Shuffle and Mate: A Dynamic Model for Spatially Structured Evolutionary Algorithms

Carlos M. Fernandes[1,2], Juan L.J. Laredo[3], Juan Julian Merelo[2], Carlos Cotta[4], Rafael Nogueras[4] and Agostinho C. Rosa[1]

[1] LaSEEB-ISR-IST, University of Lisbon, Portugal
[2] Department of Architecture and Computer Technology, Univ. of Granada, Spain
[3] Faculty of Sciences, Technology and Communications, Univ. of Luxembourg
[4] Departamento de Lenguages y Ciencias de la Computación, Univ. of Malaga, Spain
{cfernandes,acrosa}@laseeb.org, juan.jimenez@uni.lu,
{jjmerelo,rafael.nogueras}@gmail.com, ccottap@lcc.uma.es

Abstract. This paper studies a self-organized framework for modeling dynamic topologies in spatially structured Evolutionary Algorithms (EAs). The model consists of a 2-dimensional grid of nodes where the individuals interact and self-organize into clusters. During the search process, the individuals move through the grid, following a pre-defined simple rule. In order to evaluate the model, a dynamic cellular Genetic Algorithm (dcGA) is built over the proposed topology and four different movement rules are tested. The results show that when the ratio between the number of nodes in the grid and the population size is above 4:1, the individuals self-organize into highly dynamic clusters and significantly improve results attained by standard cGAs with static topologies on a set of deceptive and multimodal functions.

1 Introduction

In panmictic populations, every individual is allowed to interact with every other individual. Standard Evolutionary Algorithms (EAs) mimic this strategy for parent selection and recombination, but large-scale problems or deceptive functions with multiple local optima may require other type of population structures. In recent years, non-panmictic EAs [10], which restrict the interaction according to a pre-defined or evolving structure, are gaining increasing attention by the community.

In non-panmictic EAs, the population structure specifies a network of acquaintances over which individuals can interact (i.e., mating or selection is restricted to neighborhoods within the network). These non-panmictic EAs are also known as *spatially structured* EAs [10], a category that includes fine-grained approaches such as cellular EAs (cEAs) [1] and coarse-grained approaches such as island models [2]. In cEAs, the population is distributed in a grid and the interaction is restricted to the individuals' neighborhood. In island EAs, different subpopulations evolve isolated from each other and occasionally exchange individuals using a predefined strategy which specifies the rate and quantity of information to transfer.

T. Bartz-Beielstein et al. (Eds.): PPSN XIII 2014, LNCS 8672, pp. 50–59, 2014.

The main disadvantage with island and cellular EAs is that their base-structures, which have a great influence on the algorithm performance, require extra designing and tuning effort. In the case of island models, this added complexity translates in deciding policies for the migration frequency, selection and replacement of migrants and the topology itself. As for traditional cEAs, they use static structures that impose a rigid connectivity between the individuals.

Furthermore, even though cEAs may achieve a better management of the genetic diversity in the population when compared to panmictic EAs, the balance between exploration and exploitation may not be sufficient for problems with deceptive or epistatic fitness landscapes. Since the population is globally connected, information may spread quickly and local optima can easily take over the entire population. The investigation in this paper is an attempt to design a simple dynamic topology for cEAs, with a varying neighborhood degree and an intrinsic clustering behavior that approaches the cEA to an island model, keeps genetic diversity at a higher level and prevents sub-optimal solutions to take over the population.

In the proposed topology, n individuals are distributed in a 2-dimensional $m-$nodes grid where it holds $m > n$. Every time-step, each individual tries to recombine with one of the individuals in its Moore neighborhood (if there are any). Furthermore, the structure is dynamic: in each time-step, every chromosome updates its position by moving to a neighboring node (if there are empty nodes in the individual's neighborhood), according to a pre-defined rule that selects the destination. The position update rule, which is implemented locally and without any knowledge on the global state of the system, can be based on stigmergy [5] or Brownian movements.

When stigmergic behavior is induced by a stigmergic rule (i.e., individuals communicate via the environment), different niches of individuals appear and disappear at run-time. This clustering behavior is an emergent property of the model, and the resulting cEA has certain resemblance with an island model, with dynamic clusters (or sub-populations) of individuals with varying size.

