VOLUME 18, N° 4 2024
Journal of Automation, Mobile Robotics and Intelligent Systems

Influence of Migration on Efficacy and Efficiency of Parallel Evolutionary Computing

Submitted: 22nd September 2023; accepted: 10th April 2024

Sylwia Biełaszek, Leszek Rutkowski, Aleksander Byrski

DOI: 10.14313/JAMRIS/4-2024/27

Abstract:
Metaheuristics, such as evolutionary algorithms (EAs), have been proven to be (also theoretically, see, for example, the works of Michael Vose [1]) universal optimization methods. Previous works (Zbigniew Skolicki and Kenneth De Jong [2]) investigated impact of migration intervals on island models of EAs in their works. Here we explore different migration intervals and amounts of migrating individuals, complementing Skolicki and DeJong's research. In our experiments, we use different ways of selecting migrants and pave the way for further research, e.g., involving different topologies and neighborhoods. We present the idea of the algorithm, show experimental results.

Keywords: *parallel evolutionary computing, metaheuristics, migration*

1. Introduction

Since seminal work of Wolpert and Mac Ready [3] and formulation of No Free Lunch theorem we know that each metaheuristic algorithms must be properly parameterized in order to make it feasible for a particular problem. Researchers such as Sudholt, Cantu-Paz or Skolnicki and De Jong worked on the parallel model of an evolutionary algorithm (EA) [4] [5] arguing that decomposition of population increases diversity and the efficacy of the whole algorithm. We have been examining this problem and this paper is devoted actually to extend the results described by Skolnicki and De Jong described in [2].

While studying the influence of various migrations sizes (migration rate) and migration intervals on island models, researchers noticed that the migration interval seemed to be a dominating factor to the best solution found, too frequently migrations cause islands to dominate others and lose global diversity, too rarely mirations perform degraded performance due to slow convergence, but even small migrations already make a significant impact on the result of an island model.

Therefore, we focused on further parametrization of the parallel EA striving towards checking what kind of configurations will help us reaching better efficacy, using the popular multimodal benchmarks as case studies.

The paper is organized as follows: In the next chapter we introduce parallel models of EAs. In the third chapter we describe considered algorithm. In chapter four we present the results and discuss them. In the fifth chapter we come to the conclusion.

2. Parallel Models of Evolutionary Algorithms

EAs are well known and widely decribed in the literature [6–8] as probabilistic optimization methods inspired by biological analogies (natural evolution). The essence of EAs is to combine the phenomenon of random (undirected) genotype changes with strictly directed environmental pressure on the phenotype. It is a powerful method for solving a large scale of problems that can be described in an appropriate form. It consists of choosing the right type of algorithm, designing the method of coding solutions (creating a solution space of the problem) and constructing the objective function. To find a solution, we need to know almost nothing about the function being optimized ("black box"). There may even be no objective function at all: we can use evolutionary algorithms even when the only thing we can say about the points in the state space is which of the two points is better (tournament selection).

The scheme of the classical evolutionary algorithm includes the creation of an initial population consisting of random individuals, the use of genetic operators (i.e. certain transformations of the genetic code of individuals), calculating the value of the objective function of individuals, selection. The operations described above takes place in a cycle that ends when the specified termination condition is met. The final population in each cycle becomes the current for next one and evolution continues. The algorithm stops at the user's request, after a certain time, certain number of solution evaluations or when a certain solution quality threshold is reached. The algorithm is non-deterministic (random action of mutation, crossing and selection), we have no guarantee that the solution found is optimal, but they give a high probability that the result will be close to the optimal one and we will get it in a time that satisfies us. The genetic operators: mutation, crossover and selection can be used in different variants.

The basic and the necessary elements of an EA are:

- *Individual – an exemplary solution of the problem – placed in a certain environment to which he may be better or worse adapted. The "goal" of evolution is to create an individual that is as well adapted to a given environment as possible.*
- *Phenotype – characteristics of a given individual. In the case of EAs, these are the parameters (features) of the solution that are subject to evaluation.*
- *Genotype – a complete and unambiguous description of an individual contained in its genes.*
- *Chromosome – the place where the genotype of an individual is stored.*
- *Population – a group of individuals living in a common environment and competing for its resources.*
- *Solution coding – a way of storing any acceptable solution to a problem in the form of an individual's genotype (e.g. a string of bits).*
- *Function of adaptation (fitness) – a function that allows to determine its quality for a given individual (from the point of view of the problem being solved). Its values are real non-negative and a higher value of the function always means that a given individual is better. In the case of natural evolution, the equivalent of such a function is the general assessment of an individual's adaptation to a given environment. In practice, this function is usually a slight modification of the objective function of the problem being solved.*
- *Genetic operators: mutation, crossover and selection can be used in different variants*

2.1. Main kinds of PEA's

A significant improvement in the operation of the EA is obtained by using a parallel EA model (PEA). The basic idea is to divide a task into subtasks, and to solve them simultaneously using multiple processors.

Realization takes place as work on single population or on several relatively isolated populations, using massively parallel computer architectures or multicomputers with fewer and more powerful processing elements.

There are three main types of parallel EAs; there are global single-population master-slave EAs, single-population fine-grained EAs, and multiple-population coarse-grained EAs.

