Swarm and Evolutionary Computation 58 (2020) 100721

Contents lists available at ScienceDirect

Swarm and Evolutionary Computation

journal homepage: www.elsevier.com/locate/swevo

Population sizing of cellular evolutionary algorithms

Carlos M. Fernandes [a,*], Nuno Fachada [b], Juan L.J. Laredo [c], J.J. Merelo [d], Agostinho C. Rosa [a]

[a] LARSyS: Laboratory for Robotics and Systems in Engineering and Science, University of Lisbon, Portugal
[b] HEI-Lab - Digital Human-Environment and Interaction Lab, Lusófona University, Lisbon, Portugal
[c] LITIS University of Le Havre Le Havre, France
[d] Department of Computer Architecture, University of Granada, Spain

ARTICLE INFO

Keywords:
Spatially structured evolutionary algorithms
Cellular evolutionary algorithms
Optimal population size
Event takeover values

ABSTRACT

Cellular evolutionary algorithms (cEAs) are a particular type of EAs in which a communication structure is imposed to the population and mating restricted to topographically nearby individuals. In general, these algorithms have longer takeover times than panmictic EAs and previous investigations argue that they are more efficient in escaping local optima of multimodal and deceptive functions. However, most of those studies are not primarily concerned with population size, despite being one of the design decisions with a greater impact in the accuracy and convergence speed of population-based metaheuristics. In this paper, optimal population size for cEAs structured by regular and random graphs with different degree is estimated. Selecto-recombinative cEAs and standard cEAs with mutation and different types of crossover were tested on a class of functions with tunable degrees of difficulty. Results and statistical tests demonstrate the importance of setting an appropriate population size. Event Takeover Values (ETV) were also studied and previous assumptions on their distribution were not confirmed: although ETV distributions of panmictic EAs are heavy-tailed, log-log plots of complementary cumulative distribution functions display no linearity. Furthermore, statistical tests on ETVs generated by several instances of the problems conclude that power law models cannot be favored over log-normal. On the other hand, results confirm that cEAs impose deviations to distribution tails and that large ETVs are less probable when the population is structured by graphs with low connectivity degree. Finally, results suggest that for panmictic EAs the ETVs' upper bounds are approximately equal to the optimal population size.

1. Introduction

In standard Evolutionary Algorithms (EAs) [4], individuals are selected to the mating pool with a probability proportional to their fitness. Once in the pool, they can recombine with any other, regardless of genotypic characteristics of the individuals or topological properties of the population. In EAs terminology, a population in which the individuals are not structured by any network of acquaintances and are free to mate with any other individual is called a panmictic population. Accordingly, the algorithm is termed *panmictic EA*.

Spatially structured EAs [21] are of a different kind: interactions between individuals are constrained by a network that connects the members of the population and mating or selection is restricted to neighborhoods within (and defined by) that network structure. Spatially structured EAs can be divided into two generic classes: *cellular EAs* (cEAs) [15] and *island models* [5]. This paper is restricted to the particular case of cEAs.

Takeover time is the time for a single solution to take over the entire population. It is often argued that cEAs provide a better sampling of the search space and improve the performance of panmictic EAs [3]. In fact, the efficiency of cEAs has been systematically demonstrated [2,3,21]. The reason for the alleged better performance of cEAs may be the fact that they reduce genetic diversity loss: individuals only interact with a restricted number of other individuals; therefore, good solutions diffuse through the network at a slower rate, displaying longer takeover times. The convergence is also slower, but the algorithm is less likely to converge to local optima.

Another advantage of spatially structured EAs is their adequacy for parallelization. Actually, the primary motivation for the design of non-panmictic EAs was to develop frameworks for distributed computing, different from the standard master-slave model [12]. In the island model, which is also known as *distributed model* [5], the population is divided into a predefined number of subpopulations and each one can be distributed over different processors that exchange information

* Corresponding author.
E-mail address: cfernandes@laseeb.org (C.M. Fernandes).

https://doi.org/10.1016/j.swevo.2020.100721
Received 16 September 2019; Received in revised form 4 May 2020; Accepted 26 May 2020
Available online 20 June 2020

(solutions) periodically with each other (a process called *migration*).

Cellular EAs can be parallelized by assigning each individual to one processor, but due to the typical population size of EAs (which can be over a thousand individuals for large and difficult problems), a practical approach is to split the population into groups of individuals and assign each group to a different processor [14]. The grid of individuals that structure cEAs is thus translated into a grid of processors that communicate with their neighbors when necessary.

Spatial structures introduce another level of design in the working mechanism of EAs: synchronous and asynchronous communication [1,3]. In the island model, for instance, a synchronization point can be imposed and all the islands should then move from one communication stage to another at the same time step. Another possibility is to let each island proceed with its evolutionary process independently of other islands (asynchronous model). In cEAs, the population is updated after all individuals recombine (synchronous) or each individual is replaced by its offspring immediately after recombination (asynchronous). Parallel speedup of spatially structured EAs typically depends on whether the communication strategy is synchronous or asynchronous [1].

Although parallelization can reduce the computational times required to deliver a solution, it can do nothing to correct bad design choices. If the sequential algorithm does not converge to an acceptable solution, the parallel version will very likely fail to converge also; if the sequential convergence speed is sub-optimal, the parallel EA accordingly displays sub-optimal computational times. Therefore, it is very important to study the design mechanisms of island models and cEAs and try to understand what makes an efficient spatially structured EA. Only then can we take full advantage of the parallelization potential of these algorithms.

The relationship between selection pressure, convergence speed and takeover times of cEAs has been studied in the past [1,5,9,10]. Nevertheless, the role of population size in cEAs and its relation to population structure, information diffusion, convergence speed and accuracy is still in an early stage. Since population size is a fundamental factor for the balance between convergence speed and accuracy [19], this paper continues the study reported in Ref. [8] and investigates the optimal population size and event takeover values (ETV) [23] of cEAs structured by regular graphs with different degree, under a test set with different types of problems and varying problem size.

The bisection method is used for assessing optimal population size in different fitness landscapes [19]. Under these settings, it is possible to determine the graph that maximizes the performance of the algorithm in each type of landscape, as well as an upper bound for the population size, above which computational resources are wasted. ETVs are used to investigate information (genetic material) diffusion through the population for different graphs, fitness landscapes, population size and problem size. By studying the population size and ETV values of cEAs with different degree, it is hoped that our comprehension of the mechanisms behind efficient cEAs can be improved.

The remainder of the paper is organized as follows: Section 2 gives a background review on cellular EAs, population size and takeover times, and explains the motivation for this study; Section 3 describes the methodology and experimental setup; Section 4 presents and discusses the results and Section 5 concludes the paper and outlines future lines of work.

2. Background review

A network structure that connects individuals and restricts their interaction is the main distinguishing feature of spatially structured EAs. It is therefore expectable that many lines of research in the field investigate the ability of different population structures in maintaining genetic diversity [21]. Giacobini et al. [10], for instance, give mathematical models for the selection pressure of cEAs on regular lattices. The validation of the model is made on 32×32 grids of 1024 individuals but the authors identified a breakdown of the usual logistic approximation for low-dimensional lattices. In Ref. [9], the authors investigated takeover times in random and small-world structures, but again the population size is set to a fixed value in every experiment.

In standard cEAs, the diversity promoted by a prefixed topology is uncorrelated with the fitness landscape, making their performance strongly dependent on the problem. To overcome these difficulties, complex and adaptive population structures were also studied, sometimes under recent developments in network theory [17]. Whitacre et al. [22] propose a dynamic structure with a self-organized definition of vicinity and interaction epistasis. The authors conclude that these features, when combined, lead to emergent behaviors that are absent from panmictic EAs or standard spatially structured EAs. Namely, the dynamic structure introduces an unprecedented capacity for sustainable coexistence of genetically distinct individuals within a single population, thus preventing genetic diversity loss. The population size varies within the range [50, 400], but the authors do not justify the choice of this particular interval neither they discuss its correlation to the fitness landscape.

The population size of an EA is of major importance for its efficiency because the initial population is the source of raw building blocks. Unfortunately, population sizing depends on two conflicting objectives. On the one hand, if the number of individuals is below a specific threshold, the algorithm, lacking sufficient genetic material, loses diversity prematurely and converges to local optima. On the other hand, when the population is larger than required by the problem dimension and difficulty, the algorithm converges to the global optimum with high probability but convergence speed is reduced. Although methods for EAs automatic population sizing have been discussed [19], they were seldom used within the specific context of cEAs.

Fernandes et al. [7] proposed dynamic topologies for cEAs that improve the probability of convergence to global optima. The authors performed population sizing tests and concluded that the dynamic topologies require smaller populations when compared to static cEAs. Fernandes et al. [8] investigated how population size of cEAs correlates with the population structure and the fitness landscape. The study is restricted to a single problem instance for each problem and the optimal population size is determined only for the selecto-recombinative versions of the algorithms.

This paper continues the work reported in Ref. [8]. Here, optimal population size is investigated for both selecto-recombinative and cEAs with mutation. Furthermore, scalability with problem size is also investigated. Different crossovers are tested and random graphs are also introduced in the test set. Lastly, ETVs are calculated, not only to determine the effects of structuring the population in ETVs distribution, but also to look for behavioral patterns that characterize good design choices. Nevertheless, the main goal is to investigate if cEAs are indeed more accurate and less prone to converge to local optima, even when population size is carefully tuned to near-optimal values.

3. Experimental setup

In cellular EAs, the individuals are structured by a graph, typically a ring or a grid [3], that defines each individual's neighborhood and consequently its mating pool. The objective of this work is to study the effects of the different graphs, from sparsely to fully connected (which is equivalent to a panmictic population), in the optimal population size of cEAs and consequently in their performance. Accordingly, the first step is to design graphs with increasing connectivity degree k. Most of the studies on cEAs use 1-D or 2-D grids – [2,3,16], for instance – and in fact a grid topology is not limiting [18]. However, a more general basic structure was chosen for this study, as exemplified in Fig. 1.

3.1. Population structures

Starting from a ring structure with $k = 2$, as in Fig. 1(a), k is doubled by linking each individual to the neighbors of its neighbors, thus creating regular graphs with $k = \{2, 4, 8, 16, 32\ldots\}$. Additionally, EAs with $k = n - 1$ (i.e., with panmictic populations), where n is the population size,

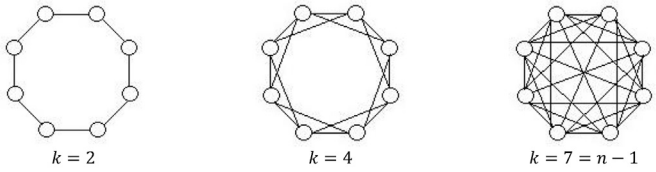

Fig. 1. Examples of regular graphs if population size is $n = 8$.

have been tested – Fig. 1(c).