We hypothesize that with this scheme the population diversity decreases at a lower rate (when compared to a standard topology), and, as a consequence, the performance of the cEA on deceptive and hard problems is improved. In this paper, the dynamic topology is tested on a cellular Genetic Algorithm (cGA). Four different strategies are described for the position update rule and the resulting algorithms are tested with a set of deceptive and epistatic functions that challenge EAs' abilities to combine building-blocks. The results show that when the ratio between the number of nodes in the grid and population size is above 4:1, the dynamic cGAs converge more often to the global optimum and significantly improve the performance of standard cGAs.

The remaining of the paper is structured as follows: Section 2 gives a background review on cEAs and on dynamic alternative topologies for cEAs; Section 3 describes the proposed system; Section 4 describes the experiments and discusses the results; Section 5 concludes the paper and outlines future lines of work.

2 Background Review

The initial objective of spatially structured EAs was to develop a framework for studying massive parallelization. However, the need to provide traditional EAs with a

proper balance between exploration and exploitation and overcome standard EAs drawbacks, like synchronicity, rigid connectivity and strong dependence on the problem, motivated several lines of research that explore the potentiality of different population structures in maintaining genetic diversity [10]. Additionally, complex population structures have been studied, some of them under the knowledge provided by recent developments in network theory.

In [1], Alba and Dorronsoro dynamically change the ratio that defines the neighborhood of interaction. Since the ratio may affect selection pressure, the authors analyze the influence of its value on the balance between exploration and exploitation. However, the base-structure of the cellular EA is maintained throughout the run. In [11], Whitacre et al. focus on two important conditions missing in EA populations: a self-organized definition of locality and interaction epistasis. With that purpose in mind, they proposed a dynamic structure and concluded that the two features, when combined, provide behaviors not present in the traditional spatially structured EAs. The most noticeable change is an unprecedented capacity for sustainable coexistence of genetically distinct individuals within a single population. The authors state that the capacity for sustained genetic diversity is not imposed on the population; instead, it emerges as a natural consequence of the dynamics of the system. Laredo et al. [7] proposed a framework for EAs based on peer-to-peer networks [9]. Within a simulated network, they model the dynamics of real networks and conclude that their system is able to achieve better performance than traditional EAs on a wide range of problems, while being scalable and resilient to the volatility of nodes in the network.

In order to deal with the specific issues that may affect the design and performance of spatially structured EAs, Fernandes et al. [3] devised a complex adaptive system to be used as a dynamic structure for populations. This model, which can be regarded as a cellular automaton [6] with short-term memory, uses stimergic communication and simple rules for movement on a grid of nodes, giving rise to self-organized clusters of particles. A noticeable feature of these clusters is that they keep evolving and changing shape, thus providing some kind of highly dynamic order. The authors demonstrate that the proposed system has indeed emergent properties that may prove useful for spatially structured EAs, or other spatially structured population-based metaheuristics.

In fact, this framework has been recently used to implement a spatially structured Particle Swarm Optimization (PSO) algorithm, in which the particles' interaction is defined by their position on the grid [4]. In this case, the position update rule is based on Brownian movement. Recently, Nogueras et al. [8] adapted the model in [3] to a spatially structured multimemetic algorithm with dynamic topology. The authors show that the dynamic topology maintains genetic diversity at a higher level and reduces the rate of convergence to local optima.

In this paper we take a different approach and test the framework as a dynamic topology for cellular Genetic Algorithms (cGAs), using position update rules that model Brownian movement and stigmergic behavior. The base model and the proposed topology are described in the following section.

3 Dynamic Topology

This section gives a formal description of the network and the transition rules that define the proposed model for dynamic population structures.

Let us consider a rectangular grid G of size $q \times s \geq \mu$. Each cell G_{uv} of the grid is a tuple $\langle \eta_{uv}, \zeta_{uv} \rangle$, where $\eta_{uv} \in \{1, \ldots, \mu\} \cup \{\bullet\}$ and $\zeta_{uv} \in (D \times \mathbb{N}) \cup \{\bullet\}$, for some domain D. The value η_{uv} indicates the index of the individual that occupies the position $\langle u, v \rangle$ in the grid. If $\eta_{uv} = \bullet$ then the corresponding position is empty. However, that same position may still have information, namely a mark (or clue) ζ_{uv}, that is placed by the individuals and provides a form of communication between them. If $\zeta_{uv} = \bullet$ then the position is empty and unmarked. Please note that when $q \times s = \mu$, the topology is the standard static 2-dimensional structure.

