

# Spatially Structured Evolutionary Algorithms: Graph Degree, Population Size and Convergence Speed

Carlos M. Fernandes, Juan L.J. Laredo and Agostinho C. Rosa

**Abstract** An evolutionary algorithm (EA) is said to be spatially structured when its individuals are arranged in an incomplete graph and interact only with their neighbors. Previous studies argue that spatially structured EAs are less likely to converge prematurely to local optima. Furthermore, they have been initially designed for distributed computing and it is often claimed that their parallelization is simpler than the equivalent non-structured algorithm. However, most of the empirical studies on spatially structured EAs use a predefined and fixed population size, whereas the full potential of this or any other any kind of EA can only be explored if the population size is properly set. This paper investigates optimal population sizes of spatially structured EAs (cellular EAs, in particular) and the relationship between that size, convergence speed and the degree of the structuring network. EAs structured by regular graphs with different degrees have been tested on different types of fitness landscapes. We conclude that in most cases graphs with low degree require smaller populations to converge consistently to global optima. However, if the population size is properly set, EAs structured by graphs with higher degrees not only converge to global optima with high probability, but also converge faster.

**Keywords** Evolutionary computation • Spatially structured genetic algorithms • Optimal population size • Distributed EAs

---

C.M. Fernandes (✉) • A.C. Rosa  
LARSyS: Laboratory for Robotics and Systems in Engineering and Science,  
University of Lisbon, Lisbon, Portugal  
e-mail: cfernandes@laseeb.org

A.C. Rosa  
e-mail: acrosa@laseeb.org

C.M. Fernandes  
Department of Computer Architecture, University of Granada, Granada, Spain

J.L.J. Laredo  
LITIS, University of Le Havre, Le Havre, France  
e-mail: juanlu.jimenez@gmail.com

# 1 Introduction

Evolutionary Algorithms (EAs) [2] are a class of metaheuristics based on the theory of evolution. Initially, an EA generates a population of solutions. Then, a set of those solutions is selected according to their fitness and recombined for generating new individuals. The new population replaces the whole or part of the parents' population and the process repeats until a stopping criterion is met. This simple procedure increases the average fitness of the population and, eventually, finds a local or global solution to the problem.

Standard EAs use what is known as panmictic populations: each individual can interact (recombine) with every other individual. However, parallel and distributed implementations of EAs may benefit from alternative, restricted forms of recombination. In recent years, spatially structured EAs [13], which restrain the interaction according to a population structure, are gaining increasing attention. The structure specifies a network of acquaintances for individuals to interact, that is, mating or selection is restricted to neighborhoods within the network structure. As argued in [2], non-panmictic EAs, such as cellular [1] or distributed EAs [3, 8], provide a better sampling of the search space and improve the performance of the equivalent panmictic EA.

This paper focuses on the particular case of spatially structured EAs called cellular EAs (cEAs). The efficiency of cEAs has been systematically demonstrated [1, 2, 13] and is attributed to their ability to maintain fitness and genetic diversity [2]. Since individuals only interact with a restricted number of other individuals, information diffuses slower through the network. This means that the balance between exploration and exploitation of panmictic EAs (under the same selection and recombination strategies) is severely altered: exploration is more intense, while exploitation takes place only in local neighborhoods. This results in higher takeover times: the diffusion of good individuals is slower. Consequently, the convergence is also slower, but the algorithm is less likely to converge to local optima.

There are several studies that investigate selection pressure, convergence speed and takeover times of cellular EAs [1, 2, 4, 6, 7]. However, to the extent of our knowledge, the relationship between population size, convergence speed, accuracy and the degree of the underlying graph has not been studied yet. Since population size is a key factor not only in the convergence speed of EAs, but also for efficient parallel implementations, we propose to investigate the optimal population size of structured EAs on regular graphs with different degree. For that purpose, we use the bisection method for assessing optimal population size in different fitness landscapes. Under these settings, we are able to determine which graph maximizes the performance of the algorithm in each type of landscape, as well as the smallest population that guarantees a high probability of convergence to the global optimum. With such knowledge, we can improve our comprehension of the mechanisms behind efficient cEAs, while optimizing the computational resources required for real-world implementations of cEAs.