Algorithm 1
cellular EA

1. $t = 0$
2. for each individual $\vec{x_i}$; $i \leftarrow 1$ to n, in population P_t
   2.1. initialize $\vec{x_i}$
   2.2. evaluate $\vec{x_i}$: $f(\vec{x_i})$
3. $t = t + 1$
4. for each individual $\vec{x_i}$; $i \leftarrow 1$ to n:
   4.1. compute neighborhood
   4.2. parent 1 is individual $\vec{x_i}$: $\vec{p_1} = \vec{x_i}$
   4.3. $\vec{p_2}$ is selected with binary tournament from the set of $\vec{p_1}$ neighbors
   4.4. two-point crossover $(\vec{p_1}, \vec{p_2}) \rightarrow (\vec{o_1}, \vec{o_2})$
   4.5. randomly select one of the offspring: $\vec{o_i}$ = random $(\vec{o_1}, \vec{o_2})$
   4.6. bit-flip mutate $\vec{o_i}$
   4.7. evaluate $\vec{o_i}$: $f(\vec{o_i})$
   4.8. insert $\vec{o_i}$ in temporary population $P_t^{'}$
5. for each individual $\vec{x_i}$; $i \leftarrow 1$ to n:
   5.1. replace $\vec{x_i}$ by $\vec{o_i}$ if $f(\vec{o_i}) \geq f(\vec{x_i})$ (maximization problems)
6. if stopping criterion is not met, go to 3

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 (*parent 1*) is recombined with one of its neighbors (*parent 2*) and from the set of two children generated by crossover one is randomly chosen and the other is discarded. Please note that generating and evaluating a single offspring as a result of recombination is a standard policy in cEAs literature, see, e.g., Ref. [3]. Other policies, such as generating two offspring, are also possible but that would result in doubling the number of evaluations per generation.

Synchronous cEAs are used: offspring are placed in a secondary population and the replacement step is only performed when that secondary population size is equal to n. Then, the fitness of parents and children are compared and offspring $\vec{o_i}$ replaces parent $\vec{x_i}$ if it codifies a better solution to the problem. The cEA was implemented by the authors and the pseudo-code is in Algorithm 1.

3.2. Bisection algorithm

Finding the optimal or near-optimal population size for a specific target-problem is a critical step in the design of any EA. In this paper, the bisection method [19] is used to determine the optimal size of the algorithms. Please note that the bisection algorithm is only applied to selecto-recombinative EAs, i.e., EAs without mutation. The objective is to determine an upper bound for the population size by establishing the smallest populations that guarantees a sufficient supply of building blocks for the EA to converge to the global optimum without requiring mutation. It is expected that smaller populations can be used effectively when mutation probability is set to non-zero values.

Bisection is described by Algorithm 2. In this paper, threshold T is set to 0.1 and initial population size n_0 to 25. Every configuration is executed 30 times before updating. The convergence criterion is met if 28 of the 30 runs converge to the global optimum. The algorithms are tested with crossover probability$p_c = 1.0$ and mutation probability (p_m) is set to 0 as required by the bisection algorithm. After determining the optimal size n_{opt}, the corresponding statistical measures are recorded.

Algorithm 2
Bisection algorithm

1. $n = n_0$
2. run EA with population size n
3. if convergence criterion is not met
   3.1. $n = 2 \times n$
   3.2. go to 2
4. $n_{min} = n/2$
5. $n_{max} = n$
6. repeat
   6.1. $n = (n_{min} + n_{\max})/2$
   6.2. run EA with population size n
   6.3. if convergence criterion is met, $n_{max} = n$
   6.4. else, $n_{min} = n$
7. until $((n_{max} - n_{\min})/n_{min}) < T$
8. compute the statistics for the problem size using $n = n_{max}$

For EAs with $p_m > 0$, the optimal population size is estimated using the standard tuning method: size is varied while other parameters remain fixed and then the performance of each configuration is evaluated. Each algorithm is executed 30 times and each run is limited to a maximum of 1,000,000 evaluations. The performance of each algorithm is measured and compared according to three criteria: convergence speed (number of evaluations required to find the global optima), accuracy (best fitness values) and robustness (success rates, i.e., the number of runs in which the global optimum is found).

3.3. Event takeover values

Introduced by Whitacre et al. [23], ETVs use information from genealogical trees to evaluate the impact of each individual on the population: the ETV of individual *id* in generation *t* measures the number of descendants of the individual that belong to the population in that generation. Equation (1) describes ETV calculation:

$$ETV_t(id) = \sum_{i=1}^{n} \sum_{j=1}^{Tobs} \varphi_{i,j} \tag{1}$$

$$\varphi_{i,j} = \begin{cases} 1 & if\ id = M_i(id_j) \\ 0 & otherwise \end{cases}$$

where n is the population size, *Tobs* the maximum size of the ascendants list and $M_i(id_j)$ is the jth position in the individual i ascendants list. In this paper, *Tobs* is limited only by the number of generations.

Genetic hitchhiking is avoided by comparing ETV_t of an individual with one of its offspring. If parent and child have the same value, the parent's ETV is set to 0. The procedure is formally described by Equation (2):

$$\begin{array}{l} if\ \left(\left(id_1 = M_i(id_j)\right)\ and\ \left(\left(id_2 = M_i(id_{j-1})\right)\right)\right. \\ and\ ETV_i(id_1) = ETV_i(id_2) \\ then\ ETV_i(id_1) = 0 \end{array} \tag{2}$$

The detailed description of ETVs calculation procedures are in Ref. [23].

3.4. Problems

The test set is composed of five functions with different characteristics: onemax (1-trap), 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, they may be deceptive or not. The trap functions in this paper are defined by:

$$F(\vec{x}) = \begin{cases} b, & \text{if } u(\vec{x}) = b \\ b - 1 - u(\vec{x}), & \text{otherwise} \end{cases} \quad (3)$$

where u ($\vec{x}$) is the unitation function and b is the subproblem size as well as the fitness of its global optimum. The trap functions are constructed by juxtaposing a number of subproblems. The total fitness is the sum of subproblems fitness values. With these definitions, order-2 are non-deceptive, order-3 are in a region between deceptiveness and non-deceptiveness and order-4 are fully deceptive.

The particular case of 1-trap is also called the onemax problem and is the prototypical simple problem in EAs with binary codification. It is described by Equation (3) when $b = 1$ and it basically consists of maximizing the number of ones in a binary string.

MMDP is a NP-hard, deceptive and multimodal problem that consists of b 6-bit subproblems with two global optima and a deceptive attractor in the middle of the fitness landscape. Each subproblem fitness value depends on the unitation function. Table 1 shows the contribution of each subproblem to the fitness of a string. Optimal solutions fitness values are equal to the number of subproblems: for instance, if the MMDP string size is 120 (which correspond to 20 subproblems), the fitness of the global optima is 20.

4. Results and discussion

Since each problem has its own characteristics and poses different difficulties to the EAs, each one is analyzed and discussed separately.

4.1. Onemax

Four onemax instances with different string size l were tested: $l = \{50, 100, 200, 400\}$. This way, population size and convergence speed scalability with problem size is also investigated. Every algorithm converged to the global optima in every instance of the problem; therefore, the only criterion used for the comparisons is the convergence speed.

Fig. 2 shows (a) the optimal population size for several cEAs (including a panmictic EA, with $k = n - 1$) as determined by the bisection method, and (b) the convergence speed (median evaluations to find optimum) of the optimal configurations. Please note that cEAs optimal populations scale better with string size than the panmictic EA. However, as seen in Fig. 2(b), the panmictic EA with optimal population size converges faster and scales better than the optimal cEAs with low k.

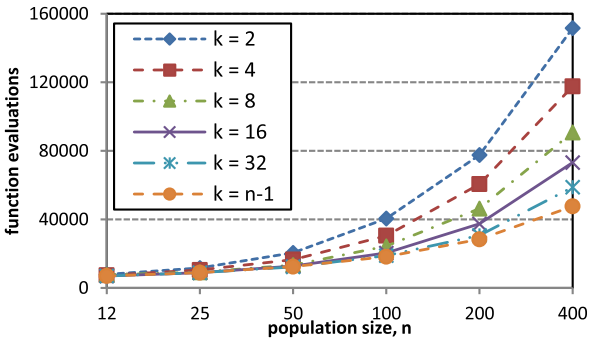

Fig. 3. Onemax: $l = 400$; $p_m = 1/l$. Convergence speed with different population size: median values over 30 runs.

Appendix A gives the numerical results of the bisection experiments (population size and evaluations for a solution), the corresponding box plots and the results of the Wittkowski statistical test (generalized Friedman rank sum test) [24]. Table A.1 shows that the optimal population of cEAs with $k = 2$ scales linearly with onemax size, but the fastest algorithm when population is set to the optimal value is cEA with $k = 32$ for $l > 100$, and panmictic EA ($k = n_{opt} - 1$) for $l = 50$ and $l = 100$. The differences between the algorithms are statistically significant, as seen in Table A.2. Figure A.1 shows the convergence speed box plots for each problem size l.

By keeping the design simple, selecto-recombinative EAs are useful for understanding some of the underlying mechanism of the algorithms. Nevertheless, EAs performance can be significantly improved with mutation, and although bisection gives an upper bound for the population size, this value is often not very informative when $p_m > 0$. This is particularly evident in the case of simple problems like onemax, in which the supply of building blocks provided by the initial population is less important than the variability introduced during the run by mutation. Therefore, in order to estimate the optimal population size for $p_m > 0$, the EAs were tested with population size set to 12, 25, 50, 100, 200 and 400, $p_m = 1/l$, and, again, string size l set to 50, 100, 200 and 400.

For every string size, $n = 12$ was the fastest population, a value that is significantly lower than the optimal populations obtained for the selecto-recombinative EAs. See Fig. 3 for the convergence speed of different populations with $l = 400$. Please note that cEAs with low k are more

Table 1
Contribution of each MMDP subproblem to the fitness.

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

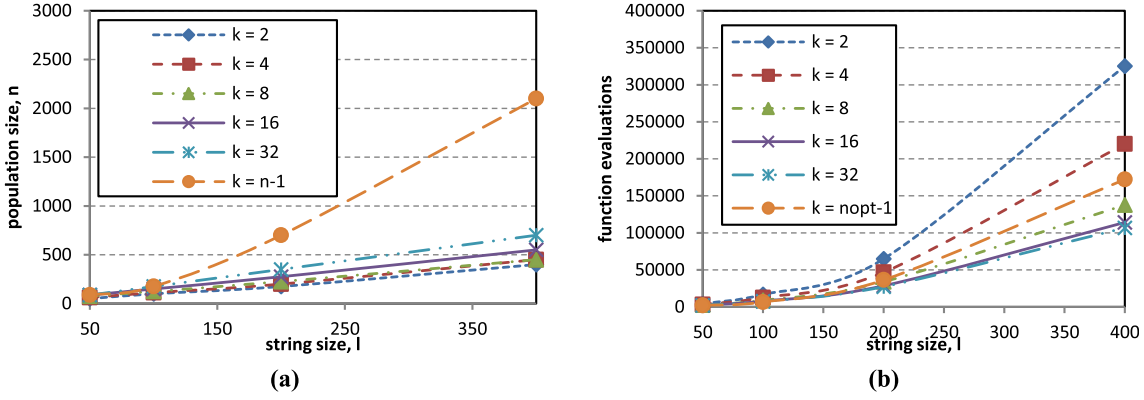

Fig. 2. Onemax. (a) Optimal population size of selecto-recombinative EAs ($p_m = 0$) with different k; (b) Convergence speed (evaluations to global optimum: median values over successful runs) with optimal population size.

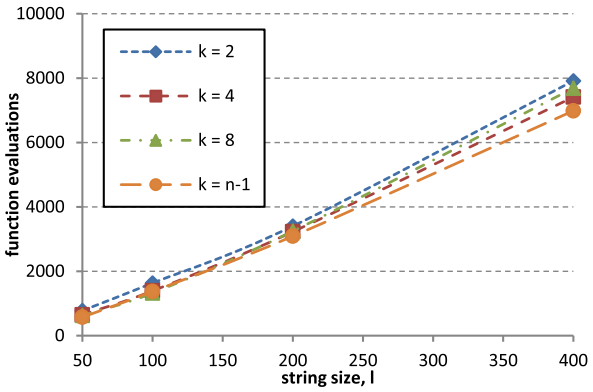

Fig. 4. Onemax: $n = 12$; $p_m = 1/l$: convergence speed with different string size.

sensitive to population size than cEAs with high k, namely $k = n-1$ (panmictic EA).