The marks are placed by individuals that occupied that position in the past and they consist of information about those individuals (captured by domain D), like their fitness ζ_{uv}^{f} or a copy of their genotype $\vec{x}$, as well as a time stamp ζ_{uv}^{t} that indicates the iteration in which the mark was placed. The marks have a lifespan of K iterations, after which they are deleted.

Initially, $G_{uv} = (\bullet,\bullet)$ for all $\langle u, v \rangle$. Then, individuals are placed randomly on the grid (only one individual per node). Afterwards, all individuals are subject to a movement phase (or position update), followed by an evolutionary phase. The process (position update and evolutionary phase) repeats until a stop criterion is met.

The evolutionary phase is the standard iteration of a cEA, comprising selection, recombination mutation and replacement. The only difference to a cEA with static structure is that in this case an individual may find empty nodes in its neighborhood, and the selection pool is restricted to the individuals that occupy adjacent nodes. If at a given time-step an individual has no neighbors, then there is no recombination event for that individual in that specific iteration.

In the position update phase, each individual moves to an adjacent empty node. Adjacency is defined by the Moore neighborhood of radius r, so an individual i at $\rho_g(i) = \langle u, v \rangle$ can move to an empty node $\langle u', v' \rangle$ for which $L_\infty(\langle u, v \rangle, \langle u', v' \rangle) \leq r$. If no empty position is available, the individual stays in the same node. Otherwise, it picks a neighboring empty node according to the marks on them. If there are no marks, the destination is chosen randomly amongst the free nodes.

We consider two possibilities for the position update phase: stimergic, whereby the individual looks for a mark that is similar to itself; and Brownian, whereby the individual selects an empty neighbor regardless of the marks. For the first option, let $\mathcal{N}\langle u, v \rangle = \{\langle u^{(1)}, v^{(1)} \rangle, \ldots, \langle u^{w}, v^{w} \rangle\}$ be the collection of empty neighboring nodes and let i be the individual to move. Then, the individual attempts to move to a node whose mark is as close as possible to its own corresponding trait (fitness or genotype) or to an adjacent cell picked at random if there are no marks in the neighborhood. This strategy leads to the self-organization of the population into dynamic clusters [3], [8]. In the alternative Brownian policy, the individual moves to an adjacent empty position picked at random. In either case, the process is repeated for the whole population. The following section describes the results attained by dynamic cGAs with stigmergic and Brownian movement.

4 Results and Discussion

In order to investigate their performance, the proposed dynamic topologies were tested on a set of functions that challenge the EAs ability to combine building-blocks and demand a careful balance between exploration and exploitation: the near-deceptive order-3 trap, the recursive epistatic H-IFF and the *needle in the haystack* Trident problem.

A trap function is a piecewise-linear function defined on *unitation* (number of ones in a string), with two distinct regions in the search space, one leading to the global optimum and the other leading to a local optimum. The trap in this test is defined by:

$$F(\vec{x}) = \begin{cases} k, & if\ u(\vec{x}) = k \\ k - 1 - u(\vec{x}), & otherwise \end{cases} \quad (1)$$

where $u(\vec{x})$ is the unitation function and k is the problem size (and also the fitness of the global optimum). With these definitions, order-3 traps are in the region between deceptive and non-deceptive, while order-2 are non-deceptive and order-4 are fully deceptive. For the experiments, an order-3 trap function was constructed by juxtaposing 100 subproblems, which corresponds to 300-bit string. The fitness of the best solution (a string of $1'$s) is 300.

Trident functions are *needle in the haystack* problems that exploit the ability of EAs to mix good but significantly different solutions. The fitness function of the Trident used in this work has two components, *base* and *contribution:* $F(\vec{x}) = \text{base}(\vec{x}) + \text{contribution}(\vec{x})$. The base depends on unitation and is described by:

$$base(\vec{x}) = \|2.\,\text{u}(\vec{x}) - \text{l}\| \quad (2)$$

where l is the chromosome length and $u(\vec{x})$ is the unitation function. The contribution rewards certain configurations of strings that have an equal number of 0's and 1's. Let L be the first half of the binary string x of length l and R the second half. The *contribution* is described by Equation 5:

$$contribution(\vec{x}) = \begin{cases} 2.l, & L = \bar{R} \\ 0, & otherwise \end{cases} \quad (3)$$

where $\bar{R}$ is the bitwise negation of R. The Trident accepts strings of length $2k$, where $k \geq 2$. For this paper, 128-bit strings were used and the optimal fitness is 256.

