

Replicator theory of Dawkins memes

Vladimír Kvasnička and Jiří Pospíchal

Department of Mathematics

Slovak Technical University

812 37 Bratislava, Slovakia

kvasnic@cvt.stuba.sk, pospich@cvt.stuba.sk

Abstract A cooperative coevolutionary model of replicators and memes is suggested, where a possibility of replicator learning is considered. A population is composed of composites that are represented by couples of a replicator and a meme. A fitness of such composite is calculated such that a gene assigned to a respective replicator is repaired by information contained from a respective meme, and then applying a short learning process a fitness is evaluated. The proposed coevolutionary method may be understood as an extension of original Hinton and Nowlan method for an explanation of the well-known Baldwin effect. Further information relevant for forthcoming evolution of replicators is conveyed by memes, it means that not only a learning process but also meme accelerate Darwinian evolution of replicators.

Keywords

coevolution, replicators, learning, Baldwin effect, memes.

1 Introduction

Darwinian evolution is a standard subject of interest in *Artificial Life* [1]. The main stimulus was observed at the end of eighties, when evolutionary algorithms suddenly emerged as a new paradigm of computer science based on the metaphor of Darwinian evolution. This paradigm may be traced back to 1932 when Seal Wright [23] has postulated an *adaptive landscape* (nowadays called the *fitness landscape or fitness surface*) and characterized Darwinian evolution as an adaptive process (nowadays we say optimization process), where the replicators of population is adapted in such a way that it reaches a local (even global) maximum on the fitness surface. Forty years later, this ingenious idea has been used by John Holland [17] as a metaphor for creation of *evolutionary algorithms*, that may be now interpreted as an abstraction of Darwinian evolution in a form of universal optimization algorithm [17] (see Dennett seminal book [9]).

In the higher degree of abstraction [1] a notion of biological individual can be identified with a replicator, which is represented by a linearly ordered genetic information content of the respective individual. Then we can talk about a population of replicators, which are replicated with a probability proportional to their fitness, while integral part of this replication process are mutations. Mutations introduce into replicators new information content, which may change fitness of offspring replicators created from parental ones by a copying process. New replicators push out from population replicators with smaller fitness. This basic reproduction cycle is continually repeated. After some time there exists is a high probability, that replicators

with new properties and substantially increased fitness emerge in population, and push out older replicators without these new properties.

Evolutionary algorithms [17] are an abstraction of the above-presented basic principles of Darwin's evolution theory. According to the way of fitness calculation three different levels of evolutionary algorithms may be considered:

- (1) *Fitness of replicators is determined exactly by their positions on the fitness landscape, i.e. it is determined entirely by replicator compositions, no other effects are included in the fitness calculation. This way of fitness calculation corresponds to standard GAs, where replicator are directly mapped on positive real numbers - fitness, without any intermediate or hidden considerations.*
- (2) *Fitness of replicator is influenced by the nearest neighborhood of their positions on the fitness landscape. It means that genotypes (assigned to respective replicators) are capable of learning in which they search through nearest neighborhood on fitness landscape. As a result of this process replicators with the same fitness can be distinguished by a presence of higher fitness positions in their nearest neighborhood, see Fig. 1. These replicators are more effective for further evolution process than those ones that are not able to look for higher fitness replicators in the nearest neighborhood. The role of learning in evolution theory is called the Baldwin effect [3], its first study by an evolutionary algorithm was done by Hinton and Nowlan [15] (cf. also refs. [4]). However, better modeling of a learning process should also represent a quality of a learning ability by a size of a neighborhood searched around a replicator. Really complex genotype then could theoretically code not only their position on a landscape, but to code their own learning algorithm as well.*
- (3) *Fitness of replicators is determined not only by the nearest neighborhood of their positions on the fitness landscape but also by the so-called memes that determine an information that is able to increase the fitness of replicators. The concept of replicator is enlarged to a couple of replicator and meme, both of them determine a fitness of replicators itself. Moreover, there should be introduced a notion of the so-called memetic interaction, where a replicator-meme complex called the donor creates a message (idea). Message is broadcasted throughout a subpopulation of the so-called recipients. These recipients incorporate the broadcasted message into their memes. Incorporation of memes into evolutionary algorithms was already studied by [6,19,21].*

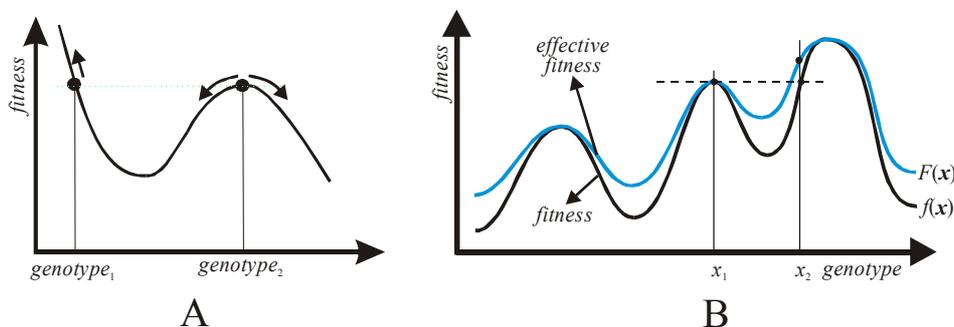


Figure 1. (A) Diagrammatic illustration of a situation when two replicators are placed on a fitness landscape so that they have the same fitness. The second replicator is situated at a local maximum and therefore it does not have a chance to increase its fitness in a nearest neighborhood. Though the first replicator has the same fitness as the second one, it is situated on the fitness landscape in such a position that in its nearest neighborhood there exist positions with greater fitness. From the standpoint of classical Darwin's evolution theory, when learning processes are fully ignored, these replicators could not be distinguished and therefore would have the same

chance to participate in a reproduction process. On the other hand, when the learning process is included into the evolution process, these replicators are distinguished, the first replicator is evaluated by a greater fitness and natural selection prefers this replicator. (B) An interpretation of Baldwin effect by making use a concept of effective fitness, which corresponds to an increased fitness due to the possible learning process. We see than on local maxima both fitness are the same.

All these three different levels of sophistication of evolutionary algorithms may be formally considered as evolutionary steps of Darwin's evolution itself. This view of the evolution distinguishes a complexity of fitness evaluation process. On the first lowest level the fitness is fully determined by replicator composition only. The second level corresponds to the previous first level that is enriched by a possibility of learning, i.e. the corresponding phenotypes are able of local search in their nearest neighborhoods. Finally, the third level is nothing but the second level enlarged by memes, i.e. phenotypes assigned to respective genotypes are not only capable of learning but also capable of employing memes in the fitness evaluation process. We have to emphasize that in all these three levels the evolution is strictly Darwinian, there is not postulated that memes or traits acquired by learning are inherited by offspring. Offspring replicators are fully determined by a reproduction process (that can be characterized as a copying of parent replicators plus small stochastic mutations). The learning process is used only for evaluation of replicators by fitness but not for their modification. For memes is situation slightly different. In the framework of memetic interaction a special type of learning process is used for the creation of a donor meme broadcasted throughout a subpopulation of recipients, where the donor meme transforms in a random way the recipient memes. This fact is in accordance with Dawkins' concept; memes may be transformed not only in the course of a reproduction process but also in the course of their lifetime.

