search performed, therefore, depends on particular characteristics of the employed swarm algorithm. For each population present in the system, a different execution flow is performed, represented by Label 1 in the chart. The usual search process employed in such algorithms consists of a group of agents moving through the search space according to a set of rules, producing a collective pattern. In our solution, these rules are encoded in each agent as a representation of the evolutionary algorithm.

After all the populations perform their search process, the artificial life attributes are analyzed, as indicated in Arrow 2. Individuals that reach an age limit threshold AT die and are removed from the population. The same occurs for individuals that have no energy left, based on an energy threshold ET. Surviving agents then interact with each other as long as there is a valid ecological relationship between them, Arrow 3 in the figure. Examples of these relationships are the *reproduction*, which occurs among individuals of the same species; and *predation*, among individuals from different ones. Finally, after Arrow 4 in the figure, an end criterion is tested.

B. *Base Algorithms*

1) Genetic Programming (GP): GP [2] is an Evolutionary Algorithm originally designed to evolve computer programs. Taking inspiration from the natural evolution process in Biology, each program is structured as a genetic programming tree (common representation) working as a genetic code. Each program (individual) is evaluated according to a fitness function. The best individuals are selected and through the reproduction of two individuals a new one is generated. Often, individuals are generated as the result of genetic operators, defined in terms of "genetic codes" coming from the parents. This process is repeated over generations and better performing individuals (better solutions) are expected to be found at the end of the process.

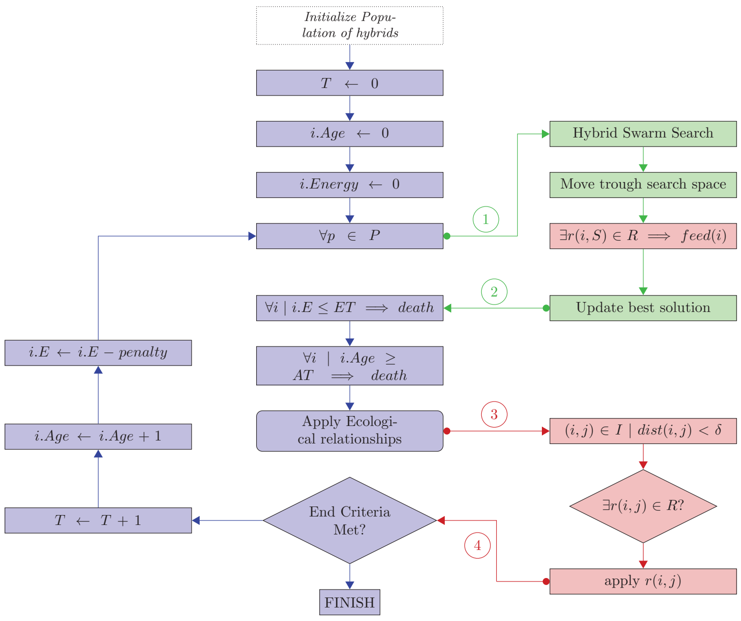
2) Particle Swarm Optimization (PSO): PSO [8] is an instance of algorithm from the Swarm Intelligence category. The algorithm takes inspiration from the collective movement of large group of animals like flocks of birds or school of fish to perform the search of solutions in optimization problems. To simulate such type of behaviour, solutions of the problem are considered particles in the space which will move according to following equations:

$$V_{i+1} = V_i + R_1 \times \phi_1 \times (P_{best} - X_i) + R_2 \times \phi_2 \times (G_{best} - X_i) \tag{1}$$

$$X_{i+1} = X_i + V_{i+1} \tag{2}$$

where X represents the position in space of each particle at a certain moment i in time, V represents a velocity function that varies according to the previous velocity and the positions of the best position found so far, P_{best} and the best position found by the whole group, G_{best}. ϕ_1 and ϕ_2 are random numbers between 0 and 1. This classic algorithm was extensively studied and several modifications have proposed in the literature. For a review about existing proposals, the reader may refer to [22].