Finally, the H-IFF function is a recursive epistatic problem with hierachical structure. The landscape requires a search for increasingly higher-order schemata, challenging the EAs' abilities to identify and combine good building blocks. The problem is defined using a recursive function. If the bit string being considered consists of all zeros or all ones, the fitness of the string is equal to its length; otherwise it has a fitness of 0. This same criterion is then applied recursively on each half of the string, until it can be subdivided no further. Adding the fitness of all substrings together yields the fitness of the whole. Formally, the HIFF fitness function can be defined as:

$$f(B) = \begin{cases} 1, & if\ |B| = 1 \\ |B| + f(B_L) + f(B_R), & if\ |B| > 1\ \ and\ (\forall_i\{b_i = 0\}\ or\ \forall_i\{b_i = 1\}) \\ f(B_L) + f(B_R), & otherwise \end{cases} \quad (4)$$

where B is a block of bits $(b_1 \ldots b_n)$, $|B|$ is the size of the block (and therefore equal to n, which must be an integer power of 2), b_i is the ith element of B, B_L and B_R are

the left and right halves of B. For the tests, a problem 128-bit strings has been constructed. The best solution has a fitness value of 1024.

With this set of functions it is possible to test the ability of the dynamic cGAs in combining the raw building blocks of the initial population and escape local optima traps. These functions challenge standard strategies, which converge very often to local optima, especially in the H-IFF and trap functions. If the proposed dynamic topology is effective in maintaining genetic diversity, then it is expected that the rate of convergence to global optima is improved.

All the cEAs used in the experiments are synchronous (i.e., the offspring are placed in a temporal population and replacement is done after every individual generates one child). Parameterization was done after [1]: the population size was set to $n = 400$; the recombination operator is the double point crossover with $p_c = 1.0$; mutation is bit-flip with $p_m = 1/l$, where l is the chromosome length; tournament selection. Only one offspring is placed in the temporary population (randomly chosen from the set of two children). In the replacement stage, the offspring replaces its parent if it is better.

The stop criteria are: to find the global optimum or to achieve a maximum of 3,000,000 function evaluations. The number of evaluations required to meet the best solution is recorded and averaged over 50 runs. A success measure (successful runs) is defined as the number of runs in which the algorithm attains the global optimum.

Four different strategies have been considered for the position update phase of the proposed algorithm. In the first one, which will be referred to as *dynamic cGA with Brownian movement* ($dcGA_b$) the individuals ignore the marks and chose randomly the destination cells amongst the empty ones in their neighborhood. In the *dynamic cGA with fitness marks* ($dcGA_f$), the individuals deposit marks with their fitness value. A similar strategy is used by the *hierarchical dynamic cGA with fitness marks* ($dcGA_{fh}$), except that in this case the individual only considers a mark if the fitness value is better than its own fitness. Finally, in the *dynamic cGA with genotype marks* ($dcGA_g$), the individuals leave copies of their genotypes in the cells, and when choosing the destination cell, the individual computes the Hamming distance between its genotype and the marks. The destination cell is then the one that minimizes the distance. The radius r of the Moore neighborhood and marks lifespan K were set to 1.

At every time-step, the individuals are ranked according to their fitness, so that the best individuals' positions are updated first. This strategy has been devised for the $DcGA_{fh}$, but in order to make fair comparisons it has been implemented in every cGA. In fact, some preliminary tests showed that ranking the individuals tends to improve the performance of the algorithms.

