DE GRUYTER
OPEN

Int. J. Appl. Math. Comput. Sci., 2015, Vol. 25, No. 3, 499–512
DOI: 10.1515/amcs-2015-0037

amcs

A STUDY ON MEME PROPAGATION IN MULTIMEMETIC ALGORITHMS

RAFAEL NOGUERAS [a], CARLOS COTTA [a,*]

[a]Department of Computer Science and Programming Languages
Higher Technical School of Computer Engineering
University of Málaga, Campus de Teatinos, 29071 Málaga, Spain
e-mail: ccottap@lcc.uma.es

Multimemetic algorithms (MMAs) are a subclass of memetic algorithms in which memes are explicitly attached to genotypes and evolve alongside them. We analyze the propagation of memes in MMAs with a spatial structure. For this purpose we propose an idealized selecto-Lamarckian model that only features selection and local improvement, and study under which conditions good, high-potential memes can proliferate. We compare population models with panmictic and toroidal grid topologies. We show that the increased takeover time induced by the latter is essential for improving the chances for good memes to express themselves in the population by improving their hosts, hence enhancing their survival rates. Experiments realized with an actual MMA on three different complex pseudo-Boolean functions are consistent with these findings, indicating that memes are more successful in a spatially structured MMA, rather than in a panmictic MMA, and that the performance of the former is significantly better than that of its panmictic counterpart.

Keywords: memetic algorithms, spatial structure, meme propagation.

1. Introduction

Four decades ago, Richard Dawkins (1976) put forward the definition of a meme in analogy to the biological concept of a gene. Memes were broadly characterized as units of imitation, that is, ideas or pieces of knowledge that jump from brain to brain, thriving and proliferating in some cases and becoming extinct in others. Even more interestingly, memes are not static objects but dynamic entities that mutate over their lifetime; these mutations can make them more interesting/useful/stronger/etc., thus boosting their propagation, or can have the opposite effect, causing that particular mutation to fade away. This plasticity explains their comparatively faster rate of adaptation with respect to biological genes.

Inspired by this notion of a meme, Moscato (1989) conceived a new optimization paradigm: memetic algorithms (MAs). MAs are a family of population-based optimization techniques that blend together ideas from different metaheuristics, most notably the orchestrated interplay between global (population-based) search and local (individual-based) search. The most popular incarnation of MAs features an evolutionary search engine endowed with local search add-ons. The notion of memetic evolution is here captured by the Lamarckian lifetime learning which solutions are subject to, via the use of some local search operators. Incidentally, the suggestion has been made (Norman and Moscato, 1989) to use the term *agent*, rather than an individual or a solution in this context, to emphasize the fact that they are active entities that purposefully try to optimize the problem under consideration. We refer to Hart *et al.* (2005), Krasnogor and Smith (2005), Moscato and Cotta (2010), Neri and Cotta (2012), or Neri *et al.* (2012) for a broader discussion of the vibrant field of MAs and their numerous applications.

While memes are typically fixed in classical MAs (i.e., they are given by the particular choice of local search operators), it was already envisioned in the 1990s that MAs could eventually work in at least two levels and two time scales (Moscato, 1999): during the short-time scale a set of agents would be searching in the search space associated to the problem, whereas during the long-time scale they would adapt their search strategies. Indeed, several models trying to make memes change during the optimization process have been proposed, constituting the central tenet in the notion of memetic computing

[*] Corresponding author

which is defined as "…a paradigm that uses the notion of meme(s) as units of information encoded in computational representations for the purpose of problem solving" (see Ong *et al.*, 2010; Chen *et al.*, 2011; Chen and Ong, 2012). These models can be accomplished at a variety of levels (Ong *et al.*, 2006). A simple possibility is the so-called 'meta-Lamarckian learning' (Ong and Keane, 2004), in which the MA has a collection of local search operators (memes) available *a priori*, and a mechanism is used to decide which of them is applied to which solution and when (note the connection with hyperheuristics (Cowling *et al.*, 2008; Chakhlevitch and Cowling, 2008)). A more complex approach features self-adaptation of the memes themselves. An example of this kind of self-adaptation is provided by multi-memetic algorithms (MMAs), in which each solution carries "genes" that indicate which local search has to be applied to it. These can range from simple pointers to existing local search operators, to the parametrization of a general local search template (Krasnogor *et al.*, 2002; Neri *et al.*, 2007) or even to the definition of a grammar to specify a new complex local search operator (Krasnogor, 2004; Krasnogor and Gustafson, 2004).

An interesting issue that arises in the context of MMAs is how memes propagate and spread over the population. While population dynamics have been well studied in the case of evolutionary algorithms (e.g., Alba and Luque, 2004; Giacobini *et al.*, 2005; Karcz-Dulęba, 2004; Rudolph and Sprave, 1995; Sarma and De Jong, 1997), the scenario is more complex in the case of memes: unlike genotypes (which correspond to solutions and thus can be evaluated according to the problem under consideration), memes can be only indirectly assessed via the effect they have on genotypes. Furthermore, memes evolve in MMAs alongside solutions by being attached to them. Since this attachment is part of the self-adaptive process, it is up to the algorithm to discover good fits between individual pairs of genotypes and memes, and this is commonly done using only information about the genetic quality of solutions (i.e., fitness information). The work presented here studies how memes propagate in such an environment driven by genetic selection and the spatial structure. To this end, we consider and analyze an idealized model of MMAs (Nogueras and Cotta, 2013). This model is described and studied in Sections 2 to 4. Later on, we turn our attention to an actual MMA in Section 5 and analyze its behavior on different test problems in order to corroborate the theoretical findings.

2. Model description

Let us consider an abstract model of MMAs in which each agent is characterized by a pair $\langle g, m\rangle \in D^2$, for some $D \subset \mathbb{R}$. The first member of the pair, g, represents the genotype, which we equate to fitness for simplicity. As for the second member, m, it represents a meme. More precisely, this value captures the *improvement potential* of that meme, that is, a measure of how good solutions can get by applying the meme. We assume there is a function $f : D^2 \to D$ monotonically increasing in its first parameter, encapsulating the application of a meme to a genotype, i.e., the effect of a single epoch of Lamarckian learning. Thus, an agent $\langle g, m\rangle$ becomes $\langle f(g, m), m\rangle$ after the application of the meme, where

$$\lim_{n\to\infty} f^n(g, m) = m \quad \text{if } g < m, \tag{1}$$

$$f(g, m) = g \quad \text{if } g \geq m. \tag{2}$$

Here $f^n(g, m)$ is the n-th composition of the function on the first argument, i.e., $f(f(\cdots f(f(g, m), m), \cdots, m), m)$. Intuitively, these conditions indicate that the fitness of a solution can be improved when a good meme is applied on it, asymptotically approaching the potential of the latter. Certainly, this is a very idealized characterization of the potential of a meme since in general this potential is not going to be absolute but may depend on a complex match between the meme, the genotype and the problem landscape, but it serves as an initial attempt to study several issues related to meme propagation in the agent pool.

The population P of the MMA is thus a collection of μ such agents, $P = [\langle g_1, m_1\rangle, \ldots, \langle g_\mu, m_\mu\rangle]$, endowed with a spatial structure that constrains agent communication. Let this spatial structure be characterized by a $\mu \times \mu$ Boolean matrix S, where S_{ij} is `true` if, and only if, the agent placed in the i-th location can communicate with the agent placed in the j location (note that exactly one agent is placed in each location). Since we are interested in observing the dynamics of the propagation of memes, we consider an extension of the selection-only model of evolutionary algorithms (i.e., using only selection/replacement and no variation operator) in which we add the local improvement stage of memetic algorithms. A scheme of the model is shown below in Algorithm 1.

Algorithm 1. Selecto-Lamarckian model.

```
for $i \in \{1, \cdots, \mu\}$ do
    INITIALIZE$\langle g_i, m_i\rangle$
end for
while $\neg$ CONVERGED ($P$) do
    $i \leftarrow$ URAND$(1, \mu)$  {Pick random location}
    $\langle g, m\rangle \leftarrow$SELECTION$(P, S, i)$
    $g' \leftarrow f(g, m)$  {Local improvement}
    $P \leftarrow$ REPLACE$(P, S, i, \langle g', m\rangle)$
end while
```

After suitably initializing the contents of the population, the algorithm engages in a cycle of selection plus improvement until the agents converge. Convergence is here approached from a memetic perspective, that is, we terminate the algorithm when the population comprises a homogeneous collection of memes (regardless of whether there is still diversity at the genetic level or not). As for the inner working of the algorithm, it resembles the new random sweep strategy of cellular automata (Schönfisch and de Roos, 1999), in which the next location to be activated is selected uniformly at random without replacement.