3) Artificial Bee Colony (ABC): ABC [13] is a Swarm Intelligence algorithm inspired by bee's strategy to build hives in positions with good quality. The algorithm can be divided into three main steps: the employed bees phase, the onlooker



Initialize Population of hybrids
$T \leftarrow 0$
$i.Age \leftarrow 0$
$i.Energy \leftarrow 0$
$\forall p \in P$
$\forall i \mid i.E \leq ET \implies death$
$\forall i \mid i.Age \geq AT \implies death$
Apply Ecological relationships
End Criteria Met?
FINISH
$T \leftarrow T+1$
$i.Age \leftarrow i.Age+1$
$i.E \leftarrow i.E - penalty$
1
2
3
4
Hybrid Swarm Search
Move trough search space
$\exists r(i,S) \in R \implies feed(i)$
Update best solution
$(i,j) \in I \mid dist(i,j) < \delta$
$\exists r(i,j) \in R?$
apply $r(i,j)$

Fig. 2: EvoCycle evolution process.

bees phase, and the scouter bees phase. Initially, a certain number of hives is positioned randomly on the search space. Then, in the employed phase, a bee visits one of the hives. Therefore, for each hive in the population, a new hive position is created by modifying one of the coordinates of the best hive position. This modification comes from the weighted sum of the best position and the difference between the current visited hive and a randomly selected one, according to the following equation:

$$v_{ij} = x_{ij} + \phi_{ij} \times (x_{ij} - x_{kj}) \tag{3}$$

where j is the randomly chosen dimension of the hives, V_i is the new generated hive, x_i is the hive i currently visited, and x_k is a randomly chosen hive different from i. ϕ is a random number between -1 and 1. After the modification, if the new hive created possesses a better fitness, than the currently visited, the new one replaces it in the population.

Next, the onlooker bee phase takes place. It has a similar process with the difference that instead of modifying a position from a hive each at a time, the change occurs in a hive selected proportionally to the quality, according to:

$$p_i = \frac{fit_i}{\sum_{N=1}^{SN} fit_n} \tag{4}$$

where p_i is the probability if a hive being chosen to be visited given its own fitness fit_i in relation the sum of the fitness of all the hives.

The last step is the scouter bee phase, which is responsible for checking if hives were visited more than a predefined number of times. If this is true for a particular hive, this hive is moved to a random position.

At each iteration, to the hives are assigned an energy level corresponding to the position they are occupying according to the following equation:

$$E_i = \begin{cases} \frac{1}{fit+1} & \text{if } fit > 0 \\ 1 + fit & \end{cases} \tag{5}$$

where fit is the corresponding fitness of the position the hive i is placed at a given iterarion.

4) EvoCycle-PSOGP: By using the proposed EvoCycle framework, we proposed an algorithm using PSO as the swarm intelligence searcher and GP trees as the codification of the PSO's agents velocity functions. In this formulation, GP operators are used to modify the genetic codes whenever a reproduction occurs. In this setting, each hybrid agent, referred to as PSOGP from now on, moves through the search space following its own velocity function. Each hybrid obtains an amount of energy proportional to its position. The ones that are in better positions will then have more chances of surviving and reproducing, leading to more chances for their genetic codes become prevalent in the population and therefore be present in future generations.

IV. Predation-aware EvoCycle (PA-EvoCycle)

The process of predation has no unique definition in Ecology literature due to different assumptions taken when trying to define this concept. In this paper, in a predation, the individual of one species feeds on individual of another one but not necessarily kills the prey [23]. By adopting this definition, both herbivore and parasitic would be considered as predation.

The general execution is basically the same until reaching the red part in the main flow of the EvoCycle framework (Figure 2), which represents any ecological relation between individuals in a general way. Here we represent this step more specifically as the predation relation.

We propose the use of two different populations for implementing a predation relationship, aiming to create a more diverse virtual ecological environment in which the predation algorithm is implemented through the interaction of populations implemented using different base swarm algorithms.

In our formulation, the predator population is implemented using a hybrid swarm algorithm implemented using PSO and GP, as described in the EvoCycle-PSOGP section. The prey population, in turn, is implemented using the ABC algorithm. The ABC population searches for suitable locations to build hives based on the quality of the search space. The hives accumulate energy at each iteration, also in proportion to the quality of the position. These hives are the primary food source of the PSOGP individuals. The PSOGP individuals move in the search space, each one according to a different velocity function defined by a GP tree. These individuals depend on energy to live and reproduce, and such energy can be acquired by feeding from the ABC hives. These individuals are able to feed from the hives if they are close enough. This condition depends on a proximity threshold. The idea is that after a search process of the ABC, the individuals, which are able to find hives to feed themselves, survive in future generations and perform a local search in the vicinity of the hive position.