Fig. 4 displays the convergence speed of the cEAs with $n = 12$ when the string size varies. Results with $k = 16$ and $k = 32$ are not plotted because for $n = 12$ they are equivalent to the panmictic EA with $k = n-1$. Observing the box plots in Fig. 5 and the statistical tests in Table 2, it becomes clear that cEAs do not outperform the panmictic EA on onemax problems.

The importance of setting population size to near-optimal values when comparing EAs is exemplified by the cEAs convergence speed when $n = 400$ (Fig. 6). In this case, the panmictic EA scales and performs (significantly) better than the other EAs. A comparison with population set to $n = 400$ would conclude erroneously that the panmictic EA is faster than every cEA.

4.2. 2-Trap

2-trap functions were tested with $l = \{50, 100, 200, 400\}$. Fig. 7 shows (a) the optimal population size of selecto-recombinative EAs, and

Table 2
Number of evaluations rank and P-value of the Wittkowski test (generalized Friedman rank sum test) for the 1-trap problem with $n = 12$ and $p_m = 1/l$; in superscript the configurations to which the differences are not significant (after pairwise comparisons using paired Wilcoxon signed-rank and corrected for multiple comparison with the Holm method).

	$k = 2$	$k = 4$	$k = 8$	$k = n-1$	*P-value*
$l = 50$	3.5	2.5^{8}	$2.2^{4;\ n-1}$	1.8^{8}	0.0
$l = 100$	$3.3^{\ n-1}$	$2.3^{8;\ n-1}$	$2.1^{4:\ n-1}$	$2.4^{2;4;8}$	0.0
$l = 200$	$3.1^{4;8}$	$2.4^{2;8;\ n-1}$	$2.4^{2;4;\ n-1}$	$2.1^{4;8}$	0.0
$l = 400$	$2.8^{4;8;\ n-1}$	$2.4^{2;8;\ n-1}$	$2.5^{2;4:\ n-1}$	$2.2^{2;4;8}$	0.1

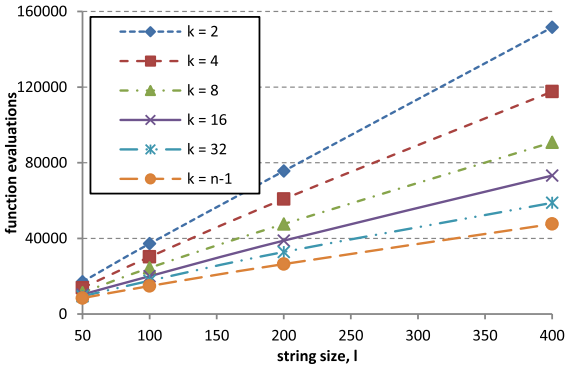

Fig. 6. Onemax: $n = 400$; $p_m = 1/l$: Convergence speed with different string size.

(b) the corresponding convergence speed of the optimal configurations. Again, cEAs populations scale better than the panmictic population. However, as with the onemax problem, the panmictic EA convergence speed is not inferior to the structured algorithms: Fig. 7(b) shows that $k = n-1$ structure converges faster than cEAs with low connectivity degree.

Table A.3 in the Appendix shows the numerical results of the

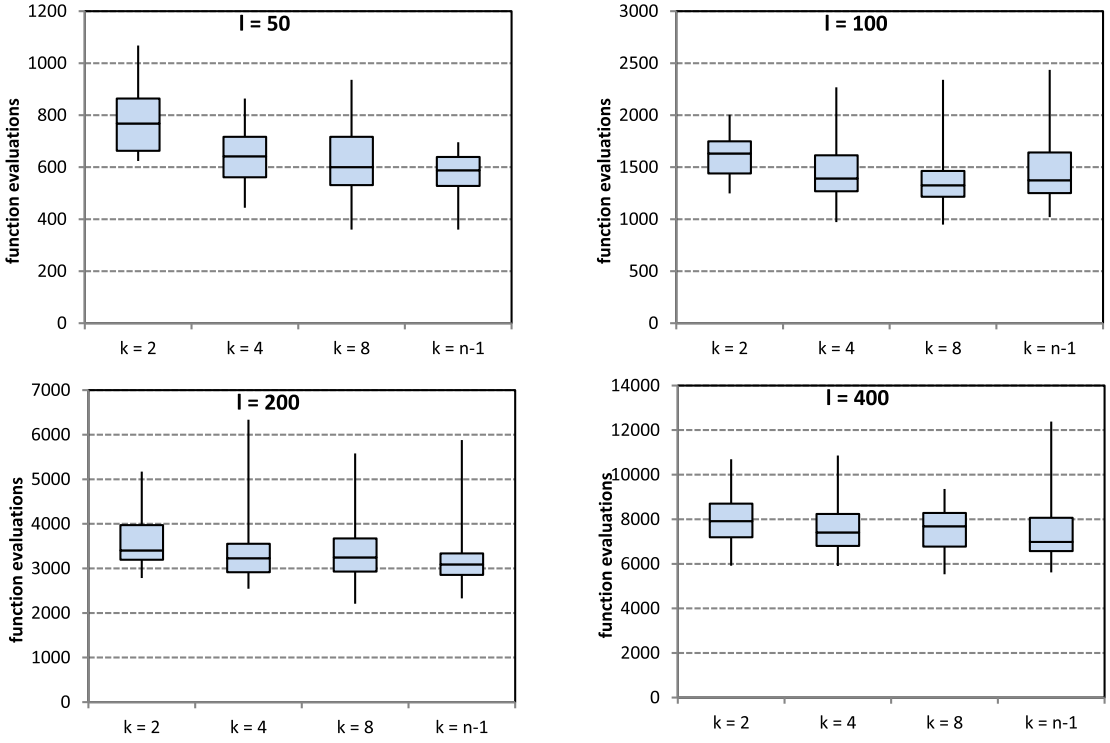

Fig. 5. Onemax: $n = 12$; $p_m = 1/l$: convergence speed box plots with different problem size.

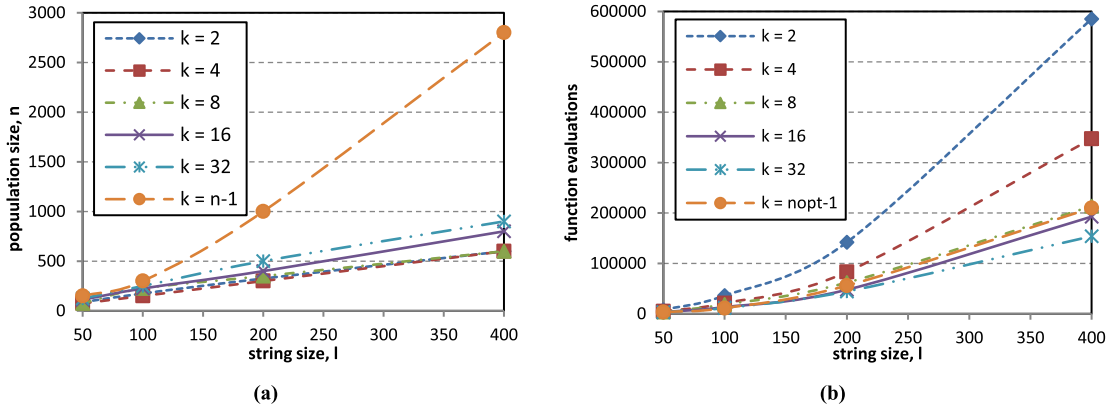

Fig. 7. 2-trap. (a) Optimal population size for selecto-recombinative EAs ($p_m = 0$) with different k; (b) Convergence speed (evaluations to global optimum: median values over successful runs) with optimal population size.

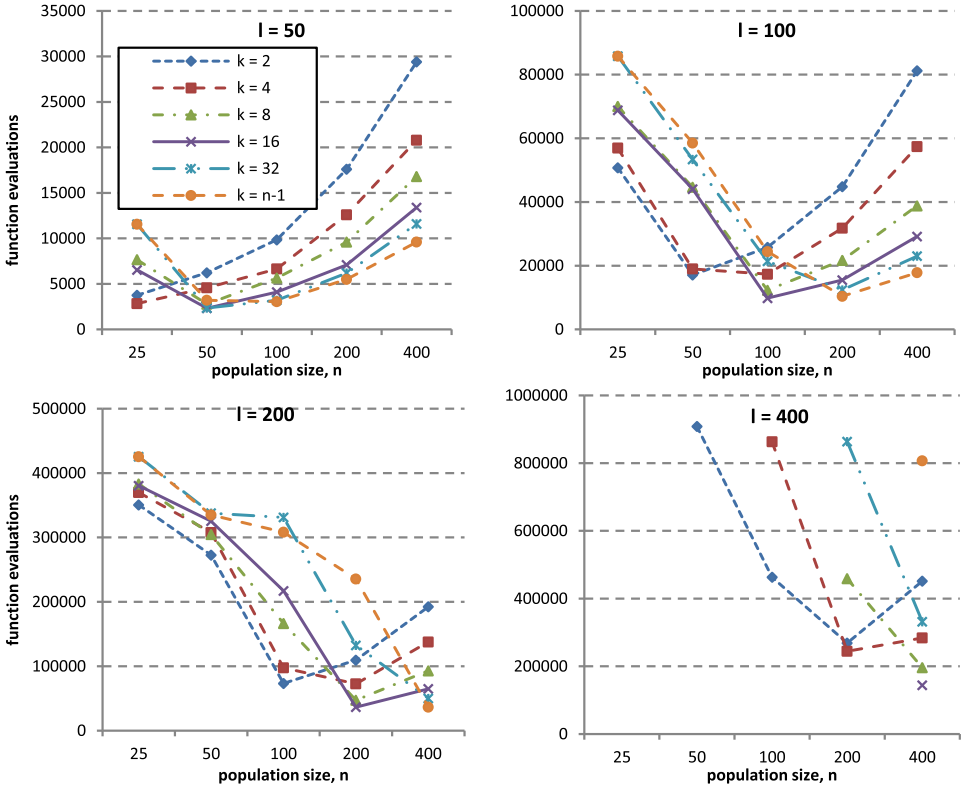

Fig. 8. 2-trap: $p_m = 1/l$. Convergence speed.

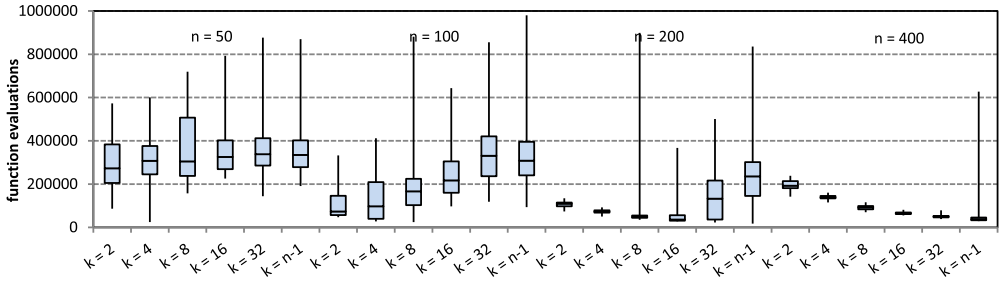

Fig. 9. 2- traps with $l = 200$. EAs with $p_m = 1/l$. Convergence speed box plots with different population size n.

Table 3
2-traps with $l = 200$. EAs with $p_m = 1/l$. Pairwise comparisons using paired Wilcoxon signed-rank test between panmictic EA with $n = 400$ and $k = n-1$ and all the other configurations: symbol + means that panmictic EA with $n = 400$ and $k = n-1$ is significantly better, – means that it is significantly worse and ≈ means that there are no significant differences between the configurations.

