Comparison of the tournament-based convection selection with the island model in evolutionary algorithms

Maciej Komosinski Konrad Miazga

Institute of Computing Science
Poznan University of Technology
Piotrowo 2, 60-965 Poznan, Poland

Abstract

Convection selection is an approach to multipopulational evolutionary algorithms where solutions are assigned to subpopulations based on their fitness values. Although it is known that convection selection can allow the algorithm to find better solutions than it would be possible with a standard single population, the convection approach was not yet compared to other, commonly used architectures of multipopulational evolutionary algorithms, such as the island model. In this paper we describe results of experiments which facilitate such a comparison, including extensive multi-parameter analyses. We show that approaches based on convection selection can obtain better results than the island model, especially for difficult optimization problems such as those existing in the area of evolutionary design. We also introduce and test a generalization of the convection selection which allows for adjustable overlapping of fitness ranges of subpopulations; the amount of overlapping influences the exploration vs. exploitation balance.

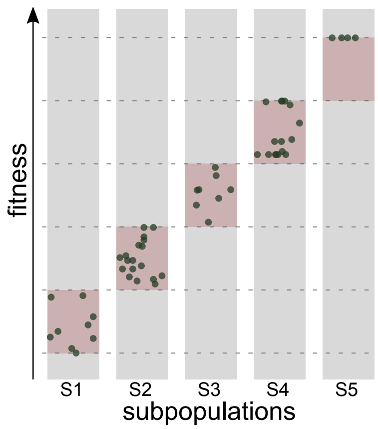
1 Introduction

In our recent paper [8] we have investigated the method of splitting the population into subpopulations based on the fitness values of solutions from that population; this idea was first introduced in [13]. This population-splitting scheme was named the *convection distribution* when the subpopulations were distributed (or the *convection selection* in the case of single-threaded algorithms) because it facilitates continuous evolutionary progress just like a convection current or a conveyor belt: each subpopulation always tries to independently improve genotypes of a specific fitness range which ensures more fitness diversity and avoids the domination of (and the convergence towards) the current globally best genotypes [4]. When the distribution of fitness values in subpopulations is inspected and visualized in time, occasional, short ascending trends (convections) are visible in the entire range of fitness values [13, see Fig. 9].

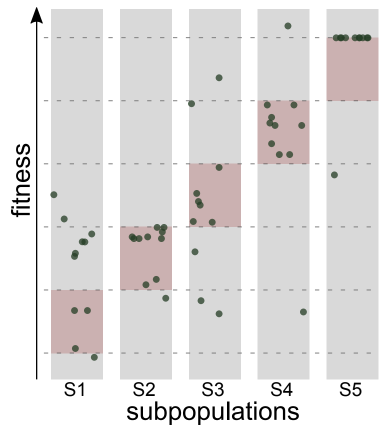
Convection selection techniques may be perceived as super- (or meta-) selection techniques, because they determine which individual should be assigned to which subpopulation, yet within these subpopulations traditional selection schemes are still employed. Therefore, convection selection can be combined with any traditional selection method, resulting in convection tournament selection, convection roulette selection, etc. In this work we will discuss one-level convection selection (i.e., a population divided into sub-populations), but this technique can be employed on multiple levels with subpopulations recursively embedded in each other.

Although the original paper [13] focused on a parallel implementation of the convection distribution and therefore it didn't compare the results obtained with the convection approaches with the results from single-population evolutionary algorithms, that comparison was provided

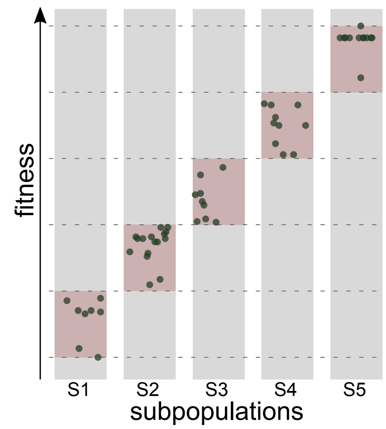
This paper appeared in *Journal of Computational Science*, 2018.
http://dx.doi.org/10.1016/j.jocs.2018.10.001



(a) The state of the population right after some n-th merge event.



(b) The state of the population after a period of independent evolution within each subpopulation, right before the $(n+1)$-th merge event.



(c) The state of the population right after the $(n+1)$-th merge event.

Figure 1: An illustration of changes in the population structure in time for the *EqualWidth* convection approach. Each of the five subpopulations is displayed in a separate column, starting from the worst subpopulation (on the left) to the best subpopulation (on the right). Each dot represents one solution, and its vertical position represents its quality. Although the fitness ranges of subpopulations are always disjoint immediately after a merge event, they may start to overlap again during periods of independent evolution. Animation available at `http://www.cs.put.poznan.pl/mkomosinski/convection.html`.

later [8] and showed that convection selection is superior not only to a random splitting of the population into subpopulations, but also to a classical, single-population evolutionary algorithm. Moreover, these experiments demonstrated that the crossover operator is not necessary for the success of the convection approach.

The superiority of the convection selection with respect to a single-population architecture is not a surprise, as it is known that given the same computational cost, multipopulational evolutionary algorithms [21, 18, 15] can sometimes yield better results in optimization tasks than standard sequential evolutionary algorithms. This is mostly due to the exchange of individuals between independent subpopulations which can increase the diversity within these subpopulations and help solutions escape from local optima. Local exchange of individuals and the use of multiple subpopulations leads to increased exploration of the search space, which is often desirable [18].

One of the most popular approaches to parallel evolutionary computations is the *island model* [16] which is based on the idea of punctuated equilibria [5]. In the island model, the algorithm maintains a number of independently evolving subpopulations (sometimes called *islands*), which are connected according to some specific topology. Once every so often (with a regular frequency or once the subpopulations start to converge), each subpopulation migrates (copies) a subset of its solutions (often selected with regard to their quality or uniqueness) to the "neighboring" subpopulations. To keep the size of each subpopulation constant, these immigrant solutions replace some solutions in the target subpopulation (often the ones with low quality or high similarity to others). Although due to the separation of subpopulations each of them loses diversity and converges toward some region of a solution space, systematic migrations prevent the convergence and help reignite evolution and escape from local optima.