The first type of parallel models is the master slave [9]. It's an easy to implement and very efficient method of parallelisation where we use a single panmitic population, just like in a simple EA, but evaluation of the individuals and genetic operators is parallel. This model does not assume anything about the underlying computer architecture. Each individual may compete and mate with any other (thus selection and mating are global). Selection and crossover consider the entire population. In this model master stores the population, and slaves evaluate the fitness of an individual which is independent from the rest of the population, and assigning a fraction of the population to each of the processors available. Communication between master and slave occurs only when each slave receives its subset of individuals to evaluate and when the slaves return the fitness values. The algorithm is usually synchronous.

The second type, fine-grained parallel EAs [10] consist of one spatially-structured population that limits the interactions between individuals. Selection and mating are restricted to a small neighborhood, but neighborhoods overlap permitting some interaction among all the individuals. The ideal case is to have only one individual for every processing element available. The most popular structures used for this model are ring, torus, cube or hypercube. This model is suited for massively parallel computers.

Third, the most popular method of parallel implementation of EAs is multiple-population EAs [4]. It consist in few relatively large subpopulations which exchange individuals occasionally in process named migration, controlled by several parameters.

They are known as "distributed" EAs, because they are usually implemented on distributed-memory MIMD computers (possibly also using VLSI circuit synthesis, GPGPU or HPC) or the "island model" because relatively isolated demes we can call "islands". They are also called coarse-grained EAs, since the computation to communication ratio is usually high.

The main idea is that copy of the best individual found in each deme is sent to all or one of its neighbors after every generation.

It is possible to use different approaches to solve this problem [11], such as work with isolated demes and with a "delayed" migration scheme in which communications began only after the demes were near convergence (very high migration rate). In this case the solution found by isolated demes was much lower than that reached with a single large population, however, this delayed scheme found solutions of the same quality as the panmictic population and as multiple demes with frequent migrations. We can also migrate solutions between demes after the demes converged completely [12, 13].

Sometimes migration happens at regular intervals, and sometimes [13] migration occurs after the demes converged completely (the author used the term "degenerate") with the purpose of restoring diversity into the demes to prevent premature convergence to a low-quality solution.

In practice we take a few conventional (serial) EAs, run each of them on a node of a parallel computer, and at some predetermined times, or number of carried evaluations in an EA, exchange a few individuals.

Most of the time, populations are in equilibrium (i.e., there are no significant changes in its genetic composition), but that changes on the environment can start a rapid evolutionary change. Therefore, the arrival of individuals from other populations can punctuate the equilibrium and trigger evolutionary changes.

2.2. PEA's Main Parameters

Main parameters of migration are the topology that defines the connections between the subpopulations, migration rate that controls how many individuals migrate, and migration interval that affects the frequency of migrations, the number of islands, and the populations sizes.

Researchers [11, 14] are trying to find relationships between important PEA parameters. It is difficult and differs for different methods of emigration and immigration and the problem being solved.

Finding these dependencies would be also helpful in estimating the optimal number of processors for solving problems in the parallel model.

So it is worth to test it for manz different settings.

Also different ways of creating fitness and mutation and crossover functions, using a single population or multiple subpopulations (dems), different ways of exchanging migrants and how selection is applied (globally or locally) were investigated [15].

2.3. Topologies

Many researchers have struggled with the topic of island model communication topology. It is a major factor in the cost of migration. Densely connected topology may promote a better mixing of individuals, but it also entails higher communication costs. The general trend is to use static topologies that are specified at the beginning, but some analyse [4] of the design and expected optimization times depending on the topology lead to changes during the execution of the algorithm according to the settings. We call it dynamic topology schemes. It speeds up the optimization. Migrants are sent to demes that meet some criteria.

[16] also studied the size of the connection topology impact and the appropriate topology choices for different applications. Migration topology rankings (more precisely, preorders) were built for a different number of islands, different optimization problems and different basic algorithms.

There are some unresolved questions in this model, such as:

- *what is the level of communication necessary to make a parallel EA behave like a panmictic EA?*
- *what is the cost of this communication?*
- *is the communication cost small enough to make this a viable alternative for the design of parallel EAs?*

2.4. EA's Sequential Contra Parallel Versions

Sequential EAs are very effective in many applications. However, there are [15] problems in their use that can be solved with PEA.

It also happens that sequential EAs can get trapped in a sub-optimal region of the search space.

PEAs can search different subspaces of the search space in parallel, thus reducing the likelihood of being trapped by low-quality subspaces.

Migration of individuals between populations may increase the selection pressure [5]. This has the desirable consequence of speeding up convergence, but it may result in an excessively rapid loss of variation that may cause the search to fail.

For example, sometimes problems require the use of very large populations [17], and the memory needed to store each individual can be significant. In some cases, this prevents an application from running efficiently on a single machine, so some parallel form of EA is necessary. After dividing it in sub-population, as different islands retain a degree of independence and thus explore different regions of the search space, the probability of an improved score increases. When the performance of a split evolutionary algorithm is similar to that of a large population, the use of migration makes the performance equal to or exceed that of a large population.

2.5. Dynamics of PEA's

Skolicki and deJong [18] described the mechanism of improving the results of pea operation, using the two-level dynamics of the island model, dividing it into levels: local on each island and inter-island interactions. These two evolutionary processes interact and may contribute to the overall outcome to varying degrees. But it is important to choose the number of islands (global) and population size (local level) on these islands accordingly. And to choose a migrant at the right moment, so that he is good enough and admitted at the right time so that he does not disrupt the processes taking place there.

