

Professor Vasile GEORGESCU, PhD
University of Craiova
E-mail: vasile.georgescu@feaa.ucv.ro

DESIGNING AND TUNING ADAPTIVE SYSTEMS THROUGH EVOLUTION, LEARNING AND MEME TRANSMISSION: BIOLOGICAL AND COMPUTATIONAL PARADIGMS

***Abstract.** With the contemporary advances in Computational Intelligence (CI), a formidable gain in diversity and effectiveness becomes more obvious. However, this machinery that has been successfully engaged in a huge transitive work seems to apparently lose the natural inclination towards reflexivity and introspection and thus appears to miss the big picture. Although biological evolution paradigms were fruitfully inspirational for artificial evolution theories and a massive part of CI appears to have bio-mimetic roots, it is no less true that biologists and computer scientists produce two parallel series of research and literature. The aim of this paper is to provide more insight into how computational paradigms drawn their inspiration from evolutionary biology. Intelligent adaptive systems, as artifacts designed by humans, can be largely seen as an effort of mimicking intelligent biological behaviors that were shaped, developed, adapted and tuned by natural evolution, learning mechanisms or enculturation. The latter concept is supported by the new field of memetics that promotes an evolutionary vision of culture, studying the replication, spread and evolution of memes, as cultural replicators. The main focus is on describing these mechanisms and the ways of applying them to designing and tuning intelligent adaptive systems. The approach based on memetic algorithms is finally benchmarked against a similar approach based on genetic algorithms, for tasks in machine learning such as evolving type-2 fuzzy inference systems.*

***Keywords:** Biological evolution, Bio-inspired metaheuristics; Hybridizations; Memetics; Memetic vs. Evolutionary algorithms: design differences and predictive performances.*

JEL classification: C22, C45, C51, C53, C63 G17

1. INTRODUCTION

Concepts and principles drawn from evolutionary biology were highly influential for the advancement of computational intelligence in general, and evolutionary computation in particular. The latter deals explicitly with building, applying and studying algorithms based upon the Darwinian principles of natural selection.

However, Darwinian evolution is far from being the only theory of interest for computational intelligence, inspired by evolutionary biology. The modern evolutionary synthesis also integrates complementary approaches to natural selection, such as the Baldwin effect, or the Waddington's genetic assimilation. The capability of the system to adapt through evolution is enriched, in this way, by the dimension of learning, which can be best defined in terms of phenotypic plasticity. In other words, learning is an aptitude carried out and (eventually)

refined by an individual during his lifetime span, whereas evolution is responsible for the changes in the genotypes that make up a population from generation to generation. The evolution and learning can equally be viewed as search processes; the genotypic evolution of the population is then associated to the global search and the phenotypic plasticity of individuals is associated to the local search.

It is largely assumed that there are “synergistic” effects between evolution and learning. Evolution shapes and tunes the ability of an adaptive system to perform learning. Conversely, learning may accelerate evolution at least under certain conditions.

Even Lamarckian evolution has become attractive for those approaches to evolutionary computing that attempt to boost the cognitive capabilities of individual through methods and principles derived from the Dawkins’ theory of cultural transmission (also known as memetics). Indeed, along with individual learning, cultural transmission is another important mechanism that can be exploited for enhancing the performance of adaptive systems. From the memetics viewpoint, it is argued that artificial life, unlike organic life, tolerates the inheritance of acquired characteristics in terms of simulation. Whether Lamarckian or Baldwinian evolution is more appropriate for carrying out cultural transmission is still under debate.

2. BIOLOGICAL THEORIES OF EVOLUTION

2.1. Lamarckian evolution: the inheritance of acquired characteristics

The French zoologist Jean-Baptiste Lamarck was led by the study of fossils to the conclusion that organisms have to change their behavior to survive, when environments change. Those features needed for survival are developed in each individual through use and/or non-use (e.g., enlarge or shrink), during his lifetime. Then the newly characteristics acquired by individuals are somehow passed on to their offspring, who can continue that development. New species may appear, eventually, over many generations, provided that enough differences have been developed. For example, if a giraffe stretched its neck for leaves, its offspring would inherit the longer neck, and continued stretching would make it longer and longer over several generations.

Lamarck's evolutionary theory was exposed in his book “*Philosophie Zoologique*” (1809). Translated in modern terms, his claim is that the phenotype acts on the genotype. Since there is no positive evidence of the inheritance of acquired characteristics, most biologists now consider that his theory is not substantiated: the characteristics acquired during the lifetime span of an individual are not directly inheritable.

2.2. Darwinian evolution: natural selection

Charles Darwin presented his theory of evolution in a famous book titled “*On the Origin of Species*” (1859). According to Darwin, natural selection is the process where organisms with higher reproductive success generate offspring, and

hence, propagate their feature through time. Only adapted features and behaviors are reinforced and selected. Natural selection operates at a population level, rather than on individuals themselves, and is grounded on three basic mechanisms: selection, mating and mutation. Selection relies on the capability of individuals to adapt and survive, when some environmental pressure challenges such capability. The genetic material is swapped between the individuals of a species by mating. This tends to homogenize the gene pool of the population. Mutations introduce the necessary variability in the spring, allowing for evolutionary diversification. Mutations can eventually lead to more fit individuals.

Changes in environment generate a selective pressure: the environment selects features contributing to survival, and tends to eliminate the others. Those with traits which help survival tend to survive and have more offspring, who inherit those traits (“survival of the fittest”, or natural selection). The modern evolutionary synthesis is nowadays called Neo-Darwinism and encompasses historical Darwinism and genetics (unknown in Darwin’s days).