V. Simulation Protocol

This section presents performed simulations aiming to validate the proposed PA-EvoCyle framework.

A. Research Questions

We hypothesize that the predation ecological relation may be used to shape a specialization of populations involved so that individuals behave as effective solution searchers. In our evaluation protocol, we address this hypothesis from different perspectives, listed below as research questions:

Q-1: Would the predation relationship shape the behavior of predators and preys? Would their interaction lead to better results in optimization problems? We hypothesise that predators that have more energy, would more likely survive and reproduce if they find preys with more energy.

Q-2: The search strategies are dependent on the genes used to encode the possible patterns. Which genes are the most significant in the search process? Inserting genes previously not present in the original search would change the search in which way?

Q-3: Each population uses different strategies and characteristics in the search process. In this way, would the exchange of information help to improve the search process?

B. Protocol

To address Q-1, we consider the use of two different populations performing a search, aiming to find the best possible solution of an optimization problem. The configuration defined in the Pa-EvoCycle section is used in this experiment where a PSOGP population predates an ABC population.

We use GP trees to encode the velocity functions of PSO agents. In this way, the parameters of the PSO velocity equation are considered as the genes, and are modeled as the leaves of GP trees. To address question Q-2, we analyse the variance of the frequency of such genes in the predator population along several iterations. We assess the impact of including an extra gene, named "Hpos," in the velocity function of PSO, which encodes the position of the hive that is currently feeding an PSO agent.

Since more than one population is moving through the search space, information about the best locations found in the evolutionary process are kept by each population apart form each other. Question Q-3 is intended to assess if the simulation results would improve if information about the best locations is shared by both populations. In this case, we compare an implementation that considers shared information with another that does not. This is done by sharing the global best position of the PSOGP with the best hive position of ABC.

In summary, the simulations are populations of PSOGP predating ABC ones, in four different configurations, with/without HPOS and with/without information exchange.

C. PA-EvoCycle Simulation

The framework was simulated using continuous optimization functions, which work as the search space and base environment for the swarm ecosystem. By applying a function to the position of each individual we obtain a function value. Also, there is an objective value for each function. The difference between the function value for the best found position from the objective value of each function is the error of a simulation. Here we use the BBOB noiseless functions benchmark implementation for the 2019 COCO framework.We performed searches using 5, 10, and 20 dimensions on the first 15 instances of each problem. The functions used are: F1 (Sphere), F8 (Rosenbrock), F13 (Sharp Ridge), and F21 (Gallagher's Gaussian 101-peaks).

D. Implementation Issues

The EvoCycle implementation is composed of several algorithms working in different layers. This section presents the values used in the parameters of each algorithm.

- *PSO configuration:* The intial PSO configuration used in the experiments is presented in Table I. The initial population is high to promote a high diversity in the genotypes of the populatin. All individuals start in a random position and with velocity equal zero.
- *GP configuration*: The genetic code used to implement the velocity functions of the PSOGP hybrids, as well as the operators are presented in Table I. The leaf nodes are the literals from the original PSO velocity equation and constants. The "Hpos" is used depending on the experiment configuration. The internal nodes are simple mathematical operators.
- *Alife configuration:* All the PSOGP particles have a life cycle defined in terms of the energy they possess. The life threshold in Table I presents the minimum energy

TABLE I: Parameters used in the EvoCycle Framework.

GP configuration	
Max Depth	3
Method	Ramped
Internal Nodes	$+, -, \times$
Leaf Nodes	$X_i, V_i, P_{best_i}, G_{best}, H_{pos}*$
Leaf Nodes	$0.5, -0.5, 1.0, -1.0$
Leaf Nodes	random values between 0 and 1

PSO configuration		ALife configuration	
Population Size	2000	Life Threshold	0
Position	Random	Distance Threshold	0.5
Velocity	0	Reproduction Threshold	top 20%
		Iterations to Reproduce	10

value for a particle to survive. The Reproduction Threshold determines that only 20% of the individuals with more energy can reproduce. Furthermore, each particle is allowed to reproduce every 10 iterations. Particles are considered close to each other if their distance is less than 0.5 in each dimension.