2.6. Our Inspirations

In [2], researchers have experimentally studied the influence of various migrations sizes (migration rate) and intervals on island models using a set of special functions. They notice that the migration interval seems to be a dominating factor, with migration size generally playing a minor role with regard to the best solution found. Too frequently migrations cause islands to dominate others and lose global diversity, even small migrations already make a significant impact on the behavior of an island model but rare migrations cause a degraded performance due to the slow convergence.

While examining the behavior of the island model, we identified the need to study the impact of migrant selection strategies on the quality of the solution. In our approach, we copy selected migrants (not relocate, like most of researchers before) and join the population on the target island where they take part in genetic operations. At the end, they are subjected to selection together with the rest of the population, i.e., we allow them to take part in genetic operations together with the residents of the island.

Using different migrant selection strategies, we found that they had a noticeable impact on the results obtained. We used two strategies to select the best and most distant migrants. The control strategy of random selection did not give such significant improvement. There has been a big progress with many settings (different migration intervals and number of migrants) regarding the migration interval and the size of the migrant group.

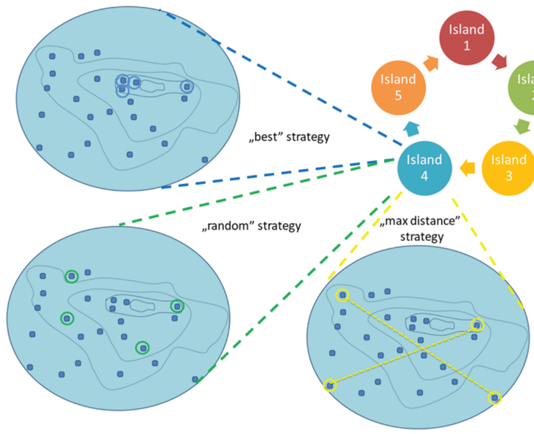


Figure 1. Three ways of selecting migrants studied. The colors used are referenced to bar results charts

3. The Considered Algorithm

In the island model, we use three migrant selection strategies: "best" and "max distance" ("mDist") without repeats, and "random". Those strategies will be discussed in detail later.

In our approach migrations consist of copying (not moving) selected individuals from the source island on destination island. Four migration intervals were used every 5th, 15th, 25th and 35th evaluation, and there were four migrant group size settings: 1, 4, 9 and 12 individuals.

Migrant admission strategy was to join them to the population on destination island, then using crossover and mutation operators, then selection.

Three migrant selection strategies without repetitions (shown in Fig. 1) were used:
- *"best" – n individuals were selected in order of fitness values, starting from the best one*
- *"random" – n random individuals were randomly selected*
- *"mDist" – Euclidean distances of all individuals were tested in pairs and then the farthest ones were selected in pairs (if possible) until the number of migrants set in the parameters was exhausted*

Each of the algorithm settings were tested 10 times and the results were averaged.

4. Experimental Results

All computations were performed on a PC workstation with Windows 10 Intel Core i5-2520M 2.50 GHz, 8 GB RAM memory and using Intel HD Graphics 3000 graphic card.

The algorithm was created using the jMetalPy framework, and the movement of migrants between the islands was created using rabbitMQ, Docker and Pika. We investigate Rastrigin and Sphere (De Jong) problems in dimension 200. The topology consists of 5 islands connected in a ring with two-way traffic. The population size on each island is 16 and the offset is 4. We study the results of the algorithm for one island by reference. Then the population has a cardinality of 80 and an offset of 20.

When comparing evolutionary algorithms running on different hardware, it is important to have an equal number of evaluations performed in the algorithms. This is because environment parameters are different, e.g., the size of the operating memory, which affects the speed of the computer.

Similarly, this is true when comparing an algorithm running on one island to one running on five islands. But it's not only the number of evaluations that matters. It is also important that the number of cycles of the algorithm is equal on one island and each of the five islands. We achieved this by equalizing the population and offset on one island and the total number on many islands, as we described above.

The end criterion was the maximum number of evaluations. For the five islands model: 15,000 for the Sphere and 20,000 for the Rastrigin, and for one island, 75,000 for the Sphere problem and 100,000 for the Rastrigin problem. This means that not only the stop condition, but also the size of the population are the same in the compared studies.

In Figures 2 and 3 (for the Rastrigin and Sphere problems with different migration strategies, respectively), the color scale (from the worst result – dark red to the best result – dark green) shows the average results of various settings: the migration interval and the size of the migrant group. We can see that the red color values are grouped in the same area and the green color values in approximate areas.

The same thing, but in the form of a 3D spatial graph, can be seen in charts 4 and 5 (for the Rastrigin and Sphere problems with the examined settings of the migration interval and the number of migrants, respectively). The best results are those with the lowest position in the drawings, and the worst are those placed highest. The worst results are in the same place in all the drawings, and the best ones occupy similar areas. We can see a similar inclination and shape of the planes created for these samples – which indicates that the size of the migrant groups and the intervals work similarly for these different problems and strategies.