Recently, memes [2,5] belong to very controversial concepts of theory of human culture. This concept was initially introduced by evolutionary biologist Richard Dawkins in his seminal book *The Selfish Gene* [8] as an information unit copied from a human brain to another human brain by imitation. Moreover, he postulated that memes have properties of replicators and therefore they may be a subject of evolution. Memetics (a science about memes) offers conceptually simple explanations about the nature and evolution of human culture, a paradigm of memes as replicators looks very attractive for audience mostly outside of social sciences. On the other hand, the memetics is strongly rejected by many social scientists as a theoretical approach based on dubious postulates and superficial analogies. Hot disputes are continued at the following three directions:

- (1) *Whether culture is properly seen as composed of independently transmitted information units,*
- (2) *whether memes have the necessary qualification to serve as replicators, and*
- (3) *whether evolutionary approaches such as memetics offer the most natural framework for a theory of culture.*

Aunger's *Darwinizing Culture* (Aunger, 2001) is the first book to attempt a through critical and conservative appraisal of the potential of memetics. There are summarized the points of agreement and/or disagreement on memetics and concluded with some suggestions for the way forward, particularly with respect to the means by which empirical research in this area may be undertaken.

The purpose of this communication is to suggest a replicator model of coevolution of replicators and memes [21], where it is postulated that a memetic environment may change selection of replicators; in other words, the replicator fitness is affected by memes. In our

approach memes are not independent from replicators, they form couples composed of a replicator and a meme (these couples are called the m-replicators). A subject of Darwinian evolution is a population composed of these couples. The memes coevolve simultaneously with replicators; a Darwinian selection there exists only for replicator-meme composites.

2 Artificial chemistry and a metaphor of chemostat

Let us consider a *chemostat* (a population) [10,13,20] composed of formal objects called the replicators. It is postulated that the chemostat is not spatially structured (in chemistry it is said that the reactor is well stirred). Replicators are represented by formal structured objects (e.g. token strings, rooted trees, λ -expressions, etc.). An interaction between replicators is potentially able to transform information, which is stored in the composition of the replicators. Therefore a chemical reaction (it causes changes in the internal structure of reacting replicators) can be considered as an act of information processing. The capacity of the information processing depends on the complexity of replicators and “chemical” reactions between them.

General ideas of the chemostat approach will be explained by an example of chemostat as a binary function optimizer that resembles many features of molecular Darwinian evolution. Let us consider a binary function – fitness

$$f : \{0,1\}^n \rightarrow [0,1] \quad (1)$$

This function $f(\mathbf{g})$ maps replicators \mathbf{g} represented by binary strings $\mathbf{g} = (g_1, g_2, \dots, g_n) \in \{0,1\}^n$ of the length n onto real numbers from the closed interval $[0,1]$. We look for an optimal solution

$$\mathbf{g}_{opt} = arg \max_{\mathbf{g} \in \{0,1\}^n} f(\mathbf{g}) \quad (2)$$

Since the cardinality of the set $\{0,1\}^n$ of solutions is equal to 2^n , a CPU time necessary for solution of the above optimization problem grows exponentially

$$t_{CPU} \approx 2^n \quad (3)$$

it means that the solution of the binary optimization problem (2) belongs to a class of hard numerical NP-complete problems. This is the main reason why the optimization problems (10) are solved by the so-called evolutionary algorithms [17], that represent very efficient numerical techniques how to solve binary optimization problems. The purpose of this subsection is to demonstrate that a metaphor of replicator provides an extremely simply but still efficient stochastic optimization algorithm.

Let a chemostat be composed of replicators that are realised by binary strings $\mathbf{g} = (g_1, g_2, \dots, g_n) \in \{0,1\}^n$. A “monomolecular” reaction is considered



where the formed replicator \mathbf{g}' substitutes a randomly selected replicator \mathbf{g}'' from the chemostat. Formally, a random selection of a replicator from the population P is realised by an operator $O_{select}(P)$, i.e. $\mathbf{g}'' = O_{select}(P)$. A term $f(\mathbf{g})$ assigned to the chemical reaction (4) is interpreted as a probability (rate constant) of a performance of reaction (4). In evolutionary algorithms a selection pressure in population of solutions (replicators) is created by a reproduction process based on replicator fitness. Replicators with a greater fitness have the greater chance to take part in a reproduction process (a measure of quality of replicators); on the other hand, replicators with a small fitness are rarely used in the reproduction process.

This simple manifestation of Darwin's natural selection ensures a gradual evolution of the whole population. In the present approach the mentioned principle of fitness selection of molecules is preserved, but it is now combined with an additional selection pressure due to a constancy of number of molecules in the chemostat. A molecule incoming to the reaction is randomly selected from the chemostat. After an evaluation of a quality of the selected molecule it is then stochastically decided whether the reaction is performed or not (see Algorithm 1), and moreover, the resulting molecule substitutes another randomly selected molecule. Finally, we specify a product \mathbf{g}' from the right-hand side of (4) as a mutation [17] of an incoming molecule \mathbf{g}

$$\mathbf{g}' = O_{mut}(\mathbf{g}) \quad (5)$$

where O_{mut} is a stochastic mutation operator that changes single bits with a probability P_{mut} . A pseudo Pascal code for the replicator algorithm is presented in Algorithm 1.

Algorithm 1.

```

P:=randomly generated chemostat of replicators  $\mathbf{g}$ ;
for epoch:=1 to epochmax do
begin  $\mathbf{g}$ :=Oselect(P);
      if random<prob( $\mathbf{g}$ ) then
        begin  $\mathbf{g}'$ :=Omut( $\mathbf{g}$ );
           $\mathbf{g}''$ :=Oselect(P);
          P:=(P+{ $\mathbf{g}''$ })-{ $\mathbf{g}'$ };
        end;
      end;
end;

```

The algorithm is initialized by a chemostat composed of randomly generated molecules (binary strings of the fixed length). A copy of a randomly selected molecule \mathbf{g} is transformed with a probability $\text{prob}(\mathbf{g})$ (specified by its functional value $f(\mathbf{g})$) by a mutation operator onto a new molecule \mathbf{g}' . The mutation operator causes an actual change only very rarely. This new mutated molecule substitutes a randomly selected molecule.

As an illustrative example we will study the chemostat approach specified for a simple unimodal function (fitness) determined over binary strings of the length 4. Let us postulate that a chemostat is formed by a multiset composed of binary strings of the length 4

$$P = \{\dots, (1100), \dots\} \subset \{0,1\}^4 \quad (6)$$

Each binary vector α is evaluated by a rational number from the closed interval $\langle 0,1 \rangle$

$$\text{real}(\mathbf{g}) = \frac{1}{2^4 - 1} \text{int}(\mathbf{g}) \quad (7)$$

where $\text{int}(\mathbf{g})$ is a nonnegative integer assigned to \mathbf{g} . A rate constant f assigned to the binary string \mathbf{g} is specified as follows

$$f(\mathbf{g}) = \frac{1}{2} (1 - \sin(2\pi \cdot \text{real}(\mathbf{g}))) \quad (8a)$$

with an optimal solution $\mathbf{g}_{opt} = (1011)$, where $\text{real}(\mathbf{g}_{opt}) = 11/15$ and $f(11/15) = 0.9973$. The probability of replication $\text{prob}(\mathbf{g})$ is specified as follows

$$\text{prob}(\mathbf{g}) = \exp\left[\alpha(f(\mathbf{g}) - f_{max})\right] \quad (8b)$$

where f_{max} is maximal value (upper bound) of the binary function $f(\mathbf{g})$.