The remainder of the paper is structured as follows: Sect. 2 gives a background review on cellular EAs; Sect. 3 describes the methodology used in this study; Sect. 4 presents and discusses the results; Sect. 5 concludes the paper and outlines future lines of work.

## 2 Background Review and Motivation

Genotypic representation, operators, selection schemes and population size are typical EAs moduli that require design choices. Population size, in particular, must be set to a minimal size that guarantees a sufficient supply of raw building blocks. If the population is too small, the algorithm loses diversity prematurely and converges to local optima. Conversely, if the population is excessively large, the convergence speed of the algorithm may be affected. The population must grasp a proper balance between genetic diversity and convergence speed, and methods have been devised for determining the minimal size that assures a high probability of convergence to global optima [12]. These methods can be applied to any kind of EA, including cEAs.

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 motivated several lines of research that explore the potentiality of different population structures in maintaining genetic diversity [13]. The primary focus of the field has been on static regular lattices: every individual has a fixed number of potential interaction partners. Giacobini et al. [7] present mathematical models for the selection pressure of cEAs on regular lattices. The experiments confirmed the theoretical results. The validation of the model has been made on  $32 \times 32$  grids (1024 individuals), but the authors identified a breakdown of the usual logistic approximation for low-dimensional lattices.

Alba and Dorronsoro [2] dynamically change the ratio that defines the neighborhood of interaction in cEAs. Since the ratio may affect selection pressure, the authors analyze its influence on the balance between exploration and exploitation. However, the base-structure of the cEA (i.e. a grid lattice) is maintained throughout the run and the population size is set to fixed value for all problems and configurations of the algorithm.

Standard cEAs have some drawbacks: synchronicity (in most cases) and a strong dependence on the problem since the genetic diversity promoted by a prefixed topology is uncorrelated with the problem structure. In order to overcome these limitations, complex population structures have been also studied, sometimes using recent developments in network theory [10]. Giacobini et al. [6] studied takeover times in random and small-world structures. Again, the population size is set to a fixed value in every experiment. Whitacre et al. [14] focus on two important conditions missing in EA populations: a self-organized definition of locality and interaction epistasis. With that purpose in mind, they propose a dynamic structure and conclude that these two features, when combined, provide behaviors not

observed in the canonical EAs or traditional spatially structured EAs. The most noticeable change in the behavior is an unprecedented capacity for sustainable coexistence of genetically distinct individuals within a single population. The population size varies on the range [50, 400], but the authors not give a reason for choosing this interval. Fernandes et al. [5] proposed dynamic and partially connected ring topologies for cEAs. The structures improve the rate of convergence to global optima when compared to cEAs with standard topologies on quasi-deceptive, deceptive and NP-hard problems. In this case, the authors conducted optimal population size tests, demonstrating that the proposed topologies require smaller populations when compared to traditional cEAs.

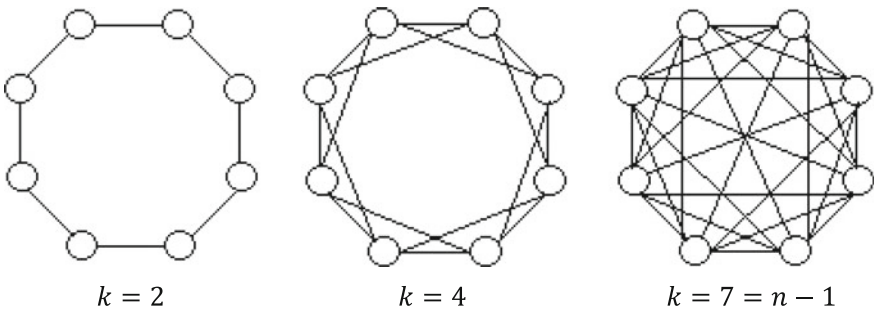
Our purpose is to investigate how population size of cEAs correlates with the structure and the fitness landscape. Since takeover times decrease with graph degree, it is expected that structures with higher degrees require larger populations. However, since good solutions diffuse more quickly when the individuals have more neighbors, it is possible that larger populations required by higher degree graphs converge faster than smaller populations in less connected structures.