While in the island model the exchange of solutions is limited in its scope and subpopulations are interchangeable, in convection approaches each subpopulation attends to solutions of a different fitness range, and the process of merging and dividing populations can have a high impact on the structure and the content of each subpopulation, as shown in Fig. 1. In this paper we compare the convection selection schemes to the island model in order to discover differences in performance between the convection approach and the algorithms commonly used in parallel evolutionary computation. We also examine the effects of a generalization of the convection selection which allows for overlapping fitness ranges of subpopulations.

2 Methods

All the experiments described in this paper were performed using Framsticks software [14, 12]. Framsticks allows to evolve bodies and brains of 3D designs (agents) towards a goal specified by some fitness function. This area of application of evolutionary algorithms benefits the most from selection schemes that improve the performance yet are still computationally inexpensive. This is because optimization tasks in evolutionary design are extremely difficult and solutions are very complex due to sophisticated genotype-to-phenotype mappings, so calculating complex properties of such solutions or estimating their similarity [7] is usually very costly and should be avoided if possible [13]. Additionally, Framsticks has a dedicated *fn* genetic representation (a vector of constrained real values) and genetic operators for numerical optimization with source available [10].

2.1 Fitness functions

Two kinds of fitness functions were used as test functions in the experiments.

1. Mathematical benchmark functions. For these optimization problems, the crossover operator was implemented as a weighted sum of two real-valued vectors, with one of the parents assigned a random weight $w_1 \in [0, 1]$ and the other parent assigned the weight $w_2 = 1 - w_1$. Mutations were implemented as adding a random, Gaussian-distributed value with zero mean to each of the variables in the parent solution. The standard deviation of these values was constant in time and was defined separately for each optimization problem. Any variable x_i in the solution vector exceeding its allowed range $[x_{i,min}, x_{i,max}]$ was reflected inwards as follows [1, 2, 10]:

$$x_i = \begin{cases} x_{i,min} + (x_{i,min} - x_i) & \text{if } x_{i,min} - x_i > 0, \\ x_{i,max} - (x_i - x_{i,max}) & \text{if } x_i - x_{i,max} > 0, \\ x_i & \text{otherwise.} \end{cases}$$

   Should such a single reflection be insufficient to make the value fall into the allowed range (which might happen for narrow intervals and large standard deviations), the value was wrapped around using the modulo operation. The fitness functions used in our experiments (formulated as maximization problems) were the 10-dimensional versions of F_7 (Shifted Rotated Griewank's Function without Bounds), F_{10} (Shifted Rotated Rastrigin's Function), and F_{11} (Shifted Rotated Weierstrass Function) from the CEC 2005 set of mathematical benchmark problems [19]. The standard deviation for mutations applied to each variable in the vector was 10.0 (for F_7), 0.3 (for F_{10}), and 0.05 (for F_{11}), normalized by diving by the square root of the problem dimensionality, i.e., by $\sqrt{10}$.

2. Framsticks-based evolutionary design [6] problems. We have used two fitness functions that differ in the difficulty of optimization: *velocity* (a harder problem) and *height* (an easier problem). The *velocity* criterion is used to evolve individuals that move fast on land (so that body and brain are co-evolved and must cooperate), whereas *height* is used to evolve static structures (their neural network and the corresponding genetic mutations are disabled) with the center of mass as elevated as possible.

   As in previous experiments [13, 8], the default *f1* genetic encoding was employed [9, 12]. This encoding is a direct mapping between symbols and parts of a 3D structure: 'X'

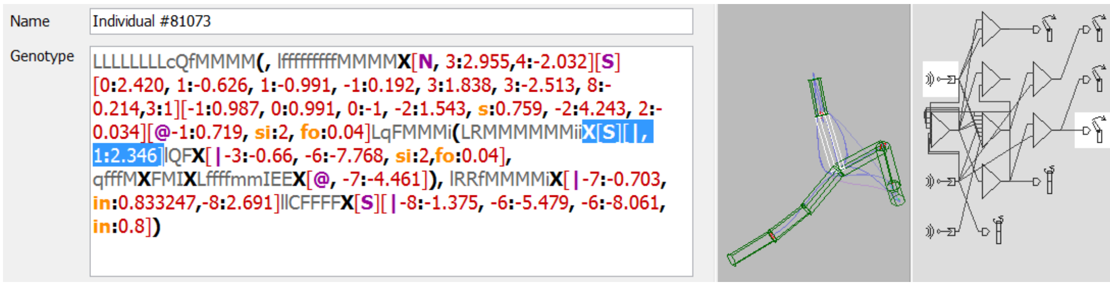


Figure 2: A sample Framsticks genotype in the *f1* representation encoding a three-dimensional structure ("body") with a coupled control system – an artificial neural network ("brain"). A part of the genotype is selected, and the corresponding parts of "body" and "brain" are highlighted in white [11, 14].

represents a rod (a stick), parentheses encode branches in the structure, and additional characters influence properties like length or rotation. Neurons are described in square brackets and index numbers in their connections are relative, so the information about connections is local and persists when a part of a genotype is cut out. The encoding is able to represent tree-like 3D body structures and neural networks of arbitrary topology (Fig. 2). Mutations modify individual aspects of the agent by adding or removing parentheses in random locations in the genotype, by adding and removing random symbols that affect the structure, by adding and removing neurons and connections, and by adding random Gaussian-distributed values to neural weights.

2.2 Evolutionary architecture

For mathematical fitness functions, evolution was started from the population filled with random solutions (drawn homogeneously from the allowed range of values). For both evolutionary design fitness functions, evolution was started from the population filled with simplest individuals (i.e., 'X' in the *f1* encoding) subjected to five random mutations to provide some initial diversity. In all the experiments, the steady-state (also known as "incremental") evolutionary algorithm [20] was used with 50% crossover and 50% mutation probabilities.

2.2.1 Convection selection

In the convection selection schemes, solutions (genotypes, individuals) are first sorted according to their fitness. Then each subpopulation receives a subset of solutions that fall within a range of fitness values associated with that subpopulation. In the experiments reported in this paper, we consider two methods of determining fitness ranges. In the first method denoted *EqualWidth* (Fig. 3c), the entire fitness range is divided into intervals of equal size; there are as many intervals as there are subpopulations. If there are no solutions in some fitness range, the corresponding subpopulation receives all solutions from the nearest non-empty lower fitness interval. In the second method denoted *EqualNumber* (Fig. 3d), once the solutions are sorted according to their fitness, they are distributed into as many sets as there are subpopulations so that each subpopulation receives the same number of solutions.