The chemostat is composed of 1000 randomly generated binary strings and the mutation operator O_{mut} is specified by a 1-bit probability $P_{mut}=10^{-5}$. Obtained numerical results are displayed in Fig. 3. We see that those binary strings are spontaneously emerging in the chemostat, which correspond to a suboptimal solution with rational numerical value closely related to $real(\mathbf{g}_{opt})=0.25$. Main results of this Section may be summarized as follows: (1) *A metaphor of replicators offers an effective stochastic optimisation algorithm, where a proof of its convergence to a global solution immediately follows from Eigen's theory of replicators [11,16,18], where it is analytically proved that there exists a unique asymptotically stable solution with the greatest rate constant*, and (2) *if the probability of mutations is a very small number, then the obtained results are very similar to those ones obtained by replicator equations with mutations* (see Fig. 2).

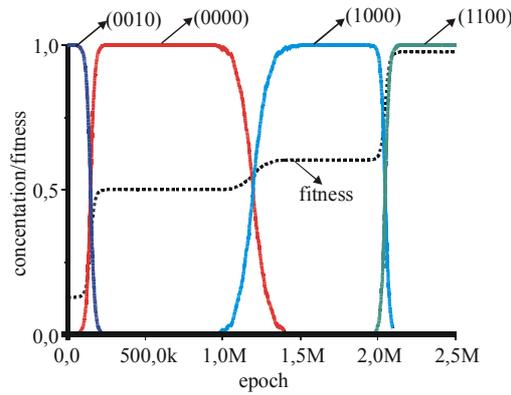


Figure 2. Plot of frequencies of appearance of some dominant binary strings (of 16) of the length 4. The chemostat was initiated by randomly generated 1000 binary strings, at the beginning of process chemostat was composed entirely of strings (0010), after 2.5×10^6 time steps most dominant final solution is $\mathbf{g}_{fin}=(1100)$, where $real(\mathbf{g}_{opt})=12/15$ and $f(12/15)=0.9755$. This final solution is juxtaposed to the optimal solution $\alpha_{opt}=(1011)$, where $real(\mathbf{g}_{opt})=11/15$ and $f(11/15)=0.9973$, but with a great Hamming distance $d=3$ (in theory of GA this effect is called the Hamming's cliff). This relatively great size of Hamming distance is the main reason why the algorithm is unable to achieve the global solution (1011).

Replicators with learning

Let a replicator $\mathbf{g} = (g_1 g_2 \dots g_n)$ be a linear string of the length n composed of symbols 0, 1, and # [15,19]

$$\mathbf{g} \in \{0, 1, \#\}^n \quad (9)$$

An *optimal replicator* \mathbf{g}_{opt} is randomly generated bit string of the length n

$$\mathbf{g}_{opt} = (g_1^{opt} g_2^{opt} \dots g_n^{opt}) \in \{0, 1\}^n \quad (10)$$

A *chemostat* (population) P is a multiset composed of p replicators

$$P = \{\mathbf{g}_1, \mathbf{g}_2, \dots, \mathbf{g}_p\} \quad (11)$$

Its *cardinality* $p=|P|$ is kept fixed throughout all generations of chemostat algorithm. Each replicator $\mathbf{g} \in P$ is evaluated by fitness. This evaluation process is connected with the *learning process* and it is determined as follows: Let us define a neighborhood of a given replicator $\mathbf{g} \in P$

$$U(\mathbf{g}) = \{\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_q\} \subseteq \{0, 1\}^n \quad (12)$$

This neighborhood is composed of q binary vectors that are randomly created from the reference replicator \mathbf{g} so that its hash symbols are randomly substituted either by 0 or 1

$$\mathbf{h} = (h_1 h_2 \dots h_n) \in \{0, 1\}^n \quad (13a)$$

$$h_i = \begin{cases} g_i & (\text{if } g_i = 0 \text{ or } g_i = 1) \\ \text{random}(2) & (\text{if } g_i = \#) \end{cases} \quad (13b)$$

The cardinality $q=|U(\mathbf{g})|$ is called the *learning size*. For the studied case there was only one optimal replicator \mathbf{g}_{opt} . A function $d(\mathbf{g})$, for $\mathbf{g} \in P$, corresponds to a minimal distance between the optimal replicator \mathbf{g}_{opt} and binary vectors from the neighborhood $U(\mathbf{g})$

$$d(\mathbf{g}) = \min_{\mathbf{y} \in U(\mathbf{g})} (\mathbf{y} - \mathbf{g}_{opt}) \quad (14)$$

A fitness of the replicator \mathbf{g} is determined as follows

$$f(\mathbf{g}) = f_{min} + (f_{max} - f_{min}) e^{-\xi d(\mathbf{g})} \quad (15)$$

where ξ is the so-called *slope coefficient* that determines a sharpness of the fitness function. The above specified evaluation of population replicators by fitness has a strong stochastic character. In particular, the neighborhood $U(\mathbf{g})$ is randomly constructed from the reference replicator \mathbf{g} . This stochastic character of the fitness evaluation is suppressed if the learning size $q=|U(\mathbf{g})|$ is increased so that there exists a real chance to find the global minimum of (14).

The learning process (i.e. evaluation of replicators by fitness) is mainly determined by the cardinality q of the neighborhood $U(\mathbf{g})$ and the slope coefficient ξ . The cardinality q corresponds to a number of trial events in the learning process when we construct a binary vector \mathbf{h} from the given replicator \mathbf{g} so that the binary vector \mathbf{h} has a smallest Hamming distance from the optimal binary vector \mathbf{g}_{opt} . Increase of the cardinality q offers to the learning process a bigger chance to find the best binary vector with the smallest possible Hamming distance from \mathbf{g}_{opt} . The slope coefficient ξ is a positive real number that may be interpreted as a selector of acceptance of binary vectors that are created from the given replicator \mathbf{g} in the framework of the learning process. If this parameter tends to infinity, then the learning process evaluates by the maximal fitness the binary vectors identical to the optimal binary vector \mathbf{g}_{opt} , all other binary vectors are evaluated by the minimal fitness. On the other hand, if the slope coefficients ξ approaches zero, the learning process evaluates all binary vectors by values of fitness close to the maximal value f_{max} . The above-discussed properties of the fitness are formally expressed by

$$\lim_{\xi \rightarrow \infty} f(\mathbf{g}) = \begin{cases} f_{max} & (\text{if } \mathbf{g} = \mathbf{g}_{opt}) \\ f_{min} & (\text{if } \mathbf{g} \neq \mathbf{g}_{opt}) \end{cases} \quad (16a)$$

$$\lim_{\xi \rightarrow 0} f(\mathbf{g}) = f_{max} \quad (16b)$$

The learning process can be interpreted as a searching in a nearest neighborhood of the given replicator (cf. eqs. (10) to (12)), where the replicator hash symbols are randomly substituted by 0s or 1s symbols. After Hinton and Nowlan [15] this is a "needle in a haystack" search problem, since there is only one correct solution in a space of 2^n solutions. If the learning is included, then chemostat algorithm can be loosely interpreted as a search method, where some one advises to the "replicator" in which direction the needle (correct solution) is hidden. More realistic interpretation of chemostat algorithm with learning is presented in Fig. 1, where the learning process is interpreted as a method that is able to give information about the nearest neighborhood on fitness landscape, i.e. whether replicator fitness can be potentially increased.