### 3 Methodology

In order to investigate the optimal population size of different types of graphs, we have implemented cEAs with increasing degree. Most of the studies on spatially structured EAs on regular graphs use 1-D or 2-D grids—see [1, 2, 9]. In fact, a grid topology does not restrict the study [11]. However, we have chosen a more general basic structure, exemplified in Fig. 1.

Starting from a ring structure ( $k = 2$ ) the degree is doubled by linking each individual to its neighbors' neighbors, creating regular graphs with  $k = \{2, 4, 8, 16, 32 \dots\}$ . Additionally, EAs with  $k = n - 1$  (i.e., with panmictic populations), where  $n$  is the population size, have been tested.

This study is restricted to synchronous cEAs, i.e., offspring are placed in the secondary population and replacement is made when the size  $n'$  of the offspring



**Fig. 1** Regular graphs for population size  $n = 8$

population is equal to the size  $n$  of the parents population. The selection scheme is the binary tournament, two-point crossover is the recombination method and bit-flip is the mutation type. In each iteration, each individual (parent1) is recombined with one of its

---

**Algorithm 1: cellular EA**


---

1. For each individual  $i \leftarrow 1$  to  $n$ :
    - 1.1. Initialize individual  $i$
    - 1.2. Evaluate individual  $i$ :  $f(\vec{x}_i)$
  2. For each individual  $i \leftarrow 1$  to  $n$ :
    - 2.1. Compute neighborhood
    - 2.2. Parent 1 is individual  $i$
    - 2.3. Parent 2 selected with binary tournament from the set of parent1 neighbours
    - 2.4. Crossover (parent 1, parent 2)
    - 2.5. Select randomly one of the offspring: offspring  $i$
    - 2.6. Mutation (offspring  $i$ )
    - 2.7. Evaluate offspring  $i$ :  $f(\vec{x}'_i)$
    - 2.8. Insert offspring  $i$  in temporary population  $P_t$
  3. For each individual  $i \leftarrow 1$  to  $n$ :
    - 3.1. Replace individual  $i$  by offspring  $i$  if  $f(\vec{x}'_i) > f(\vec{x}_i)$  (maximization problems)
  4. If the stopping criterion is not met, go to 2
- 

---

**Algorithm 2: Bisection method**


---

1. Start with small  $n$
  2. Double  $n$  until EA convergence criteria is met
  3.  $(\min, \max) = (n/2, n)$
  4. repeat until  $(\max - \min) / \min < T$ 
    - $n = (\min + \max) / 2$
    - if  $n$  leads to convergence criteria
    - then  $\max = n$  else  $\min = n$
  5. Compute the statistics for the problem size using population size =  $\max$
- 

neighbors (parent2). From the set of two children generated by crossover, one is randomly chosen and replaces parent1 if its fitness is higher. The pseudo-code of the cEA is in Algorithm 1.

Finding an appropriate population size for a given problem is critical for the performance of any EA. To determine the optimal population size of the cEAs we have used a selectorecombinative version of the algorithms (i.e., without mutation) and the bisection method [12]. Please note the bi-section method is performed in EAs without mutation. The objective is to determine the minimal population size that guarantees a sufficient supply of building blocks for the search process to converge to the global optimum without needing mutation. Then, it is expected that smaller populations can be used effectively when mutation probability is set to a non-zero value.