In both of the tested convection-based approaches (i.e., *EqualWidth* and *EqualNumber*), the underlying traditional selection mechanism was the tournament selection. The logic of these approaches was implemented as follows. Every $R \cdot N$ evaluations (where N is the size of the entire population, and R is the migration period scaling factor which defines how frequently subpopulations should merge, e.g. $R = 2$ means that between any two merging events $2N$ new solutions will be evaluated), M subpopulations are merged and then all individuals from the complete (merged) population are again split into M subpopulations according to the applied

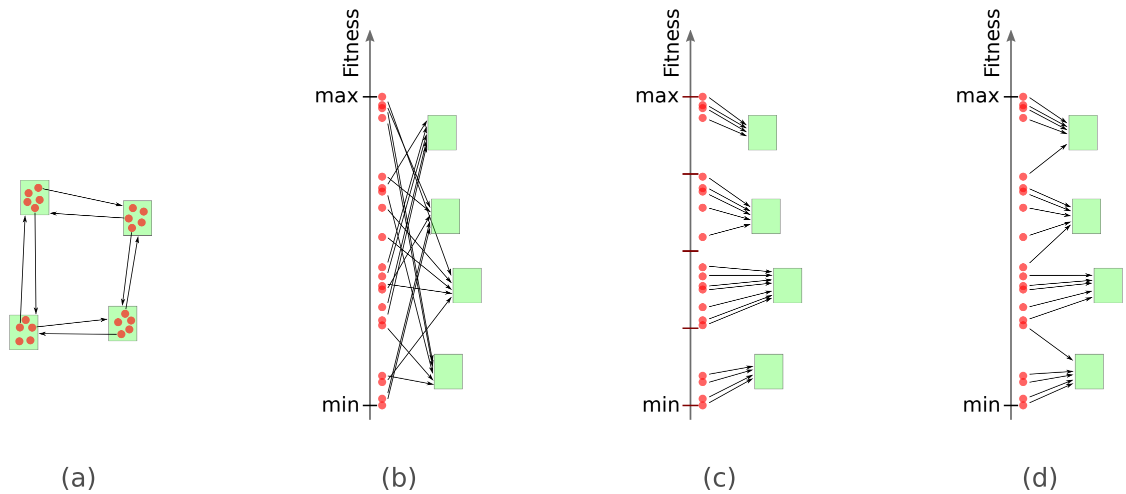


Figure 3: An illustration of four ways of assigning solutions to subpopulations. Out of these four, in this paper we examine (a), (c) and (d); we do not examine (b) as it has proved to be inferior [8]. The fitness of 20 solutions is shown as red circles, and 4 subpopulations are depicted as green boxes. (a) The island model (the solutions with no arrows do not change their assignment during a migration, and the assignment to subpopulations does not depend on the fitness of solutions). (b) Random assignment of solutions to subpopulations. (c) Convection selection with fitness intervals of equal width (*EqualWidth*). (d) Convection selection with fitness intervals yielding equal number of solutions (*EqualNumber*).

super-selection scheme (*EqualWidth* or *EqualNumber*). After that, the algorithm cycles through all subpopulations in sequence so that each subpopulation in turn becomes the current one.

Every iteration, the steady-state evolutionary algorithm selects parent solutions from the current subpopulation (using tournament selection with the tournament of size t), creates a new child solution by applying a random genetic operator (mutation or crossover) to the parent solution(s), and adds the newly created offspring to the current subpopulation. Once this new individual has been evaluated, the negative selection process removes randomly one individual from a random subpopulation, so the size of the complete population remains constant. Then, the next subpopulation in sequence becomes the current one. After all subpopulations have been processed, the cycle starts again unless it is time to repeat the merge-and-divide procedure.

2.2.2 The island model

In the implementation of the island model (denoted further as *IslandModel*) used in the experiments reported here, the full population of solutions is divided into M equally sized subpopulations (islands). The topology determines the neighbor subpopulations for each subpopulation. We used a ring topology, which means that each subpopulation S is assigned a unique number from the range $[1; M]$ where M is the number of subpopulations, and the neighborhood relation depicted as the $\rightleftharpoons$ symbol is $S_1 \rightleftharpoons S_2 \rightleftharpoons S_3 \rightleftharpoons ... \rightleftharpoons S_M \rightleftharpoons S_1$. The ring topology is one of the most popular topologies in the island model, whose sparse connectivity delays the convergence of the complete population. The subpopulations evolve separately (the negative selection process always removes one random solution from the current, not from a random subpopulation like it was implemented in the convection approaches) for the total of $R \cdot N$ evaluations ($R \cdot \frac{N}{M}$ per subpopulation, where $\frac{N}{M}$ is the size of every subpopulation). Then, each subpopulation migrates 10% of its randomly selected solutions to each of its neighbors (each neighbor receives a different, random set of migrant solutions). The proportion of 10% is more than necessary to prevent convergence within subpopulations [17], however it is the lowest proportion for which there is always at least one migrant solution per a pair of neighboring subpopulations. During every migration event, each subpopulation sends just as many emigrants to its neighbors as many immigrants it takes from them, so the sizes of subpopulations remain unchanged. Although we select the migrating solutions randomly with a uniform distribution, in practice these choices are usually made based on the quality of the solutions (i.e., bad solutions are replaced by the good neighbor solutions), which as a result leads to an increased selective pressure [3]. In the *IslandModel* experiments, migrant solutions are selected randomly because we intend to compare different approaches over very long periods of evolution, and therefore preventing premature convergence of the *IslandModel* algorithm is desirable (in our experiments we prefer to eventually obtain a very good solution over obtaining a reasonable solution quickly).

3 Experiments

There are three main goals of the experiments described in this paper. The ideas behind them and the parameters used are both described below.