3. Meme propagation

Having defined the general model in the previous section, let us consider some qualitative features of meme propagation that can be extracted from it. Let us assume that selection is done by binary tournament, i.e., once a location i selected, a neighboring location j is selected from $\mathcal{N}(i) = \{j \mid S_{ij}\}$, and the agent with the best fitness is retained. Regarding replacement, let us assume that the improved agent replaces the agent that lost the previous tournament.

We are interested in analyzing the number of copies of each meme in the meme pool, so let us denote by $N(m, g, t)$ the number of instances of meme m attached to genotype g at time t, assuming for simplicity that D is some discrete domain. If we divide this quantity by the pool size μ, we obtain $p(m, g, t)$, the fraction of the population comprising meme m attached to genotype g at time t. In each passing iteration of the system the number of copies can be estimated as

$$\begin{aligned} &N(m, g, t+1) \\ &\qquad = N(m, g, t) + C(m, g, t) - D(m, g, t), \end{aligned} \tag{3}$$

where $C(m, g, t)$ and $D(m, g, t)$ represent the expected number of copies of meme m attached to genotype g that are created or destroyed at time t. The creation of a new copy can be accomplished by the combined effect of selection of a suitable agent with meme m and the application of the meme to the corresponding genotype. Let us express this as

$$C(m, g, t) = \sum_{g'} \sigma(m, g', t) p(g' \xrightarrow{m} g), \tag{4}$$

where $\sigma(m, g', t)$ is the probability of selecting an agent carrying meme m and genotype g' at time t and $p(g' \xrightarrow{m} g)$ is the probability that the application of meme m on genotype g' results in genotype g. The first quantity can be computed as the probability of the binary tournament picking two agents with meme m and genotype g or only one agent with this structure but with a better fitness than its competitor:

$$\begin{aligned} &\sigma(m, g, t) \\ &= p(m, g, t)^2 + 2p(m, g, t)\,[1 - p(m, g, t)] \\ &\quad \times \frac{\sum_{m'} \sum_{g' < g} p(m', g', t)}{1 - p(m, g, t)}, \end{aligned} \tag{5}$$

where the last factor is the probability that the fitness of the competitor is worse than g provided it is not an $\langle g, m \rangle$ agent. This expression assumes that the global distribution of memes/genotypes across the whole population is the same as for local neighborhoods. Obviously, this holds for the panmictic case in which any two agents are neighbors, so we can assume this case initially and consider it a first approximation towards more general situations.

Concerning the destruction of a copy of a particular pair meme/genotype, it can arise via the selection of such a pair and the subsequent application of local improvement (which will alter the genotype) or via replacement by an agent of higher fitness. The first case also requires that the other agent chosen in the tournament be a copy of the same pair, so that it is later substituted by the improved agent. Thus,

$$D(m, g, t) = \sum_{g' \neq g} p(m, g, t)^2 p(g \xrightarrow{m} g') + \tilde{\sigma}(m, g, t)\,. \tag{6}$$

The replacement probability $\tilde{\sigma}(m, g, t)$ can be expressed as

$$\begin{aligned} \tilde{\sigma}(m, g, t) &= 2p(m, g, t)\,[1 - p(m, g, t)] \\ &\quad \times \frac{\sum_{m'} \sum_{g' > g} p(m', g', t)}{1 - p(m, g, t)}. \end{aligned} \tag{7}$$

Let us now consider the evolution of the system in the early-term and mid-term, before a particular meme starts to saturate the population. In this situation memes are widely spread across genotypes, so $p(m, g, t) \ll 1$ and hence we can neglect quadratic terms $p(m, g, t)^2$. We thus have

$$\sigma(m, g, t) = 2p(m, g, t) \sum_{m'} \sum_{g' < g} p(m', g', t), \tag{8}$$

$$\tilde{\sigma}(m, g, t) = 2p(m, g, t) \sum_{m'} \sum_{g' > g} p(m', g', t). \tag{9}$$

Substituting back into Eqns. (4) and (6), we get

$$\begin{aligned} C(m, g, t) &= 2 \sum_{g', m'} p(m, g', t) \\ &\quad \times \sum_{g'' < g'} p(m', g'', t) p(g' \xrightarrow{m} g), \end{aligned} \tag{10}$$

$$D(m, g, t) = 2p(m, g, t) \sum_{m'} \sum_{g' > g} p(m', g', t). \tag{11}$$

Since $p(g' \xrightarrow{m} g) = 0$ for $g < g'$ or $m < g$, Eqn. (10) reduces to

$$C(m,g,t) = 2 \sum_{g''<g'\leqslant g} \sum_{m'} p(m,g',t)p(m',g'',t)p(g' \xrightarrow{m} g). \tag{12}$$

If $m \leq g$, then $p(g' \xrightarrow{m} g)$ is 1 if $g' = g$ and 0 otherwise. Subsequently, the difference $\Delta_g^m(t) = C(m,g,t) - D(m,g,t)$ is in this case

$$\begin{aligned}\Delta_g^m(t) &= 2p(m,g,t)\sum_{m'}\sum_{g''<g} p(m',g'',t) \\ &\quad - 2p(m,g,t)\sum_{m'}\sum_{g''>g} p(m',g'',t) \\ &= 2p(m,g,t)\sum_{m'}\Big[\sum_{g''<g} p(m',g'',t) \\ &\quad - \sum_{g''>g} p(m',g'',t)\Big].\end{aligned} \tag{13}$$

Focusing on the sign of the difference in the expression above, we essentially obtain that *inert* memes (i.e., memes that can no longer improve their hosts) can thrive by hitchhiking, that is, if they attach themselves to agents above the median of the population.

Let us, on the other hand, consider the case $m > g$. In this situation, $p(g' \xrightarrow{m} g)$ is 1 if $g' = f^{-1}(g,m)$ and 0 otherwise, where we denote by $f^{-1}(g,m)$ the genotype value such that $f(f^{-1}(g,m),m) = g$. Using g^{-m} as a shorthand for $f^{-1}(g,m)$,

$$\begin{aligned}\Delta_g^m(t) &= 2p(m,g^{-m},t)\sum_{m'}\sum_{g''<g^{-m}} p(m',g'',t) \\ &\quad - 2p(m,g,t)\sum_{m'}\sum_{g'>g} p(m',g',t) \\ &= \sum_{m'}\left[2p(m,g^{-m},t)\sum_{g''<g^{-m}} p(m',g'',t)\right. \\ &\quad \left. -2p(m,g,t)\sum_{g'>g} p(m',g',t)\right].\end{aligned} \tag{14}$$

The sign of this expression depends on the balance between the goodness of genotypes in the basin of attraction of g and the badness of g itself (in both cases, goodness/badness relative to the rest of the population). Note that in general there is a reinforcement between these quantities in the sense that, the better a genotype in the basin of attraction of g, the better we can expect g to be. This does not just mean that *active* memes proliferate more and more when they attach themselves to good solutions as one would expect, but also that memes with high potential can find their way to the final stages of the evolution provided they have enough time to improve their hosts (recall that the goodness of solutions evolves with time as an effect of the application of the meme).

This suggests that models with slower genetic convergence can have a beneficial effect on the propagation of good memes, allowing the latter enough time to express themselves in the population and overcome the hitchiking effect of bad memes. The next section provides a more quantitative analysis of this effect via numerical simulations.

4. Numerical simulations

The numerical experimentation aims to empirically explore the dynamics of meme propagation and how they are affected by factors such as the population size, the relative improvement potential of memes and the underlying spatial structure of the population. Regarding population sizes, we have considered values $\mu \in \{100, 256, 400, 625\}$. These values cover a broad range of population sizes and are also perfect squares, which is important in connection with one of the spatial structures considered, namely, a square toroidal grid with a von Neumann neighborhood: two locations (i,j) and (i',j') are connected if their Manhattan distance $|i-i'| + |j-j'| \leqslant r$, where r is the neighborhood radius. We have considered $r = 1$, which leads to the traditional North-South-East-West (plus the current location) neighborhood. The other spatial structure considered is the panmictic model in which all locations are connected. In either case, we have considered the function

$$f(g,m) = \begin{cases} g & \text{if } g \geq m, \\ (g+m)/2 & \text{if } g < m \end{cases} \tag{15}$$

to represent the action of memes. Intuitively, this function provides smaller improvements for increasingly good genotypes, much like it often happens in practice. All experiments are averaged over 100 runs in order to obtain representative results. Each run is terminated upon convergence of memes, which for simplicity is determined when all memes are equal to 2 decimal positions.