2.3. Baldwinian evolution: phenotypic plasticity (adaptation through learning)

The American psychologist James Mark Baldwin, in his paper “*A New Factor in Evolution*” (1896), was concerned with the effects of lifetime learning (as a form of phenotypic plasticity) on the genotypic evolutionary process. The Baldwin effect (BE) shows how the phenotype is shaped by the interaction between lifetime development of individuals and evolution. Two adaptive processes are at the origin of the BE: the genotypic evolution of the population of individuals with learning capabilities; the phenotypic plasticity of individuals, defined as the ability of an organism to adapt to its environment through learning, during its lifetime. This plasticity of individuals dictates which ones will be able to survive and produce offspring, thus determining the course of evolution. BE postulates that trait plasticity itself would be favored by evolution when those plastic responses increased individual fitness. The presence of lifetime learning can influence the selective pressure for genetic traits. The coupling between phenotypic plasticity and genotypic accommodation gives rise to adaptive evolution. Organisms that possess heritable learning capabilities have a better chance to survive certain environment effects. Therefore, learning tends to accelerate evolution by raising the reproductive fitness. Behaviors that are learned or developed over the lifetime span of an individual become instinctual through their gradual encoding in the genome over many generations, via mutations or incremental improvements.

Unlike disproved Lamarckism, the BE does not require the direct inheritance of acquired characteristics. What is actually inherited is the capability to learn (i.e., to acquire those characteristics). This makes BE completely consistent with Darwinian inheritance mechanisms.

Typical examples of BE are: ability to learn, to increase muscle strength with exercise, to tan with exposure to sun.

2.4. Waddington's theory: genetic assimilation and phenotypic rigidity (canalization)

Genetic assimilation is a related, but different evolutionary theory, proposed by Conrad Hal Waddington. It occurs when a behavior that was once acquired in the phenotype becomes specified in the genotype. According to Waddington, the main deficiency of Darwinism is that all the variability observed in nature is unlikely to be produced only by mutations that occur at random. More importance should be paid to the effects induced by the environment in evolution.

Waddington (1942) defined genetic assimilation as a process “by which a phenotypic character, which initially is produced only in response to some environmental influence, becomes, through a process of selection, taken over by the genotype, so that it is formed even in the absence of the environmental influence which had at first been necessary”. In other words, since the phenotypes induced by the environment are adaptive, the selection on the developmental system tends to reduce the phenotypic plasticity (i.e., the responsiveness to the environment). As a consequence, after a number of generations of exposure to the environmental stimulus, the induced phenotype becomes *inherited (genetically assimilated)*. This reduction in lifetime plasticity is known as *canalization*, or *phenotypic rigidity*.

If *phenotypic plasticity* is not beneficial (i.e., have fitness costs), then genetic assimilation could be favored. Then it acts in the opposite sense compared to the Baldwin effect, by decreasing plasticity (i.e., increasing canalization) in a population subject to a given range of environmental conditions, provided that an increase in canalization is adaptive.

In many situations, organisms can benefit from phenotypic rigidity. They may slowly evolve rigid mechanisms that replace or augment their plastic mechanisms. Learning by trial-and-error (e.g., learning to avoid snakes) is typically a risky phenotypic plasticity. There can be advantages if the learned behavior is genetically assimilated and hence replaced with an instinctive response to such dangerous environmental challenges.

2.5. Comparing the Baldwin effect and genetic assimilation

Both the Baldwin and Waddington theories are based on the assumption that natural selection acts upon favorable mutations, and are thus both fully compatible with neo-Darwinian evolution.

In the Baldwin effect, selection may act in two ways: (1) by changing the mean trait values in the population, but without changing the level of plasticity; (2) by increasing the level of plasticity when acting on the phenotype: the individuals that are positively selected are the most plastic ones, possessing the most extreme phenotype.

By contrary, genetic assimilation should act to increase canalization (i.e., decrease plasticity) in a population subject to a given range of environmental conditions, provided that an increase in canalization is adaptive. The assumption associated with genetic assimilation is that plasticity itself will evolve.

It is possible to have both the Baldwin effect and genetic assimilation occurring simultaneously for a specific trait in different lineages or for different traits within a lineage. They can equally promote evolutionary diversification in nature. Phenotypic plasticity allows expansion into novel environments, whereas genetic assimilation allows adaptive genetic divergence among populations.

3. EVOLUTION, LEARNING AND ENCULTURATION (MEME TRANSMISSION)

There are three main explanatory mechanisms of the way a system is adapting to its environment: evolution, learning and enculturation (meme transmission).

3.1. Evolution

Evolution is a slow adaptive process that acts incrementally and through which the basic characteristics of a species are dynamically adjusted. It is based on the Darwinian principle of natural selection (survival of the fittest), operates at a population level, rather than on individuals themselves, and is grounded on three principles: variation, competition and heritability. In the modern evolutionary synthesis, genetics is the biologically adequate theory of heredity. Genes provide the mechanism of natural selection and are passed on by survival of a species. They thus underpin Darwinism, forming a keystone of evolutionary theory. While evolution is ubiquitous and fundamental to all of biology, it also plays a crucial role in intelligent systems since it has shaped both very biological systems and the computationally intelligent systems that mimic the former. Within the latter it is best viewed as a type of optimization process in time and space. The environment is the fundamental driver of this optimization process.

3.2. Learning

Learning is based on the connectionist model of the brain and is concerned with the gradual improvement of the adaptation capability to the environment, at an individual level, in an attempt of tuning the structure and behavior characteristics of the individual. It is local in space and time, since it occurs in a single individual, during its lifetime, and allows for more plasticity in how an individual can better cope with its environment.

The evolutionary value of learning consists in accelerating the evolution of an adaptive trait. This typically happens in a dynamical environment, where phenotypic plasticity (such as the learning capability) is beneficial for evolution and can smooth the fitness landscape. The Baldwin effect provides the explanatory framework for this case.

However, learning is not always advantageous. It is often expensive, since requires experiments, involving acquisition of data and sensors. Static environments typically favor instinctive mechanisms, which are less expensive than learned ones. A trait that initially has been carried out by a learning

mechanism may eventually become common and robust in the population. This gives evolution a long enough time to find an instinctive mechanism for replacing the learned one. Prototypical for this case is the Waddington's theory of genetic assimilation (canalization).

In designing artificial adaptive systems, one can chose to follow the principle of Lamarckian evolution and to directly integrate learned behaviors into genotype, by building a mapping from phenotype to genotype, if possible.

3.3. Contrasting evolution and learning

Evolution and learning have different goals. The goal of evolution is to maximize fitness. Learning is used, instead, by individuals in an attempt to help them achieving their immediate goals, which may not match with the "goals" of evolution.