3.1 A multi-parameter comparison of convection selection and the island model

The first goal is to provide a comparison between the quality of solutions obtained by the two convection approaches and by the island model. As each of these three approaches may need different parameter values to achieve their best performance (and these best parameter values may also differ between problems), to avoid any biases we have decided to conduct this comparison so that each approach is tested for a number of different optimization goals, for every combination of the parameter values described below. We record fitness value of the best solution in the population every 1000 evaluations (new solutions), so we can plot how the quality of the best solution was changing in time during the evolution. Then, the series of fitness values are averaged separately for each combination of optimization goals, parameter values, and approaches. Finally, for each combination of the approach and the fitness function, we create a new series composed of the best average fitness values in time, among all tested parametrizations. This means that for each of the approaches such a final series represents the highest average fitness that was obtained for any of the tested combinations of parameter values. This allows us to compare the best performance of different approaches in a parameter-agnostic manner.

In each of the evolutionary runs, $2 \cdot 10^6$ (for evolutionary design problems: *velocity* and *height*) or $5 \cdot 10^5$ (for mathematical problems: *Rastrigin*, *Griewank* and *Schwefel*) individuals were evaluated – so that the number of evaluations differed for different fitness functions, but not for different approaches. Three approaches were compared: *EqualWidth*, *EqualNumber* and *IslandModel*. For each of these approaches, all the combinations of the following sets of parameter values were tested: average subpopulation size $\frac{N}{M} \in \{10, 20, 50\}$ (the *average* size is mentioned, because the size of subpopulations in the *EqualWidth* approach cannot be directly controlled), the number of subpopulations $M \in \{5, 10, 25\}$, tournament size $t \in \{2, 5\}$, and the number of individual evaluations between merging the subpopulations (given as the multiple of the size of the full population) $R \in \{2, 10, 25\}$. For the *IslandModel* approach with the ring topology, the number of solutions migrating between the pairs of neighboring subpopulations was set to 10% of the size of a subpopulation.

Such a setup means that to obtain one result (i.e., the best fitness value from one evolutionary run) for each combination of parameter values, $3 \times 3 \times 2 \times 3 = 54$ independent evolutionary runs were needed per one tested approach (162 runs in total). Since the evolutionary process is non-deterministic, to obtain averages and standard deviations these runs were repeated 20 times for each combination of parameter values, which yielded 3240 independent evolutionary runs performed for each of the five fitness functions that were considered.

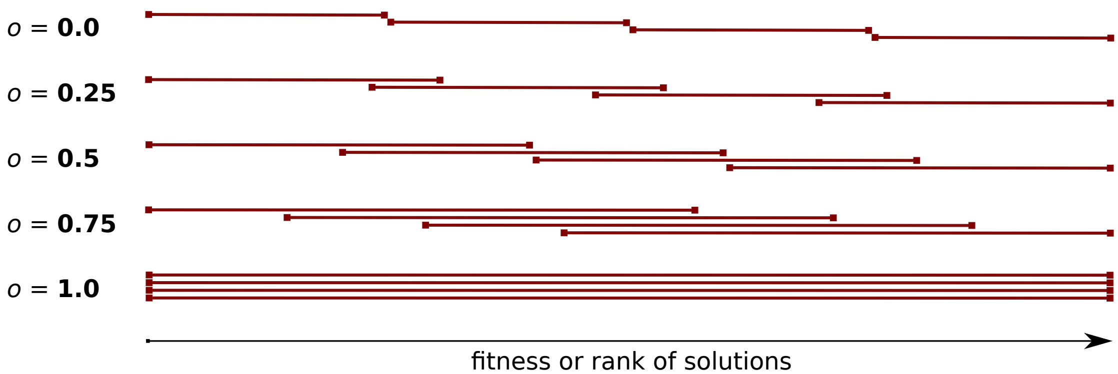


Figure 4: The effect of the parameter o (the amount of overlapping) on the fitness or rank ranges of different subpopulations in the convection approaches.

3.2 The influence of individual parameters on the quality of obtained solutions

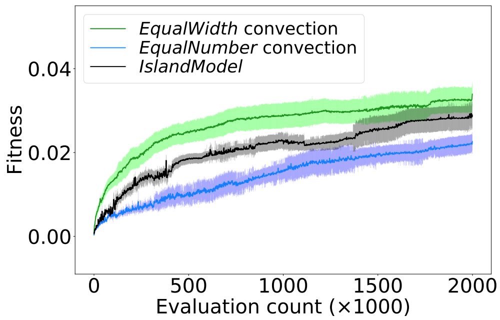
The second goal is to examine the way in which the performance of the tested approaches depends of the values of their parameters. Although a similar analysis was made before [8], there was no crossover operator used in the previous study, the number of evaluations per evolutionary run was half of the currently used, and there were no mathematical benchmark functions tested. This analysis will be based on the results of the experiments described in Sect. 3.1.

3.3 Overlapping ranges of fitness values in convection selection

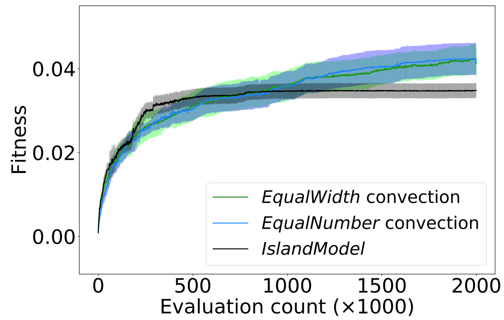
In this work we introduce and test a generalization of the convection approaches, where overlapping fitness ranges among subpopulations are allowed. By default, in convection approaches the ranges of fitness values assigned to subpopulations are fully disjoint (although they may start to overlap with each other during periods of independent evolution between the merge events). In this generalization, we introduce the parameter o which specifies the degree to which the fitness ranges (or rank ranges in the case of *EqualNumber*) of subpopulations should overlap. The effect of this parameter on the subpopulations is illustrated in Fig. 4; $o = 0$ means that there is no overlap (as would be the case without this generalization), whereas $o = 1$ means that fitness ranges of all subpopulations fully overlap (and so the fitness range of each of them is the same, which is similar to the *Random* approach tested in earlier experiments [8]). When the solutions are distributed among subpopulations and some solution lies in the overlapping range of fitness, it is assigned randomly to any of the subpopulations that share this fitness range. A low degree of overlap will lead to extensive exploration of the solution space at the expense of exploitation, while a high degree of overlap will be better for exploiting promising areas of the search space at the cost of lower exploration. We compared the final fitness values obtained for different fitness functions, for $o \in \{0.0, 0.25, 0.5, 0.75, 1.0\}$, $t \in \{2, 5\}$, $M = 10$, $N = 200$ and $R = 10$. Each combination of parameters was tested in 10 independent runs, for the total of 100 evolutionary runs. Due to high computational cost of all the experiments, this generalization was tested only in the *EqualWidth* approach.