Let us firstly analyze meme propagation as a function of the relative improvement potential of memes at the beginning of the run. For this purpose, we take $D = [0,1]$ and consider a scenario in which genotypes and memes are randomly initialized in this range, and another scenario in which genotypes take initial values in $[0, 0.5]$ whereas memes are randomly sampled from $[0,1]$.

Figure 1 shows the distribution of memes at each

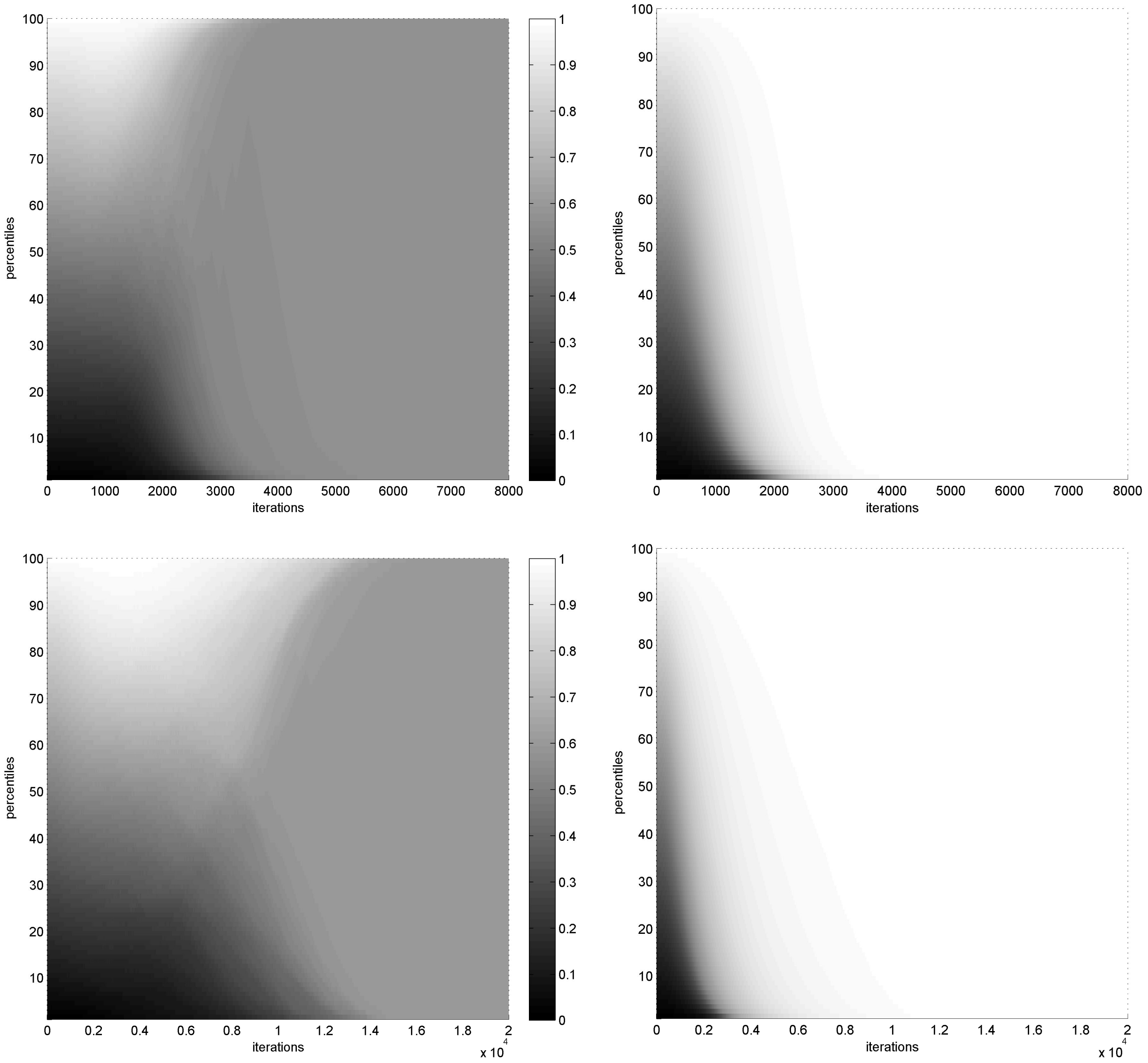

Fig. 1. Meme maps for simulations with $\mu = 625$. The upper row corresponds to panmictic connectivity and the lower row to a von Neumann neighborhood. Similarly, the left column corresponds to genotypes initialized in $[0, 1]$ and the right one to initialization in $[0, 0.5]$ (memes are initialized in [0,1] in both cases). Lighter shades of gray indicate higher meme values. The evolution of the algorithm is depicted in each subfigure from left to right, each vertical slice representing the distribution of memes at a certain time t. Note the different scale on the x-axis.

time step (the lighter the shade of gray[1], the higher the meme value). Focusing firstly on the upper row (panmictic topology), note the clearly different behavior depending on genotype initialization. When genes and memes are both initialized in $[0, 1]$, the algorithm seldom converges on a high-potential meme. Actually, such memes temporarily proliferate in the initial stages of the algorithm but are later driven to extinction by memes hitchhiking on high quality genotypes to which they stuck by chance. The situation is quite different when genotypes are initially drawn from $[0, 0.5]$: in this case the algorithm does gradually converge to the upper part of the meme distribution, with low-potential memes quickly disappearing from the population. A more detailed perspective on this is provided by Fig. 2, in which qualified runtime distributions (QRTDs) (Hoos and Stützle, 2004) are provided. These indicate the probability of a certain target (in this case, convergence to a meme in a desired percentile) being reached as a function of

[1] A color version of this figure can be accessed at `http://dx.doi.org/10.6084/m9.figshare.1235204`.