In Figures 6 and 7 (for the Rastrigin and Sphere problems, respectively), the bar graphs show the comparison of average final results of the five-island and one-island settings. Here we can clearly observe how many attempts of the tests on five islands (interval, size of the group of migrants) led to success with a given migrant selection strategy, i.e. improved the result of the single-island model (its graph is to the left of the bar of one island). For example, the strategy "best" and "mDist" in many settings (interval, size of the group of migrants) led to an improvement in the result in relation to the single-island model. And in the "random" strategy – comparative, because the migrating individuals were simply drawn randomly, it did not improve the result of one island at all (for Sphere) or almost at all (for Rastrigin).

		migration interval			
		5	15	25	35
number of emigrants	1	60,05018	62,24134	65,83849	67,3321
	4	56,13038	60,55656	59,63388	67,85675
	9	147,1152	63,50758	66,54987	68,98297
	12	203,4451	66,638	64,72388	66,62612

		migration interval			
		5	15	25	35
number of emigrants	1	61,57406	63,74214	67,41727	68,83856
	4	62,79272	58,71623	65,47896	65,38775
	9	145,3055	61,89432	65,39729	66,28892
	12	201,2937	64,39775	64,30907	66,21031

(a) Rastrigin with “best” migration strategy (left) and with “mDist” (right)

		migration interval			
		5	15	25	35
number of emigrants	1	68,43988	67,67216	69,45889	72,94368
	4	63,77661	65,40143	69,11512	67,24009
	9	152,7634	66,37153	67,87853	68,23155
	12	198,4316	75,06816	65,00696	71,01671

(b) Rastrigin with “random” migration strategy

Figure 2. Average results achieved for Rastrigin problem with different migration strategies. Results tabulated by migration intervals and numbers of migrants. Color scale and intensity – worst result – dark red, best result – dark green

		migration interval			
		5	15	25	35
number of emigrants	1	1,322557	1,4741	1,666194	1,706987
	4	1,132882	1,22046	1,472165	1,489834
	9	7,823738	1,459737	1,504647	1,637578
	12	12,77929	1,669268	1,631477	1,60942

		migration interval			
		5	15	25	35
number of emigrants	1	1,533697	1,771442	1,834546	1,832084
	4	1,442026	1,498892	1,599757	1,880713
	9	7,904653	1,422301	1,648423	1,827603
	12	12,64878	1,509175	1,660833	1,698813

(a) Sphere with “best” migration strategy (left) and with “mDist” (right)

		migration interval			
		5	15	25	35
number of emigrants	1	1,915516	1,864151	2,11241	2,062294
	4	2,022016	1,854306	1,853639	1,919819
	9	7,720189	1,626553	1,777396	1,933809
	12	12,13308	1,782286	1,815817	1,869938

(b) Sphere with “random” migration strategy

Figure 3. Average results achieved for Sphere problem with different migration strategies. Results tabulated by different migration intervals and numbers of migrants. Color scale and intensity – worst result – dark red, best result – dark green

Table in Figure 8 describes (for each problem, each way of selecting migrants, different migration intervals and numbers of migrants) ranking (in plus or minus) of the results obtained in relation to the corresponding result obtained on one island.

The following symbols have been used in the table Figure 8:

- *0 – reference point – result for a model with 1 island*
- *positive numbers – position of five island model with specific parameters in ranking of results better than appropriate “reference” on one island*
- *negative numbers – position of five island model with specific parameters in the ranking of results worse than the result of appropriate “reference” on one island*

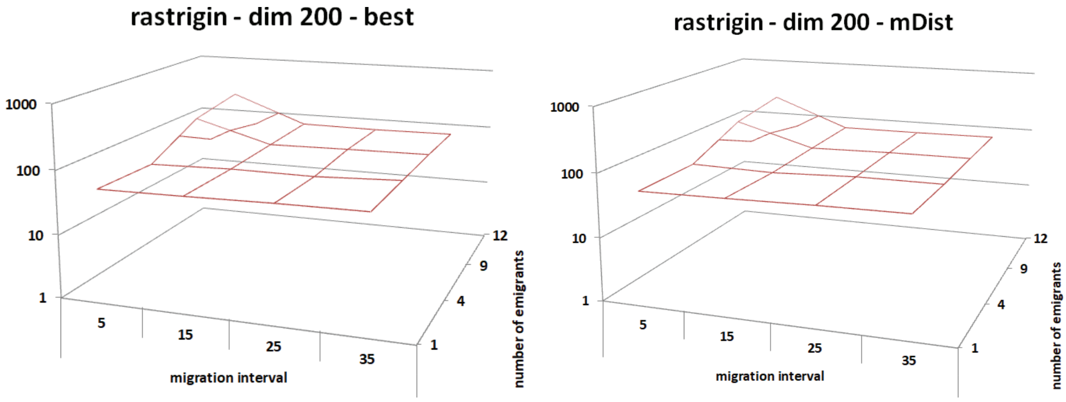
The first three columns in the Table 8 describe algorithm start parameters: the number of islands, the migration interval and the number of migrating individuals.

Figures 12 and 13 show the runs of the 10-trial average for each setting (interval, migrant group size) in the winning strategies for a given problem.

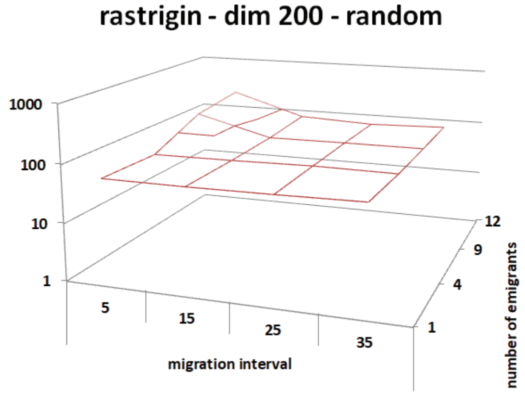
For easier understanding of Figure 8, please compare Figure 13 showing all line results of Sphere 200 in “best” version, it’s values shown in Figure 3a and the forth column of the Results table Table 8, i.e., Sphere 200 / “best” column. Find the result obtained on one island in the graph and notice that subsequent results have the same number in the table as the distance of their graph from the graph obtained on one island.