The bisection method is a simple yet effective technique used to determine the optimal population size of selectorecombinative EAs and it is described by Algorithm 2. For this particular case the threshold  $T$  was set to 0.1 and initial population size was set to 200. Every configuration was run 30 times before updating and the convergence criteria is met if 29 of those 30 runs converge towards the global optimum. The algorithms were tested with  $p_c = 1.0$ . Mutation probability was set to 0. After determining the optimal population size, that configuration was executed for 50 times and the number of evaluations necessary to reach the optimum was averaged over the successful runs.

## 4 Experimental Setup and Results

The algorithms were tested with onemax, 2-trap, 3-trap, 4-trap and MMDP. A trap function is a piecewise-linear function defined on *unitation* (the number of ones in a binary string) that has two distinct regions in the search space, one leading to the global optimum and the other leading to a local optimum. Depending on its parameters, trap functions may be deceptive or not. The trap functions in these experiments are defined by:

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

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, order-2, -3 and -4 trap functions were constructed by juxtaposing, respectively, 150, 75 and 60 subproblems, generating 300- (2-trap), 150- (3-trap) and 120-bit (4-trap) problems. The fitness values of the best solutions are, respectively, 300, 150 and 120.

The onemax problem is the 1-trap function and consists in maximizing the number of ones in a binary string. The size of the string in the onemax problem used for this study is  $l = 400$ , corresponding to an optimal fitness of 400.

The MMDP is an NP-hard, deceptive and multimodal. It consists of  $k$  6-bits subproblems with two global optima and a deceptive attractor in the middle of the fitness landscape. Each subproblem fitness values depend on the unitation function. Table 1 shows the contribution of each subproblem to the fitness value of a string. For the experiments, 120-bit strings were used. Optimal solutions have fitness values of 20. Table 2 summarizes the test set.

**Table 1** MMDP. Contribution of each subproblem configuration to the fitness value

$u(\vec{x})$	0	1	2	3	4	5	6
$F(\vec{x})$	1.000000	0.000000	0.360384	0.640576	0.360384	0.000000	1.000000

**Table 2** Functions: type, string size and best solution fitness

Function	Type	String size	Best fitness
onemax	Non-deceptive	400	400
2-trap	Non-deceptive	300	300
3-trap	Nearly deceptive	150	150
4-trap	Deceptive	120	120
MMDP	Deceptive	120	20

**Table 3** Optimal population size

	onemax	2-trap	3-trap	4-trap	MMDP
$k = 2$	400	500	350	400	500
$k = 4$	450	500	350	450	500
$k = 8$	500	550	350	500	500
$k = 16$	550	700	400	550	500
$k = 32$	800	750	500	550	550
$k = 64$	1000	1000	650	650	650
$k = 128$	1200	1300	700	750	750
$k = n - 1$	2200	2275	1100	1200	800

First experiments determine the optimal population size of cEAs with  $k = \{2, 4, 8, 16, 32, 64, 128, n - 1\}$ , where  $n$  is population size. Results are in Table 3. As expected, optimal size increases with the degree of the underlying structures.

Table 4 shows the average number of evaluations required to reach the global optimum when the population size is set to the values found previously and shown