Evolution has produced a type of trade-off between instinctual traits (taken over by the genome) and traits learned via the interaction of the organism with its environment. An appropriate amount of learning has evolved to help ensure reproductive success.

Both the evolution and learning appear to search the fitness landscape for a better position. The current location and the shape of the fitness landscape dictate the right trade-off between evolution, which performs a kind of global search, and learning, which performs a kind of local search.

There are different time scales to which evolution and learning operate.

Phenotypic plasticity is more advantageous in a dynamic environment, where evolution cannot adapt fast enough and learning may be used by individuals as a compensatory mechanism. Thus, learning can increase the variability in the population, which subsequently can accelerate evolution. It actually smoothes the slope of the fitness landscape. This is the specific context in which synergetic effects between evolution and learning normally appear.

In a static environment, instead, evolution accommodates better to the time scale dimension and thus can adapt, favoring phenotypic rigidity (instinctual traits are not penalized). There are some risks that come with learning, since it strongly relies on experience, and is hence more stochastic than the instinct. While the slope of the fitness landscape is already smooth, learning may have little advantage over instinct. Instincts are brittle and thus more stable. In such specific contexts there are no synergetic effects. The seminal work "How Learning Can Guide Evolution" by (Hinton and Nowlan, 1987) provided much insight into the interplay between evolution and learning.

3.4. Enculturation (meme transmission)

Culture is an essential part of the human adaptation. Individual learning can be complemented by cultural transmission that has more global influences among large social groups. Dawkins (1976) coined the term *meme* for the basic unit of cultural transmission, analogous to the gene in biological evolution. Memes live in human brains and compete with each other for resources such as time and

storage space, under the pressure of limiting factors. Time is perhaps the subject of most heavy competition. Some memes are more successful in the meme-pool than others. Particular qualities that allows for high survival value among memes are longevity, fecundity, and copying-fidelity. Memes propagate themselves in the meme pool by spreading from brain to brain via colonization and a replication process, based on imitation. A meme colonizes a human brain and tends to mature into an idea. Subsequently, that idea may be eventually communicated to another person, by means of a communication process that involves transmitting the meme. The analogy between memes and genes suggests that memes are better referred to as the devices by which we express our ideas and are analogous to genotypes, whereas the ideas themselves are analogous to phenotypes. Meme transmission gives rise to a form of evolution that is fully compatible with the Darwinian evolution. The transmitted meme results from the mutation and recombination with other memes in the host brain.

Since ideas (identified with the phenotypes of memes) have themselves a certain degree of plasticity, the Baldwin effect is also a good candidate for evolving memes. The ideas that populate the host brain are more or less consistent with each other, more or less appealing, and more or less plastic. The fitness landscape is defined by the degree to which they fit into the ecology of the brain. Plastic ideas have the potential to adapt, but they have to be processed by the host brain and thus consume brain time. The ideas derived directly from memes are not plastic at all (are just imitations), but benefit from the fact that need not to be processed and thus do not consume brain time.

Ideas that are highly consistent with the cultural environment are devised as meme and enter the meme transmission process.

The generic denomination of *Memetic Algorithms*, also called the *Cultural Algorithms*, is used to encompass a class of metaheuristics that have been inspired by Dawkins' notion of meme. A memetic algorithm is a kind of evolutionary algorithm that makes use of local search, wherein evolution and learning are combined using the Lamarckian or Baldwinian strategy. It actually combines local search heuristics with genetic operators.

3.5. Differences between biological and artificial evolution

An apparently surprising fact is that in the evolution of biological systems, there is no fitness function. Nature is content to produce descendants. Regulation is due to environmental pressure (external conditions that tend to influence the individuals' development and behavior) as well as phylogeny, and morphogenesis, which are supported by some "rationales" internal to the genomes.

On the other hand, in artificial evolution, someone is almost obliged to define fitness functions, from the moment when she or he is interested in measurements in the phenotypic space (the performance, among others).

While biological evolution uses a fundamental intermediate level for morphogenesis (the protein space), artificial evolution manipulates non-living

objects or structures, and works in a very short time. In addition, it generally works with a fixed genomic structure, very useful for calculations and direct genotype-phenotype correspondence.

4. METAHEURISTICS AND HYBRIDIZATION STRATEGIES

4.1. Generalities

Metaheuristics are some form of stochastic computational approach to an optimization problem, aiming at finding the global optimum, while avoiding being trapped in other local optima. They work in an iterative way as a means to gather information, to explore the search space and to cope with problems such as combinatorial explosion. Many of them are often inspired by natural systems in many fields such as evolutionary biology (e.g., evolutionary and genetic algorithms), physical systems (e.g., simulated annealing, gravitational search algorithm), ethology (e.g., ant colony algorithm, particle swarm optimization) and so on.

Metaheuristics are organized around the following three concepts: - diversification (exploration); - intensification (exploitation); - storage (learning).

Diversification or exploration is the process that directs the procedure to gather new information about the problem to be optimized. The simplest diversification strategy is to periodically restart the search process from a randomly generated solution or judiciously choosing the set of admissible solutions in a region not yet visited.

On the other hand, intensification or exploitation uses information already collected to explore in details the areas considered promising in the search space. Its implementation lies, most often, in the temporary widening of the current solution neighborhood.

As for the storage, it is the support of learning, which allows the algorithm to only consider the areas where the global optimum is likely to be found, thus avoiding local optima that are good solutions, but not the best possible solutions. Thus, alternating intensification, diversification and storage, metaheuristics operate in a progressive and iterative way. The initial step is often selected randomly and the final step is often set by means of a stop criterion. All metaheuristics are based on the balance between intensification and diversification of the search. Otherwise, there will be too rapid convergence to local optima by lack of diversification, or too long exploration by lack of intensification.

Taking into account the way of manipulating solutions, there are three fundamental approaches: local search (trajectory-based), constructive and population-based ones.