Finally, we have to specify chemostat algorithm (outlined in the previous chapter, see Algorithm 1) applied to populations of replicators described in the first part of this chapter. An operation of *mutation* applied to a replicator \mathbf{g} is performed in such a way that going successively throughout all parts of replicator we change with small probability $P_{mut}^{(replication)}$ replicator entries to one of their complements, e.g. symbol 1 is changed randomly with the same probability either to 0 or to #.

Chemostat algorithm with learning features specified in this chapter has the following set of parameters:

- (1) n , replicator length ($n=20$)
- (2) p , population size ($p=500$)
- (3) $P_{mut}^{(replication)}$, replication mutation probability ($P_{mut}^{(replication)}=0.01$)
- (4) t_{max} , number of epochs ($t_{max}=10^6$)
- (5) q , learning size ($q=10$)
- (6) ξ , slope parameter for fitness calculation ($\xi=0.1 \dots 1.0$)
- (7) α , slope parameter for replication probability $prob(\mathbf{g})$ ($\alpha=2.0$)
- (8) f_{max} , maximal value of fitness ($f_{max}=2.0$)
- (9) f_{min} , minimal value of fitness ($f_{min}=1.0$)
- (10) initial population P is randomly generated so that its replicators contains 50% symbols #, 25% symbols 0, and 25% symbols 1

where in parentheses are given numerical values of parameters used in our illustrative calculations.

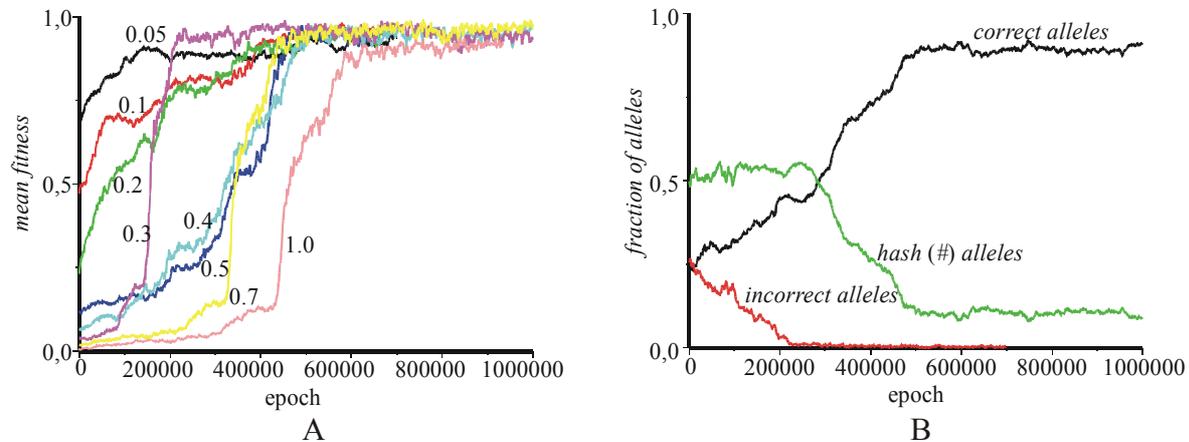


Figure 3. (A) Eight plots of mean fitness vs. epochs for different values of slope parameters, from promoted learning ($\xi=0.05$) to suppressed learning ($\xi=1.0$). Other parameters are specified at the end of Chapter 3. We see that if the slope parameter ξ tends to greater values, then the learning process is more and more suppressed, the period when mean fitness is equal to the minimal value of fitness ($f_{min}=1.0$) is longer. If the slope parameter ξ is greater than a threshold value (say $\xi=3.0$), then the learning is entirely suppressed, the mean fitness in the course of whole evolution is equal to f_{min} (the optimization strength of the algorithm too small to show here). (B) Plot of fraction of correct alleles, incorrect alleles, and hash symbol (#) alleles algorithm with learning (slope parameter is $\xi=0.5$). The fraction (frequency) of fixed correct alleles increased over epochs and the fraction of fixed incorrect alleles decreased almost to zero. On the other hand, the frequency of hash alleles (that can be substituted by 0s or 1s symbols in the learning process) stayed relatively very high. After Hinton and Nowlan [15] this may be caused by too insignificant selective pressure to fix all the hash alleles. This selective pressure may be substantially increased, e.g. by lowering the learning size q (it determines the cardinality of neighborhood $U(\mathbf{x})$, where the learning process is performed). According to our numerical experiences, when the learning size is set $q=1$, then fraction of hash alleles is monotonously decreasing in the course of evolution almost to zero value.

Various numerical results of our model calculations are outlined in Fig. 3. This figure is very instructive for interpretation of Baldwin effect in the framework of our chemostat calculations. Five plots of mean fitness for different slope parameters ξ are presented in Fig. 3, diagram A. We see that the learning may be substantially suppressed by increasing the slope parameter ξ . The same effect may be also achieved by decreasing the learning size q , the learning is suppressed when the cardinality of the neighborhood is very small (i.e. $q \rightarrow 1$), i.e. the minimization problem (12) is restricted only to a few binary vectors. An evolution of fractions of alleles (assigned to 0, 1, and # symbols) is displayed in Fig. 3, diagram B. Very interesting observation is a decrease of the fraction of hash symbols in the course of evolution from the initial 50% value to the final 15% value. This means that the used chemostat algorithm with learning simulates properly classical Baldwin effect; it is able of a genetic encoding of a trait that originally had to be learned. This fact is manifested by monotonously decreasing value of fraction of hash symbols in the population in the course of its evolution. Hash symbols in replicators are introduced as learnable replicator alleles, their decreasing fraction means that some of them were fixed by 0s or 1s alleles, i.e. learnable traits are fixed genetically.

3 M-replicators

In the previous chapter we have formulated simple chemostat algorithm with a possibility to simulate learning process in the course of the evolution process (Baldwin effect). The learning process consists in a replacing hash # symbols in replicators by 0s or 1s. These values are substituted randomly and in a subset of these binary vectors we look for a binary vector with minimal Hamming distance from the given optimal replicator. The learning process assigns to each replicator a binary vector that is the best one of all possible vectors that can be randomly constructed from the replicator. If we compare this version of the chemostat algorithm with a chemostat algorithm where a learning process is considerably suppressed (e.g. slope parameter ξ is used with high numerical value or the learning size q is of very small value), then we see that an effectiveness of the chemostat algorithm with learning is greater than those one with suppressed learning, see Fig. 2.