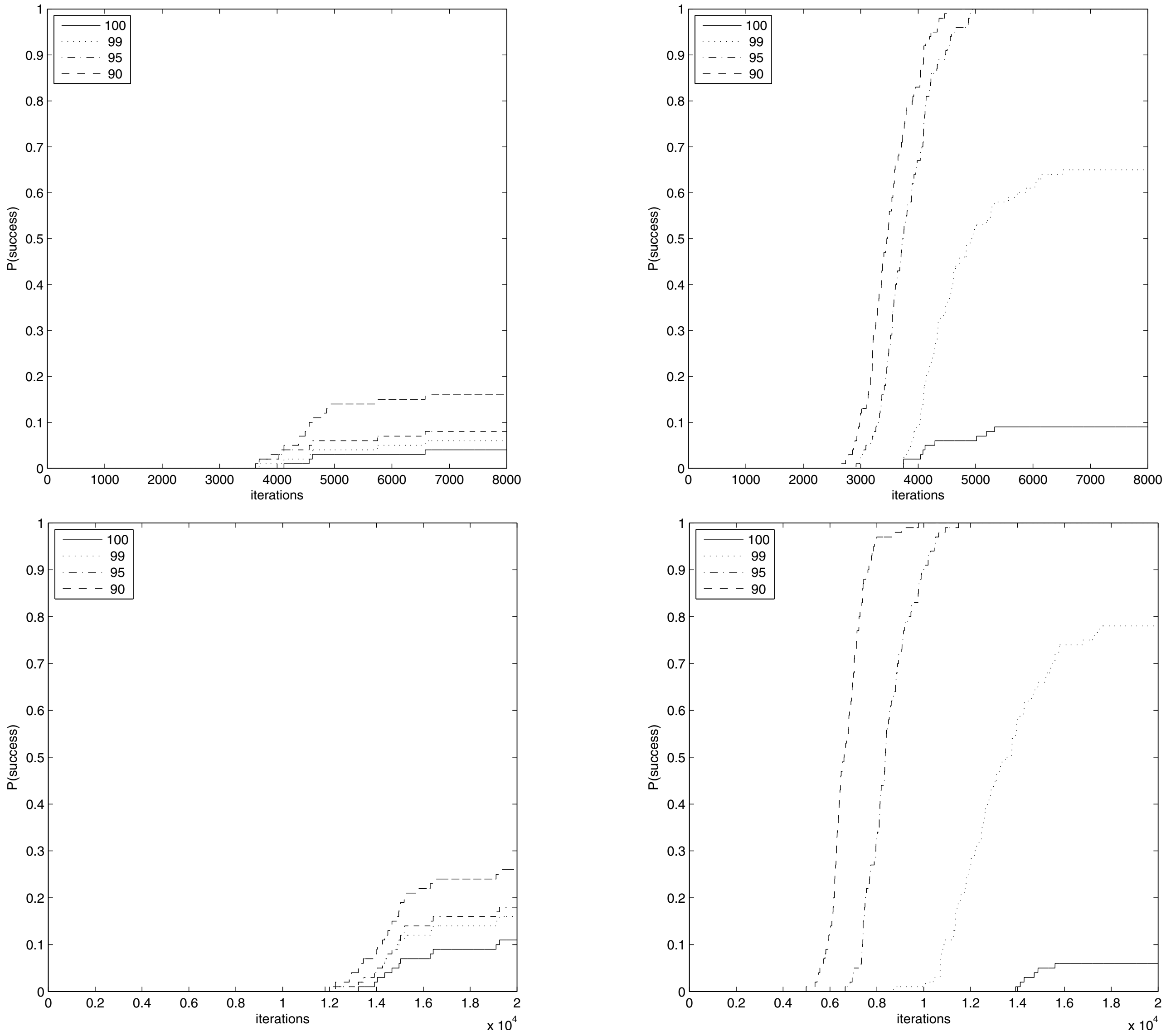

Fig. 2. Qualified runtime distributions for simulations with $\mu = 625$. The upper row corresponds to panmictic connectivity and the lower row to a von Neumann neighborhood. Similarly, the left column corresponds to genotypes initialized in $[0, 1]$ and the right one to initialization in $[0, 0.5]$ (memes are initialized in [0,1] in both cases). The curves indicate the probability that the population converges to a meme in the i-th initial percentile of the population as a function of the number of iterations. Note the different scale on the x-axis.

the number of iterations. Observe how the probabilities are below 10% for memes above the 95% percentile in the first scenario, whereas this probability is 100% in the second scenario. In the latter a spurious match between a very good genotype and a bad meme cannot happen since these very good solutions do not exist initially. Furthermore, high-potential memes by initially attaching themselves to bad genotypes can highly improve the quality of the latter in the initial steps, thus increasing their chances of survival.

Let us now turn our attention to the effect of the spatial structure. The bottom row of Fig. 1 shows the distribution of memes for the case of a von Neumann neighborhood. Note how a pattern similar to the panmictic case is observed with respect to genotype initialization. A more detailed inspection reveals several differences, though. Firstly, observe how the convergence is slower in this case (e.g., the scale in the x-axis is larger). This is a well-known effect of using the spatial structure and is commonly exploited in the context of evolutionary algorithms to promote diversity and thus decrease the chances of getting stuck in local optima (Dorronsoro and

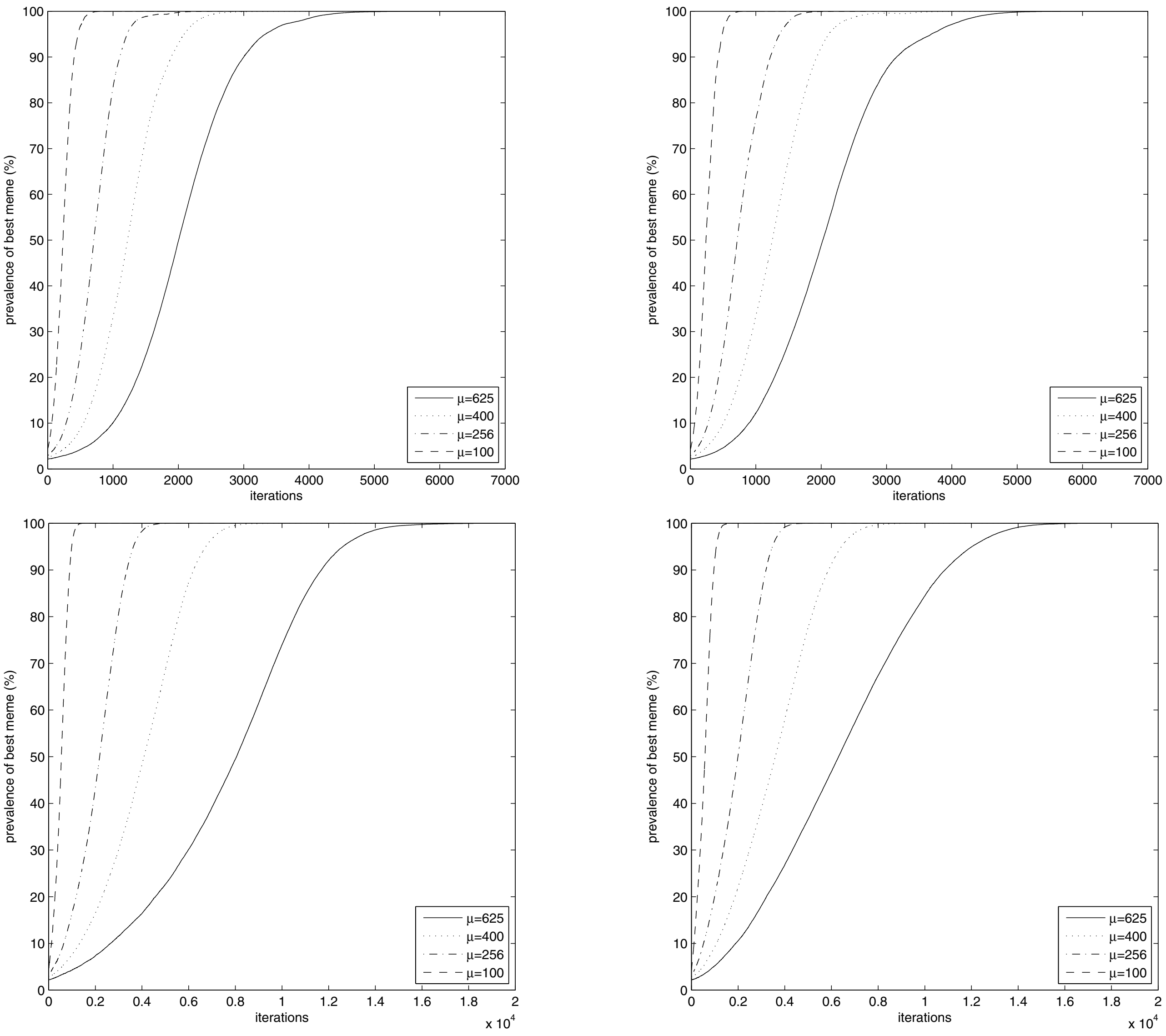

Fig. 3. Growth curves for different population sizes. The upper row corresponds to panmictic connectivity and the lower row to a von Neumann neighborhood. Similarly, the left column corresponds to genotypes initialized in $[0, 1]$ and the right one to initialization in $[0, 0.5]$ (memes are initialized in [0,1] in both cases). Note the different scale on the x-axis.

Alba, 2008; Tomassini, 2005). In the case of MMAs this has an additional advantage: a slower convergence increases the lifespan of individual memes, thus giving them more chances to improve their hosts if they have the potential to do so. Hence, the algorithm is more robust and can better cope with hitchhiking inert memes. This can be seen in the meme map in the bottom row of Fig. 1 by a larger prevalence of lighter-gray areas, and more clearly in the QRTDs (bottom row of Fig. 2), e.g., the 95% percentile is reached with nearly 20% probability in the case of $[0, 1]$-initialization (cf. below 10% in the panmictic case), and the 99% percentile is reached with nearly 80% probability for $[0, 0.5]$-initialization (cf. about 65% in the panmictic case). A sign-rank test (Wilcoxon, 1945) indicates that the difference in the final percentile reached is statistically significant in both cases ($\alpha = .05$).