The local search approach is based on making small changes iteratively to a single solution, called the current solution, and constructing a trajectory in the space of solutions, by attempting to move towards optimal solutions. From a starting point x_0 , the search consists of passing stepwise from solution to solution, in a neighborhood consisting of all solutions. Often the local search operators stop

when a locally optimal solution is found. But accepting this type of solution is not always satisfactory. It is so important to get out of these local minima by allowing the local search operator to find points for which the new solution chosen will be better than the previous ones. This is the principle adopted for the Hill Climbing, Simulated Annealing, or Tabu Search algorithms.

The constructive approach is based on constructing solutions from their constituents, by adding the best possible element to a partial solution, iteration by iteration. Greedy algorithms belong to this class and have GRASP (Greedy Randomized Adaptive Search Procedure) as an important representative (Feo and Resende, 1995). Another example is Ant Colony Optimization (ACO) (Dorigo et al., 1996) that mimics the foraging behavior of ants.

The population-based (or perturbation) approach relies on selecting and then combining existing solutions, iteratively, into new ones, from a set called population.

4.2. Evolutionary algorithms: representation and design

Evolutionary Algorithms (EAs) are a special class of metaheuristics, founded on the principles of artificial evolution. Historically, several families of EAs were developed independently. The *Evolution Strategies* (ESs) was proposed by I. Rechenberg starting from 1965, to solve continuous optimization problems. Fogel, Owens and Walsh, in the 1960s, conceived *Evolutionary Programming* (EP) as an artificial intelligence method for the design of finite state automata. Genetic Algorithms (GAs) were proposed by J. Holland (in 1973-1975) for combinatorial optimization and later popularized by D.E. Goldberg (1989). Genetic Programming (GP) appeared initially as a subdomain of GAs, and was matured by J. Koza (1992). GP is specifically concerned with the automatic construction of programs and can be seen as artificial evolution of programs represented as trees. Differential Evolution (DE), proposed by Price and Storn (1995), is one of the EAs that have experienced great development in recent years. DE is a stochastic meta-heuristic for multidimensional function optimization, originally designed for continuous, unconstrained problems, but can be extended to handle mixed-variable problems, as well as non-linear constraints. It is inspired by GAs and ESs, combined with a geometric search technique.

Formally, EAs are stochastic global optimization algorithms of order 0: no property of continuity or differentiability is necessary for the smooth running of the method, only the knowledge of the values of the function to be optimized at the sampling points is required.

We seek to optimize a function f with real values, defined on a search space S . A specific vocabulary is used:

- the objective function f is called *fitness (performance)* function;
- the points of the search space S are called *individuals* ;
- the tuples of individuals are called *populations* ;
- we are talking about a *generation* for the main loop of the algorithm.

The evolution time is supposed to be discrete, and we denote by P_t the population of fixed size N at the generation t .

The environment pressure, which is simulated using the fitness function f , Darwinian principles of natural selection and uncontrolled variations are implemented in the algorithm as follows:

- initialization of the population P_0 by choosing N individuals in S , generally by drawing at random with a uniform probability on S ;
- assessment of individuals from P_0 (calculation of the values of f for all individuals);
- the generation t builds the population P_t from the population P_{t-1} :
 - selection of the best performers of P_{t-1} in the sense of f (most suitable reproduce);
 - application of genetic operators (with a given probability) to the selected parents, which generates new individuals, the children. We speak of mutation for unary operators, and of crossover for binary (or n -ary) operators. Note that this step is always stochastic;
 - evaluation of children;
 - replacement of the P_{t-1} population by a new population created from the children and/or old parents of the P_{t-1} population by means of Darwinian selection (the most adapted survives).
- evolution stops when the desired level of performance is achieved, or a fixed number of generations has passed without improving the best performer.

The main component of the algorithm, which is actually prior to all the others, is the representation of the search space. In many cases, the search space is totally determined by the problem: it is the space S on which the objective function f is defined. But it is always possible to transport one's problem into a skillfully chosen space, in which it will be easier to define efficient genetic operators. This space is then called *genotypic space*, and the initial search space S , in which the performance of individuals is calculated, is called *phenotypic space*. Historically, the *bitstring* representation of the search space, $S = \{0,1\}^N$ has been initially used by the school of GAs (a string of bits being assimilated to a chromosome). Alternatively, S can be represented by real-valued vectors, i.e., as a bounded or non-bounded subset of R^n , which is the n -dimensional real vector space.

The various stages of the algorithm can be divided into two groups: those related to artificial Darwinism (selection and replacement), which depend only on the values taken by f , and not on the chosen representation (i.e., genotypic space); and those intimately related to the nature of this research space. Thus, initialization and genetic operators are specific to each genotype, but do not depend on the objective function f (this is the Darwinian principle of uncontrolled variations).

Typically, selection and crossover operations are exploitation steps, while initialization and mutation are exploration steps (although certain EA variants deviate from this general pattern). A trade-off between exploration and exploitation

can thus be obtained by playing on the various parameters of the algorithm (probabilities of application of the operators, selection pressure, etc.).

The essential difference between the selection step and the replacement step is that the same individual can be selected several times during the selection step (which corresponds to having several children), while during the replacement stage, each individual is selected once (and he or she survives) or not at all (and he or she disappears forever). Furthermore, the replacement procedure may involve either the children alone or the previous population as a whole.

There are two categories of selection or replacement procedures: *deterministic* and *stochastic*.

In deterministic selection, the individuals are first evaluated with the fitness function. The less successful individuals are totally eliminated from the population, and the best individual is always selected. It is said that this selection is *elitist*. *Deterministic tournament* selection uses only comparisons between individuals and does not even require sorting of the population. It has a parameter of arity T , the size of the tournament. To select an individual one draws T evenly in the population, and selects the best of these T individuals.

In stochastic selection we resort to a stochastic procedure, which gives a chance to the less efficient individuals. It also may happen that the best individual is not selected, and that none of the children achieves a performance as good as that of the best parent. Roulette wheel selection consists of giving each individual a probability of being selected proportional to his performance. *Stochastic tournament* selection is probably the best current selection mode to finely adjust the selection pressure, and the fastest also. It consists in giving a real parameter t between 0.5 and 1, in choosing uniformly 2 individuals (with or without replacement) and in retaining only the best with probability t .

An *evolution engine* is a union of a *selection procedure* and a *replacement procedure*.