Similarly, when considering Rastrigin problem, Figure 12, Figure 2a and Table 8 (the seventh column, ie. Rastr 200 / “best” column) should be considered.

The best, taking into account migrants selection strategies, were: “best”, then “mDist” and finally “random”.

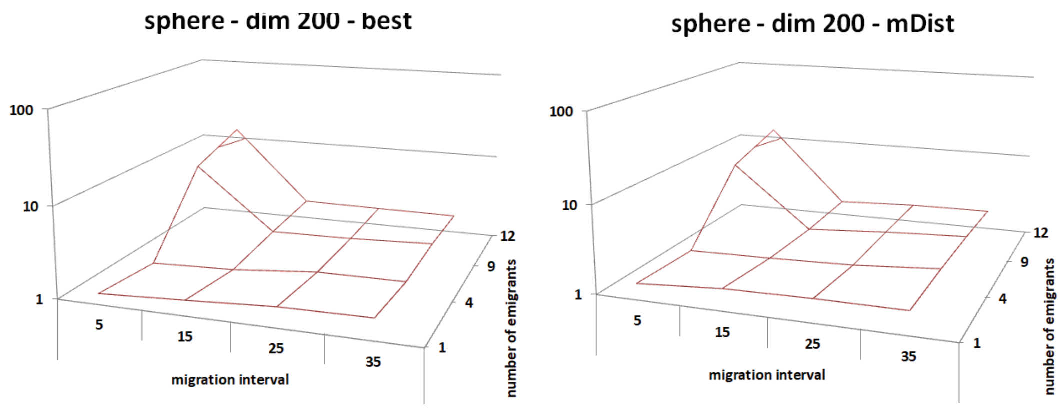


(a) "best" (left) and "mDist" (right) migrant selecting strategy

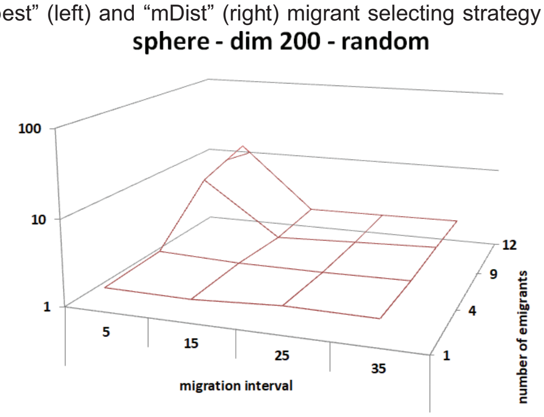


(b) "random" migrant selecting strategy

Figure 4. Comparative results of Rastrigin problem, with the examined settings of the migration interval and the number of migrants

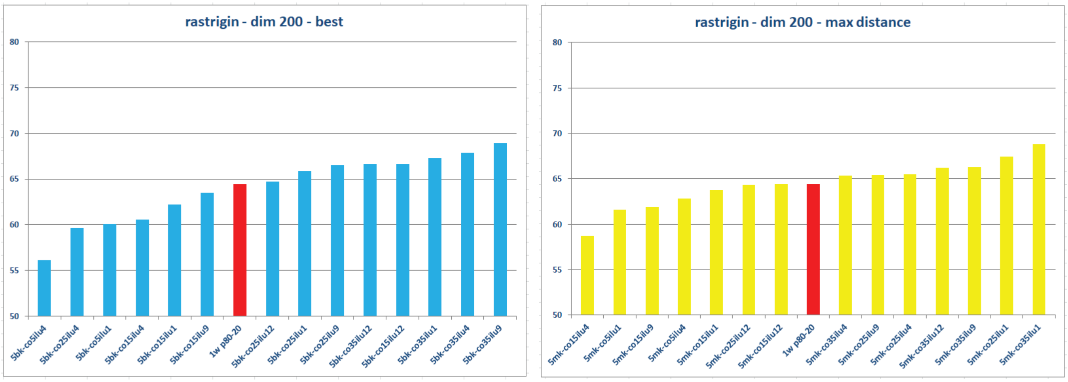


(a) "best" (left) and "mDist" (right) migrant selecting strategy

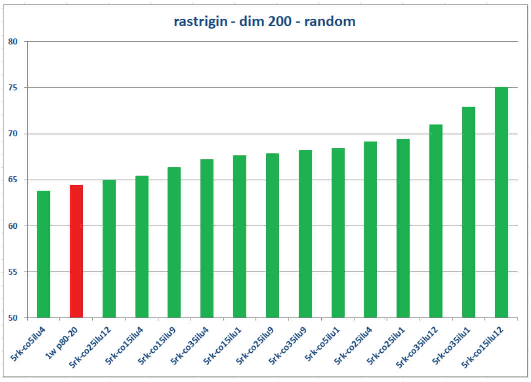


(b) "random" migrant selecting strategy

Figure 5. Comparative results of Sphere problem, with the examined settings of the migration interval and the number of migrants