4 Results

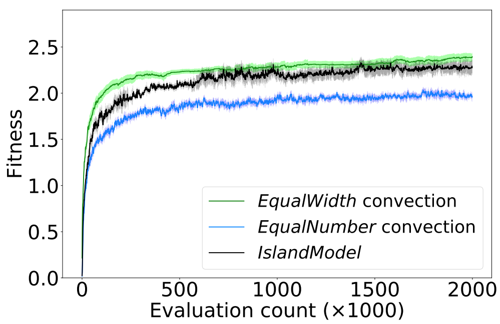
This section presents the results of the experiments and provides their analysis and interpretation.



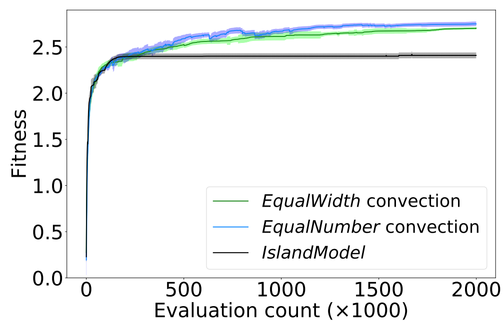
(a) *velocity*, $t = 2$



(b) *velocity*, $t = 5$



(c) *height*, $t = 2$



(d) *height*, $t = 5$

Figure 5: Comparison of the performance of convection approaches and the island model for the evolutionary design fitness functions. Each series consists of the high bound (i.e., best) of the average fitness value obtainable for a given selection scheme for all of the tested sets of parameter values. The band around each series represents a quarter of the standard deviation for that series (the entire standard deviation is not shown to avoid overlapping bands and improve the readability of the plots).

4.1 A multi-parameter comparison of convection selection and the island model

The comparison of the best average fitness value for any tested parametrization is split into two separate comparisons: one for a low selective pressure ($t = 2$), and one for a higher selective pressure ($t = 5$). Note that this split is only done here to have a better insight into how the selective pressure influences results yielded by the considered approaches; in other analyses reported in this paper the tournament size was treated just like the other examined parameters, and in the overall "best performance" analyses we considered the better result of the two tournament sizes.

The results for both evolutionary design fitness functions (*velocity* and *height*) are shown in Fig. 5. It can be observed that in a given time ($2 \cdot 10^6$ evaluations) all the compared approaches manage to find better results when the selective pressure is higher ($t = 5$). However, while the increased selective pressure leads to a quick convergence for the *IslandModel* approach (after which the improvement stops), this effect is not observed for the convection selection approaches. This is most likely due to the structure of the population in the convection-based approaches, which counteracts the potentially negative effect of the high selective pressure: poor solutions are spared by being assigned to subpopulations where they do not need to compete with the best solutions found so far, and therefore the complete population never truly converges to a single local optimum. Instead, the solutions that fall off the peak in the fitness landscape are given a chance to climb once again, possibly in the direction of another local optimum.

Because of the mechanism that allows for escaping from local optima that is inherent to convection approaches, in our experiments with the evolutionary design fitness functions these approaches always obtain better results than *IslandModel* for the higher selective pressure (Figs. 5b and 5d). However, in experiments with a low selective pressure where the populations are expected to be more diverse, the *IslandModel* can obtain better results than the *EqualNumber* approach. Nevertheless, even then *IslandModel* still looses (albeit only slightly) to *EqualWidth* which finds the highest fitness solutions for both functions in the low selective pressure environment.

The results of the performance comparison for mathematical benchmarks are shown in Fig. 6. For all three functions (F_7, F_{10}, F_{11}), every tested method tends to eventually converge to comparable results. However, for the F_7 function, the population converges more quickly for the island model than for the convection selection. In additional experiments performed with a more limited mutation operator (that only modified a single value in a vector instead of all values) and less randomized crossover, no approach turned out to be clearly better than any other approach; in some cases the convection selection managed to find better solutions compared to the island model. This suggests that the convection selection is better-suited for rugged, highly difficult fitness landscapes, while the island model may converge faster for relatively simpler problems.

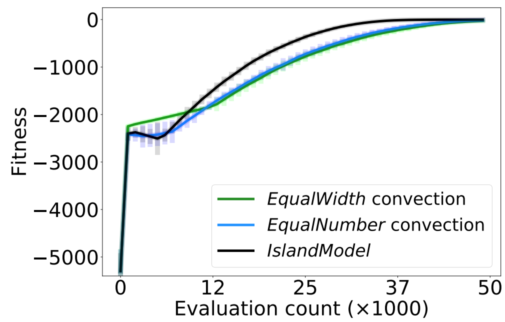
4.2 The influence of individual parameters on the quality of obtained solutions

The analysis of the influence of individual parameters on the quality of obtained solutions for the evolutionary design functions is presented in Fig. 7. For every combination of a fitness function and an approach, a gradient of improvement can be seen; the improvement is visible as circles getting filled when progressing in some direction in the space of parameters. Each of such gradients indicates how the values of the parameters should be changed in order to obtain better and better results. Despite slight differences between different combinations of the evolutionary design fitness functions (rows) and approaches (columns), improvement can be noticed in the direction of the higher number (M) of larger ($\frac{N}{M}$) subpopulations, with longer periods between merge events (R). Similar observations are true for the three considered mathematical benchmark functions, although for each of them the improvements can be noticed in different directions. The results of these experiments suggest that further extensions of the ranges of tested parameter values would be reasonable.