Finally, we consider the takeover time, namely, the time required for a meme (not necessarily the best one as shown previously) to completely dominate the population. Figure 3 shows the growth curves, depicting the percentage of the meme pool occupied by the most repeated meme (note that the most repeated meme need not be the same throughout a run; we simply count the number of copies of the most repeated one at each time step). These curves exhibit the typical shape of the

Table 1. Fitting growth curves to a logistic function. For each algorithm configuration, the scale parameter α and the mean squared error are shown.

		population size							
		$\mu = 100$		$\mu = 256$		$\mu = 400$		$\mu = 625$	
topology	G	α	mse	α	mse	α	mse	α	mse
panmictic	0.5	81.63	0.000042	210.79	0.000069	324.65	0.000064	521.93	0.000017
	1.0	77.71	0.000008	187.90	0.000030	297.90	0.000027	462.87	0.000010
von Neumann	0.5	186.82	0.000366	578.91	0.000554	1046.43	0.000361	1975.00	0.000318
	1.0	168.57	0.000322	585.80	0.000532	1057.52	0.000537	1945.63	0.000705

well-known logistic model $f(t) = 1/(1 + Ke^{-t/\alpha})$. Indeed, such a model was proposed early on in the literature by Sarma and De Jong (1997) in the context of spatially structured evolutionary algorithms. While by no means the only alternative (see, e.g., Giacobini *et al.*, 2003) it serves as a good starting point for quantifying the growth of the dominant meme. Qualitatively, we observe, as expected, the well-known pattern of slower convergence for increasing population sizes and for the von Neumann topology (Giacobini *et al.*, 2005) as opposed to the panmictic population. From a quantitative point of view, we have fitted the growth data to a logistic curve to identify the scale factor α that renders the number of iterations dimensionless. The resulting data are shown in Table 1. As can be seen, the fit is quite good, yielding very low mean squared errors. The scale parameters are quite similar for variants with the same topology, and are about 2–5 times larger for the von Neumann topology than for the panmictic case, corresponding to the relative takeover time which can be seen in Fig. 3. With respect to the population size, the increase in the scale parameter admits a linear interpolation $\alpha = a + b\mu$, yielding values of $b = 0.84$ and $b = 0.74$ for the panmictic case and $b = 3.43$ and $b = 3.40$ for grid topology with the von Neumann connectivity.

5. Experimental results

Having analyzed the idealized model of MMAs, we now turn our attention to an operative algorithmic incarnation of MMAs so as to validate the previous findings. The MMA considered will be described next; subsequently, we present the problem benchmark used in the experimentation; finally, we report on and analyze experimental results.

5.1. Multimemetic algorithm.

We have considered an MMA following the framework defined by Smith (2008; 2012). To be more precise, we have considered a population of individuals carrying binary genotypes and single memes, which represent rewriting rules. Such rules are of the kind $A \rightarrow C$, where both A and C are patterns of a certain length. These patterns are expressed as strings composed of symbols from a ternary alphabet $\Sigma = \{0, 1, \#\}$, where '$\#$' represents a wildcard symbol. Given a certain genotype $b_1 b_2 \cdots b_n$ a rule $A \rightarrow C$ could be potentially applied in any position i in which the antecedent $A = a_1 \cdots a_r$ of the rule matches the contents of the genotype (i.e., $b_i b_{i+1} \cdots b_{i+r-1} = a_1 \cdots a_r$). The action of the rule would be to rewrite that portion of the genotype, implanting the consequent $C = c_1 \cdots c_r$ of the rule (i.e., setting $b_i b_{i+1} \cdots b_{i+r-1} \leftarrow c_1 \cdots c_r$). In this process, the wildcard '$\#$' is interpreted as follows:

- In the antecedent of the rule it denotes a 'don't-care' symbol, meaning that it can match both '0' and '1'.
- In the consequent of the rule it denotes a 'don't-change' symbol, meaning that the corresponding position of the genotype is left unchanged.

For instance, let a genotype be 11101100, and let a rule be $10\# \rightarrow 0\#1$. A possible application of the rule could be the following:

$$11\overbrace{101}^{A}100 \xrightarrow{\text{rule}} 11\underbrace{001}_{C}100.$$

In this example, there is another potential application point, namely, the sixth position, resulting in the genotype 11101001. In general, the order in which genotype bits are scanned is randomized to avoid positional bias. When a match is found, the resulting neighboring genotype is generated and evaluated according to the objective function under consideration. In order to keep the total cost of the process under control, a parameter w is used to determine the maximal number of rule applications. The best neighbor generated (if better that the current genotype) is kept. Let us finally note in passing that, by virtue of the pattern definition used, the whole process of genotype rewriting by means of rules can be regarded as a heuristic way of manipulating schemata, injecting instances of new schemata in the place of extant ones whenever the fitness function dictates the former are better.

Apart from the use of memes embedded within individuals, our MMA otherwise resembles a steady-state memetic algorithm in which parents are selected using binary tournament, recombination, mutation and local search (conducted using the meme carried out by the individual as illustrated before). Then they are used to generate the offspring, which replaces the worst parent (following our model presented in Section 3).

5.2. Test suite. In order to test the MMA, we have considered three different problems defined on binary strings, namely, Deb's trap function (Deb and Goldberg, 1993), the massively multimodal deceptive problem (MMDP) (Goldberg *et al.*, 1992), and Watson *et al.*'s (1998) hierarchical-if-and-only-if (H-IFF) function. Let us describe each of them separately in the following.

Deb's 4-bit fully deceptive function (TRAP henceforth) is defined so as to have a single global optimum in a fairly isolated location and a local optimum surrounded by increasingly fit individuals (thus deceiving gradient-based methods to follow the path towards this local optimum). More precisely, TRAP is defined as a function of unitation as

$$f(b_1 \cdots b_4) = \begin{cases} 0.6 - 0.2 \cdot u(b_1 \cdots b_4) & \text{if } u(b_1 \cdots b_4) < 4, \\ 1 & \text{if } u(b_1 \cdots b_4) = 4, \end{cases} \tag{16}$$

where $u(s_1 \cdots s_i) = \sum_j s_j$ is the unitation (number of 1s in a binary string) function. The function above is used as the basic block upon which to build a higher-order problem by concatenating k such blocks, and defining the fitness of a $4k$-bit string as the sum of the function value for all blocks/subproblems. In our experiments we have considered $k = 32$ subproblems (and hence *opt* $= 32$).

The next function is the MMDP. This is a bipolar deceptive function with two global optima far apart from each other (at the extreme unitation values), and a local deceptive attractor at the halfway point. Placing the deceptive attractor here implies that there are massively more local optima than global optima ($\binom{L}{L/2}$ local vs. 2 global). The precise definition of the basic MMDP block (for 6 bits) is as follows:

$$f(b_1 \cdots b_6) = \begin{cases} 1, & u(b_1 \cdots b_6) \in \{0, 6\}, \\ 0, & u(b_1 \cdots b_6) \in \{1, 5\}, \\ 0.360384, & u(b_1 \cdots b_6) \in \{2, 4\}, \\ 0.640576, & u(b_1 \cdots b_6) = 3. \end{cases} \tag{17}$$

Again, we concatenate k copies of this basic block to create a harder problem. We have considered $k = 24$ (thus, *opt* $= 24$).

Lastly, the H-IFF function is a recursive epistatic problem whose hierarchical structure forces the search algorithm to move from searching combinations of bits to searching combinations of increasingly higher-order schemata (and thus challenging the abilities of the algorithm to identify and combine good building blocks). This function is defined for binary strings of 2^k bits by means of two auxiliary functions:

$$f : \{0, 1, \bullet\} \to \{0, 1\}. \tag{18}$$

$$t : \{0, 1, \bullet\} \to \{0, 1, \bullet\}, \tag{19}$$

where '•' is used as a *null* or NaN value. More precisely, let us define these two latter functions as follows:

$$f(a, b) = \begin{cases} 1, & a = b \neq \bullet, \\ 0, & \text{otherwise}, \end{cases} \tag{20}$$

$$t(a, b) = \begin{cases} 0, & a = b = 0, \\ 1, & a = b = 1, \\ \bullet, & \text{otherwise}. \end{cases} \tag{21}$$

Intuitively, f is the function used to score the contribution of building blocks and t is a function used to capture their interaction. They are used as follows:

$$\text{H-IFF}_k(b_1 \cdots b_n) = \sum_{i=1}^{n/2} f(b_{2i-1}, b_{2i}) + 2\text{H-IFF}_{k-1}(b'_1, \cdots, b'_{n/2}), \tag{22}$$

where

$$b'_i = t(b_{2i-1}, b_{2i}) \tag{23}$$

and

$$\text{H-IFF}_0(\cdot) = 1. \tag{24}$$

This recursive definition leads to a hierarchical tree-like dependency structure: increasingly large building blocks are considered as we move upwards in the tree, and the weight of their interaction is correspondingly greater. We have considered $k = 7$ (i.e., 128-bit strings), which, using our definition above of the HIFF function, implies *opt* $= 576$.