- *Feeding configuration:* Bees accumulate energy in the hives according to the equation $H_{it} = H_{it-1} + E(h)$, where H_{it} is the amount of energy accumulated at iteration it and $E(h)$ is the amount of energy obtained by the hive following Equation 5. The PSOGP agents feed when they are close to a hive (defined in terms of a predation distance threshold). They accumulate energy is computed by $E_{it} = E_{it-1} + \frac{H}{H_f+d}$, where E_i is is the energy accumulated in iteration it, H is the energy accumulated until iteration it in the hive that feeds the particle, H_f is the number of particles feeding that hive, and d is the L2 distance between the particle and the hive.

VI. Results and Discussion

Figure 3 presents the results of our simulations on functions F1, F8, F13, and F21. These figures present the error (objective minus the best value) obtained for each instance in a log10 scale. The x-axis refers to different instances, while the y-axis refers to the error.

From Figure 3, we observe that there was a difference in the error results of the four configurations. The first observation is that the exchange of global best position found by each of the populations led to the discovery of regions with low errors in more instances as encoded in the yellow and purple lines for functions F1, F8, F13, and F21. The second observation is that even when improvements were obtained, there were large fluctuations on the error values for the configurations that consider the use of shared information regarding the global best. We can also observe that the introduction of the “Hpos” gene produced similar results as the ones yielded by the configuration that does not consider such gene. These results suggest that the use of “Hpos” does not lead to improvements (Question Q-2).

We also analyzed the variation of the population size throughout the iterations of the executions along with the errors. Figure 4 presents the population growth of PSOGP predator population in two scenarios. The first scenario, with no exchange of information between predator and prey populations, indicates that the individuals could not locate ABC hives to feed from and the population perished before 100 iterations. Even after the predator population was extinguished, the ABC algorithm kept running but the error dropped just by a little amount (order of 10^2). In the other scenario, the PSOGP population was able to locate ABC hives, fed from them, and gained energy to reproduce. In this case, the population kept growing. Besides that, the interaction of the two populations indicates that positions discovered by predators also benefited the ABC population since the errors decreased much more over generations (order of 10^{-8}). These results suggest that the interaction between populations improve the search capability (Question Q-1). From Figure 4, we can also observe that the exchange of information (bottom plot of the figure) benefited both populations in the search process.

To determine which genes were more relevant (as posed in Question Q-2) we analyze the change of frequencies of the genotypes through several iterations, in the cases of success, with and without the “Hpos” gene. As the particles of the population are moving through the search space following the PSO equation, different combinations of literals determine a movement towards a resulting direction, as a phenotype. We counted the number of times each of these phenotypes appears in the velocity equation of every individual, at each iteration.

By analyzing the phenotype count at the end of the execution, we can check which phenotypes were not eliminated in the search process. With the persistent phenotypes in the end, we can verify if there were any that prevailed over the others in the population. Based on the counting, we selected the most frequent phenotypes to analyze their variance in the whole execution. Figures 5a and 5b present the variance in the frequency of phenotypes in the F8 function with 10 dimensions execution per 100 iterations in a stacked histogram. Each column of the histogram presents the total number of individuals of the population from the iteration 100 to the end of execution. Each column is then divided, with each part being the number of individuals which contain a given phenotype.

From Figures 5a and 5b, we can observe that there were genes more important when the predation process occured. In both cases (with and without “Hpos”), we observe that moving towards the best known location from the current position ($Gbest - Pos$) dominated the population. This result suggests that the predation could produce specific behaviour in the population as posed in Question Q-1. With the dominant phenotypes, we could also observe the emergence of different movement behaviours. From the list in Figure 5a, there was the dominance of five different genotypes, which are associated with four phenotypes. The first phenotype is the aforementioned ($Gbest - Pos$). The second phenotype makes particles to keep their movement towards the same direction they