The components of the algorithm that depend intimately on the chosen representation are, on the one hand, the *initial population*, obtained by sampling, as uniformly as possible, the search space S , and on the other hand the *genetic operators*, which create new individuals from the selected parents. We distinguish *crossover* operators (binary, or more generally n -ary) and *mutation* operators, unary. The *crossover* operator exchanges genetic material between parents. The *mutation* operator allows visiting all the search space. It should be quasi-ergodic, in the sense that any point of the search space can be reached in a finite number of mutations. The confrontation of early artificial evolution theories with empirical results has given rise to debates about the representation and the respective importance of the crossover and mutation operators. In GAs, crossover is the main search operator ensuring recombination. On the contrary, ESs and EP mainly use mutation. The first versions of ESs and EP did not include any crossover. In the same way that the GA look at the mutation as a repair operator (re-injection of elementary bricks lost due to sampling errors in the population), the ES consider

the crossover as a repair operator. It serves to extract the similarities of the good individuals of the population to repair the excess of noise introduced by the mutations, which increases the speed of convergence.

Facilitating the adjustment of parameters is one of the stakes of current theoretical research on EAs. Recent work on the convergence of these algorithms provided a rich theoretical framework, not only reduced to parameter adjustment techniques, but also a more in-depth understanding of when and why an EA is effective. The design of self-adaptive algorithms, where the parameter adjustment is automatic, brings us closer to the perspective of a more flexible and autonomous design of our optimizers: to let the evolution take care of itself.

4.3. Hybridization strategies

Hybridization consists of combining the characteristics of two different methods to get the advantages of both methods.

According to the taxonomy proposed by Talby (2002), hybridization of metaheuristics with themselves involves two main classifications: a hierarchical classification and a flat classification.

The hierarchical classification of metaheuristics is characterized by the level and method of hybridization. The hybridization may be low-level, or high-level. In the low-level hybridization, a metaheuristic replaces one operator of another method that encompasses. By contrary, in the high-level hybridization, each metaheuristic preserves its integrity during hybridization. Each level of hybridization generates two types of cooperation, namely, the relay mode and the co-evolutionary mode. In the relay mode, the methods are performed sequentially, that is to say the output of the first method is the input of the next method. When the different methods work in parallel to explore the search space, this mode is called co-evolutionary. The combination of modes and levels gives four classes of hybridizations: low-level relay hybridization, low-level co-evolutionary hybridization, high-level relay hybridization and high-level co-evolutionary hybridization.

The **low-level relay hybridization** encompasses the class of single solution metaheuristics, in which another method is incorporated to form a new algorithm.

The **low-level co-evolutionary (teamwork) hybridization** consists of incorporating one or more single solution metaheuristics into a metaheuristic with a population of solutions. The advantage of this type of hybridization is to offset the operating power of local search and the exploration power of a global search.

The **high-level relay hybridization** occurs when metaheuristics are used sequentially that is to say the final solution(s) of the first metaheuristic is/are the initial solution(s) of the next metaheuristic. In this procedure, all methods remain intact.

In the case of **high-level co-evolutionary (teamwork) hybridization**, the employed metaheuristics work in parallel, exchanging information between them to find the optimal solution of the problem.

The flat classification of metaheuristics is characterized by the type of hybridization methods, their scope and the nature of their purposes. Depending on the type of hybridization, there are homogeneous hybrid methods, where the algorithms are based on the same metaheuristic, and heterogeneous hybridized methods, where the metaheuristics are different.

The global hybridization occurs when all hybridized methods are applied to the entire search space. By contrast, the partial hybridization decomposes a problem into sub-problems where each one has its own search space.

4.4. There is “No-Free-Lunch”

Theoretical and empirical results came to support the idea that looking for a metaheuristics that is claimed to be generically the winner in competition with all the others is unrealistic. In their famous theorem of the “No-Free-Lunch” (NFL), Wolpert and McReady (1995) showed that there is no optimizer that outperforms its competitors on average, on all optimization problems. What an optimization algorithm has gained on a problem (compared to all other algorithms) is lost on another problem. However, we must be careful not to naively interpret the NFL by concluding that it is useless to improve the optimization methods since they will never do better than, for example, a random search. Actually, this theorem states that optimizers perform differently on different classes of functions. To be relevant, optimization research must link the algorithm to the problem.

5. MEMETIC ALGORITHMS

It is now well established that hybridization of evolutionary algorithms (EAs) with other techniques can greatly improve the efficiency of search. EAs that have been hybridized with local search techniques (LS) are often called *Memetic Algorithms* (MAs). MAs were introduced by Moscato (1989, 1999) and further developed by Merz (2000). They were also inspired by the concept of a *meme* coined by Richard Dawkins (1976). A meme is taken to represent a learning or development strategy. Thus, in the case of MAs, memes can be thought of as representing alternative improvement strategies that could be applied to solutions, where these strategies may be imitated, improved, modified, etc.

In Talbi’s hierarchical taxonomy of metaheuristics, MAs could be placed within the class of low-level co-evolutionary (teamwork) hybrids. In other words, an MA is an EA that includes one or more local search phases within its evolutionary cycle. The rationale behind this hybridization is to compensating for the deficiency of EAs in local exploitation and the inadequacy of LSs in global exploration.

MAs actually combine a Darwinian evolutionary approach to optimization with a Lamarckian or Baldwinian adaptation (learning) approach, by embedding a LS into a GA. For example, if we re-inject the improved individuals into the population, it is Lamarckism. Baldwinism, meanwhile, is implemented in an AE by

improving the individual through a local search, then evaluating its fitness after modification, but leaving its genome intact. The Lamarckian evolution allows a faster convergence than the Darwinian evolution, but the risk of being blocked in a local optimum is higher. In Baldwinian evolution, an individual who can potentially lead to a good solution is more likely to be chosen, and this approach avoids a precipitous convergence towards a local optimum (the evaluation of individuals after a local search results in smoothing the fitness function, which makes it easier to find good solutions). Thus, a memetic model of adaptation exhibits the plasticity of individuals that a strictly genetic model fails to capture.