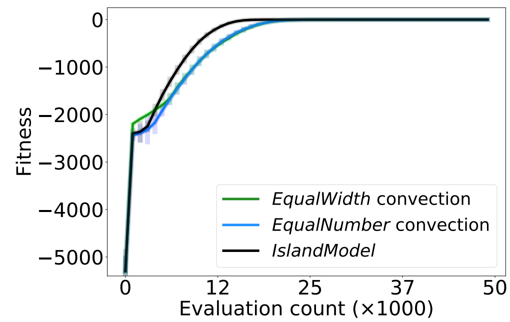
4.3 Overlapping ranges of fitness values in convection selection

The results of experiments for different amounts of the overlap parameter o are shown in Fig. 8. The value of $o = 0.0$ (no overlap) could be expected to be the most efficient case, because each subpopulation deals with a separate range of fitness. However, this specific case yielded best results only for the *velocity* fitness function with $t = 2$ (i.e., an extremely hard optimization problem with a slow convergence, high exploration and low exploitation caused by a low selection pressure). The other interesting case is $o = 1.0$ which, for the evolutionary design problems, always lead to extreme (worst or best) fitness values: in most cases these values were worse than they would be if there was any overlap present (Figs. 8a, 8b and 8d), but for some configurations they were in fact better (Fig. 8c and some mathematical functions). This may suggest that the optimization process is most sensitive to changes in the amount of overlapping in the range of the highest values of this parameter – in particular, near full overlapping for $o = 1.0$.

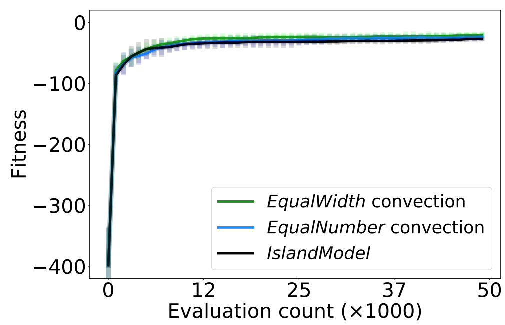
Comparing Figs. 5 and 8, it seems that the performance of the *EqualWidth* approach for $o = 1.0$ is analogous to the performance of the *IslandModel* approach – they both tend to converge prematurely for the same combinations of fitness functions and selective pressures. This similarity may be explained by the implementation of the *IslandModel* in which the migrating solutions



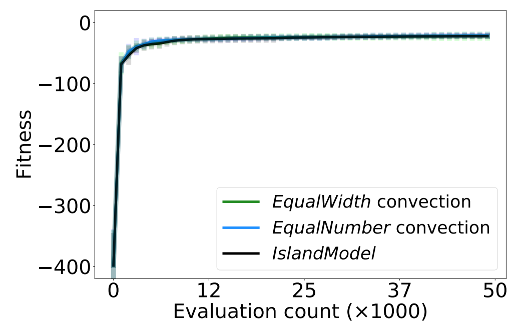
(a) CEC 2005 F_7 , $t = 2$



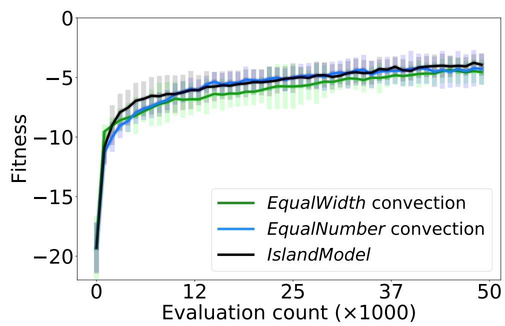
(b) CEC 2005 F_7 , $t = 5$



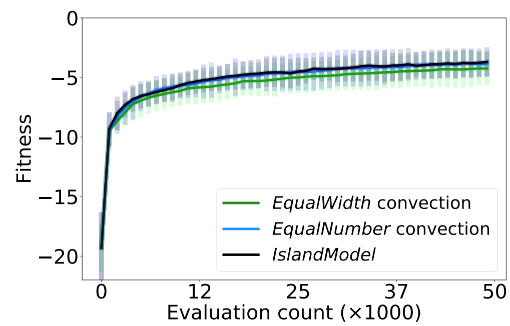
(c) CEC 2005 F_{10} , $t = 2$



(d) CEC 2005 F_{10} , $t = 5$

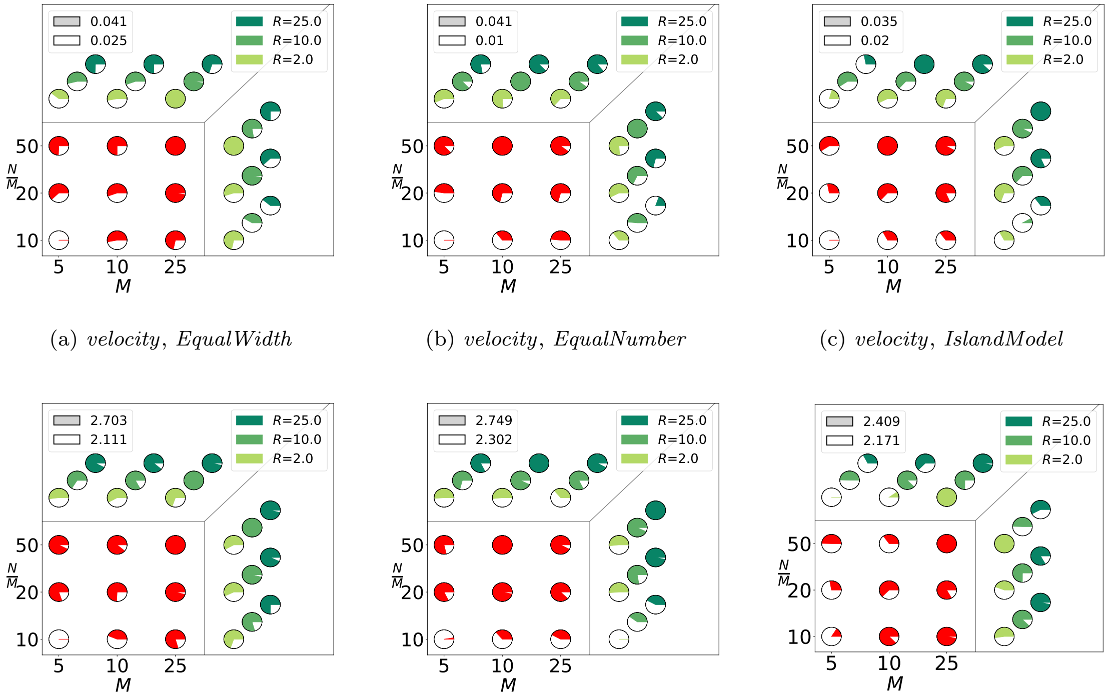


(e) CEC 2005 F_{11} , $t = 2$



(f) CEC 2005 F_{11} , $t = 5$

Figure 6: Comparison of the performance of convection approaches and the island model for the mathematical fitness functions. Each series consists of the high bound (i.e., best) of the average fitness value obtainable for a given selection scheme for all of the tested sets of parameter values. The band around each series represents one standard deviation for that series.