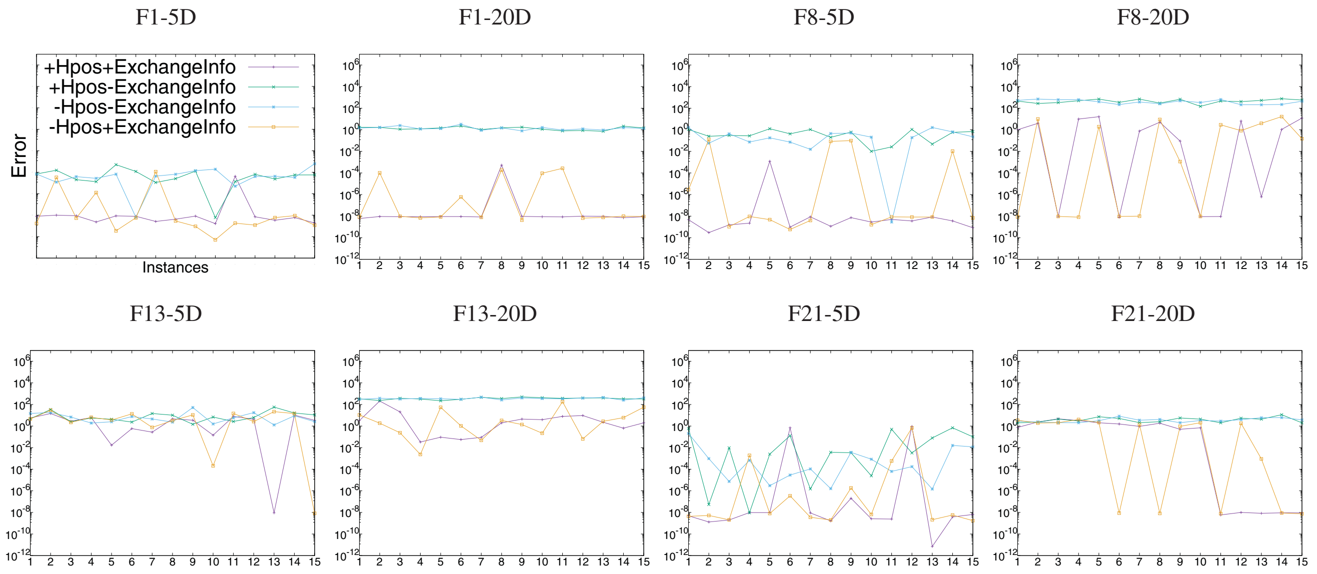


Fig. 3: Simulations of the model using different configurations. Each line of the figure corresponds to the functions F1, F8, F13, and F21. The columns of the right are the 5 dimensions results, and the left 20 dimensions. On the x-axis are each of the 15 instances from each problem. On the y-axis the errors are represented in the interval $(10^{-12}, 10^{6})$, with a log10 scale. The purple line with '+' uses the Hpos gene, with information exchange between predators and preys. The green line with 'x' also uses the Hpos gene but no information exchange. The blue line with '*' does not use the Hpos gene nor the information exchange. On the yellow line with squares the Hpos gene is absent but there is information exchange.

were going. This phenotype is associated with two genotypes: $(Pos + Vel)$ and (Vel^2). The third is a phenotype that makes the particle to move in an opposite direction $(Pos - Vel)$. The last one is a movement around the current position $(POS - R)$. With regard to the use of the "Hpos" gene (Figure 5b), besides the $(Gbest - Pos)$ predominance, we can also observe a movement towards the hive the particle was feeding from $(Hpos - Pos)$ and the exploration of direction between the best known location and the hive position $(Gbest - Hpos)$.

VII. Conclusions

In this work, we presented an ecosystem of swarm-based algorithms with elements of artificial life and the presence of different populations that interact with each other through a predation ecological relationship. In our proposal, we used a hybridization of PSO with GP, named PSGOGP to allow individuals to evolve their own PSO velocity equations through iterations. A predation relation was implemented in which the PSOGP feeds from ABC. In the simulations, we were able to find that the interaction through a predation relationship of the PSOGP and ABC populations led to the emergence of movement behaviours in the PSOGP populations. We analyzed the variance of specific genotypes through the iterations of the search process and found out that a movement pattern that leads the particles in the direction of the best global position were dominant in different scenarios. We also observed that an information exchange of that best global position was crucial for the survival of the predator PSOGP population.