A common design for MAs is to apply LS to members of the EA population after recombination and mutation, in order to intensify the exploitation of the best search regions gathered during the global sampling done by the EA. The choice of the local optimization heuristic depends on the problem at hand.

6. COMPARING GAs AND MAs WHEN EVOLVING TYPE-2 FUZZY LOGIC SYSTEMS

In Georgescu (2017) we introduced a new approach to evolving an Interval Type-2 Fuzzy Logic System (IT2FLS) for bankruptcy prediction, by using GAs. In this section, we extend our work to using MAs and contrast these two alternative approaches in terms of design differences and predictive performance.

6.1. Brief presentation of type-2 fuzzy logic systems

The advantage of an IT2FLS is that of representing and capturing uncertainty with more degrees of freedom, which allows it, presumably, to outperform its type-1 counterpart. The GA-based approach has been specifically designed to solve a mixed integer optimization problem that is consistent with our strategy of training an *IT2FLS* from scratch. This led us to use genetic operators such as *Tournament selection*, *Extended Laplace crossover* and *Power mutation*. See Georgescu (2017), for more details.

Fig. 1 shows the primary and secondary membership functions of a general type-2 fuzzy set, where *UMF* and *LMF* are the Upper Membership Function and Lower Membership Function, respectively, and the hashed region is the so called *footprint of uncertainty* (FOU).

An IT2FLS is similar to a type-1 FLS, containing a fuzzifier, rule base, fuzzy inference engine, and output processing. Additionally, a type-2 FLS has a type-reducer in the output processing, which has the ability to generate a type-1 fuzzy set from a type-2 fuzzy set. The defuzzifier then can defuzzify this type-1 fuzzy set to a crisp number. The Karnik-Mendel (KM) algorithms and their extended counterparts (EKM) are the most known algorithms for the centroid type-reduction and defuzzification stage. Alternative approaches, based on faster approximations have been also proposed, such as the *Wu-Tan* and the *Nie-Tan* methods.

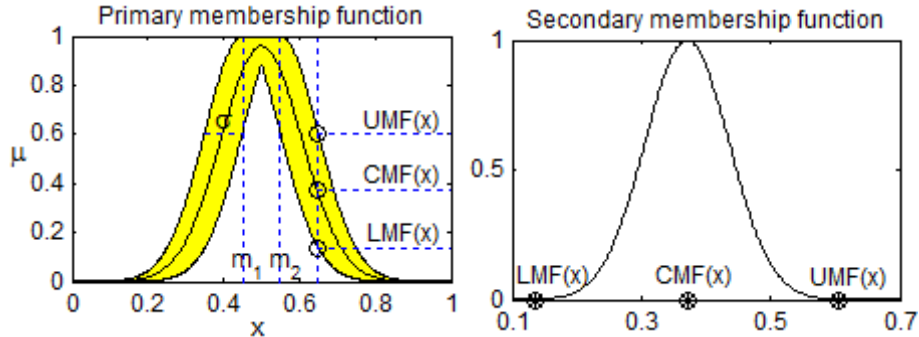


Figure 1. Primary and secondary membership functions of a general type-2 fuzzy set

6.2. Representation and design with GAs

The rationales behind using GA as a metaheuristic for optimization are that it benefit from specialized genetic operators, such as *Tournament selection*, *Extended Laplace crossover* and *Power mutation*, which were specially designed to handle optimization problems with integer variables. This is crucial when evolving an *IT2FLS* from scratch, meaning that the shapes and positions of the *IT2MFs*, along with the rules of an *IT2FLS*, are simultaneously evolved.

Four types of Interval Type-2 Membership Functions (*IT2MFs*) are used in our implementation: *IT2zmf*, *IT2gaussmf*, *IT2trimf* and *IT2smf*. We encode the type of *IT2MFs* by an integer, $tmf \in \{1,2,3,4\}$. Let $[x_a, x_b] \subseteq X$ be the interval spanned by an *IT2FS*. If we denote by $m = (x_a + x_b)/2$, $s = (x_b - x_a)/2$ and e , respectively, the *middle point*, the *spread* and the *extent of uncertainty*, each of the four *IT2MFs* mentioned above can be uniquely encoded by means of four parameters: (tmf, m, s, e) . See Fig. 2.

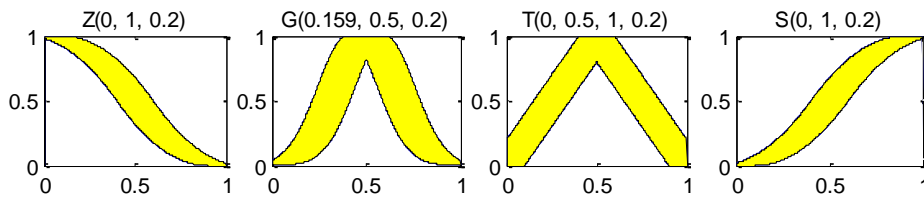


Figure 2. *IT2MFs*, with $x_a = 0, x_b = 1$ and $e = 0.2$, or $m = (x_a + x_b)/2 = 0.5$, $s = (x_b - x_a)/2 = 0.5$.

In order to evolve an *IT2FLS* by means of a GA, we need to encode each candidate solution of the optimization problem into a chromosome. Let us assume that the *IT2FLS* consists of NI input variables, one output variable and NR rules. We consider that the universes of discourse of all input and output variables are

covered by fuzzy partitions with the same number of *IT2FSs*, say NF . The meaning of each gene depends upon its position in the chromosome. The first $(NI+1) \cdot NF \cdot 4$ genes will encode the fuzzy space defined by all the *IT2FSs* that generate the fuzzy partitions of the input and output variables. Each fuzzy rule in the rule base is also encoded by $(NI+1)$ genes, with values represented by an integer in the following set: $\{-NF, -NF+1, K, -1, 0, 1, K, NF-1, NF\}$. An integer in the first NI positions will indicate an *IT2FS* selected for the corresponding input variable, while an integer in the last position, $NI+1$, will indicate an *IT2FS* selected for the output variable. The integer 0 points out the absence of a certain variable in that rule and a negative integer shows that the negation of an *IT2FS* is to be selected for the corresponding variable in the rule. A rule without a nonzero antecedent and consequent part is not a feasible rule and will not be included in the rule base.