	$k = 2$	$k = 4$	$k = 16$	$k = 16$	$k = 32$	$k = n-1$
$n = 50$	+	+	+	+	+	+
$n = 100$	≈	≈	+	+	+	+
$n = 200$	≈	≈	≈	≈	≈	≈
$n = 400$	≈	≈	≈	≈	≈	n/a

Table 4
2-traps with $l = 400$. EAs with $p_m = 1/l$. Successful runs (out of 30) of each algorithm.

	$n = 25$	$n = 50$	$n = 100$	$n = 200$	$n = 400$
$k = 2$	0	1	10	28	30
$k = 4$	0	0	7	25	30
$k = 8$	0	0	2	21	30
$k = 16$	0	0	0	14	29
$k = 32$	0	0	0	6	21
$k = n-1$	0	0	0	0	9

experiment for determining the optimal size and Figure A.2 the box plots of the resulting data. Like for onemax, the fastest configurations are the cEAs with higher connectivity and the panmictic EA. Panmictic EA ranks first when $l = 100$ (but with no significant differences to the cEA with $k = 32$, as shown in Table A.4).

2-traps were also tested with $p_m > 0$. Population size was set to 25, 50, 100, 200 and 400, $p_m$to $1/l$, and string size l to 50, 100, 200 and 400. Fig. 8 plots the convergence speed for different strings: the population size that optimizes convergence speed depends on the degree k and the difference between cEAs with low k and the EA with $k = n-1$ is evident.

Unlike onemax problems, the optimal population size obtained by the bisection algorithm for 2-traps can help tuning the population for mutational EAs. In general, the best population size for the cEAs with mutation was found to be in the range $[n_{opt}/2, n_{opt}]$, where n_{opt} is the optimal population size of the selecto-recombinative version of the cEAs found by the bisection method.

Fig. 9 shows the box plots of results attained by the algorithms with the $l = 200$ instance of the problem ($p_m = 1/l$). Every configuration converges to the global optimum in every run, therefore comparisons

Table 5
3-trap, $l = 120$, $p_m = 1/l$. Successful runs (SR) and median function evaluations (FE).

		$n = 25$	$n = 50$	$n = 100$	$n = 200$	$n = 400$	$n = 800$
$k = 2$	SR	0	0	19	30	30	30
	FE	–	–	56,100	96,500	177,600	340,000
$k = 4$	SR	0	0	0	30	30	30
	FE	–	–	–	65,500	116,200	213,600
$k = 8$	SR	0	0	0	27	30	30
	FE	–	-	–	38,600	76,400	141,600
$k = 16$	SR	0	0	0	0	30	30
	FE	–	–	–	–	48,800	93,200
$k = 32$	SR	0	0	0	14	30	30
	FE	–	–	–	20,800	38,400	70,800
$k = n-1$	SR	0	0	0	7	24	30
	FE	–	–	–	340,000	26,000	43,200

between the algorithms can be made considering only convergence speed. The optimal population size depends on the connectivity degree: cEA with $k = 2$ optimizes its performance with $n = 100$, while $k = 32$ and $k = n-1$ best results are attained with $n = 400$. These results again stress the importance of setting the appropriate population size. However, the most interesting outcome is the panmictic EA performance: panmictic EA with $n = 400$ convergence speed is better or at least statistically equivalent to other configurations, as shown in Table 3. Like for onemax, there are no evidences that cEAs outperform panmictic EA in 2-trap functions.

The convergence speed for some configurations of the algorithms when $l = 400$ is not represented in Fig. 8. This is because when l is set to 50, 100 and 200, every algorithm attained 100% success rates in every instance of the problem, but when $l = 400$ smaller populations do not find the global optima within the 1,000,000 function evaluations stop criteria. To complement the results in Fig. 8 for $l = 400$, Table 4 shows the number of successful runs of each algorithm for 2-traps with $l = 400$. From the analysis of the results it appears that the panmictic EA requires populations with more than 400 individuals to optimize its performance. In fact, tests with larger populations showed that the EA with $k = n-1$ optimizes its performance with $n \geq 1600$.

4.3. 3-Trap

To test selecto-recombinative EAs on 3-trap functions, string size was set to $l = \{30, 60, 120, 240\}$. Optimal populations are presented in Fig. 10(a). Like in previous experiments, cEAs populations scale better with problem size, but as seen in Fig. 10(b), the convergence speed of panmictic EA is better than cEAs with low k. Numerical results and box plots are in Appendix A.

With mutation, results are similar to those obtained with 2-traps:

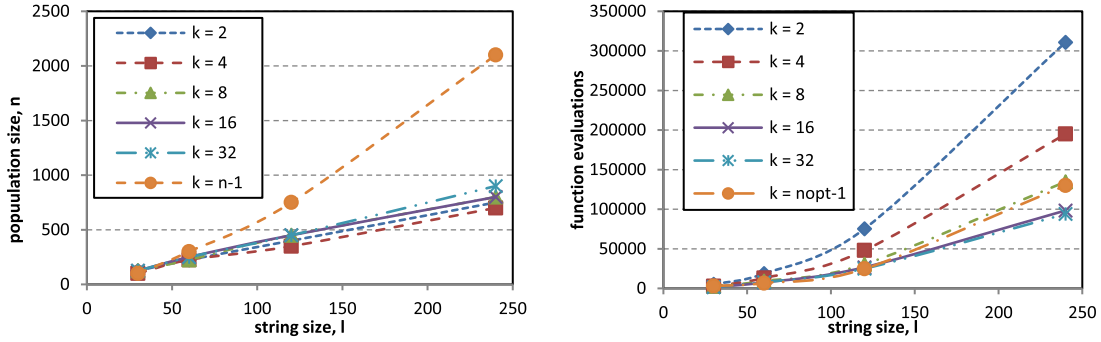

Fig. 10. 3-trap. (a) Optimal population size for selecto-recombinative EAs ($p_m = 0$) with different k, (b) Convergence speed (evaluations to global optimum: median values over 30 runs) with optimal population size.

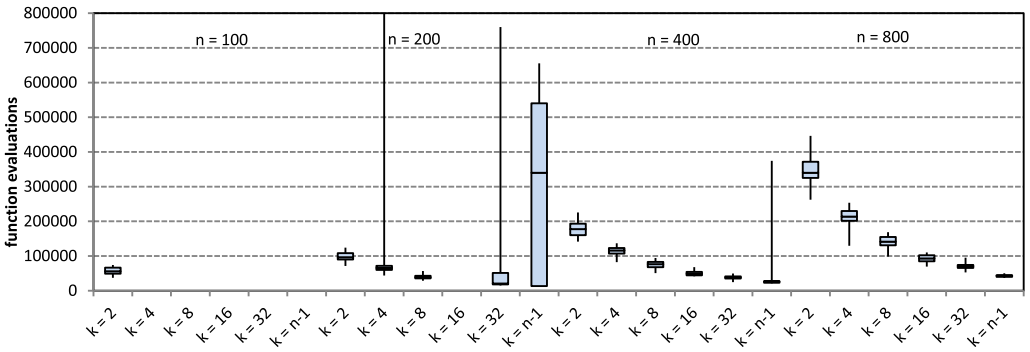

Fig. 11. 3- traps, $l = 120$, $p_m = 1/l$. Convergence speed box plots with different n.

Table 6
3- traps, $l = 120$, $p_m = 1/l$. Pairwise comparisons using paired Wilcoxon signed-rank test between panmictic EA with $n = 800$ and all the other configurations: symbol + means that panmictic EA with $n = 800$ and is significantly better, – means that it is significantly worse and ≈ means that there are no significant differences between the configurations.

	$k = 2$	$k = 4$	$k = 8$	$k = 16$	$k = 32$	$k = n - 1$
$l = 25$	+	+	+	+	+	+
$l = 50$	+	+	+	+	+	+
$l = 100$	+	+	+	+	+	+
$l = 200$	+	+	+	+	+	+
$l = 400$	+	+	+	+	≈	≈
$l = 800$	+	+	+	+	+	n/a

population size that optimizes convergence speed depends on k, but if set to their optimal size, panmictic and structured populations perform similarly. Table 5 gives the median number of evaluations required by the best configuration of each EA when $l = 120$, while Fig. 11 shows the box plots for each EA configuration.

Table 6 shows the results of pairwise comparison between panmictic EA with $n = 800$ and all the other configurations of the algorithm. Panmictic EA with $n = 800$ is significantly better than all the other configurations, except panmictic EA with $n = 400$ and cEA with $n = 400$ and $k = 32$. Again, it is demonstrated that when the population size optimized, the panmictic EA is hardly outperformed by structured EAs. The claim that structured EAs reduce genetic diversity loss is confirmed by the optimal population sizes: cEAs with low connectivity require smaller population to converge consistently to the global optimum. However, if the panmictic EA is properly tuned, its performance is equal or better than the structured EAs.

4.4. 4-Trap and MMDP

MMDP and 4-trap problems are the most challenging fitness landscapes in the test set. Their local optima and deceptiveness require a careful balance between exploration and exploitation and genetic diversity is very important to avoid premature convergence to sub-optimal solutions. First, let us look at the performance of selecto-recombinative EAs. Fig. 12 shows that panmictic populations scale worse than structured populations (numerical results in Appendix A). However, as in non-deceptive and quasi-deceptive problems, panmictic EAs convergence speed scales better or equivalent to cEAs – see Fig. 13 for the convergence speed of 4-trap problems.

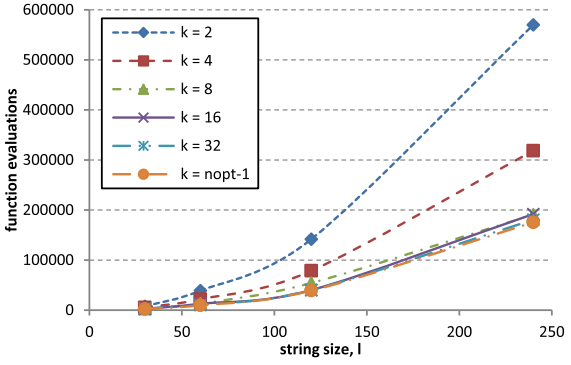

Fig. 13. 4-trap: $p_m = 0$; Convergence speed (evaluations to global optimum: median values over 30 runs). Population size as in Fig. 12(a).

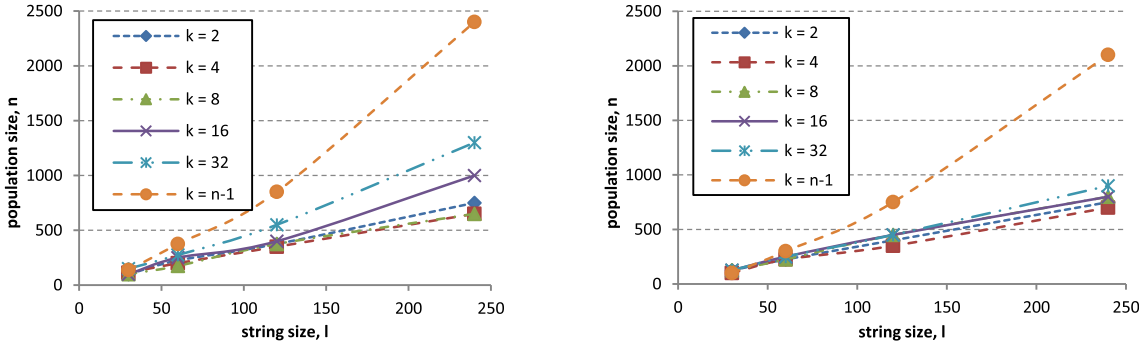

Fig. 12. Optimal population size for selecto-recombinative cEAs: (a) 4-trap; (b) MMDP.