**Table 4** Convergence speed: average number of evaluations and standard deviation

	onemax	2-trap	3-trap	4-trap	MMDP
$k = 2$	319,986.21	348,483.33	133,712.07	146,560.00	168,266.67
	$\pm 17,253.56$	$\pm 22,773.05$	$\pm 16,085.69$	$\pm 18,423.25$	$\pm 29,590.81$
$k = 4$	219,930.00	222,433.33	86,205.00	104,167.24	108,866.67
	$\pm 12,784.26$	$\pm 26,610.77$	$\pm 10,812.42$	$\pm 12,786.66$	$\pm 14,151.84$
$k = 8$	157,233.33	147,836.21	54,961.67	73,206.90	66,100.00
	$\pm 7,747.89$	$\pm 1,1261.46$	$\pm 5,178.70$	$\pm 10,743.37$	$\pm 8,515.10$
$k = 16$	<b>114,210.34</b>	129,173.33	41,701.67	51,645.00	44,724.14
	<b><math>\pm 5,271.88</math></b>	$\pm 7,495.93$	$\pm 4,602.52$	$\pm 4,915.85$	$\pm 5,630.89$
$k = 32$	122,560.00	<b>99,795.00</b>	38,683.33	40,425.00	35,806.90
	$\pm 5,366.15$	<b><math>\pm 5,847.96</math></b>	$\pm 3,100.29$	$\pm 4,175.58$	$\pm 2,988.51$
$k = 64$	117,933.33	104,068.97	40,913.33	<b>37,812.07</b>	33,979.31
	$\pm 4,448.42$	$\pm 4,008.30$	$\pm 2,966.80$	<b><math>\pm 2,395.00</math></b>	$\pm 3,254.16$
$k = 128$	120,331.03	110,196.67	<b>37,727.59</b>	38,700.00	33,725.00
	$\pm 3,887.07$	$\pm 4,601.69$	<b><math>\pm 1,947.32</math></b>	$\pm 3,275.69$	$\pm 2,904.63$
$k = n - 1$	181,462.07	160,463.33	51,920.00	49,646.67	<b>33,296.55</b>
	$\pm 5,010.59$	$\pm 3,995.89$	$\pm 3,792.95$	$\pm 3,989.5$	<b><math>\pm 2,142.51</math></b>

in Table 3. The optimal convergence speed (in bold) is attained with highly connected graphs. The panmictic population ( $k = n - 1$ ) does not attain the best convergence speed values in every function but it is significantly better than lower degree graphs ( $k = 2$  and  $k = 4$ ) in every fitness landscapes. (In this study, Mann-Whitney U tests were performed at the 5% level of significance to determine if two distributions of numerical results are significantly different). Furthermore,  $k = n - 1$  attains the best performance in the MMDP problem.

The following experiment was performed to stress out the importance of determining the optimal population size of EA for a particular fitness landscape.

**Table 5** Fitness (median, best and worst values) and success rates (SR)

		onemax $n = 200$	2-trap $n = 250$	3-trap $n = 175$	4-trap $n = 200$	MMDP $n = 250$
$k = 2$	Median	400	300	150	120	20
	Best	400	300	150	120	20
	Worst	400	300	149	118	19.64
	SR	30	30	28	24	23
$k = 4$	Median	400	300	150	120	20
	Best	400	300	150	120	20
	Worst	400	300	148	118	19.64
	SR	30	30	25	23	22
$k = 8$	Median	400	300	150	119	20
	Best	400	300	150	120	20
	Worst	400	300	148	118	19.64
	SR	30	30	20	12	23
$k = 16$	Median	400	300	149	119	20
	Best	400	300	150	120	20
	Worst	400	300	146	116	19.64
	SR	30	30	7	7	22
$k = 32$	Median	400	300	147	117.5	20
	Best	400	300	150	120	20
	Worst	400	300	144	115	19.28
	SR	30	30	1	1	20
$k = 64$	Median	400	300	146	116	19.64
	Best	400	300	149	120	20
	Worst	400	300	142	113	19.28
	SR	30	30	0	1	6
$k = 128$	Median	400	300	145	116	19.64
	Best	400	300	148	120	20
	Worst	400	300	140	112	18.20
	SR	30	30	0	1	16
$k = n - 1$	Median	400	300	146	116	19.64
	Best	400	300	149	119	20
	Worst	400	300	142	113	18.92
	SR	30	30	0	0	9