How to intensify the chemostat algorithm with already included learning process? Further possibility is to include the so-called memetic approach initially introduced into evolutionary biology by Dawkins [8]. We use very simplified model of memes so that it can be easily introduced to chemostat algorithms with learning. In our approach memes are linear strings (of the same length as replicators) and each replicator is automatically accompanied by a meme. The memes are carriers of information about best local solutions that have been already achieved (see Chapter 1). Employing of memes is very important in evaluation process as a further source of information about a best solution that should be found. At the beginning of evolution the memes are empty (i.e. they have zero information content), in the course of evolution process they may receive information in the framework of the so-called memetic interaction. Here, a quasirandomly (with high fitness) selected replicator–donor and its meme creates a message - meme broadcasted through whole population, and some replicators–recipients accept this meme and incorporate it in their memes. In the reproduction process, created offspring obtain memes from one of two parental replicators declared as a mother (we say that offspring are educated by mothers).

A *meme* is a string composed of n symbols 0, 1, and *

$$\mathbf{m} = (m_1 m_2 \dots m_n) \in \{0, 1, *\}^n \quad (17)$$

A *width* of the meme \mathbf{m} , denoted by $|\mathbf{m}|$, is determined as a number of nonstar symbols, e.g. for a meme $\mathbf{m}=(01**01***)$ its width is $|\mathbf{m}|=4$. In our forthcoming considerations memes are used as an auxiliary vehicle of information used in the learning process of replicators. If the width of a meme is zero, then we say that this meme does not contain any useful information and its application in the framework of a learning process is irrelevant.

A *meme replicator* (*m-replicator*) is determined as an ordered couple of a replicator and a meme, both of the same length

$$\boldsymbol{\alpha} = (\mathbf{g}, \mathbf{m}) = ((g_1 g_2 \dots g_n), (m_1 m_2 \dots m_n)) \in \{0, 1, \#\}^n \times \{0, 1, *\}^n \quad (18)$$

where \mathbf{g} is a *replicator part* of $\boldsymbol{\alpha}$ and \mathbf{m} is a *meme part* of $\boldsymbol{\alpha}$. A population P is a multiset composed of p m-replicators

$$P = \{\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \dots, \boldsymbol{\alpha}_p\} \subseteq \{0, 1, \#\}^n \times \{0, 1, *\}^n \quad (19)$$

The learning process of fitness evaluation of simple replicators, specified in the previous chapter, will be now generalized for m-replicators. This learning process consists of two stages.

In the *first stage*, the meme part \mathbf{m} of $\boldsymbol{\alpha}$ is used for a repair of the replicator part \mathbf{g} of $\boldsymbol{\alpha}$, replicator hash symbols are substituted by meme nonstar symbols. In particular, let $\hat{\mathbf{g}}$ be a replicator created from the original replicator \mathbf{g} by a repair process performed with respect to the meme part \mathbf{m}

$$\hat{\mathbf{g}} = (\hat{g}_1 \hat{g}_2 \dots \hat{g}_n) = O_{\text{repair}}(\mathbf{g}; \mathbf{m}) \quad (20a)$$

$$\hat{g}_i = \begin{cases} g_i & (\text{if } g_i \neq \# \text{ or } m_i = *) \\ m_i & (\text{if } g_i = \# \text{ and } m_i \neq *) \end{cases} \quad (20b)$$

For better understanding of this repair process let us consider an m-replicator

$$\boldsymbol{\alpha} = \left(\underbrace{(00\#\#1\#0\#)}_g \underbrace{(11***0**)}_m \right) \quad (21)$$

The repaired replicator $\hat{\mathbf{g}}$ is determined by (18a-b) as follows

$$\begin{aligned} \mathbf{g} &= (00\#\#1\#0\#) \\ \mathbf{m} &= (11***0**) \\ \hline \hat{\mathbf{g}} &= (00\#\#100\#) \end{aligned} \quad (22)$$

where in the sixth position the lower condition in (18b) is satisfied, i.e. hash symbol # is substituted by 0.

In the *second stage*, the repaired replicator $\hat{\mathbf{g}}$ is used as a reference replicator for construction of the neighborhood

$$U(\hat{\mathbf{g}}) = \{\hat{\mathbf{g}}_1, \hat{\mathbf{g}}_2, \dots, \hat{\mathbf{g}}_q\} \subseteq \{0, 1\}^n \quad (23)$$

where binary vectors $\hat{\mathbf{g}}_i$ are created from the repaired replicator $\hat{\mathbf{g}}$ so that its hash symbols are randomly substituted by 0 or 1 symbols, see eq. (11b). Applying the same procedure as in the standard chemostat algorithm with learning, we look for a minimal Hamming distance between binary vectors from the neighborhood and the optimal replicator \mathbf{g}_{opt} (cf. eq. (12))

$$\hat{d}_{\min} = \min_{\hat{\mathbf{h}} \in U(\hat{\mathbf{g}})} (\hat{\mathbf{h}} - \mathbf{g}_{\text{opt}}) \quad (24)$$

Finally, a fitness assigned to an m-replicator α is determined in the same way as in eq. (13)

$$f(\alpha) = f_{min} + (f_{max} - f_{min})e^{-\xi \hat{d}_{min}} \quad (25)$$

Summarizing, an evaluation of m-replicators by fitness is performed formally by the same procedure as simple replicators (see previous chapter) except for the fact that meme part is used for a repair process of the replicator part. Memes are carriers of additional information that is used for a repair process that precedes the learning process. For a useful meme the resulting value d_{min} may be smaller than in the case when the repair process is not used (or it is used but a width of the meme part is zero).

Let us postulate three elementary “reactions”, which are fundamental for the proposed “chemostat” evolutionary algorithm over a population of m-replicator:

(1) **Replication**, an m-replicator (g, m) is replicated to another m-replicator (g', m') (see fig. 4)

$$\underbrace{(g, m)}_{\text{parent}} \rightarrow \underbrace{(g, m)}_{\text{parent}} + \underbrace{(g', m')}_{\text{offspring}} \quad (26)$$

Components of the offspring m-replicator (g', m') from the right-hand side of (4) are closely related to the respective parent components (usually with a small “distance”) and are specified as follows:

$$g' = O_{mut}^{(replicator)}(g) \quad (27a)$$

$$m' = O_{mut}^{(meme)}(m) \quad (27b)$$

where $O_{mut}^{(replicator)}$ ($O_{mut}^{(meme)}$) is a replicator mutation operator (Holland, 1975) (specified by a probability $P_{mut}^{(replicator)}$)

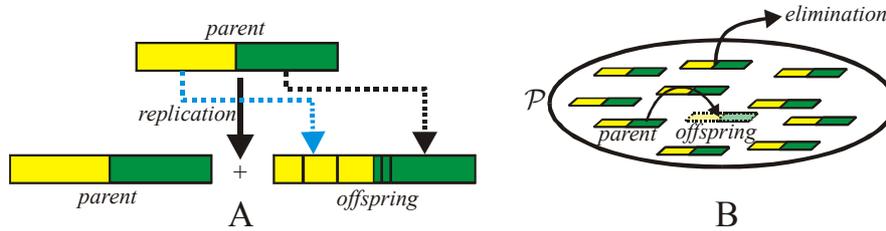


Figure 4. Diagrammatic interpretation of the replication process (4). (A) Randomly selected parent m-replicator is replicated, both components of an offspring m-replicator are specified as slightly mutated versions of parent counterparts (manifested by an appearance of thin vertical columns in the offspring). (B) Schematic outline of the replication process in a population, randomly selected parent m-replicator is replicated (with mutations) to an offspring m-replicator, and moreover, randomly selected m-replicator is eliminated from the population.