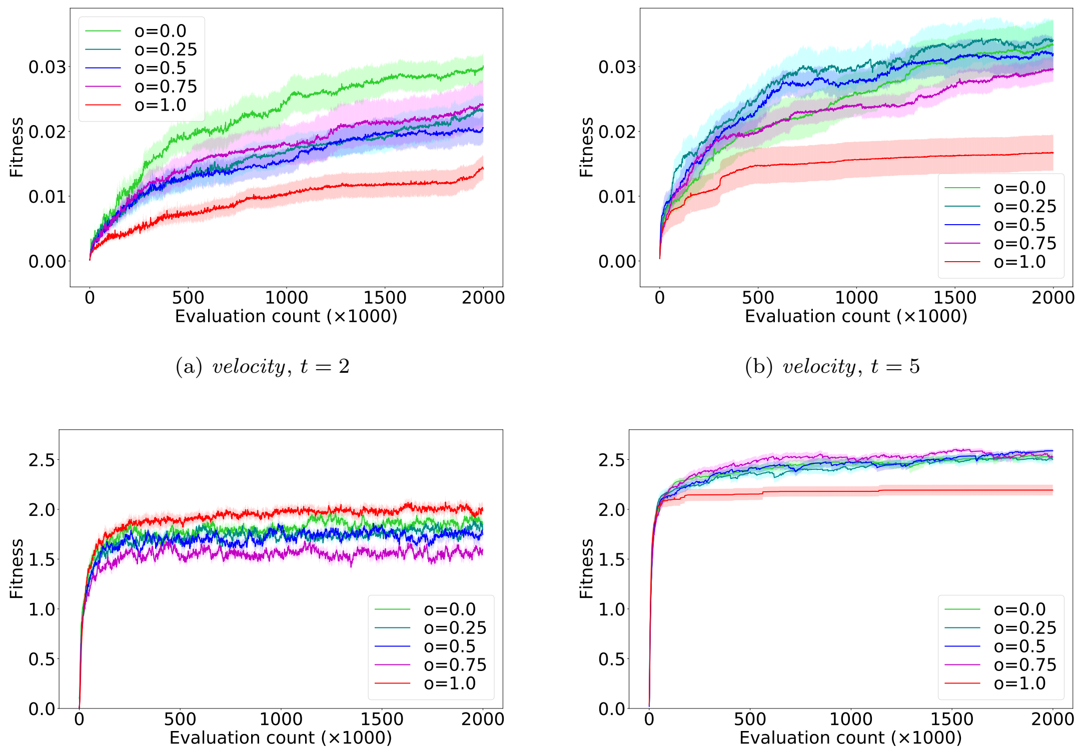
(a) *velocity*, *EqualWidth* (b) *velocity*, *EqualNumber* (c) *velocity*, *IslandModel*

(d) *height*, *EqualWidth* (e) *height*, *EqualNumber* (f) *height*, *IslandModel*

Figure 7: The maximal value of the best fitness obtained after $2 \cdot 10^6$ evaluations (averaged per parametrization), for any combination of values of any two (out of three) parameters, with the third one marginalized out. The three parameters included in the plots are $\frac{N}{M}$ (the average size of a subpopulation), M (the number of subpopulations) and R (the scaling factor for time between the merge events). Empty circles represent the minimal fitness value present in each chart and full circles represent the maximal fitness value in each chart. The actual minimal and maximal fitness values are shown in the legend. For example, a half full circle similar to the one for $M = 10$ and $\frac{N}{M} = 20$ in Fig. 7a indicates that the best examined parametrization incorporating these specific parameter values obtained on average the fitness value of about $\frac{0.025+0.041}{2} = 0.033$.

are selected randomly. Random selection is similar to maximal overlapping of subpopulations in convection approaches, which can be seen as disabling the principal convection mechanism, so the fitness of a solution does not affect the subpopulation to which this solution will be assigned. This impairment of the convection approaches makes them more alike the *IslandModel* and improves their results where optimization problems are not very hard and fast convergence, low exploration and high exploitation are preferred. However, in such cases *IslandModel* may still sometimes obtain better results than impaired convection approaches, because it exchanges a small number of solutions between subpopulations only locally. This local exchange is enough to disrupt the convergence within each subpopulation [17], but *IslandModel* still maintains the genetic diversity between different subpopulations quite well, contrary to the *EqualWidth* approach with fully overlapping ranges.

The influence of overlapping can be interpreted in terms of balancing the exploration and exploitation aspects of evolutionary algorithms. Low degree of overlapping increases the exploration of the search space and simultaneously lowers the exploitation (because only a small proportion of the entire population works with the best solutions), while a high degree of overlapping does the opposite. Since most of the parameter values for the experiments with overlapping reported



(a) *velocity*, $t = 2$

(b) *velocity*, $t = 5$

(c) *height*, $t = 2$

(d) *height*, $t = 5$

Figure 8: Comparison of the performance of convection approaches for different amounts of an overlap o. Each series consists of the high bound (i.e., best) of the average fitness value obtainable for a given selection scheme for all of the tested sets of parameter values. The band around each series represents a quarter of the standard deviation for that series (the entire standard deviation is not shown to avoid overlapping bands and improve the readability of the plots).

here were set to be always the same (as described in Sect. 3.3), as such they likely were not optimal for every tested combination of the fitness function and the size of the tournament, which is confirmed by Fig. 7 and Sect. 4.2. In the situation where the same values of parameters are used independently of the difficulty of the given optimization problem, some degrees of overlapping are better than others in alleviating the problem of non-optimal ratio of exploration and exploitation – with low values of o facilitating exploration and high values of o facilitating exploitation.