In future work, we intend to use more than one population of hybrids to examine the variation of evolvable genotypes in multiple populations.

References

[1] J. H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*. Cambridge, MA, USA: MIT Press, 1992.

[2] J. R. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. Cambridge, MA, USA: MIT Press, 1992.

[3] H.-G. Beyer and H.-P. Schwefel, "Evolution strategies – a comprehensive introduction," *Natural Computing*, vol. 1, pp. 3–52, Mar 2002.

[4] K. V. Price, *Differential Evolution*, pp. 187–214. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013.

[5] K. O. Stanley and J. Lehman, *Why Greatness Cannot be Planned: The Myth of the Objective*. Springer International, 2015.

[6] D. Simon, "Biogeography-based optimization," *Evolutionary Computation, IEEE Transactions on*, vol. 12, no. 6, pp. 702–713, 2008.

[7] R. S. Parpinelli and H. S. Lopes, "A computational ecosystem for optimization: review and perspectives for future research," *Memetic Computing*, vol. 7, no. 1, pp. 29–41, 2015.

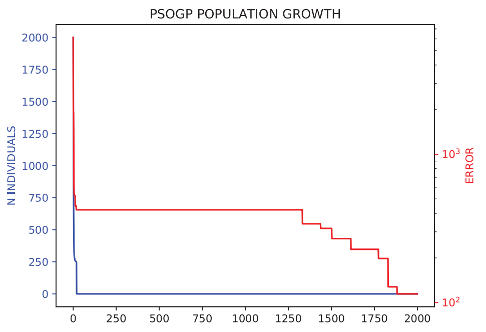
[8] R. C. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proceedings of the sixth international symposium on micro machine and human science*, vol. 1, pp. 39–43, New York, NY, 1995.

[9] R. Poli, W. B. Langdon, and O. Holland, "Extending particle swarm optimisation via genetic programming," in *European Conference on Genetic Programming*, pp. 291–300, Springer, 2005.

[10] D. Samarasinghe, E. Lakshika, M. Barlow, and K. Kasmarik, "Automatic synthesis of swarm behavioural rules from their atomic components," in *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO 18, (New York, NY, USA), p. 133140, Association for Computing Machinery, 2018.

[11] A. Neupane, M. A. Goodrich, and E. G. Mercer, "Geese: Grammatical evolution algorithm for evolution of swarm behaviors," in *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO 18, p. 9991006, 2018.

With **no** information exchange



With information exchange

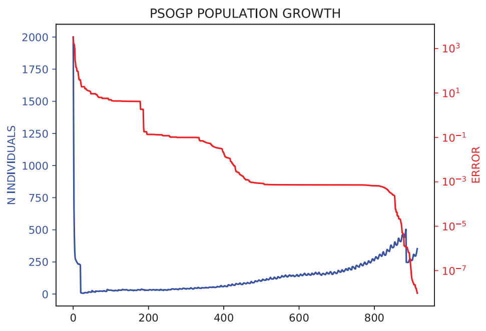
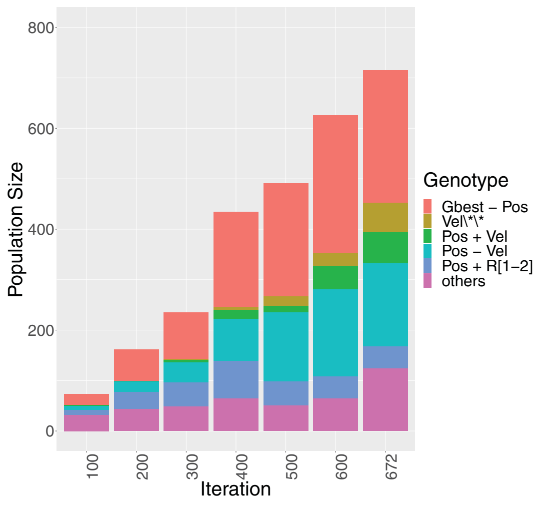
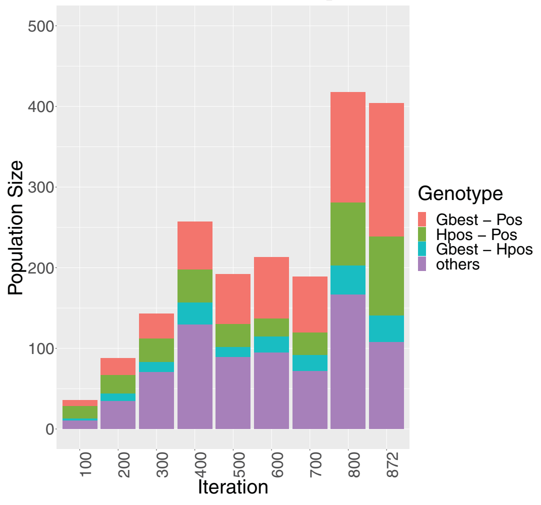