Finally, each chromosome consists of $(NI+1) \cdot NF \cdot 4 + (NI+1) \cdot NR$ genes and is part of a population of chromosomes represented in the hyperspace of potential solutions. First, an initial chromosome population is randomly generated. Then, the population will undergo genetic operations such as *selection*, *crossover*, and *mutation* to evolve and optimize chromosomes. Every chromosome is assigned a fitness value, and then a selection operator is applied to choose relatively 'fit' chromosomes to be part of the reproduction process. The crossover and mutation operators are closely related to the encoding scheme of *MFs* and rules. The fitness function uses the error rate between desired outputs and estimated outputs. Therefore, the fitness value is evaluated based on simulating the *IT2FLS* on a given learning dataset. See Georgescu (2017) for more details.

6.3. Extension from GA to MA, by embedding the Simulated Annealing Algorithm into GA

As an alternative to GA we use now a MA, where the local search improvement is performed by embedding a Simulated Annealing Algorithm into GA.

The Simulated Annealing (SA) algorithm was inspired by the process of annealing in metallurgy, where a material is heated and slowly cooled under controlled conditions, in order to reduce the possible defects in the material by increasing the size of the crystals in its structure. This results in an improvement of its strength and durability. The energy and mobility of atoms are increased by heating, while a slow cooling allows a new low-energy configuration to be found and exploited.

SA is an adaptation of the Metropolis-Hastings Monte Carlo algorithm. The use of SA for solving combinatorial optimization problems is much more recent and dates back to the 1980s (Kirkpatrick, 1983; Cerny, 1985).

The idea is to perform a movement according to a probability distribution that depends on the quality of the different neighbors: the best neighbors have a higher probability; the worst ones have a lower probability. A parameter, called the

temperature and denoted by T , is used. For higher T , all neighbors have about the same probability of being accepted. For lower T , a movement that degrades the cost function has a low probability of being chosen. For $T = 0$, no degradation of the cost function is accepted. The temperature varies during the search: T is high at the beginning, then decreases progressively towards 0.

The pseudo-code of SA is given below:

```

Let  $s = s_0$ 
For  $k = 0 : k_{\max}$  (exclusive):
     $T \leftarrow \text{temperature}(k/k_{\max})$ 
    Pick a random neighbour,  $s' \leftarrow \text{neighbour}(s)$ 
    If  $P(E(s), E(s'), T) \geq \text{rand}(0, 1)$ :
         $s \leftarrow s'$ 
Output: the final state  $s$ .
    
```

Notations: s is the current state; s' is the new (candidate) state; $E(s)$ is the energy of state s ; $E(s')$ is the energy of state s' ; k_{\max} is the maximum number of steps; $P(E(s), E(s'), T) \geq \text{rand}(0, 1)$.

The call $\text{rand}(0, 1)$ returns a random number uniformly distributed in $[0,1]$.

Two similar *IT2FLS* architectures have evolved: one using GA (called *GA-IT2*) and another using MA (*MA-IT2FLS*). Their performances were tested and compared by considering a problem of bankruptcy prediction. A sample of 130 Romanian companies has been drawn from those listed on Bucharest Stock Exchange (BSE). The binary variable to be predicted is *entering insolvency or not*. As predictors, a selection of the most relevant 8 financial ratios has been used.

A particular configuration of an *IT2FLS* evolved using our available training dataset is shown in Fig. 3. It consists of 8 rules, 8 antecedent variables and one consequent variable. Our experimental setup was to use the same training dataset in order to evolve 100 configurations for both the *GA-IT2FLS* (evolved with GA) and *MA-IT2FLS* (evolved with MA) and to compare their in-sample and out-of-sample average classification error rates.

The results are presented in Table 1. We conclude that the percentage of misclassified companies, which results, in average, using *MA-IT2FLS* is significantly lesser than that using *GA-IT2FLS* for both the in-sample and out-of-sample average classification error rates. This actually means that MAs outperforms GAs when used to evolve *IT2FLS*s.

Table 1. In-sample and out-of-sample average classification error rates

	<i>GA-IT2FLS</i>	<i>MA-IT2FLS</i>
In-sample average classification error rate	6.14 %	5.83%
Out-of-sample average classification error rate	7.35 %	6.67%

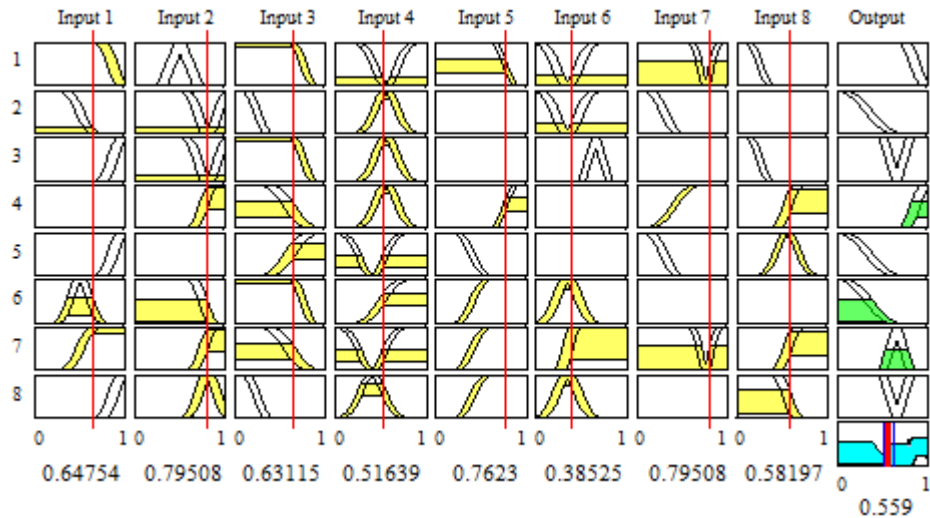


Figure 3. Rule viewer of the IT2FLS

7. CONCLUSIONS

The design of so-called “intelligent systems” through models inspired by biological intelligence has been highly successful in the last decades. Computational intelligence is a branch of AI that is concerned with the study of adaptive mechanisms allowing for intelligent behavior in complex and changing environments. There are different visions underlying the notion of intelligence in humans, animals and machines. However, all computational paradigms are deeply rooted in biological paradigms, from basic adaptive mechanisms in simple organisms, passing through evolutionary and learning processes that undergo all forms of life, to cognition and human thinking processes.