5.3. Results. The experiments have considered the MMA described in Section 5.1 using a population size of $\mu = 256$, binary tournament selection, recombination probability $p_X = 1.0$ and mutation probability $p_M = 1/\ell$, where ℓ is the length of genotypes. Genotype recombination is done using single-point crossover. As for the memes, with probability p_X the offspring will inherit the meme carried by the best of the parents (otherwise, it inherits the meme of its first parent). Memes are defined as rules of length $r = 3$ and are applied with parameter $w = 5$. We consider a maximum number of 25,000 evaluations for 128-bit TRAP and H-IFF and 50,000 evaluations for a 144-bit MMDP. In all cases the cost of

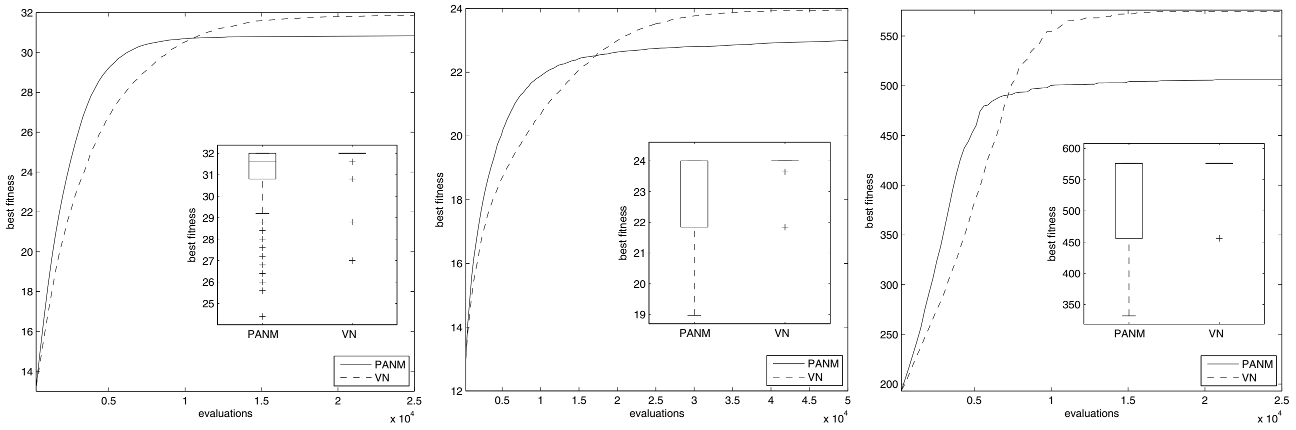

Fig. 4. Best fitness (averaged for 100 runs) of MMAs with panmictic and von Neumann connectivity. The inset shows the distribution of fitness values attained. From left to right: TRAP, the MMDP and H-IFF.

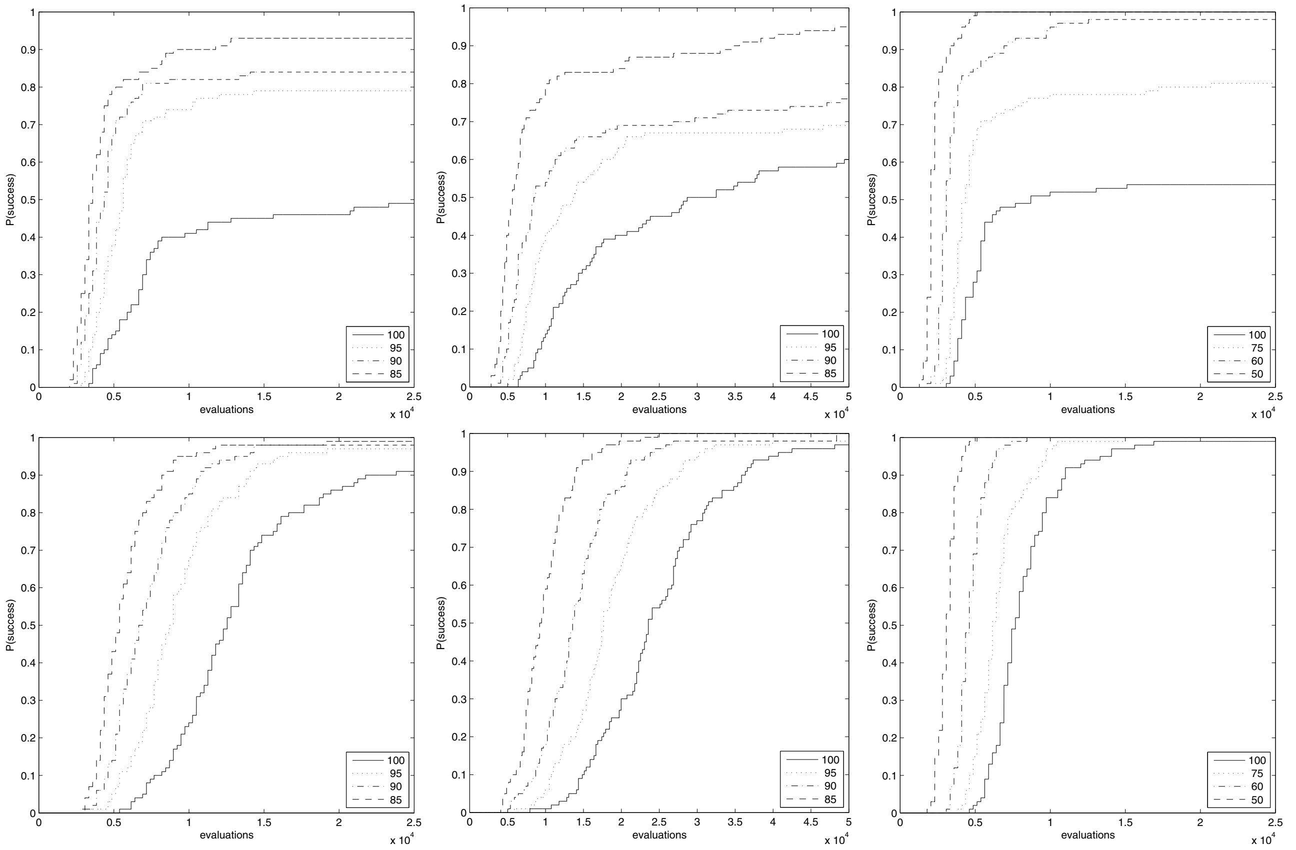

Fig. 5. Qualified runtime distributions for MMA variants. The top row corresponds to panmictic connectivity and the bottom one to a von Neumann neighborhood. Similarly, the leftmost column corresponds to TRAP, the middle one to the MMDP and the rightmost one to H-IFF. The curves indicate the probability that the population converges to a solution whose quality is $i\%$ of the optimum (i.e., p indicates an approximation of $(100-p)\%$), as a function of the number of iterations. These approximation percentages are different in the case of H-IFF due to the particularities of the distribution of values in this function (bigger gaps between the optimal solution and suboptimal ones).

Table 2. Results (averaged for 100 runs) of MMAs with panmictic and von Neumann connectivity. The number of times the optimum is found (n_{opt}), the median ($\tilde{x}$), mean ($\bar{x}$) and standard error of the mean ($\sigma_{\bar{x}}$) are shown.

	TRAP			MMDP			H-IFF		
	n_{opt}	$\tilde{x}$	$\bar{x} \pm \sigma_{\bar{x}}$	n_{opt}	$\tilde{x}$	$\bar{x} \pm \sigma_{\bar{x}}$	n_{opt}	$\tilde{x}$	$\bar{x} \pm \sigma_{\bar{x}}$
panmictic	49	31.6	30.8 ± 0.2	60	24.0	23.0 ± 0.1	54	576.0	506.1 ± 8.1
von Neumann	91	32.0	31.9 ± 0.1	97	24.0	24.0 ± 0.0	99	576.0	574.8 ± 1.2

meme application is accounted as a partial evaluation (as the fraction of subproblems that need being re-evaluated after modifying the genotype), thus effectively measuring the overhead related to using them. As in Section 4, two algorithmic variants are considered: an MMA with panmictic population and with a spatially structured (toroidal square grid with the von Neumann topology) population. All experiments with either algorithm and test problem are replicated one hundred times.