5 Conclusions and further work

In this paper we provided a comparison between the convection-based selection schemes in multipopulational evolutionary algorithms and the island model approach, which is widely used in parallel and distributed evolutionary computation. We also examined a generalization of the convection approaches, where the overlap between fitness ranges of different subpopulations is allowed.

The experiments demonstrated that convection selection is superior to the island model for hard optimization problems such as evolutionary design, and can yield comparable results for simpler optimization problems, such as the three considered mathematical problems. This is likely because convection selection can work more efficiently than the island model by preventing the convergence of the population. On the other hand, allowing fitness ranges in subpopulations to overlap makes it possible to balance the exploration–exploitation tradeoff, including the extreme case of full overlap where all subpopulations share the same fitness range similarly to the island model.

The results and analyses presented here were based on experiments that were extremely time consuming due to a large number of combinations of parameter values tested. Still, we would like to extend ranges of parameters, as current results indicate that this may lead to more interesting findings and more general conclusions. Another goal is to develop adaptive convection that adjusts its parameters during evolution, which is motivated by the difficulty of predicting optimal values of the parameters a priori and the infeasibility of searching though all of them. Finally, we would like to establish a theoretical basis for the convection selection/distribution schemes in order to identify the properties of the problems for which these schemes are especially efficient.

References

[1] Seth Bullock. Are artificial mutation biases unnatural? In *European Conference on Artificial Life*, pages 64–73. Springer, 1999. `doi:10.1007/3-540-48304-7_11`.

[2] Seth Bullock. Smooth operator? Understanding and visualising mutation bias. In *European Conference on Artificial Life*, pages 602–612. Springer, 2001. `doi:10.1007/3-540-44811-X_68`.

[3] Erick Cantú-Paz. Migration policies, selection pressure, and parallel evolutionary algorithms. *Journal of Heuristics*, 7(4):311–334, 2001.

[4] Matej Črepinšek, Shih-Hsi Liu, and Marjan Mernik. Exploration and exploitation in evolutionary algorithms: a survey. *ACM Computing Surveys (CSUR)*, 45(3):35, 2013.

[5] Connie J. G. Gersick. Revolutionary change theories: A multilevel exploration of the punctuated equilibrium paradigm. *Academy of management review*, 16(1):10–36, 1991.

[6] Philip F. Hingston, Luigi C. Barone, and Zbigniew Michalewicz. *Design by evolution: advances in evolutionary design*. Springer, 2008.

[7] Maciej Komosinski. Applications of a similarity measure in the analysis of populations of 3D agents. *Journal of Computational Science*, 21:407–418, 2017. URL: `http://dx.doi.org/10.1016/j.jocs.2016.10.004`, `doi:10.1016/j.jocs.2016.10.004`.

[8] Maciej Komosinski and Konrad Miazga. Tournament-based convection selection in evolutionary algorithms. *PPAM 2017 proceedings, Lecture Notes in Computer Science*, 10778:466–475, 2018. `doi:10.1007/978-3-319-78054-2_44`.

[9] Maciej Komosinski and Adam Rotaru-Varga. Comparison of different genotype encodings for simulated 3D agents. *Artificial Life Journal*, 7(4):395–418, Fall 2001.

[10] Maciej Komosinski and Szymon Ulatowski. Framsticks SDK (Software Development Kit). URL: `http://www.framsticks.com/sdk`.

[11] Maciej Komosinski and Szymon Ulatowski. Genetic mappings in artificial genomes. *Theory in Biosciences*, 123(2):125–137, September 2004. URL: `http://dx.doi.org/10.1016/j.thbio.2004.04.002`, `doi:10.1016/j.thbio.2004.04.002`.

[12] Maciej Komosinski and Szymon Ulatowski. Framsticks: Creating and understanding complexity of life. In Maciej Komosinski and Andrew Adamatzky, editors, *Artificial Life Models in Software*, chapter 5, pages 107–148. Springer, London, 2nd edition, 2009. URL: `http://www.springer.com/978-1-84882-284-9`.

[13] Maciej Komosinski and Szymon Ulatowski. Multithreaded computing in evolutionary design and in artificial life simulations. *The Journal of Supercomputing*, 73(5):2214–2228, 2017. URL: `http://www.framsticks.com/files/common/MultithreadedEvolutionaryDesign.pdf`, `doi:10.1007/s11227-016-1923-4`.

[14] Maciej Komosinski and Szymon Ulatowski. Framsticks web site, 2018. URL: `http://www.framsticks.com`.

[15] Gabriel Luque, Enrique Alba, and Bernabé Dorronsoro. Parallel genetic algorithms. *Parallel metaheuristics: A new class of algorithms*, pages 107–126, 2005.

[16] W. N. Martin, J. Lienig, and J. P. Cohoon. Island (migration) models: evolutionary algorithms based on punctuated equilibria. In T. Bäck, D. B. Fogel, and Z. Michalewicz, editors, *Evolutionary Computation 2: Advanced Algorithms and Operators*, chapter 15, pages 101–124. Institute of Physics Publishing, Bristol, 2000.

[17] Zbigniew Skolicki and Kenneth De Jong. The influence of migration sizes and intervals on island models. In *Proceedings of the 7th annual conference on Genetic and evolutionary computation*, pages 1295–1302. ACM, 2005.

[18] Dirk Sudholt. Parallel evolutionary algorithms. In *Springer Handbook of Computational Intelligence*, pages 929–959. Springer, 2015.

[19] P. N. Suganthan, N. Hansen, J. J. Liang, K. Deb, Y. P. Chen, A. Auger, and S. Tiwari. Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization. Technical report, Nanyang Technological University, 2005.

[20] Gilbert Syswerda. A study of reproduction in generational and steady state genetic algorithms. *Foundations of genetic algorithms*, 2:94–101, 1991.

[21] Marco Tomassini. Parallel and distributed evolutionary algorithms. In K. Miettinen et al., editors, *Evolutionary Algorithms in Engineering and Computer Science*, chapter 7, pages 113–133. Wiley, 1999.