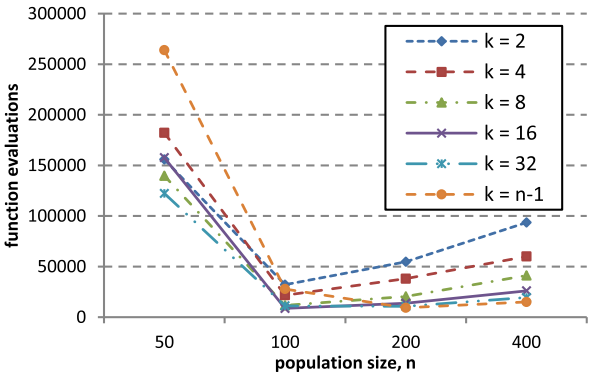

Fig. 14. MMDP: $l = 60$. EAs with $p_m = 1/l$. Convergence speed with different population size.

When $p_m = 1/l$, the optimal population, as in previous experiments, depends on the degree k. With smaller populations, low k cEAs perform better than the panmictic EA, but if the population for the latter is set to near-optimal values its performance is equivalent or better than cEAs. Fig. 14 is an example of such behavior: for MMDP with $l = 60$, the panmictic EA is clearly outperformed by the cEAs if $n = 50$, but if $n = 200$ the panmictic EA is not only faster than the other algorithms with the same population size, but it is also faster considering the whole range of population sizes in the test (50, 100, 200 and 400).

Fig. 15 shows the box plots of MMDP configurations with $n = 100$ and $n = 200$ and Table 7 shows the respective pairwise statistical test comparing the panmictic EA with $n = 200$ with other configurations. The tests demonstrate that the panmictic EA is significantly better or equivalent to the other configurations. Again, no evidences were found that structured populations perform better than panmictic EAs.

The results presented above indicate that the panmictic EA is as accurate as and faster than cEAs in several instances of the problems. If it is true that cEAs reduce genetic diversity loss rate – and optimal population size tests in this paper confirm this hypothesis –, it seems that panmictic EAs, if properly tuned and supplied with sufficient raw building blocks (i.e., with an appropriate population size), are able to balance exploration and exploitation, succeed in maintaining genetic diversity and perform better or similarly to cEAs.

Table 7
MMDP; $l = 60$; $p_m = 1/l$. Pairwise comparisons using paired Wilcoxon signed-rank test between panmictic EA with $n = 200$ and other configurations: symbol + means that panmictic EA with $n = 200$ and is significantly better, – means that it is significantly worse and ≈ means that there are no significant differences between the configurations.

	$k = 2$	$k = 4$	$k = 8$	$k = 16$	$k = 32$	$k = n - 1$
$n = 100$	+	+	≈	≈	≈	≈
$n = 200$	+	+	+	≈	≈	n/a

4.5. cEAs with higher connectivity degree

In the experiments described in this section, cEAs were tested with k ranging from to 2 to 32, which include the typical connectivity degrees found in cEAs literature: ring ($k = 2$), von Neumann neighborhood ($k = 4$) and Moore neighborhood ($k = 8$). Furthermore, bisection tests started with population size 25, which excluded cEA with k above that value. However, for large populations it is possible to further increase k values. In this section, MMDP problems were tested with cEAs with higher connectivity degree.

As seen in Section IV. D, for MMDP problems with $l = 60$ and EAs with $n = 100$ and $n = 200$, no statistical differences were observed between cEAs with large k and panmictic EAs. However, it is possible that k values between $k = 32$ and $k = n - 1$ achieve even better results. In order to test this hypothesis, an additional experiment was performed using cEAs with $k = 64$ and $k = 128$ (in this case only for n = 200). The

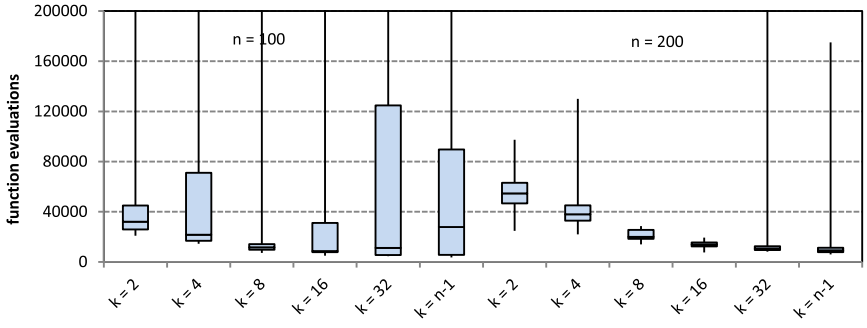

Fig. 15. MMDP with $l = 60$. EAs with $p_m = 1/l$. Convergence speed box plots with $n = 100$ and $n = 200$.

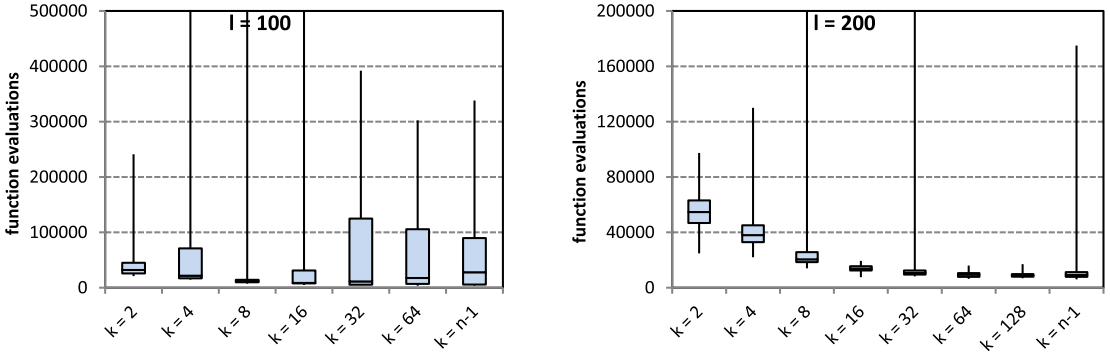

Fig. 16. MMDP: $l = 60$. EAs with $p_m = 1/l$. Convergence speed box plots with $n = 100$ and $n = 200$.

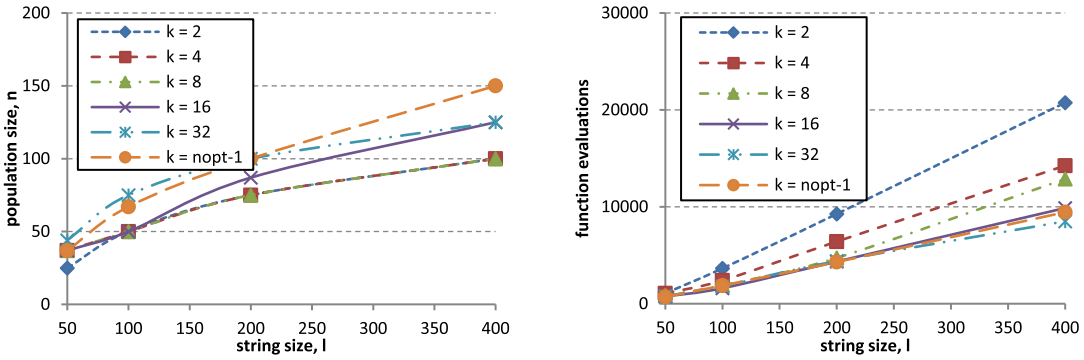

Fig. 17. Onemax. EAs with uniform crossover: (a) Optimal population size of selecto-recombinative EAs ($p_m = 0$) with different k; (b) Convergence speed (evaluations to global optimum: median values over successful runs) with optimal population size.

resulting convergence speed box plots are shown in Fig. 16.

It seems that in this case structures with higher connectivity do not improve significantly the performance of cEAs with $k = 32$ and EAs with $k = n - 1$ and in fact pairwise comparisons using paired Wilcoxon signed-rank test confirm that there are no statistical differences between the algorithms. However, these results do not invalidate the hypothesis that for larger populations or other types of problems, structured cEAs with higher connectivity may indeed improve panmictic EAs.

4.6. Crossover

For the previous experiments, the EAs were designed with 2-point crossover operators. However, it is has been demonstrated [20] that for trap problems different operators result in significantly different performance. With onemax problems, for instance, uniform crossover scales much better than 1-point or 2-point, while for deceptive problems like 4-trap and MMDP, uniform crossover scales worse, because it is too disruptive to recombine the building blocks as a whole [20]; accordingly, 1-point crossover is expected to scale better than 2-point in deceptive fitness landscapes.

In order to investigate the general behavior of cEAs and the robustness of previous conclusions on the relative performance of the algorithms with different types of crossover, the EAs were tested with uniform crossover for onemax problems, and 1-point crossover for MMDP.

Fig. 17 shows (a) the optimal population size as determined by the bisection method and (b) the convergence speed (median evaluations to find optimum) of the optimal configurations – please compare with Fig. 2. As expected [20], the EAs convergence speed scale linearly with problem size. Nonetheless, what is most important for the investigation in this paper is that the relative behavior observed with 2-point crossover is maintained when switching to uniform crossover – please compare the

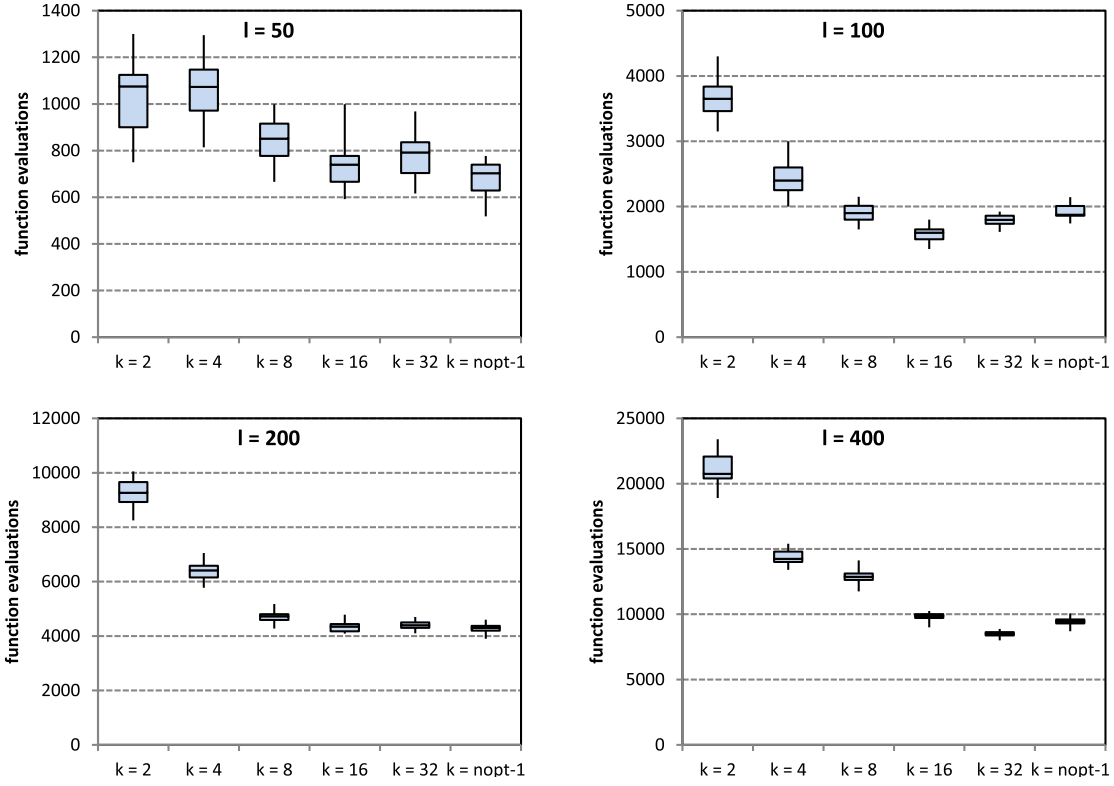

Fig. 18. Onemax. EAs with uniform crossover ($p_m = 0$): convergence speed box plots for each problem size.