Let us firstly observe the evolution of the best fitness in Fig. 4. The panmictic MMA (PANM) exhibits a faster initial convergence but it is quickly surpassed by the von Neumann MMA (VN), which manages to keep steadily improving for a longer time, finally attaining notably better solutions—see the boxplots in the insets of Fig. 4, showing the distribution of fitness values achieved by each algorithm, and the detailed numerical data in Table 2. This improvement is consistent across all the problems in the test suite and can be shown to be statistically significant to a level of $\alpha = 0.05$ using a Wilcoxon ranksum test. A complementary perspective on this improved convergence of the VN is shown in Fig. 5. By inspecting the QRTDs we can observe how there is a marked improvement on the success probability in the case of the VN with respect to PANM in all three problems. Not only is the VN capable of finding the optimal solution more often (91%–99% success rate for the VN vs. 49%–60% for PANM), but it is also capable of approaching a high quality solution on a more frequent basis. Consistent with the findings of the idealized model, this can be due to the higher diversity of memes which are capable of sustaining a fruitful Lamarckian search for a longer time. The evolution of diversity (measured in terms of the entropy of memes in the population) is depicted in Fig. 6 (top). Memetic diversity decreases more gently in the case of the VN, thus giving potentially good memes more opportunities to express themselves (ultimately resulting in the improved quality of the results as mentioned before). In fact, as shown in Fig. 6 (bottom), the success rate of meme application is eventually higher in the case of the non-panmictic MMA, which helps to explain the higher fitness performance of the latter.

6. Conclusions

We have presented some initial steps in analyzing meme propagation in MMAs. Using an idealized model of genotypes and memes, we have shown that the selection intensity plays a very important role in allowing high-potential memes to proliferate. In a panmictic model, good memes will dominate the final population when the starting solutions have a substantial improvement margin on average. When this margin is smaller, average memes can hitchhike their way to the final stages of the evolution and make other comparatively better memes become extinct. In the presence of a spatial structure inducing longer takeover times (in our case, a toroidal square grid with the von Neumann topology), this hitchhiking effect is somewhat mitigated, allowing good memes to express themselves and increasing their chances for making it to the final population. These findings have been empirically corroborated by deploying an actual MMA on three different pseudo-Boolean functions, namely, concatenated trap functions (both unimodal and massively multimodal), and Watson *et al.*'s (1998) hierarchical-if-and-only-if function. The MMA with a spatially structured population has been shown to maintain a higher memetic diversity and to provide notably improved results with respect to its panmictic counterpart.

An interesting line of future research will consider other topologies and study their effect on meme propagation. We are particularly interested in the use of the island model (Cantu-Paz, 2000; Schaefer *et al.*, 2012), which can be readily deployed on distributed computational environments (Alba, 2005). Work is already in progress in this area. Looking beyond, another topic for further research is the extension of this analysis to coevolutionary memetic algorithms (Smith, 2007; 2012) in which memes are detached from genotypes and co-evolve alongside the latter in a separate population.

Acknowledgment

This work is an extended version of our previous results (Nogueras and Cotta, 2013). We acknowledge the support of Spanish MICINN under the project *ANYSELF* (TIN2011-28627-C04-01)[2], by Junta de Andalucía under the project *DNEMESIS* (P10-TIC-6083)[3] and Universidad de Málaga, Campus de Excelencia Internacional Andalucía Tech.

[2] http://anyself.wordpress.com/.

[3] http://dnemesis.lcc.uma.es/wordpress/.

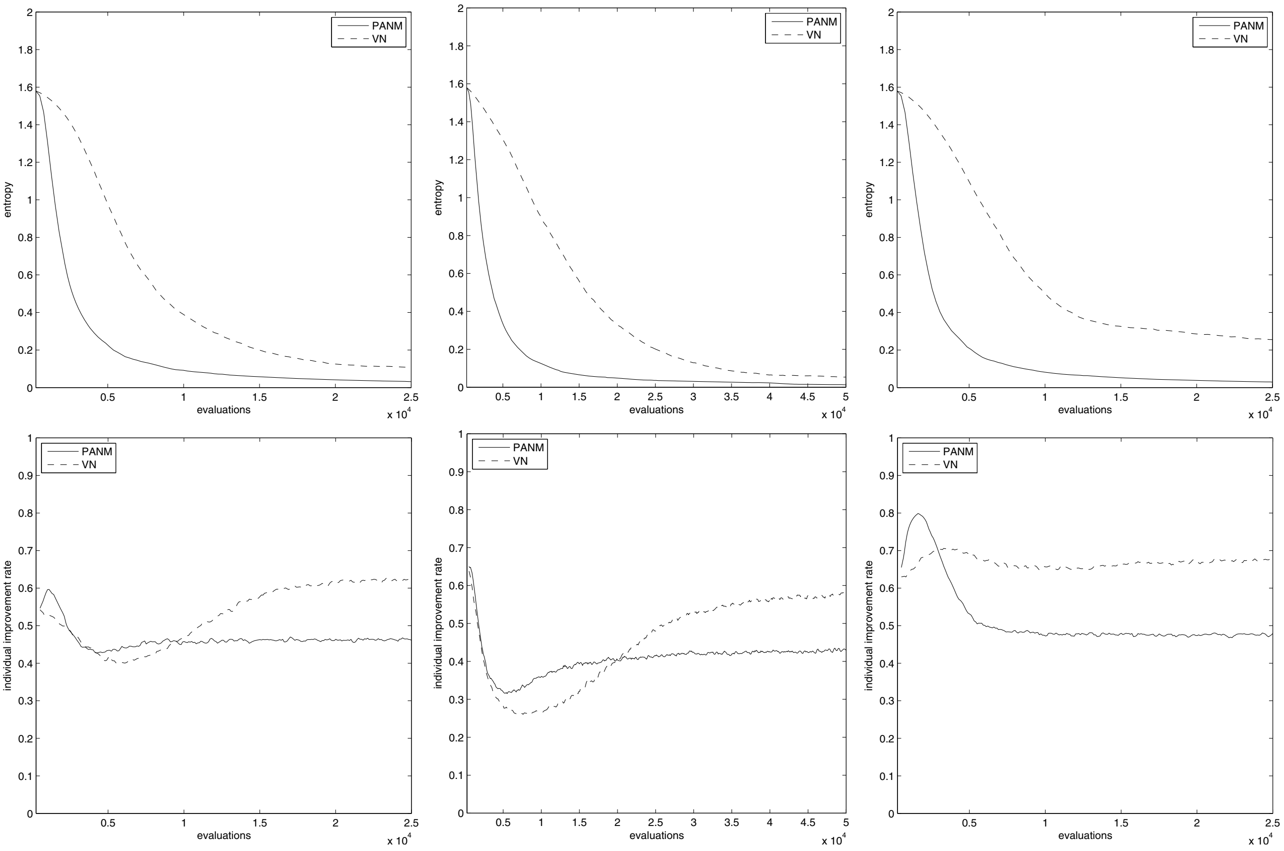

Fig. 6. Evolution of diversity in memes (top), evolution of meme success (fraction of local search stages that result in an improved individual) (bottom). From left to right: TRAP, the MMDP and H-IFF.

References

Alba, E. (2005). *Parallel Metaheuristics: A New Class of Algorithms*, Wiley-Interscience, Hoboken, NJ.

Alba, E. and Luque, G. (2004). Growth curves and takeover time in evolutionary algorithms, *in* K. Deb (Ed.), *GECCO 2004*, Lecture Notes in Computer Science, Vol. 3102, Springer-Verlag, Berlin/Heidelberg, pp. 864–876.

Cantu-Paz, E. (2000). *Efficient and Accurate Parallel Genetic Algorithms*, Kluwer Academic Publishers, Norwell, MA.

Chakhlevitch, K. and Cowling, P. (2008). Hyperheuristics: Recent developments, *in* C. Cotta, M. Sevaux and K. Sörensen (Eds.), *Adaptive and Multilevel Metaheuristics*, Studies in Computational Intelligence, Vol. 136, Springer-Verlag, Berlin/Heidelberg, pp. 3–29.

Chen, X. and Ong, Y.-S. (2012). A conceptual modeling of meme complexes in stochastic search, *IEEE Transactions on Systems, Man, and Cybernetics, Part C* **42**(5): 612–625.

Chen, X., Ong, Y.-S., Lim, M.-H. and Tan, K.C. (2011). A multi-facet survey on memetic computation, *IEEE Transactions on Evolutionary Computation* **15**(5): 591–607.