Formally, the replication process may be expressed by a stochastic operator

$$\alpha' = O_{repli}(\alpha) \quad (28)$$

In a limiting case, a resulting m-replicator $\alpha'=(g', m')$ may be simply equal to the argument $\alpha=(g, m)$. We have to emphasize that for replication processes we distinguished two type of mutation operators, the first one is a replicator mutation and the second one is a meme mutation, where both are characterized by different mutation probabilities $P_{mut}^{(gene)}$ and $P_{mut}^{(meme)}$, respectively. This distinguishing allows us to separate processes of replicator and meme mutations, it implies that we may introduce different “evolutionary” rates of replicators and memes. A probability of an application of the replication operator (an analog to the kinetic rate constant k) is specified by

$$prob_1 = \exp[\alpha(f(\alpha) - f_{max})] \quad (29)$$

where f_{max} is an estimated maximal value the fitness and $\alpha > 0$ is the so-called *slope parameter* (its greater values, $\alpha \rightarrow \infty$, cause a smallness of the probability, $prob \rightarrow 0$, for $f(\alpha) < f_{max}$). It means that a replication process is more probable for those m-replicators that have fitness closely related to its maximal value f_{max} , such an assumption is of great importance for an accomplishing a global solution (or its tightly related approximation) of the optimization problem (3) by the proposed evolutionary algorithm based on the metaphor of “chemostat” (Dittrich, 1999; Gillespie, 1977).

(2) **Interaction**, two m-replicators (\mathbf{g}, \mathbf{m}) (called the donor) and (\mathbf{g}', \mathbf{m}') (called the acceptor) mutually interact such a way that there exists an oriented “memetic” transfer of a meme from the donor to the acceptor (see fig. 5)

$$\underbrace{(\mathbf{g}, \mathbf{m})}_{donor} + \underbrace{(\mathbf{g}', \mathbf{m}')}_{acceptor} \rightarrow \underbrace{(\mathbf{g}, \mathbf{m})}_{donor} + \underbrace{(\mathbf{g}', \mathbf{m}'')}_{modified\ acceptor} \quad (30)$$

An acceptor memetic part of is created from the transferred donor meme by its mutation

$$\mathbf{m}'' = O_{mut}^{(meme)}(\mathbf{m}) \quad (31)$$

The modified acceptor is composed of a meme, which is originated from the donor and modified by a mutation. This interaction process corresponds to one of fundamental properties of memetic systems [2,5], in particular, memes are spreading out through population not only “vertically” by replication, but also “horizontally” when a donor offers its meme to other acceptors by the interaction process. The interaction process may be expressed by a stochastic operator

$$(\mathbf{g}', \mathbf{m}'') = O_{interact}((\mathbf{g}, \mathbf{m}), (\mathbf{g}', \mathbf{m}')) \quad (32)$$

In a limiting case, if this operator could not applied to arguments, then the resulting m-replicator ($\mathbf{g}', \mathbf{m}''$) is simply equal to the second argument (acceptor) (\mathbf{g}', \mathbf{m}'). In order to get a complete specification of the interaction operator, we have to introduce a probability of its application to arguments

$$prob_2 = \max\{0, 1 - \exp[\alpha(f(\alpha') - f(\alpha))]\} \quad (33)$$

and $\alpha > 0$ is the so-called *slope parameter* (its greater values cause a dichotomic behavior (either zero or unit) of the probability. The probability is positive if a fitness of donor is greater than a fitness of acceptor, $f(\alpha') > f(\alpha)$, and it is proportional to a difference $f(\alpha) - f(\alpha')$. It means that there exists a “one-way” memetic transfer from a donor has a greater fitness than the acceptor.

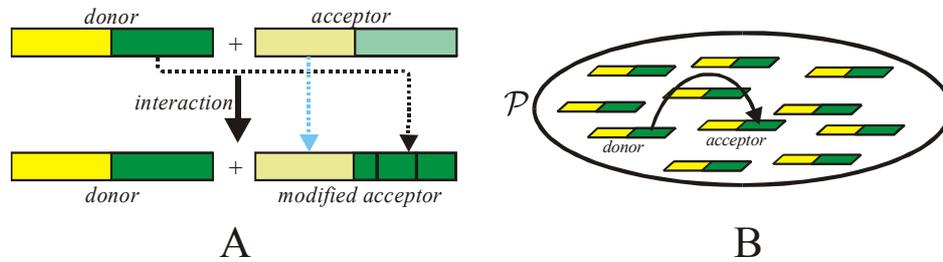


Figure 5. Diagrammatic interpretation of the interaction process (24). (A) Randomly selected donor transfers its meme \mathbf{m} to a randomly selected acceptor and then this meme is locally optimized with respect to the original acceptor replicator \mathbf{g}' . (B) Schematic outline of the interaction process in a population, two randomly selected m-replicators “interact” such that a meme from the donor is transferred (represented by an oriented arrow) to the acceptor and it is locally optimized with respect to the acceptor replicator.

(3) **Extinction**, an m-replicator (\mathbf{g}, \mathbf{m}) is removed from the population \mathcal{P}

$$\underbrace{(\mathbf{g}, \mathbf{m})}_{\text{parent}} \rightarrow \emptyset \quad (34)$$

Each replication process is automatically accompanied by the extinction. Since the replication increases the number of m-replicators in population about one (cf. eq. (4)), an application of extinction (a randomly selected m-replicator is eliminated from the population) ensures a constant number of particles in chemostat (i.e. the size of population is invariant in the course of evolution)

In the proposed evolutionary algorithm based on a metaphor of chemostat (Dittrich, 1999; Gillespie, 1977) a selection pressure in population of m-replicators is created by a replication and interaction processes based on fitness. M-replicators with a greater fitness have the greater chance to take part in a replication or interaction process (a measure of quality of m-replicators); on the other hand, m-replicators with a small effective fitness are rarely used in the replication process or as a donor in the interaction. This simple manifestation of Darwin's natural selection ensures a gradual evolution of the whole population. In the present approach the mentioned principle of fitness selection of m-replicators is preserved, but it is now combined with an additional selection pressure due to a constancy of number of m-replicators in the chemostat. An m-replicator outgoing from the replication reaction eliminates a randomly selected m-replicator. Moreover, we have to distinguish between a performance of replication and interaction processes; the replication process should be applied with substantial higher frequency than interaction process, that is simple replications of parents onto offspring are more frequent than transfers of a meme from donors to acceptors (see Algorithm 2).