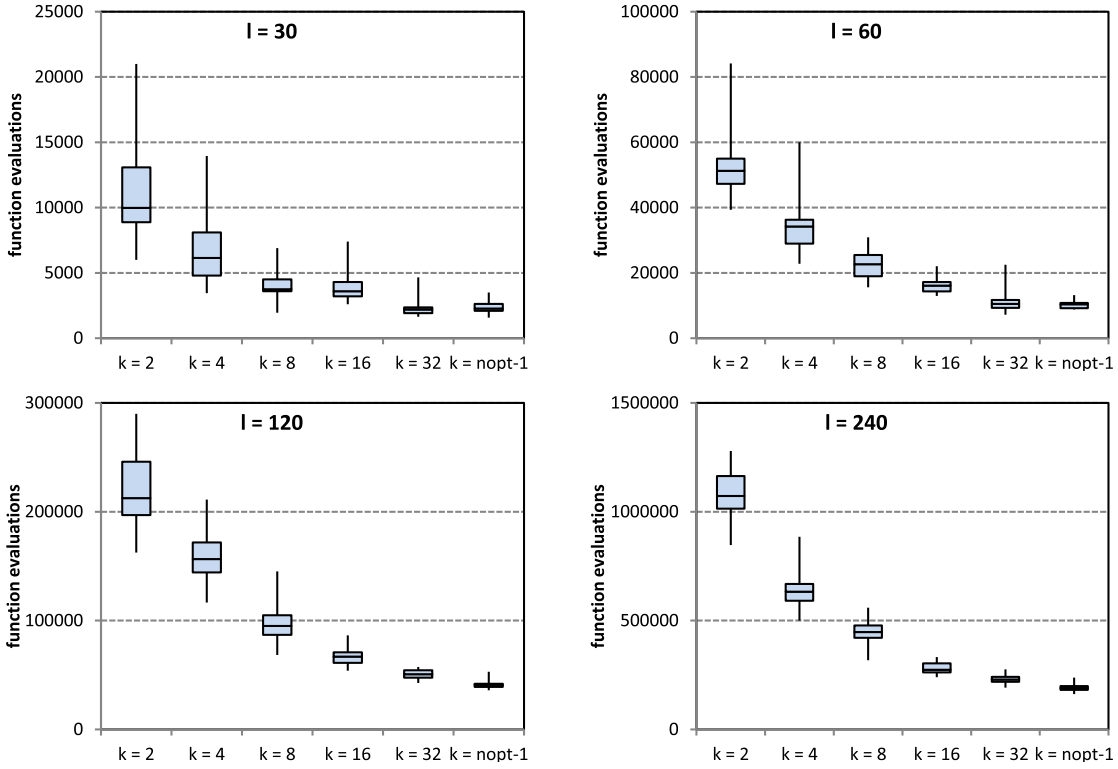

Fig. 19. MMDP. EAs with 1-point crossover ($p_m = 0$): convergence speed box plots for each problem size.

box plots in Fig. 18 to those in Figure A.1 in the Appendix.

For MMDP, the convergence speed box plots are in Fig. 19. Like with 2-point crossover, the panmictic EAs perform significantly better than low k cEAs and are competitive with larger k cEAs. Again, the relative performance is similar when using different types of crossover – please compare Fig. 19 with Figure A.5 in Appendix.

Finally, MMDP was tested with 1-point crossover and $p_m = 1/l$. A summary of the results is in Fig. 20. The convergence speed curves are similar to those in Fig. 14 (MMDP and EAs with 2-point crossover) and the box plots show that the best convergence speed is attained by the panmictic EA, meaning that for the MMDP with uniform crossover the conclusions regarding the EAs performance are similar to those in Section IV.D.

In conclusion, the type of crossover affects the performance of the EAs and the best option depends on the type of problem. However, the relative performance of the different cEAs and the panmictic EA does not seem to be dependent on the type of crossover, both for selecto-recombinative and mutation-EAs.

4.7. Random graphs

Regular graphs are useful for empirical and theoretical studies of cEAs performance and dynamic behavior under different connectivity degrees. However, in real world implementations of distributed EAs, it is not always possible to guarantee a regular network structure. Take for instance the extreme case of peer-to-peer networks, in which nodes connect and disconnect without central coordination [13]. Although a comprehensive investigation of cEAs with random and dynamic networks is beyond the scope of this paper, it is important in the context of the present work to confirm if the cEAs behavior reported in previous sections is also observed when the networks are provided with some kind of randomness.

There are many models of random graphs. The most simple is to start with n nodes and then establish successive edges between randomly chosen nodes. However, this model does not guarantee that after generating $k \times n$ edges every node has at least one connection (or, if starting with $n' > n$ edges, connecting n edges at random results in the

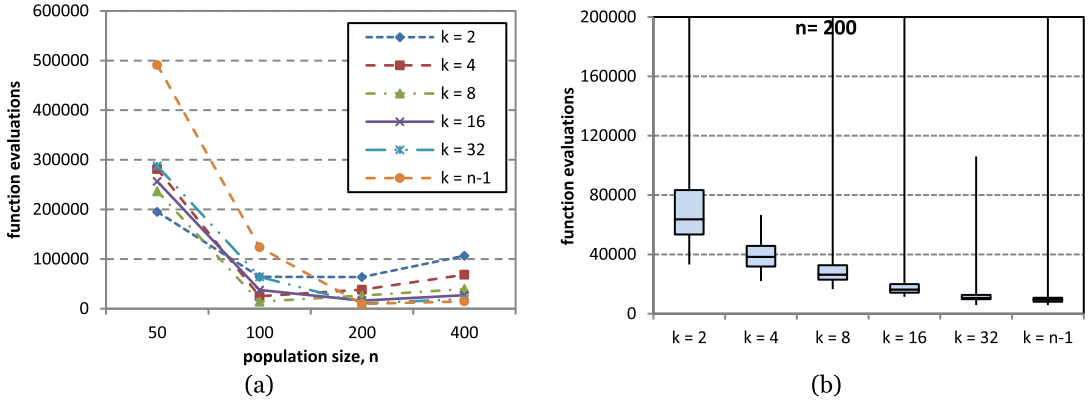

Fig. 20. MMDP, $l = 60$. EAs with 1-point crossover ($p_m = 1/l$): a) convergence speed with different population size; b) convergence speed box plots for $n = 200$.

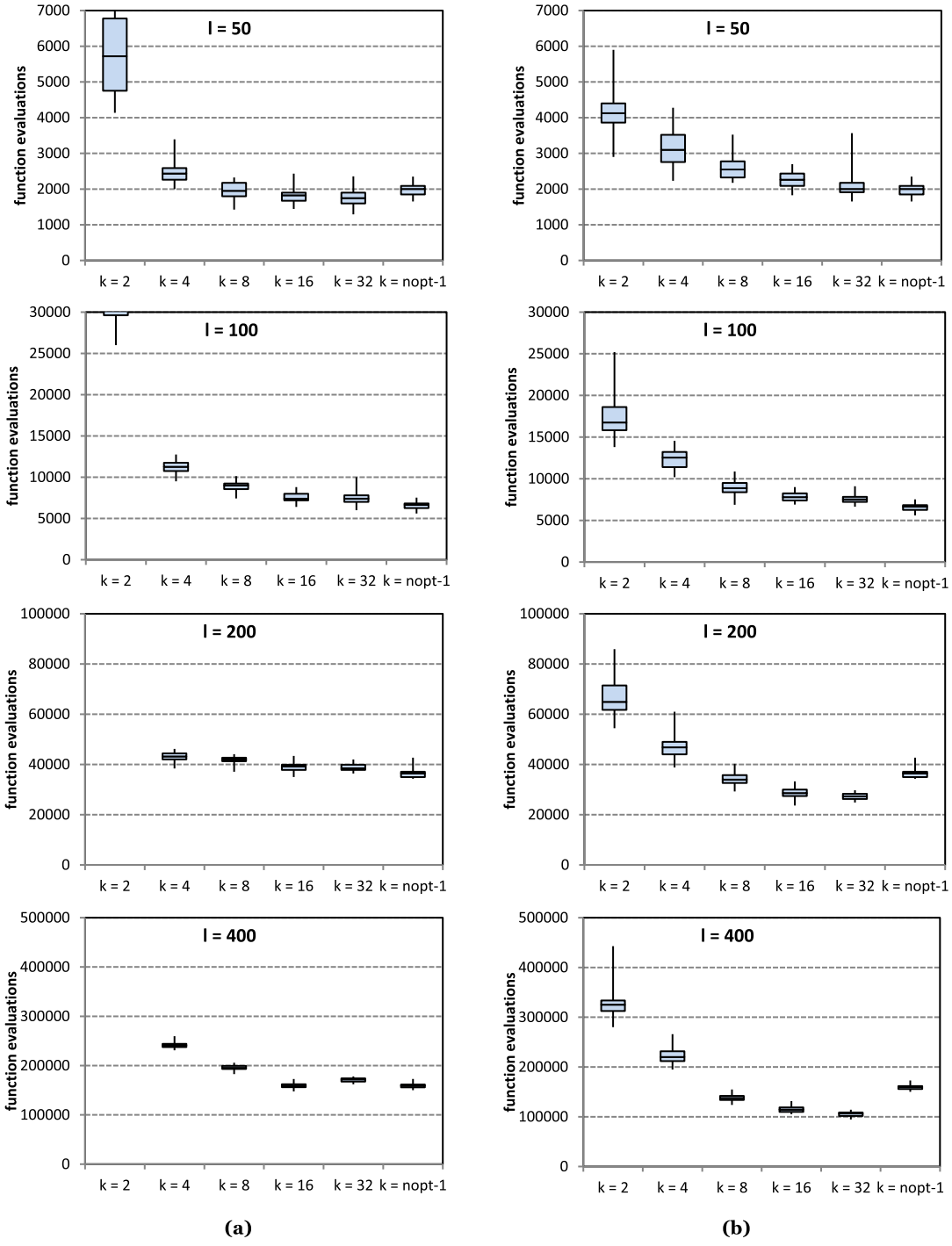

Fig. 21. Onemax. EAs with $pm = 0$. Box plots: (a) random graphs; b) regular graphs. (Some measures related to $k = 2$ configurations fall out of the range for the sake of visualization.)

desired average connectivity degree). Therefore, in order to compare the performance of cEAs and panmictic EAs on regular and random graphs, the latter were generated using a simple model: starting with n nodes, edges created by randomly selecting two nodes, while guaranteeing that each one of n nodes of the graph has at least $k/2$ edges. Graphs with 2, 4, 8, 16 and 32 average edges per node were generated and tested with selecto-recombinative EAs on onemax and MMDP problems. Like in previous sections, the bisection method was used to determine the optimal population size of selecto-recombinative EAs and then convergence speed was recorded and statistical measures taken.

A summary of the results attained by the algorithms on onemax problems is presented in Fig. 21. Convergence speed box plots of EAs with regular graphs are also shown for comparison. The first conclusion is that cEAs scale worse when structured by random graphs. This is particularly evident for cEAs with $k = 2$, but random graphs with higher k also perform worse than regular graphs on larger onemax problems. When compared to cEAs on random graphs, panmictic EAs are definitely the best choice for solving onemax problems.