Cowling, P., Kendall, G. and Soubeiga, E. (2008). A hyperheuristic approach to schedule a sales submit, *in* E. Burke and W. Erben (Eds.), *PATAT 2000*, Lecture Notes in Computer Science, Vol. 2079, Springer-Verlag, Berlin/Heidelberg, pp. 176–190.

Dawkins, R. (1976). *The Selfish Gene*, Clarendon Press, Oxford.

Deb, K. and Goldberg, D.E. (1993). Analyzing deception in trap functions, *in* L.D. Whitley (Ed.), *Second Workshop on Foundations of Genetic Algorithms*, Morgan Kaufmann, San Francisco, CA, pp. 93–108.

Dorronsoro, B. and Alba, E. (2008). *Cellular Genetic Algorithms*, Operations Research/Computer Science Interfaces, Vol. 42, Springer, New York, NY.

Giacobini, M., Alba, E. and Tomassini, M. (2003). Selection intensity in asynchronous cellular evolutionary algorithms, *in* E. Cantú-Paz *et al.* (Eds.), *Genetic and Evolutionary Computation Conference, GECCO 2003*, Lecture Notes in Computer Science, Vol. 2723, Springer-Verlag, Berlin/Heidelberg, pp. 955–966.

Giacobini, M., Tomassini, M., Tettamanzi, A. and Alba, E. (2005). Selection intensity in cellular evolutionary algorithms for regular lattices, *IEEE Transactions on Evolutionary Computation* **9**(5): 489–505.

Goldberg, D.E., Deb, K. and Horn, J. (1992). Massive multimodality, deception, and genetic algorithms, *Parallel Problem Solving from Nature, PPSN II*, Elsevier, Brussels, pp. 37–48.

Hart, W., Krasnogor, N. and Smith, J. (2005). *Recent Advances in Memetic Algorithms*, Studies in Fuzziness and Soft Computing, Vol. 166, Springer-Verlag, Berlin/Heidelberg, pp. 3–27.

Hoos, H. and Stützle, T. (2004). *Stochastic Local Search: Foundations & Applications*, Morgan Kaufmann Publishers Inc., San Francisco, CA.

Karcz-Dulęba, I. (2004). Time to the convergence of evolution in the space of population states, *International Journal of Applied Mathematics and Computer Science* **14**(3): 279–287.

Krasnogor, N. (2004). Self generating metaheuristics in bioinformatics: The proteins structure comparison case, *Genetic Programming and Evolvable Machines* **5**(2): 181–201.

Krasnogor, N., Blackburne, B., Burke, E. and Hirst, J. (2002). Multimeme algorithms for protein structure prediction, *in* J. Merelo *et al.* (Eds.), *Parallel Problem Solving From Nature VII*, Lecture Notes in Computer Science, Vol. 2439, Springer, Berlin, pp. 769–778.

Krasnogor, N. and Gustafson, S. (2004). A study on the use of "self-generation" in memetic algorithms, *Natural Computing* **3**(1): 53–76.

Krasnogor, N. and Smith, J. (2005). A tutorial for competent memetic algorithms: Model, taxonomy, and design issues, *IEEE Transactions on Evolutionary Computation* **9**(5): 474–488.

Moscato, P. (1989). On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms, *Caltech Concurrent Computation Program, Report 826*, California Institute of Technology, Pasadena, CA.

Moscato, P. (1999). Memetic algorithms: A short introduction, *in* D. Corne, M. Dorigo and F. Glover (Eds.), *New Ideas in Optimization*, McGraw-Hill, Maidenhead, pp. 219–234.

Moscato, P. and Cotta, C. (2010). A modern introduction to memetic algorithms, *in* M. Gendreau and J.-Y. Potvin (Eds.), *Handbook of Metaheuristics*, International Series in Operations Research & Management Science, Vol. 146, Springer, New York, NY, pp. 141–183.

Neri, F. and Cotta, C. (2012). Memetic algorithms and memetic computing optimization: A literature review, *Swarm and Evolutionary Computation* **2**: 1–14.

Neri, F., Cotta, C. and Moscato, P. (2012). *Handbook of Memetic Algorithms*, Studies in Computational Intelligence, Vol. 379, Springer-Verlag, Berlin/Heidelberg.

Neri, F., Tirronen, V., Kärkkäinen, T. and Rossi, T. (2007). Fitness diversity based adaptation in multimeme algorithms: A comparative study, *IEEE Congress on Evolutionary Computation, CEC 2007, Singapore*, pp. 2374–2381.

Nogueras, R. and Cotta, C. (2013). Analyzing meme propagation in multimemetic algorithms: Initial investigations, *Proceedings of the 2013 Federated Conference on Computer Science and Information Systems, Cracow, Poland*, pp. 1013–1019.

Norman, M. and Moscato, P. (1989). A competitive and cooperative approach to complex combinatorial search, *Proceedings of the 20th Informatics and Operations Research Meeting, Buenos Aires, Argentina*, pp. 3.15–3.29.

Ong, Y.-S. and Keane, A. (2004). Meta-Lamarckian learning in memetic algorithms, *IEEE Transactions on Evolutionary Computation* **8**(2): 99–110.

Ong, Y.-S., Lim, M.-H. and Chen, X. (2010). Memetic computation-past, present and future, *IEEE Computational Intelligence Magazine* **5**(2): 24–31.

Ong, Y.-S., Lim, M.-H., Zhu, N. and Wong, K.-W. (2006). Classification of adaptive memetic algorithms: A comparative study, *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* **36**(1): 141–152.

Rudolph, G. and Sprave, J. (1995). A cellular genetic algorithm with self-adjusting acceptance threshold, *1st IEE/IEEE International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications, London, UK*, pp. 365–372.

Sarma, J. and De Jong, K. (1997). An analysis of local selection algorithms in a spatially structured evolutionary algorithm, *in* T. Bäck (Ed.), *7th International Conference on Genetic Algorithms*, Morgan Kaufmann, San Francisco, CA, pp. 181–186.

Schaefer, R., Byrski, A. and Smołka, M. (2012). The island model as a Markov dynamic system, *International Journal of Applied Mathematics and Computer Science* **22**(4): 971–984, DOI: 10.2478/v10006-012-0072-z.

Schönfisch, B. and de Roos, A. (1999). Synchronous and asynchronous updating in cellular automata, *BioSystems* **51**(3): 123–143.

Smith, J.E. (2007). Coevolving memetic algorithms: A review and progress report, *IEEE Transactions on Systems, Man, and Cybernetics, Part B* **37**(1): 6–17.

Smith, J.E. (2008). Self-adaptation in evolutionary algorithms for combinatorial optimisation, *in* C. Cotta, M. Sevaux and K. Sörensen (Eds.), *Adaptive and Multilevel Metaheuristics*, Studies in Computational Intelligence, Vol. 136, Springer, Berlin/Heidelberg, pp. 31–57.

Smith, J.E. (2012). Self-adaptative and coevolving memetic algorithms, *in* F. Neri, C. Cotta and P. Moscato (Eds.), *Handbook of Memetic Algorithms*, Studies in Computational Intelligence, Vol. 379, Springer-Verlag, Berlin/Heidelberg, pp. 167–188.

Tomassini, M. (2005). *Spatially Structured Evolutionary Algorithms*, Natural Computing Series, Springer-Verlag, Berlin/Heidelberg.

Watson, R.A., Hornby, G.S. and Pollack, J.B. (1998). Modeling building-block interdependency, *in* A. Eiben, T. Back, M. Schoenauer and H.-P. Schwefel (Eds.), *Parallel Problem Solving from Nature, PPSN V*, Lecture Notes in Computer Science, Vol. 1498, Springer-Verlag, Berlin/Heidelberg, pp. 97–106.

Wilcoxon, F. (1945). Individual comparisons by ranking methods, *Biometrics* **1**(6): 80–83.

Rafael Nogueras obtained an M.Sc. degree in computer science in 1998 from the University of Málaga, Spain. He is currently working on his Ph.D. at that university, under the supervision of Carlos Cotta, in the area of memetic algorithms.

Carlos Cotta obtained his M.Sc. and Ph.D. in computer science from the University of Málaga, Spain, in 1994 and 1998, respectively. He has held a tenured professorship at that university since 2001. His main research areas involve metaheuristic optimization, in particular hybrid and memetic approaches with the focus on both algorithmic and applied aspects (particularly combinatorial optimization) and complex systems.

Received: 6 June 2014
Revised: 26 November 2014