Algorithm 2.

```

chemostat P is randomly generated;
for t:=1 to tmax do
begin (g, m) := Oselect(P);
  if random < prob1 then
  begin (g', m') := Orepli(g, m);
    (g'', m'') := Oselect(P);
    (g'', m'') ← (g', m');
  end;
  while random < Pinteract then
  begin (g, m) := Oselect(P); (g', m') := Oselect(P);
    if random < prob2 then
    begin (g', m'') := Ointeract((g, m), (g', m'));
      (g', m') ← (g', m'');
    end;
  end;
end;
end;
```

The algorithm is initialized by a population composed of randomly generated m-replicators that are all evaluated by the fitness (see row 1). The algorithm is composed of two blocks that are activated with different probabilities, the first one (with a probability P_{repli}) for a replication (rows 5 – 8) and the second one (with a probability P_{interact}) for an interaction (rows 9-16). The replication block is initiated by a random selection (realized by an operator

O_{select}) of an m -replicator (g, m) , which is subjected stochastically (with a probability specified by $\text{prob}(g, m)$) to a replication process expressed by an operator O_{repli} . The resulting product (g', m') is evaluated by a fitness and then it is returned to the population such that it eliminates a randomly selected m -replicator (g'', m'') . The interaction block is repeated with a probability P_{interact} , two m -replicators are randomly selected and then both are undergone an interaction.

Parameters specified at the end of Chapter 3 are still valid also for the present modification. Moreover, a random generation of initial population and additional two parameters used in this version of chemostat algorithm should be specified:

- (3') $P_{\text{mut}}^{(\text{meme})}$, replication mutation probability ($P_{\text{mut}}^{(\text{meme})}=0.01$)
- (10') initial population P is composed of randomly generated m -replicators, where replicator parts contains 50% symbols #, 25% symbols 0, and 25% symbols 1 and meme parts are of zero width (i.e. they contain only star symbols)

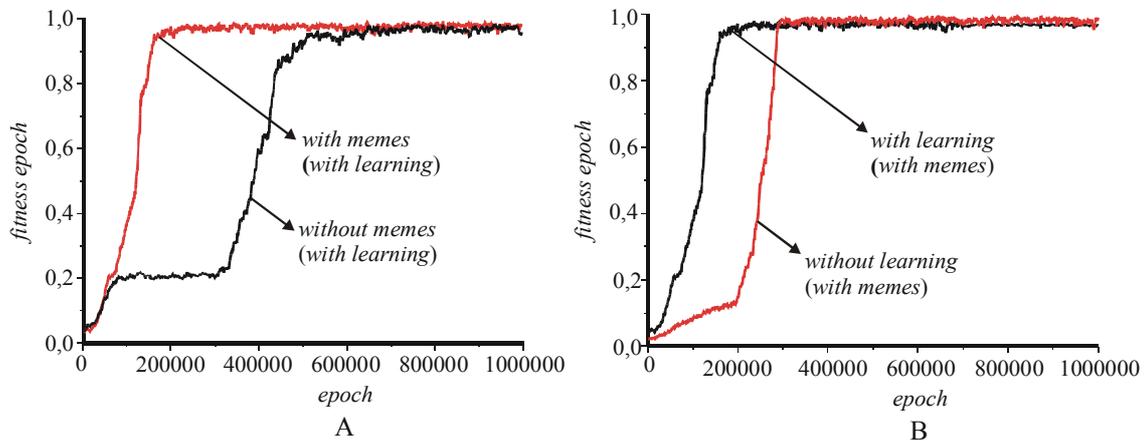


Figure 6. (A) Plot of mean fitness for two different versions of chemostat algorithm with and without a memetic interaction when learning is included in both simulations. Calculations were done for slope parameter $\xi=0.5$. We see that chemostat algorithm with memes and learning outperforms the chemostat algorithm with learning only. (B) Plot of mean fitness for two different versions of chemostat algorithm with and without learning when memes are included in simulations. Calculations were done for slope parameter $\xi=0.5$. We see that chemostat algorithm with learning and memetic information outperforms the chemostat algorithm with only memetic information.

Illustrative numerical results are presented in Fig. 4. Both diagrams in this Figure nicely illustrate an importance of an inclusion of memes into chemostat algorithm with learning. It means that memes may be considered after learning as further important factor for an acceleration of evolution of replicators. On the other hand, diagram B illustrates and importance of the learning itself in the chemostat algorithm. Summarizing both observations we may say that a simultaneous combination of learning together with memes produces very robust replicator algorithm, which is able to overcome limitations of standard Darwinian evolution that forbid a genetic transfer of acquired information “treasure” from parents onto offspring. Moreover, if the learning process is costly, then the memes might be very important for acceleration of evolution, they are able to overcome to some extent learning barriers.

4 Conclusions

Two different coevolutionary models of replicators and memes are possible (see Fig. 7). The *first* one (diagram A) corresponds to the so-called competitive coevolution of replicators and memes. In this competitive approach there exist two quasi-independent populations that are composed of replicators and memes, respectively; these populations mutually interact through fitness evaluation of both populations. For instance, it means that when replicators are evaluated by fitness, then the fitness is determined by a structure of the respective replicator and by whole population of memes, and similarly for the meme population. This fact is a main obstacle of competitive approach, an interaction with other population in the course of fitness evaluation is not uniquely specified [22], there exist many different possibilities how to realize it and most of them are very time consuming (e.g. if the fitness of a replicators depends on an interaction with all memes from other population).

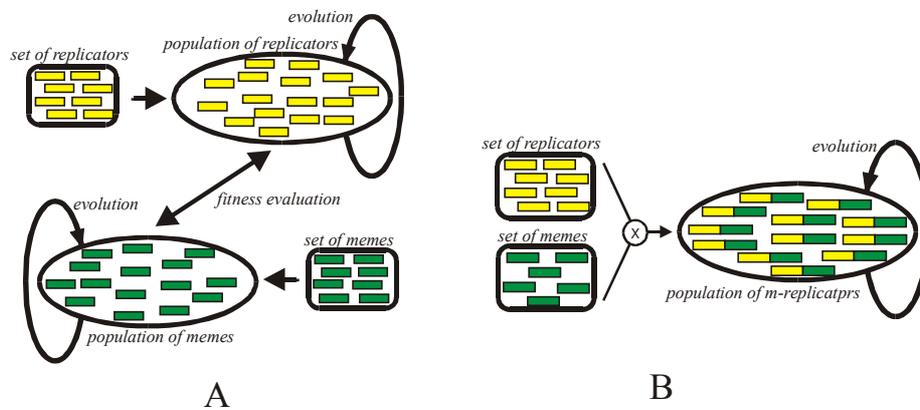


Figure 7. Two different model of coevolution of replicatoprs and memes. Diagram A represents a competitive coevolutionary model with two quasi-independet populations of replicators and memes, these populations mutually interact through fitness evaluations of replicators and memes. Diagram B represents a cooperative coevolutionary model, it contains one population composed of m-replicators (couples of replicator and meme). Fitness of m-replicators is calculated such that fitness of replicators is modified by a contribution originated by the associated meme.

The *second* possibility (used in this communication) is the so-called cooperative coevolution of replicators and memes, where we have only one common population composed of compositions of replicators and memes (called the m-replicators). This approach simply overcomes the above mentioned shortcoming with fitness evaluation of the competitive coevolution. The fitness of a respective m-replicator is calculated in such a way that its main dominant part is specified by replicator structure itself and some smaller additional terms arise from an interaction between respective replicator and meme. Moreover, the approach of cooperative evolution nicely corresponds to a restricted look on memetics, where there is postulated that memes could not exist independently from replicators, they are “stored” in a “brain” of replicators. Therefore, we believe that the used cooperative coevolutionary model is highly realistic and reflects a strong coupling between replicators and memes.