The results attained by the algorithms on MMDP problems are in Fig. 22. Contrary to onemax problems, cEA improve their convergence speed if structured by random graphs. The exception are $k = 2$ cEAs. This is probably because the random graph algorithm does not guarantee

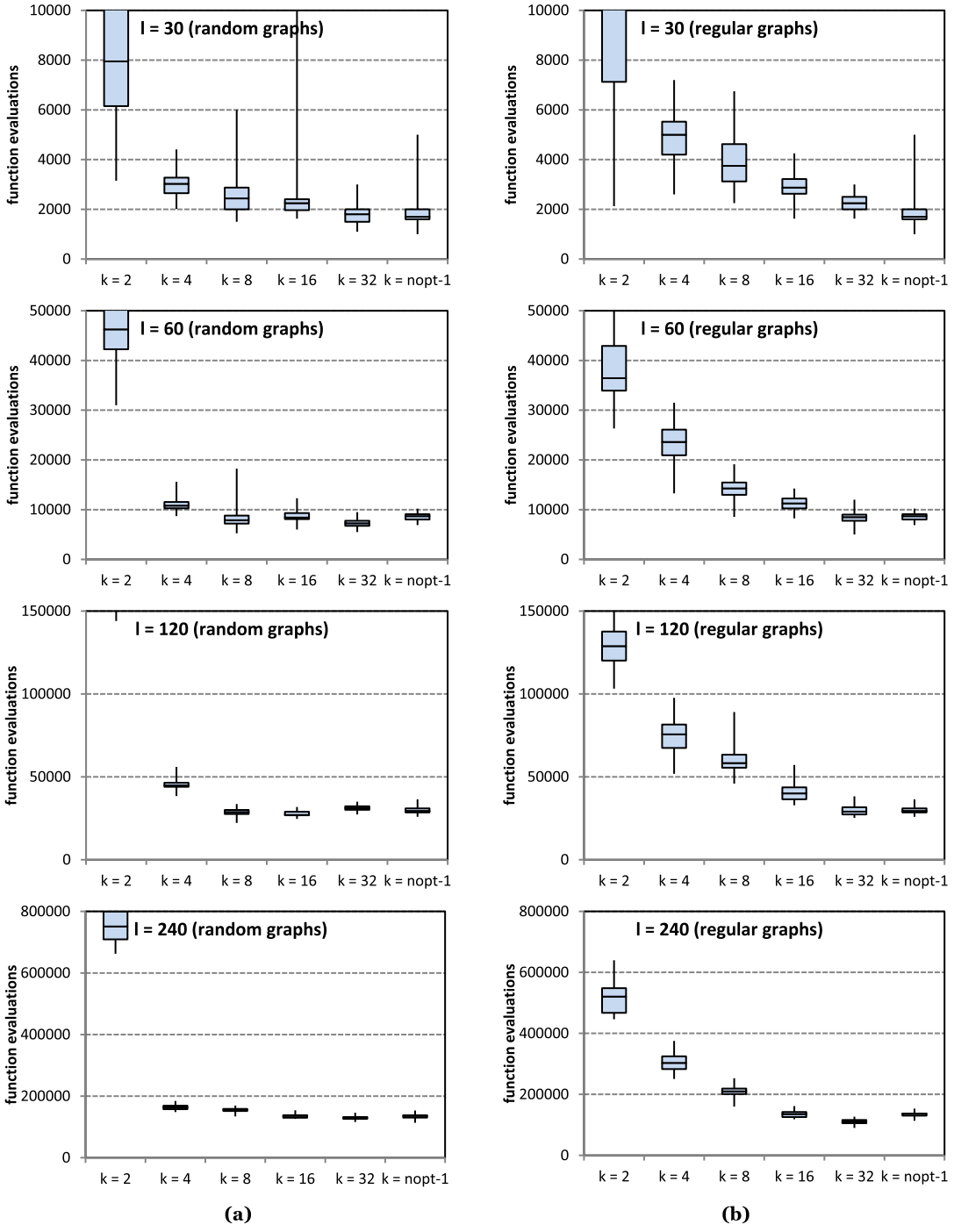

Fig. 22. MMDP. EAs with $p_m = 0$. Convergence speed box plots: (a) random graphs; b) regular graphs. ($k = 2$ configurations may fall out of the range for the sake of visualization.)

connected graphs and with such low k value it is highly probable that most of the graphs are disconnected and the population divided into subpopulations.

4.8. Event takeover values

In Ref. [23], Whitacre et al. introduced and calculated ETVs for several EA designs and fitness landscapes and concluded that panmictic populations generate ETV distributions (measuring the number of occurrences of each ETV value) that fit a power law. In addition, they claim that the distribution is not sensitive to the design choices (selection scheme, population size, etc): it consistently fits a power law. The only exception is the population structure since cEAs cause power law deviations for large ETV sizes and the effect is stronger when k is low. This means that cEAs prevent individuals to spread their genetic material through the entire population. This conclusion is consistent with theoretical and empirical results on cEAs takeover times.

In this paper, cEAs and panmictic EAs were tested and compared on five different problems with four instances each, thus making 20 different fitness landscapes. The experiments and statistical analysis of ETVs do not confirm the presence of power laws in the collected data.

Fig. 23 shows the log-log of the ETV complementary cumulative

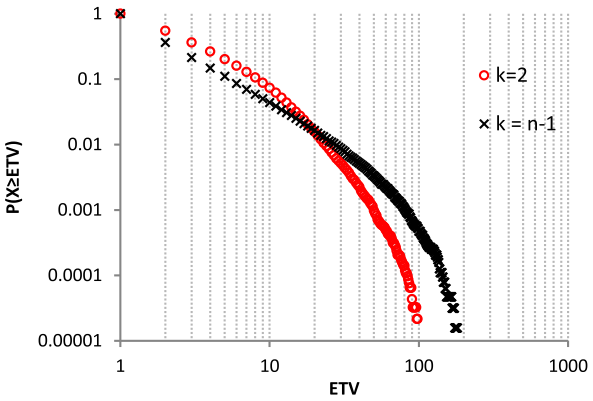

Fig. 23. MMDP: $l = 60$; $n = 200$; $p_m = 1/l$. ETV CCDF of panmictic EA ($k = n - 1$) and cEAs with $k = 2$.

distributions function (CCDF) of a cEA ($k = 2$) and a panmictic EA in an instance of the MMDP. Configurations with population size $n = 200$ optimize convergence speed (both algorithms attain the global optimum in every run) in this particular instance of MMDP.

Distributions are clearly not linear through the whole range of ETVs. For the panmictic EA, a power law model was computed and the maximum likelihood used to estimate the exponent, which is $\alpha = 2.14$, below the values in Ref. [23]. Then, goodness of fit was used to estimate where the scaling region begins. The power law hypothesis was tested using a goodness-of-fit test. Finally, a Vuong's test comparing power law and log-normal models concluded that a log-normal distribution could also produce the ETVs. As such, there is no evidence of power law in the data. The statistical tests were made with the *poweRlaw* R package [11], following the guidelines in Ref. [6].

The reason for the contradictions between the conclusions in Ref. [23] and our analysis of ETVs may lie in differences in methodology. In the early 2000s, least-squares fitting to log-log plots was still a common method to estimate the parameters of power-law distribution, and although in Ref. [23] there is not enough information on the statistical approach to ETV distributions, we may assume that the fitting was done by linear regression. In fact, the authors in Ref. [23] plot ETV distributions functions instead of CCDFs.

In Ref. [6], Clauset et al. identified the methodological error and proposed new statistical techniques for parameter estimation for power-law data. They applied their method to twenty-four real-world data sets and demonstrate that in the majority of the data set log-normal or stretched exponential distributions fit the data with equal probability as power law. Furthermore, in some cases the power law can simply be ruled out. Therefore, it is not surprising that, when analyzed with the tools in Ref. [6], ETV distributions power law fit are not significantly better than a log-normal fit. However, as seen in Fig. 23 (and the outcome is similar with other fitness landscapes), structuring the population has a clear effect on the distribution, namely for large sizes. Largest ETVs of cEAs are well below population size n, and in general they are bounded by $[1, n/2]$, while panmictic ETVs sometimes cover the whole possible range of values $[1, n]$. As $k$increases, the upper bound also increases, and for $k = 32$ its value is closer to n. Please note that these behavioural patterns were observed in every fitness landscape and also with different genetic operators.

From the large amount of data gathered with the tests, it is possible to identify other behavioural patterns. In general, increasing population size has strong effects on the distribution. Fig. 24 shows the CCDFs of panmictic EAs with different population size n on the same instance of MMDP. Every configuration finds the best solution in every run and the EA with $n = 200$ requires less evaluations to find the optimum (considering median values): $n = 200$ is therefore the panmictic EA's optimal population size (amongst the n values that were tested) for this instance of the problem.

Please note that for $n = 50$ and $n = 100$, the distributions are characterized by high probability values when the ETVs are close to the upper bound $ETV_{max} = n$: i.e, it is highly probable that descendants of a good solution takeover the entire population. With optimal size ($n = n_{opt} = 200$), that probability decreases significantly, and with above optimal population the distributions do not suffer much deviations from the $n = 200$ distribution. In fact, the ETVs of $n = 400$ and $n = 800$ populations are bounded by $[1, n_{opt}]$. These results show that ETVs distributions strongly depend on population size and that they can detect below-optimal population size.

In most of the cases studied for this paper, ETVs are bounded by $[1, n_{opt}]$. Fig. 25 shows the ETV of the panmictic EA for 2-trap with l = 200. ETVs do not grow higher than 400 which is precisely the optimal population size found empirically.

This pattern was observed for different crossover types and also in selecto-recombinative EAs. Fig. 26 shows the ETV of the selecto-recombinative panmictic EA on MMDP problems with l = 60. ETVs are clearly bounded by 300 and the optimal population size found in this case using the bisection method is 275. Finally, Fig. 27 shows the ETV of the panmictic EA with 1-point crossover when solving the MMDP problem with $l = 30$. The ETV upper bound is approximately 100 while the optimal population size is $n = 75$. These are quite interesting and promising results, suggesting that ETV may be used for population sizing of EAs.

The only exception to the described patterns was the onemax problem. This is probably due to the specific characteristics of the fitness landscape, but further investigation is needed in order to understand what are the onemax properties that cause ETVs distribution patterns to

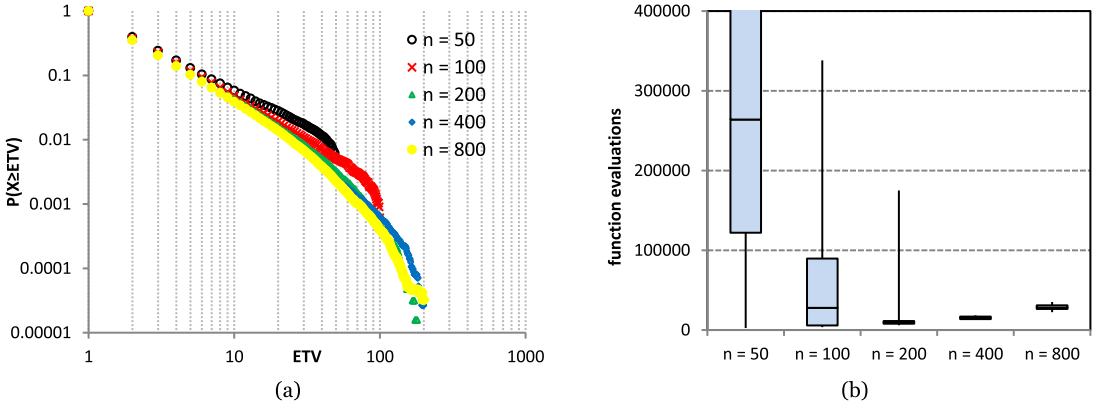

Fig. 24. MMDP: $l = 60$; $p_m = 1/l$. a) Panmictic EAs ETVs CCDFs with different population size; b) convergence speed box plots. Optimal population as determined empirically is $n = 200$.