**Table 6** Convergence speed: average number of evaluations and standard deviation

	onemax	2-trap	3-trap	4-trap	MMDP
$k = 2$	95,900.00	235,716.67	99,497.22	110,708.33	124,673.91
	$\pm 3,626.24$	$\pm 23,382.20$	$\pm 15,799.90$	$\pm 21,653.64$	$\pm 58,068.80$
$k = 4$	72,460.00	148,950.00	65,856.00	67,452.17	151,840.91
	$\pm 2,769.36$	$\pm 16,527.33$	$\pm 12,553.7$	$\pm 10,967.43$	$\pm 200,212.74$
$k = 8$	55,493.33	117,900.00	58,403.95	43,233.33	114,608.70
	$\pm 2,183.79$	$\pm 49,055.90$	$\pm 89,744.99$	$\pm 5,718.13$	$\pm 166,207.75$
$k = 16$	43,646.67	217,041.67	25,300.00	26,914.29	116,761.36
	$\pm 2,058.24$	$\pm 371,359.30$	$\pm 2,710.82$	$\pm 4,989.13$	$\pm 230,706.28$
$k = 32$	35,793.33	423,041.67	17,675.00	18,600.00	162,950.00
	$\pm 1,960.64$	$\pm 203,743.01$	–	–	$\pm 230,706.28$
$k = 64$	32,866.67	615,841.67	–	15,000.00	86,708.33
	$\pm 1,590.78$	$\pm 228,150.17$	–	–	$\pm 119,648.17$
$k = 128$	32,333.33	608,983.33	–	13800.00	287,468.75
	$\pm 2,074.03$	$\pm 239,684.09$	–	–	$\pm 304,905.86$
$k = n - 1$	31,600.00	611,441.67	–	–	127,277.78
	$\pm 1,702.74$	$\pm 232,371.70$	–	–	$\pm 153,606.36$

For each problem, the population size  $n$  of each cEA was set to  $n_{min}/2$ , where  $n_{min}$  is the population size in Table 3. Mutation probability is set to  $p_m = 1/l$ , where  $l$  is the string size, and crossover probability is  $p_c = 1.0$ . The algorithms were all run for 1,000,000 function evaluations or until reaching the global optimum. Results are averaged over 30 runs and shown in Tables 5 and 6.

Table 5 shows the median, best and worst fitness attained by each configuration in each problem, as well as the success rates (the number of runs in which the algorithm found the global optimum). The global optimum of onemax and 2-trap functions is found in every run by every cEA. These are simple and unimodal problems without local optima. Therefore, provided with variation and mutation operators and given enough time, an EA will eventually find the optimum. Comparison of performance can therefore be made using convergence speed. Table 5 shows that the convergence speed in the onemax problem increases with  $k$ . Optimal speed in 2-trap is attained with  $k = 8$ . However, better convergence speed is attained for 2-trap using larger populations—see cEAs with  $k = 32$  and  $k = 64$  in Table 4.

For the 3-trap problems, better results are clearly attained by the cEAs with optimal population size, except for  $k = 2$ , which attains a good success rate with lower convergence speed. For 4-traps, the accuracy is clearly degraded when population size is set to  $n = 200$ . The same goes for MMDP problem.

These results show that deceptive functions require a careful tuning of the population size. Furthermore, the numerical results in Tables 3, 4, 5 and 6 demonstrate that setting the population size to suboptimal values may mislead the conclusions on the performance of cEAs, mainly in deceptive and multimodal

problems. These problems require a proper balance between the initial supply of building blocks, the selection pressure and the variation operator. The results in this paper call into question the efficiency of cEAs in deceptive and multimodal problems—please remember that the most efficient EA in the tested MMDP problem is the panmictic EA. However, further tests are required in order to confirm the hypothesis.

## 5 Conclusions

This paper investigates the relationship between population size, convergence speed and graph degree of cEAs with populations structured by regular graphs. In order to determine the minimal population size that guarantees convergence to global optimum with high probability, the bisection method has been applied to cEAs with different degree. The numerical results show that graphs with lower degree require smaller populations. However, the larger populations required by graphs with higher degree converge faster to global optima. These results suggest that when the population is properly set, higher degree or even panmictic populations are more efficient than cEAs in ring structures or low degree graphs. Furthermore, conclusions on the performance of the different structures are entirely different and almost certainly misleading if the population size is set to the same value for all configurations.