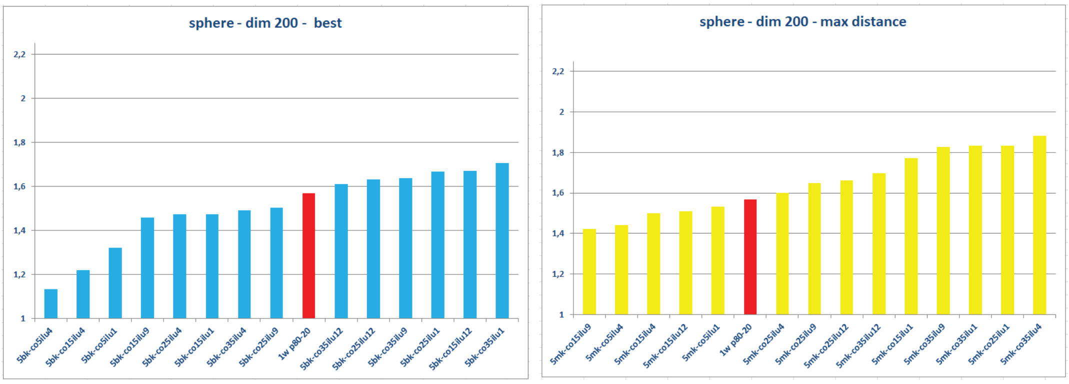


(a) “best” (left) and “mDist” (right) migrant selection strategy

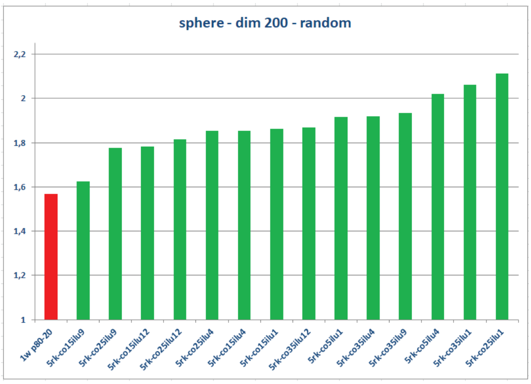


(b) “random” migrant selection strategy

Figure 6. Graphs of results of Rastrigin problem, obtained with different settings of the migration interval and the number of migrants on the five islands. For comparison – red bar – the result achieved on one island with comparable settings



(a) “best” (left) and “mDist” (right) migrant selection strategy



(b) “random” migrant selection strategy

Figure 7. Graphs of results of sphere problem, obtained with different settings of the migration interval and the number of migrants on the five islands. For comparison – red bar – the result achieved on one island with comparable settings

			Sphere200			Rastr200		
island num	interval	emigr num	best	rand	mDist	best	rand	mDist
1	-	-	0	0	0	0	0	0
5	5	1	6	-9	1	4	-8	6
5	5	4	8	-12	4	6	1	4
5	5	9	-15	-15	-15	-15	-15	-15
5	5	12	-16	-16	-16	-16	-16	-16
5	15	1	3	-7	-5	2	-5	3
5	15	4	7	-6	3	3	-2	7
5	15	9	5	-1	5	1	-3	5
5	15	12	-5	-3	2	-5	-13	1
5	25	1	-4	-14	-8	-2	-10	-6
5	25	4	4	-5	-1	5	-9	-3
5	25	9	1	-2	-2	-3	-6	-2
5	25	12	-2	-4	-3	-1	-1	2
5	35	1	-6	-13	-7	-6	-12	-7
5	35	4	2	-10	-9	-7	-4	-1
5	35	9	-3	-11	-6	-8	-7	-5
5	35	12	-1	-8	-4	-4	-11	-4

Figure 8. Ranking, for each of the examined migration intervals and the number of migrants, (in plus or minus) the results obtained in the PEA with five islands in relation to the EA result obtained on one island for the same problem and the method of selecting migrants

	rastrigin - dim 200		
one island	64,442		
result	"best"	"random"	"mDist"
five island worst	68,982	75,068	68,838
five island best	56,13	63,776	58,716
value difference	"best"	"random"	"mDist"
5 islands worst - 1 island	4,54	10,626	4,396
5 islands best - 1 island	-8,312	-0,666	-5,726
5 islands worst - 5 islands best	12,852	11,292	10,122

Figure 9. Comparing the best and worst results in five island model to one island model for different migrants selection strategies. Rastrigin problem)

In Sphere200/random, no settings connected with five islands gave results better than the one island score, and in Rastr200/random only one setting (co5ile4) gave better results when comparing analogously.

In the other versions of the migration strategy ("mdist" and "best"), for both problems, there are always some five island results that perform better than one island.

This means that our selective selection of migrants on source island is working. We achieve the best results when we select the best or more diverse individuals (the most distant in our case), the results are then significantly better than when we select migrants randomly.

The best results in the Sphere problem were achieved with the "best" strategy (settings – interval 5, group of 4 migrants), worse with the "mDist" strategy (interval 15, group of 9 migrants), and all settings of "random" strategy achieved worse results than results of one island.

	sphere - dim 200		
one island	1,568		
result	"best"	"random"	"mDist"
five island worst	1,706	2,112	1,88
five island best	1,132	1,626	1,422
value difference	"best"	"random"	"mDist"
5 islands worst - 1 island	0,138	0,544	0,312
5 islands best - 1 island	-0,436	0,058	-0,146
5 islands worst - 5 islands best	0,574	0,486	0,458

Figure 10. Comparing the best and worst results in five island model to one island model for different migrants selection strategies. Sphere problem

		number of migrants			
		1	4	9	12
migration interval	5	4	5	0	0
	15	3	4	4	2
	25	0	2	1	1
	35	0	1	0	0

Figure 11. Comparing the improved results (per six settings = 2 problems * 3 strategies) of PEA with five islands in relation to the result of EA on one island, obtained for the examined migration intervals and numbers of migrants

The best results of the Rastrigin problem were achieved with "best" strategy (settings – interval 5, group of 4 migrants), followed by "mDist" (settings – interval 15, group of 4 migrants) and only one "random" strategy setting (settings – interval 5, group of 4 migrants) was better for five islands than the results of one island.