In order to evaluate the efficiency of the algorithms, the dynamic cGAs are compared with static cGAs with Moore (cGA_M) and von Neumann (cGA_{vN}) neighbourhoods on a 20×20 grid. The evolutionary phase begins only when the average clustering degree k (the number of neighbours of an individual, including the individual itself) rises above 2.5. This *ad hoc* strategy is used for avoiding the initial distribution stage in which many individuals are still isolated (i.e., with none or only a few neighbours). Typically, the individuals start to cluster in a few generations and the evolutionary phase begins at a very early stage. Although the threshold is imposed here by a centralized decision, a local decentralized (self-organized) strategy is also possible. For instance, the evolutionary phase could be triggered individually,

Table 1. Average best fitness values (plus standard deviation)

	X×Y	H-IFF	Trident	3-trap
cGA_M	20×20	877.17±90.95	243.20±38.79	296.58±2.07
cGA_{vN}	20×20	915.84±91.86	243.20±38.79	297.70±1.71
$dcGA_b$	30×30	856.67±100.59	184.32±64.18	295.18±2.30
	40×40	905.33±99.92	235.52±47.40	296.58±1.95
	50×50	902.33±92.51	235.52±47.40	297.78±1.43
	60×60	926.67±102.08	232.96±49.68	297.76±1.70
	70×70	928.96±87.00	219.43±58.42	297.86±1.54
$dcGA_f$	30×30	871.68±109.82	194.56±64.60	294.42±2.82
	40×40	870.08±94.51	230.40±51.72	294.70±3.26
	50×50	917.76±94.67	250.88±25.34	296.54±2.19
	60×60	944.96±90.19	**256.00±0.00**	297.58±1.48
	70×70	954.88±85.37	**256.00±0.00**	298.16±1.45
$dcGA_{fh}$	30×30	862.400±97.36	192.00±64.65	294.28±2.38
	40×40	884.80±112.17	235.52±47.40	294.76±2.60
	50×50	921.60±94.80	245.76±35.08	295.90±3.09
	60×60	940.80±93.24	253.44±18.10	296.56±2.55
	70×70	965.12±79.27	248.32±30.71	297.42±1.71
$dcGA_g$	30×30	875.84±92.10	222.72±56.72	295.88±2.16
	40×40	929.60±98.38	253.44±18.10	297.14±1.83
	50×50	924.48±92.86	250.88±25.34	297.86±1.52
	60×60	947.84±84.27	**256.00±0.00**	298.68±1.12
	70×70	**979.84±77.28**	**256.00±0.00**	**298.74±1.28**

for each chromosome. However, such strategy introduces a transitory phase in which the population only recombines partially (steady-state). This could make a comparison with static strategies more difficult and potentially unfair and therefore it has been left for future work.

The objectives of the first experiment are to study the performance of the dynamic cGAs and the effects of the grid size on their behaviour. For that purpose, grids with different size have been tested, starting with a 30×30 grid. The averaged final fitness value attained by each algorithm in each function is shown in Table 1.

The first conclusion is that the Brownian version, in general, does not improve significantly the performance of the cGA_{vN} (the best static strategy). A dynamic topology *per se* is not sufficient to overcome the drawbacks of standard cEAs. Some kind of organization must take place in order to generate a better interaction between the individuals. When stigmergy is introduced in the model, the results are clearly improved, as seen in Table 1.

The dynamic topologies with stigmergic-based movement rules increase standard cGAs performance when the grid is larger than 40×40. In general, dynamic populations with stigmergic rule moving on 60×60 and 70×70 grids significantly improve the standard cGAs. For instance, the $dcGA_g$ on a 70×70 grid is significantly better than the standard cGAs in every function (according to Kolmogorov-Smirnov statistical tests with a 0.05 level of significance). The fact that smaller grids do not necessarily improve the static cGA performance suggests that it is not the movement of the individuals that makes the algorithm better in this set of functions, but instead some kind of global island-like pattern that emerges when the grid is larger.

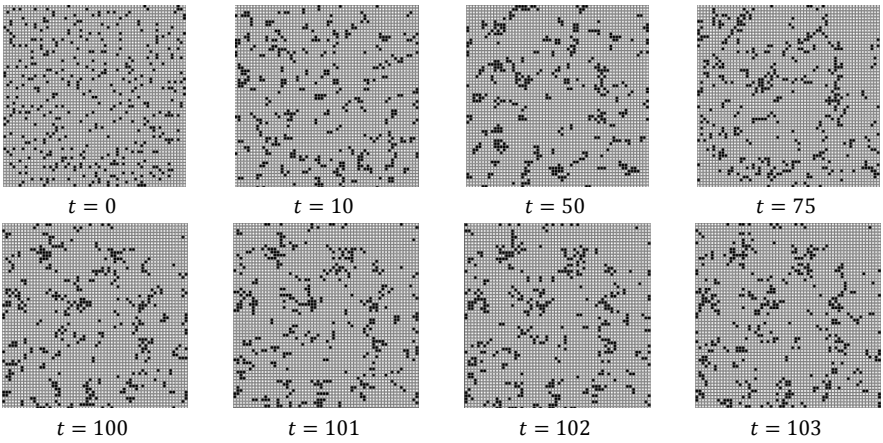

Fig. 1. Distribution of the individuals on the grid at different iterations of the search process. $dcGA_f$ and Trident function. 60×60 grid.