Fig. 4: Plots of the population growth through the iterations of a simulation. The upper figure illustrates an execution on an instance fail of the F8 function with 10 dimensions and with no information exchange. On the bottom, a success case on the same function but with information exchange. The x-axis stands for the number of iterations. The blue line is the population size in number of individuals represented on the y-axis on the left and the red line is the errors in a log10 scale represented in the y-axis on the right, with minimum of 10^2 (upper) and 10^{-9} (bottom)



(a) **Without** Hpos



(b) **With** Hpos

Fig. 5: Variance of phenotypes per iteration in function F8 (a) with and (b) without the Hpos gene with position information exchange. Each column of the histogram presents the number of the most frequent phenotypes. The phenotypes are represented by the colors in the legend on the right.

[12] T. Wang, X. Peng, Y. Wu, and J. Gao, "A GP based two-layer framework for data-driven modeling of swarm self-organizing rules," in *IEEE Congress on Evolutionary Computation, CEC 2019, Wellington, New Zealand, June 10-13, 2019*, pp. 174–181, IEEE, 2019.
[13] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm," *Journal of Global Optimization*, vol. 39, pp. 459–471, Nov 2007.
[14] S. Forrest and T. Jones, "Modeling complex adaptive systems with echo," in *Complex Systems: Mechanisms of Adaptation*, pp. 3–21, IOS Press, 1994.
[15] J. H. Holland, "Complex adaptive systems," *Daedalus*, vol. 121, no. 1, pp. 17–30, 1992.
[16] R. Gras, D. Devaurs, A. Wozniak, and A. Aspinall, "An individual-based evolving predator-prey ecosystem simulation using a fuzzy cognitive map as the behavior model," *Artificial life*, vol. 15, no. 4, pp. 423–463, 2009.
[17] M. Alfonseca and F. J. Soler Gil, "Evolving a predator–prey ecosystem of mathematical expressions with grammatical evolution," *Complexity*, vol. 20, no. 3, pp. 66–83, 2015.
[18] T. Ito, M. L. Pilat, R. Suzuki, and T. Arita, "Population and evolutionary dynamics based on predator–prey relationships in a 3d physical simulation," *Artificial life*, vol. 22, no. 2, pp. 226–240, 2016.
[19] C. Adami, J. Moore, F. Dyer, A. Hintze, and R. Olson, "Exploring the coevolution of predator and prey morphology and behavior," in *Artificial Life Conference Proceedings 13*, pp. 250–257, MIT Press, 2016.
[20] P. Sunehag, G. Lever, S. Liu, J. Merel, N. Heess, J. Z. Leibo, E. Hughes, T. Eccles, and T. Graepel, "Reinforcement learning agents acquire flocking and symbiotic behaviour in simulated ecosystems," in *The 2018 Conference on Artificial Life*, pp. 103–110, MIT Press, 2019.
[21] O. Witkowski and T. Ikegami, "How to make swarms open-ended? evolving collective intelligence through a constricted exploration of adjacent possibles," *Artificial life*, vol. 25, no. 2, pp. 178–197, 2019.
[22] Y. Zhang, S. Wang, and G. Ji, "A Comprehensive Survey on Particle Swarm Optimization Algorithm and Its Applications," *Mathematical Problems in Engineering*, vol. 2015, pp. 1–38, 2015.
[23] R. Taylor, *Predation*. Population and Community Biology, Springer Netherlands, 2013.