In the "random" strategy, individuals of different quality were randomly selected. Therefore, their results are worse than in strategies where carefully selected individuals – the best or the most diverse – migrate qualitatively or improve the diversity on the destination island.

For every issue and migration strategy, the settings with interval 5 and size of migration group 9 or 12 – always performed significantly worse than all others. It happens because in such cases migrants arriving in large numbers on the destination island hinder the evolution on this island, almost replacing its existing population These are cases of short migration intervals with a big number of migrants (close to the size of the population) at the same time.

The additional information in Figure 9 (for Rastrigin) and Figure 10 (for Sphere) show the numerical values achieved on five islands (best and worst result) and on one island and show the numerical differences between these results.

Let's look at this differences. As we can see the greatest value differences between results achieved on five islands and biggest improvement on five islands over the scores of a single island were achieved for "best" migrant selection strategy in both problems. In Rastrigin (Fig. 9) "best" surpassed the one island score by 8,13, "mDist" by 5,72, and a "random" by 0,66. In Sphere (Fig. 10) "best" surpassed one island score by 0,43, "mDist" by 0,14, and "random" by 0,05.

Figure 11 summarizes positive values in Figure 8 by rows. In each field at the intersection of the appropriate migration interval and the size of the migrant group, the number of results from five islands that achieved success appears, i.e., they were better than the result on one island (out of six possible – two problems * three strategies).

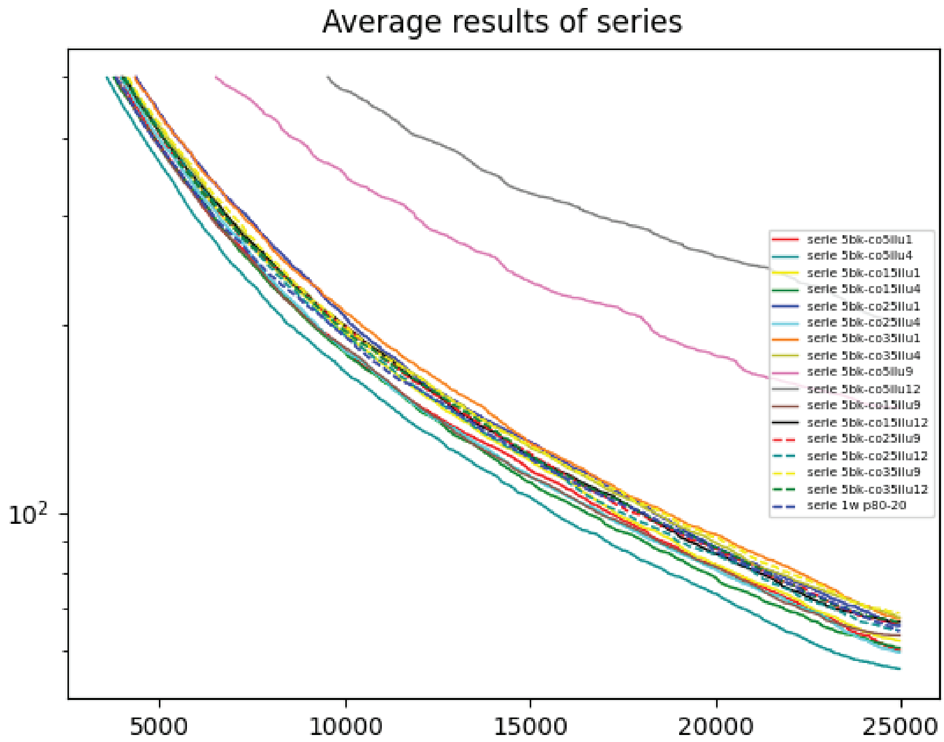


Figure 12. Averaged runs of parallel calculations on five islands and reference calculations obtained on one island for a winning migrant selection strategy ("best") for the Rastrigin problem with dimension 200