Figure 1 shows the distribution of the individuals at different iterations (t) of the search process for a grid with size 60×60. The evolutionary phase begins at $t = 50$. Clusters of individuals emerge already at an early stage. Those clusters are highly dynamic and in a few generations the global pattern radically changes (please note the distributions between iteration $t = 100$ and $t = 103$). The topology self-organizes into a kind of dynamic island model, in which the communication between the clusters is also an emergent property, arising from the global behavior of the system. After $t = 50$, when the evolutionary phase is introduced (and therefore several fitness values and genotypes are changing in each time-step), the clusters are sparser, but this an expected outcome due to the variation introduced by the evolutionary process.

Table 2 shows the number of successful runs attained by the cGAs. Again, under this criterion, the dynamic versions outperform the static topologies. The similarity-based strategy is particularly efficient, attaining the best success rates.

The previous results show that the dynamic cGAs are able to converge more often to global optimum. Therefore, they have a better balance between exploration and exploitation for these fitness functions: with the same raw building-blocks, the dynamic cGAs combine more efficiently the solutions. This is probably because the emergent structures, with their clustering degree and dynamical behaviour, are more efficient at maintaining genetic diversity. In order to investigate this hypothesis,

Table 2. Number of successful runs

	H-IFF	Trident	3-trap
cGA_M	11	45	2
cGA_{vN}	19	45	5
$dcGA_b(70 \times 70)$	21	36	8
$dcGA_f(70 \times 70)$	29	50	9
$dcGA_{fh}(70 \times 70)$	31	48	5
$dcGA_g(70 \times 70)$	37	50	13

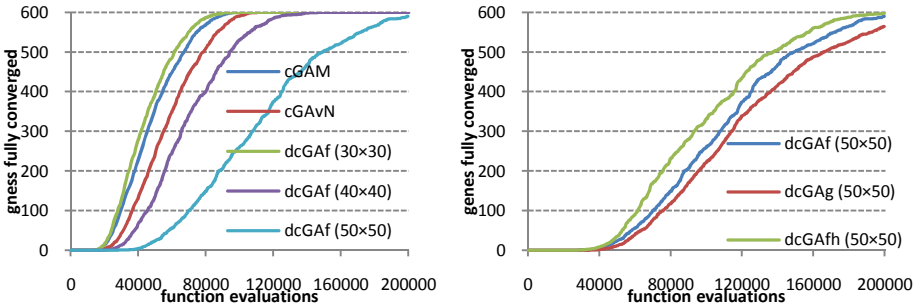

Fig. 2. Genetic diversity

the algorithms were tested without mutation and the number of genes that converge (i.e., genes with alleles 0 or 1 in the entire population) during the run was computed and plotted. The results are in Figure 2. The diversity is in fact maintained at a higher level by the structures. Furthermore, increasing the grid increases the diversity (left-hand graph in Figure 2). As for the different strategies, the best strategy (similarity-based) is also the one that maintains diversity at a higher level (right-hand graph).

Finally, since the dynamic topologies maintain genetic diversity at a higher level, therefore increasing exploration and reducing the risk of convergence to local optima, it is expected that the convergence speed is reduced, a typical payoff for increasing robustness. Table 3 shows the averaged number of evaluations required by each algorithm to reach the global optimum (only runs in which the global optimum has been found are considered). The static structures are faster, but as seen in Table 1 and Table 2, at the expenses of a significant drop of the performance levels.