This paper has been focused on describing the mechanisms of evolution, learning and meme transmission and the ways of applying them to designing and tuning intelligent adaptive systems. Nature-inspired metaheuristics are very appealing and versatile techniques to help achieving such goals. They proved to be very effective when dealing with difficult optimization problems, without having to modify the basic structure of the algorithm used and became more and more popular thanks to their ease of use in different areas. It should be noted that good performance often requires adequate formalization of the problem and intelligent adaptation of a metaheuristic. Hybridization can be used to improve the effectiveness of an optimization solver by harnessing the power of two or more metaheuristics and combining them into a single one. This may give us new ways of controlling the exploration and exploitation capabilities of the newly generated hybrid.

We also provided experimental evidence that metaheuristics can serve for the purpose of evolving either type-1 or type-2 fuzzy logic systems to perform tasks such as classification or function approximation. Although type-2 fuzzy logic systems are several times more computationally time consuming due to the inclusion of a type-reducer in the output processing, it benefit instead from the advantage of representing and capturing uncertainty with more degrees of freedom. We used two approaches to evolving *IT2FLSs* – one based on MAs and another based on GAs – and benchmarked the former against the latter. The results showed that *MA-IT2FLS* outperformed *GA-IT2FLS* for both the in-sample and out-of-sample average classification error rates.

REFERENCES

- [1] **Baldwin, J. M. (1896)**, *A New Factor in Evolution*. Am. Nat. 30:441–451;
- [2] **Beyer, H.-G., Schwefel, H.-P. (2002)**, *Evolution Strategies – A Comprehensive Introduction*. Natural Computing 1:3–52;
- [3] **Crispo, E. (2007)**, *The Baldwin Effect and Genetic Assimilation: Revisiting Two Mechanisms of Evolutionary Change Mediated by Phenotypic plasticity*; *Evolution* 61-11: 2469–2479;
- [4] **Darwin, C.R. (1859)**, *On the Origin of Species*. John Murray, London;
- [5] **Dawkins, R. (1976)**, *The Selfish Gene*. Oxford: Oxford University Press;
- [6] **Dorigo, M., Maniezzo, V., Colorni A. (1996)**, *The Ant System: Optimization by a Colony of Cooperating Agents*. IEEE Transactions on Systems, Man, and Cybernetics– Part B, 26(1), 29–41;
- [7] **Feo, T.A., Resende, M.G.C. (1995)**, *Greedy Randomized Adaptive Search Procedures*. Journal of Global Optimization, 6:109–133;
- [8] **Flatt, T. (2005)**, *The Evolutionary Genetics of Canalization*. *Q. Rev. Biol.* 80:287– 316;
- [9] **Fogel D. B. (2006)**, *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence*, Wiley;
- [10] **Georgescu V. (2017)**, *Using Genetic Algorithms to Evolve Type-2 Fuzzy Logic Systems for Predicting Bankruptcy*; *Kybernetes*, 46(1):142-156;
- [11] **Georgescu V. (2016)**, *Using Nature-inspired Metaheuristics to Train Predictive Machines*; *Economic Computation and Economic Cybernetics Studies and Research*; *ASE Publishing*; 50(2): 5-24;
- [12] **Georgescu, V. (2016)**, *Exploring the Potential of Cuckoo Search Algorithm for Classification and Function Approximation*; *Lecture Notes in Business Information Processing*, vol. 254, 186-194, Springer;
- [13] **Glover, F. (1989)**, *Tabu Search - Part 1*. *ORSA Journal on Computing*. 1 (2): 190–206;
- [14] **Goldberg, D. E. (1989)**, *Genetic Algorithms in Search Optimization and Machine Learning*; Addison-Wesley, Reading, MA;

- [15] **Hinton G.E, Nowlan S.J.(1987), *How Learning Can Guide Evolution*; Complex Systems 1, 495-502;**
- [16] **Holland, J. H. (1975), *Adaptation in Natural and Artificial Systems*; University of Michigan Press, Ann Arbor, MI;**
- [17] **Karnik, N.N., Mendel, J. M. and Qilian, L. (1999), *Type-2 Fuzzy Logic Systems*; *IEEE Trans. Fuzzy Syst.*, Vol. 7 No. 6, pp. 643–658;**
- [18] **Karnik, N.N. and Mendel, J.M. (2001), *Centroid of a Type-2 Fuzzy Set*; *Inf. Sci.*, Vol. 132 No. 1–4, pp. 195–220;**
- [19] **Liang, Q. and Mendel, J.M. (2000), *Interval Type-2 Fuzzy Logic Systems: Theory and Design*; *IEEE Transactions on Fuzzy Systems*, Vol. 8 No. 5, pp. 535-550;**
- [20] **Kirkpatrick, S., Gelatt, C. D., Vecchi, M. P. (1983), *Optimization by Simulated Annealing*; Science, New Series, Vol. 220, No. 4598, 671-680;**
- [21] **Koza, J.R. (1992), *Genetic Programming: On the Programming of Computers by Means of Natural Selection*; MIT Press;**
- [22] **Talbi, E.(2002), *A taxonomy of Hybrid Metaheuristics*. Journal of Heuristics, tome 8, pages 541-564;**
- [23] **Waddington, C. H. (1942), *Canalization of Development and the Inheritance of Acquired Characters*. *Nature* 150:563–565;**
- [24] **Wolpert D.H., Macready W.G.(1997), *No Free Lunch Theorems for Optimization*; *IEEE Transactions on Evolutionary Computation*, Vol. 1, No. 1;**
- [25] **Wu, D. and Mendel, J.M. (2009), *Enhanced Karnik-Mendel Algorithms*; *IEEE Trans. Fuzzy Syst.*, Vol. 17 No. 4, pp. 923–934.**