The study has been restricted to regular and static graphs. In the future, we intend to apply the same experimental procedure to random, small-world and dynamic structures. The takeover times of the different graphs with different population size will be investigated as well as the behavior and performance of the different structures with different string sizes (scalability tests). Finally, the implications of a proper setting of the population size in parallel and distributed cEAs will also be studied.

**Acknowledgements** First author wishes to thank FCT, *Ministério da Ciência e Tecnologia*, his Research Fellowship SFRH/BPD/111065/2015). This work was supported by FCT PROJECT [UID/EEA/50009/2013].

## References

1. Alba, E., Tomassini, M.: Parallelism and evolutionary algorithms. *IEEE Trans. Evol. Comput.* **6**(5), 443–462 (2002)
2. Alba, E., Dorronsoro, B.: The exploration/exploitation tradeoff in dynamic cellular genetic algorithms. *IEEE Trans. Evol. Comput.* **9**, 126–142 (2005)
3. Bäck, T.: *Evolutionary Algorithms in Theory and Practice*. Oxford University Press, Oxford (1996)

4. Cantú-Paz, E.: Migration policies, selection pressure, and parallel EAs. *Journal of Heuristics* **7** (4), 311–334 (2001)
5. Fernandes, C.M., Laredo, J.L.J., Merelo, J.J., Cotta, C., Rosa, A.C.: Dynamic and Partially Connected Ring Topologies for Evolutionary Algorithms with Structured Populations, *EvoApplications 2014: Applications of Evolutionary Computation*, pp. 665–677 (2014)
6. Giacobini, M., Tomassini, M., Tettamanzi, A.: Takeover time curves in random and small-world structured populations. In: *Proceedings of the 7th GECCO*, pp. 1333–1340 (2005)
7. Giacobini, M., Tomassini, M., Tettamanzi, A.G.B., Alba, E.: Selection intensity in cellular evolutionary algorithms for regular lattices. *IEEE Trans. Evol. Comput.* **9**, 489–505 (2005)
8. Laredo, J.L.J., Bouvry, P., González, D.L., Fernández de la Vega, F., Arenas, M.G., Merelo, J.J., Fernandes, C.M.: Designing robust volunteer-based evolutionary algorithms. *Genet. Program Evol. Mach.* **15**(3), 221–244 (2014)
9. Payne, J.L., Eppstein, M.J.: Emergent mating topologies in spatially structured genetic algorithms. In: *Proceedings of 8th GECCO*, pp. 207–214 (2006)
10. Réka, A., Barabási, A.-L.: Statistical mechanics of complex networks. *Rev. Mod. Phys.* **74**, 47–94 (2000)
11. Sarma J., De Jong, K.: An analysis of the effect of the neighborhood size and shape on local selection algorithms. In: *Proceedings of International Conference on Parallel Problem Solving from Nature IV*, LNCS 1141, pp. 236–244. Springer (1996)
12. Sastry, K.: Evaluation-relaxation schemes for genetic and evolutionary algorithms. M.Sc. thesis, University of Illinois, Urbana, IL, USA (2001)
13. Tomassini, M.: *Spatially Structured Evolutionary Algorithms*. Springer, Heidelberg (2005)
14. Whitacre, J.M., Sarker, R.A., Pham, Q.: The self-organization of interaction networks for nature-inspired optimization. *IEEE Trans. Evol. Comput.* **12**, 220–230 (2008)

Intelligent Distributed Computing XI

Ivanovic, M.; Bădică, C.; Dix, J.; Jovanović, Z.; Malgeri, M.;  
Savić, M. (Eds.)

2018, X, 334 p. 93 illus., 74 illus. in color., Hardcover

ISBN: 978-3-319-66378-4