5 Conclusions and Future Work

This paper investigates a dynamic cellular Genetic Algorithms (cGA) in which the individuals communicate via a grid of nodes and self-organized its structure on that grid. The global behavioral patterns emerge from local interactions defined by simple rules. When compared to static topologies, the dynamic structure maintains genetic diversity at a higher level, resulting in an improvement of the convergence rates to global optimum on a set of functions that defy the GAs abilities to combine building-blocks. Such behavior is attained when the ration between the number of nodes in the grid and the population size is above 4:1. With these settings, the distribution of

Table 3. Convergence speed (function evaluations)

	H-IFF	Trident	3-trap
cGA_M	39600.00±9248.29	44035.56±5678.77	93000.00±3400.00
cGA_{vN}	48191.30±15320.04	47377.78±5265.71	100560.00±13716.21
$dcGA_b$	49808.57±9093.29	55168.47±13663.58	108822.75±4795.37
$dcGA_f$	114608.86±36325.71	75936.81±12991.56	266100.11±48188.63
$dcGA_{fh}$	104568.00±27233.83	69546.29±9204.14	212348.20±20325.13
$dcGA_g$	127981.38±41261.71	95204.48±15756.53	298350.47±20739.19

individuals in the grid emerges into a global island-like model, highly dynamic and with frequent communication between the clusters.

Future work will be focused on the traits of the system and their effects on the behavior of the population and on the performance of the algorithm. Radius r of the neighborhood and marks' lifespan K will be investigated. Different stigmergic strategies will be tested, namely those that favor recombination between dissimilar individuals. Finally, the experiments will be extended to other type of functions (unimodal and multimodal) in order to achieve a better comprehension of the structure's working mechanism and potential as an alternative cGA network.

Acknowledgements. The first author wishes to thank FCT, *Ministério da Ciência e Tecnologia*, his Research Fellowship SFRH/BPD/66876/2009. The work was supported by FCT PROJECT [PEst-OE/EEI/LA0009/2013], Spanish Ministry of Science and Innovation projects TIN2011-28627-C04-02 and TIN2011-28627-C04-01, Andalusian Regional Government P08-TIC-03903 and P10-TIC-6083, CEI-BioTIC UGR project CEI2013-P-14, and UL-EvoPerf project.

References

1. Alba, E., Dorronsoro, B.: The exploration/exploitation tradeoff in dynamic cellular genetic algorithms. IEEE Trans. Evol. Computation 9, 126–142 (2005)
2. Cantú-Paz, E.: Migration Policies, Selection Pressure, and Parallel EAs. Journal of Heuristics, 311–334 (2001)
3. Fernandes, C.M., Laredo, J.L.L., Merelo, J.J., Cotta, C., Rosa, A.C.: Towards a 2-dimensional Framework for Structured Population-based Metaheuristics. In: Proc. of IEEE International Conference on Complex Systems, pp. 1–6 (2012)
4. Fernandes, C.M., Laredo, J.L.L., Merelo, J.J., Cotta, C., Rosa, A.C.: A Study on Time-Varying Partially Connected Topologies for the Particle Swarm. In: Proc. of the IEEE Congress on Evolutionary Computation, pp. 2450–2456. IEEE (2013)
5. Grassé, La, P.-P.: reconstrucion du nid et les coordinations interindividuelles chez bellicositermes et cubitermes sp. La théorie de la stigmergie: Essai d'interpretation du comportement des termites constructeurs. Insectes Sociaux 6, 41–80 (1959)
6. Ilachinski, A., Cellular Automata, A.: Discrete Universe, World Scientific (2001)
7. Laredo, J.L.J., Castillo, P.A., Mora, A.M., Merelo, J.J., Fernandes, C.M.: Resilience to churn of a peer-to-peer evolutionary algorithm. International Journal of High Performance Systems Architecture 1(4), 260–268 (2008)
8. Nogueras, R., Cotta, C., Fernandes, C.M., Jiménez Laredo, J.L., Merelo, J.J., Rosa, A.C.: An analysis of a selecto-lamarckian model of multimemetic algorithms with dynamic self-organized topology. In: Dediu, A.-H., Martín-Vide, C., Truthe, B., Vega-Rodríguez, M.A. (eds.) TPNC 2013. LNCS, vol. 8273, pp. 205–216. Springer, Heidelberg (2013)
9. Steinmetz, R., Wehrle, K. (eds.): Peer-to-Peer Systems and Applications. LNCS, vol. 3485. Springer, Heidelberg (2005)
10. Tomassini, M.: Spatially Structured Evolutionary Algorithms. Springer (2005)
11. Whitacre, J.M., Sarker, R.A., Pham, Q.: The self-organization of interaction networks for nature-inspired optimization. IEEE Transactions on Evolutionary Computation 12, 220–230 (2008)