What about a possibility to generalize the present model? The first approach for a generalization of the present method is a postulation that a population of m-replicators is "geographically" structured so that it will be decomposed onto disjoint subpopulations and each subpopulation has some other subpopulations as neighborhoods. There is possible to examine the effect of erecting complete or semipermeable barriers between different

subpopulations and the effect of migration. Second, until now, only one meme was attached to each replicator, it will be interesting to study when this restriction is removed and a variable number of memes is attached to replicators [21]. Third, in our calculations we kept fixed the optimal replicator \mathbf{g}_{opt} , it means that the goal of evolution remained unchanged in the course of whole evolution. Interesting results might be obtained by slight variation of this optimal replicator \mathbf{g}_{opt} , in each generation it can be mutated with very low probability, e.g. $P_{mut}^{(opt)} = 0.001$. Then, we will compare an effectivity of the algorithm with learning only and the algorithm with learning and memes. According to our preliminary results it seems, that memes might be very important for permanent adapting of the best evolutionary solution when goal of evolution (or environment where populations exist) is slightly changing. In particular, if the algorithm is performed only with learning, we have observed that the evolving system has a great inertia, it is unable to well adapt to a new evolution goal.

One of the most serious restrictions of the present theory is a postulate that a particular replicator interacts only with one meme. In many cases of theoretical interest (in theoretical memetics) this severe restriction seriously limits an applicability of the present theory; therefore we suggest its generalized version, which in some extent overcomes the mentioned restriction (see Fig. 8). The concept of m-replicator is generalised such that it is composed of more than one memes that are taken from different sets (it means that different memes may have a different semantic contents). Moreover, this simple generalization nicely demonstrates flexibility and effectivity of the present theoretical approach, which allows not only simple computational simulations, but also its power to be simply modified and generalized.

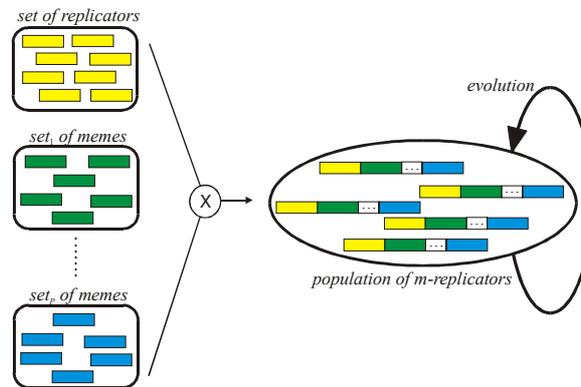


Figure 8. In the generalized approach a population is composed of m-replicators that are represented by compositions (determined by a “cross” product and called the m-replicator) of a replicator and P different kinds of memes.

The present simple coevolutionary model of replicators and memes support an intuitive idea that memes are very important acceleration factor of Darwinian evolution, when evolutionary goals are very complex tasks and offspring replicators need an advice from parents how to continue effectively the evolutionary task from their lifetime experiences. In other words, replicators are accompanied as an integral part by memes that correspond to information, which is important for quicker orientation of replicators on fitness landscape. According to meme contents replicators “know” in advance, which directions on fitness landscape are more effective than other ones with small or nothing relevance for the forthcoming evolution of replicators. If the evolutionary task is not very complicated, then an acceleration of evolution may be achieved by an inclusion of the learning into fitness evaluation process (i.e. by Baldwin effect). But, if the evolutionary goals are represented by very complex tasks, then an

inclusion of the learning into evolutionary process is insufficient, and meme must be introduced as an auxiliary tool to overcome information barrier between parents and offspring. This general observation may be considered as a main substantiation of memetic view on culture evolution [2].

Acknowledgments

This work was supported by the grants # 1/4209/97 and # 1/5229/98 of the Scientific Grant Agency of Slovak Republic.

References

- [1] Adami, C. (1999). Introduction to Artificial Life. Springer Verlag, New York,.
- [2] Aunger, R., ed. (2001). *Darwinizing Culture: The Status of Memetics as a Science*. Oxford University Press, New York,.
- [3] Baldwin, J. M. (1896). A New Factor in Evolution, *American Naturalist* **30**, 441-451.
- [4] Belew, R. K., and Mitchell, M., eds. (1996). *Adaptive Individuals in Evolving Populations: Models and Algorithms*. Addison-Wesley, Redwood City, CA.
- [5] Blackmore, S. (1999). *The meme machine*. Oxford University Press, New York.
- [6] Bull, L., Holland, O., Blackmore, S. (2000). On Meme-Gene Coevolution. *Artificial Life* **6** , 227-235.
- [7] Darwin, C. (1859). *The Origin of Species*, reprinted 1985, Penguin, London.
- [8] Dawkins, R. (1976). *The Selfish Gene*, Oxford University Press, Oxford.
- [9] Dennett, D (1995). *Darwin's dangerous idea*. The Penguin Press, Hammondsworth.
- [10] Dittrich P, Ziegler, J., Banzhaf, W. (2001). Artificial Chemistries - A Review. *Artificial Life* **7**, 225-275.
- [11] Eigen M. (1971). Self organization of matter and the evolution of biological macro molecules. *Naturwissenschaften* **58** , 465-523.
- [12] Feldman, M.W., Laland, K.N. (1996). Gene-Culture Coevolutionary Theory. *Trends in Ecology and Evolution* **11**, 453-457.
- [13] Gillespie D. T. (1977). Exact Stochastic Simulation of Coupled Chemical Reactions. *Journal of Physical Chemistry* **81**, 2340.
- [14] Goldberg, D.E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, New York.
- [15] Hinton, G. E., Nowlan, S.J. (1987). How learning can guide evolution, *Complex System* **1**, 495-502.
- [16] Hofbauer, J., Sigmund, K. (1988). *The Theory of Evolution and Dynamical Systems*. Cambridge University Press, Cambridge.
- [17] Holland, J.H. (1975). *Adaptation in Natural and Artificial Systems*, MIT Press, Cambridge, MA.
- [18] Jones B.L., Enns R.H., Rangnekar S.S. (1976). On the theory of selection of coupled macromolecular systems. *Bulletin of Mathematical Biology* **38**, 15-23.
- [19] Kvasnička, V., Pospíchal, J. (1999). Simulation of Baldwin effect and Dawkins memes by genetic algorithm. In Roy, R., Furuhashi, T. and Chawdhry, P.K. (eds.): *Advances in Soft Computing*. London: Springer-Verlag, pp. 481-496.
- [20] Kvasnička, V., Pospíchal, J. (2002). Artificial Chemistry and Molecular Darwinian Evolution in Silico. *Collection of Czechoslovak Chemical Communications* (sent for publication).

- [21] Kvasnička, V. (2002). Replicator theory of coevolution of genes and memes. *Advances in Complex Systems* (sent for publication).
- [22] Paredis, J. (1997). C7.4 Coevolutionary Algorithms. In Bäck, T., Fogel, D., Michalewicz, Z. (eds.): *The Handbook of Evolutionary Computation*. Oxford University Press, Oxford.
- [23] Wright S. (1931). Evolution in Mendelian Populations. *Genetics*, 16, 97-159.