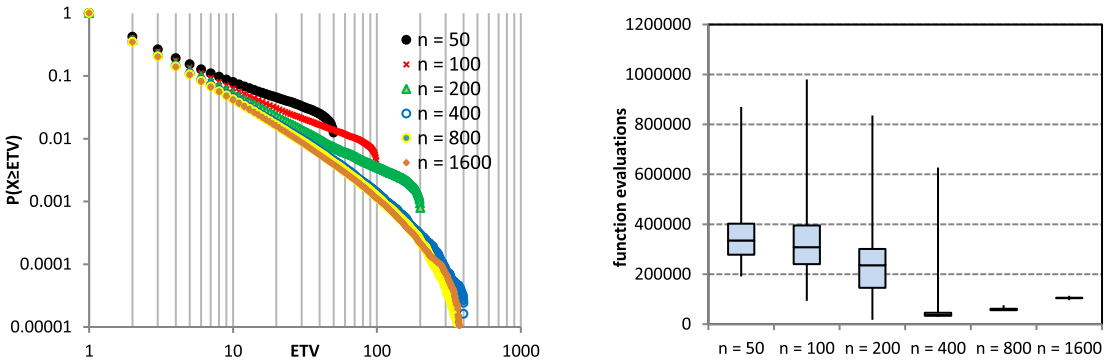

Fig. 25. 2-trap: $l = 200$; $p_m = 1/l$. a) Panmictic EAs ETVs CCDFs with different population size; b) convergence speed box plots. Optimal population as determined empirically is $n = 200$.

be so different, for instance, from those generated by EAs solving 2-traps (which are also non-deceptive fitness landscapes like onemax).

5. Conclusions

With the purpose of investigating the effects of structuring EA populations in the optimal population size and, and consequently, in the dynamics and performance of the algorithms, cEAs and panmictic EAs were tested with several fitness landscapes and parameter settings. Population sizing for selecto-recombinative EAs was performed with the bisection method while mutation-EAs population size was determined empirically with sensitivity tests.

As predicted by previous theoretical and empirical studies, the results reported in this paper show that cEAs require smaller populations to converge consistently to the global optima of different fitness landscapes. However, if the population size is properly set, panmictic populations perform equivalently or better than cEAs. The results stress the importance of tuning population size and demonstrate that if the size is not properly set the conclusions on the performance of the algorithms, namely when comparisons are at the center of the investigations, will probably be misleading. We therefore suggest using bisection to determine a population upper bound and then fine tune the population size of the mutation-EA with sensitivity tests. Another alternative (which, however, requires further research in order to confirm its robustness) is to determine the ETVs distribution of the algorithm in a particular fitness landscape and then use ETV_{max} as an upper bound for the population size. Results suggest that this procedure is effective for both selecto-recobinative EAs and mutation-EAs, and gives good approximations of the optimal population size for different crossover operators. On the other hand, the test problems use binary representation of the variables. Therefore conclusions are valid for binary EAs.

A statistical analysis of several ETV distributions generated by panmictic populations on several fitness landscapes shows that the presence of power law distributions in data cannot be claimed. These findings, however, do not diminish the importance of ETVs in the analysis of EAs dynamic behavior. The results confirm that cEAs generate deviations from the panmictic heavy-tailed distributions, reducing ETVs upper bounds, which means that they indeed restrict descendants of good solutions from dominating the entire population.

In the future, other types of graphs (random and dynamic) will be tested and compared to regular graphs. The hypothesis of using ETV as a population sizing method will be exhaustively scrutinized. Finally, since power law exponents cannot be used to assess the effects of design choices in ETVs, the goal now is to devise other statistical measures to help investigate how population structure, population size and other

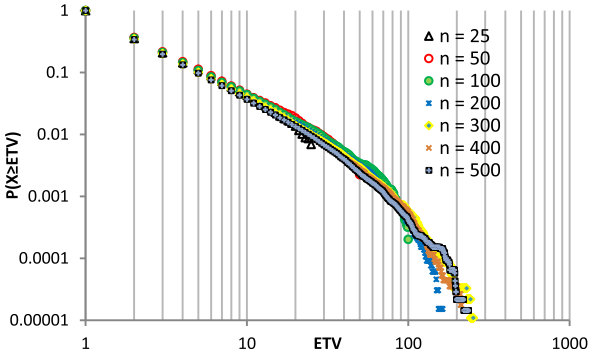

Fig. 26. MMDP with $l = 60$. Selecto-recombinative panmictic EAs ETVs CCDFs with different population size. Optimal population size as determined with bisection is $n = 275$.

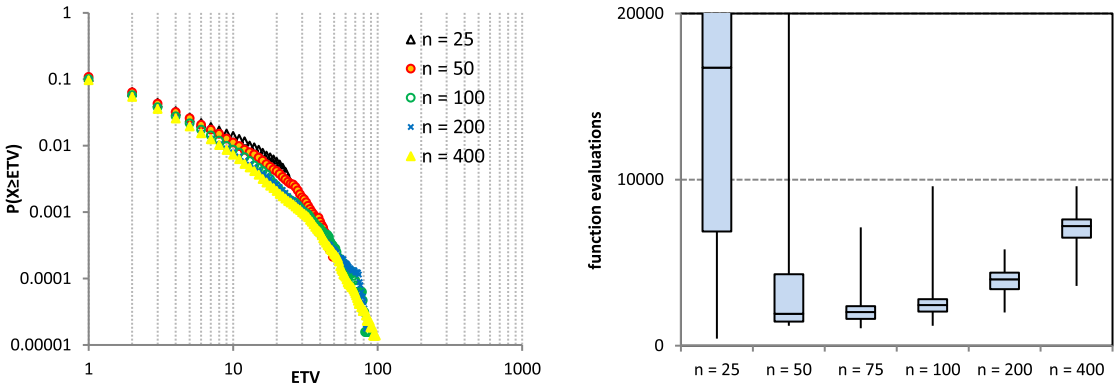

Fig. 27. MMDP with $l = 30$; $p_m = 1/l$ and 1-point crossover: a) ETVs CCDFs with different population size; b) convergence speed box plots.

parameters, affect genealogical dynamics and ETVs, and how those dynamics can be used to understand the mechanisms behind efficient EAs and population sizing.

Declaration of competing interest

☒ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Carlos M. Fernandes: Conceptualization, Methodology, Software, Writing - original draft. **Nuno Fachada:** Software, Writing - review & editing. **Juan L.J. Laredo:** Methodology, Writing - review & editing. **J.J. Merelo:** Supervision, Writing - review & editing. **Agostinho C. Rosa:** Supervision.

Acknowledgments

This work was supported by Fundação para a Ciência e Tecnologia projects UIDB/50009/2020 (LARSyS) and UIDB/05380/2020 (HEI-Lab), EPHEMECH (TIN2014-56494-C4-3-P, Spanish Ministry of Economy and Competitivity), PROY-PP2015-06 (Plan Propio 2015 UGR), project CEI2015-MP-V17 of the Microprojects program 2015 from CEI BioTIC Granada.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.swevo.2020.100721.

References

[1] A. Abdelhafez, E. Alba, G. Luque, Performance analysis of synchronous and asynchronous distributed genetic algorithms on multiprocessors, Swarm Evol. Comput. 49 (2019) 147–157.
[2] E. Alba, M. Tomassini, Parallelism and evolutionary algorithms, IEEE Trans. Evol. Comput. 6 (5) (2002) 443–462.
[3] E. Alba, B. Dorronsoro, The exploration/exploitation tradeoff in dynamic cellular genetic algorithms, IEEE Trans. Evol. Comput. 9 (2005) 126–142.
[4] T. Bäck, Evolutionary Algorithms in Theory and Practice, Oxford University Press, Oxford, 1996.
[5] E. Cantú-Paz, Migration policies, selection pressure, and parallel EAs, J. Heuristics 7 (4) (2001) 311–334.
[6] A. Clauset, C.R. Shalizi, M.E.J. Newman, Power-law distributions in empirical data, SIAM Rev. 51 (4) (2009) 661–703.
[7] C.M. Fernandes, J.L.J. Laredo, J.J. Merelo, C. Cotta, A.C. Rosa, Dynamic and partially connected ring topologies for evolutionary algorithms with structured populations, EvoAppl. 2014: Appl. Evol. Comput. (2014) 665–677.
[8] C.M. Fernandes, L.J.L. Laredo, A.C. Rosa, Spatially structured evolutionary algorithms: graph degree, population size and convergence speed, in: 11th International Symposium on Intelligent Distributed Computing, 2017.
[9] M. Giacobini, M. Tomassini, A. Tettamanzi, Takeover time curves in random and small-world structured populations, in: Proc. Of the 7th GECCO, 2005, pp. 1333–1340.
[10] M. Giacobini, M. Tomassini, A. Tettamanzi, E. Alba, Selection intensity in cellular evolutionary algorithms for regular lattices, IEEE Trans. Evol. Comput. 9 (2005) 489–505.
[11] C.S. Gillespie, Fitting heavy tailed distributions: the poweRlaw package, J. Stat. Software 64 (2) (2015) 1–16.
[12] Y.-J. Gong, W.-N. Chen, Z.-H. Zhan, J. Zhang, Y. Li, Q. Zhang, et al., Distributed evolutionary algorithms and their models: a survey of the state-of-the-art, Appl. Soft Comput. 34 (2015) 286–300.
[13] J.L.J. Laredo, P. Bouvry, D.L. González, F. Fernandéz de la Vega, M.G. Arenas, J.J. Merelo, C.M. Fernandes, Designing robust volunteer-based evolutionary algorithms, Genet. Program. Evolvable Mach. 15 (3) (2014) 221–244.
[14] G. Luque, E. Alba, B. Dorronsoro, An asynchronous parallel implementation of a cellular genetic algorithm for combinatorial optimization, Genetic and Evolutionary Computation Conference, GECCO 2009, in: Proceedings, Montreal, Québec, Canada, July 8-12, 2009.
[15] G. Luque, E. Alba, Parallel Genetic Algorithms: Theory and Real World Applications, Springer, 2011.
[16] J.L. Payne, M.J. Eppstein, Emergent mating topologies in spatially structured genetic algorithms, in: Proc. 8th GECCO, 2006, pp. 207–214.
[17] A. Réka, A.-L. Barabási, Statistical mechanics of complex networks, Rev. Mod. Phys. 74 (2000) 47–94.
[18] J. Sarma, K. De Jong, An analysis of the effect of the neighborhood size and shape on local selection algorithms, in: Proc. Int. Conf. Parallel Prob. Solving from Nature IV, LNCS 1141, Springer-Verlag, 1996, pp. 236–244.
[19] K. Sastry, Evaluation-relaxation Schemes for Genetic and Evolutionary Algorithms, MSc Thesis, University of Illinois, Urbana, IL, USA, 2001.
[20] D. Thierens, Scalability problems of simple GAs, Evol. Comput. 7 (4) (1999) 331–352.
[21] M. Tomassini, Spatially Structured Evolutionary Algorithms, Springer, Heidelberg, 2005.
[22] J.M. Whitacre, R.A. Sarker, Q. Pham, The self-organization of interaction networks for nature-inspired optimization, IEEE Trans. Evol. Comput. 12 (2008) 220–230.
[23] J.M. Whitacre, R.A. Sarker, Q. Pham, Making and breaking power laws in evolutionary algorithms, Memet. Comput. 1 (2) (2009) 125–137.
[24] K.M. Wittkowski, Friedman-type statistics and consistent multiple comparisons for unbalanced designs with missing data, J. Am. Stat. Assoc. 83 (404) (1988) 1163–1170.