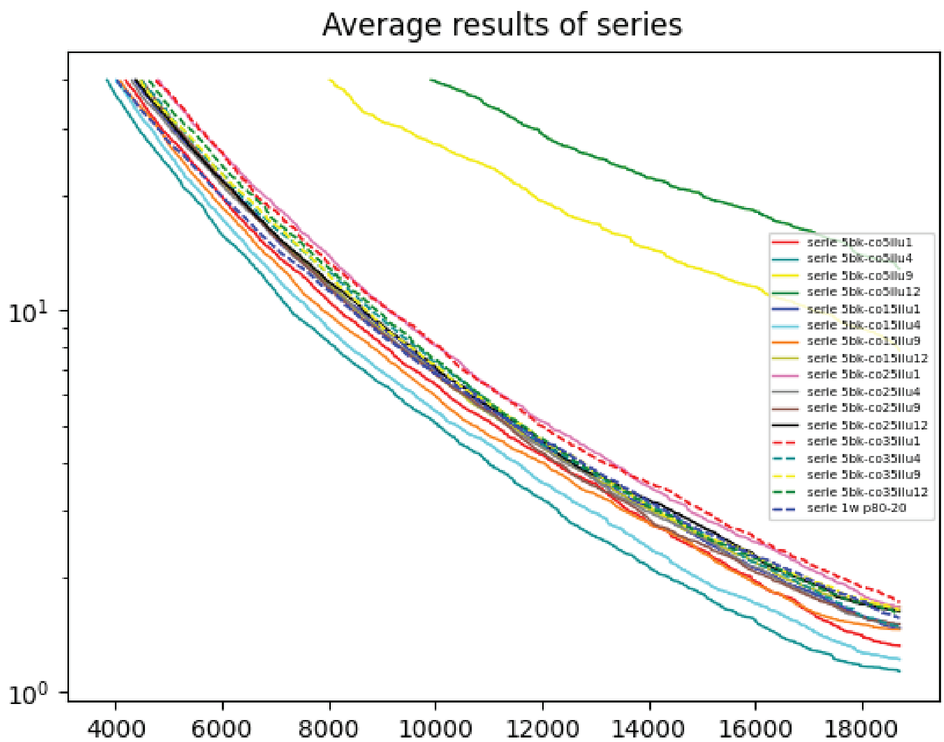


Figure 13. Averaged runs of parallel calculations on five islands and reference calculations obtained on one island for a winning migrant selection strategy ("best") for the Sphere problem with dimension 200

As we can see the best results we achieved for low number of migrants (one, four or nine) and more frequent migrations (after every 5th or every 15th evaluation). This is because these are short intervals and small groups of migrants – that is why they have the opportunity to improve the quality of the population on the target, and not disrupt the course of processes on it.

If we focus on the worst settings of the interval and the number of migrant groups, we can see that, as mentioned above – large migrations destroy the population of the destination island by replacing it, both when they are frequent (parameters – interval 5, migrant group size: 9, 12), when they are very disturbing, and rare (parameters – interval 35, migrant group size: 9, 12).

Another group of the worst attempts were in which migrations were few and rare (parameters – interval – 25, 35, migrant group size: 1).

5. Conclusion

Inspired by the works of De Jong and Skolicki, we performed an experiment for Rastrigin and Sphere problems. We used four migration intervals and four sizes of migrant groups, expanding the experiment with a study for different migration strategies.

We observed that the strategies we used for selecting migrants in many cases led to improved results compared to our reference, i.e., the EA model working as one island.

In the "best" strategy, we sent group of best individuals from the source island. So there was a possibility that on the destination island they would be good material for the further operation of the evolutionary algorithm.

Using the "mDist" strategy, we sent a group of individuals that were as distant as possible from each other in terms of genotype. Thus, they had a chance to increase diversity on the destination island, or even to restore diversity if the population on the destination island was too convergent.

We intend to continue our research on this important topic in the future. We will conduct our research using HPC on a large scale. In addition, we intend to study the behavior of a parallelized EA in an environment with delays, assuming the possibility of desynchronization and taking care of scalability. This will make it possible to compare the operation of the algorithm in contrast to coherently and synchronously operating master-slave models and the classic island model.

AUTHORS

Sylwia Biełaszek – AGH University of Science and Technology, Al. Mickiewicza 30, Krakow 30-059, Poland, e-mail: bielsyl@agh.edu.pl.

Leszek Rutkowski – AGH University of Science and Technology, Al. Mickiewicza 30, Krakow 30-059, Poland; Institute of Systems Sciences Polish Academy of Sciences, ul. Newelska 6, Warsaw 01-447, Poland, e-mail: rutkowski@agh.edu.pl.

Aleksander Byrski* – AGH University of Science and Technology, Al. Mickiewicza 30, Krakow 30-059, Poland, e-mail: olekb@agh.edu.pl.

*Corresponding author

ACKNOWLEDGEMENTS

This work was supported by NCN UMO-2019/35/O/ST6/00571 and the funds assigned by the Polish Ministry of Education and Science to AGH University of Science and Technology.

References

[1] L. D. Davis, K. De Jong, M. D. Vose, L. D. Whitley, W. Miller, Eds., Evolutionary Algorithms, Vol. 111 of The IMA Volumes in Mathematics and its Applications, New York, NY, Springer, 1999, [Online]. Available: doi: 10.1007/978-1-4612-1542-4.

[2] Z. Skolicki, K. De Jong, The influence of migration intervals on island models, 2005, pp. 1295–1302, doi: 10.1145/1068009.1068219.

[3] D. Wolpert, W. Macready, "No Free Lunch Theorems for Optimization", IEEE Transactions on Evolutionary Computation, vol. 1, no. 1, pp. 67–82, 1997.

[4] D. Sudholt, Parallel Evolutionary Algorithms, Berlin, Heidelberg, Springer Handbooks, Springer, 2015, doi: 10.1007/978-3-662-43505-2-46.

[5] E. Cantu-Paz, "On the Effects of Migration on the Fitness Distribution of Parallel Evolutionary Algorithms", no. UCRL-JC-138729, 2000, URL https://www.osti.gov/biblio/791479.

[6] D. E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, 1989, google-Books-ID: 2IIJAAAACAAJ.

[7] R. Chiong, T. Weise, Z. Michalewicz, Variants of Evolutionary Algorithms for Real-World Applications, 2012, doi: 10.1007/978-3-642-23424-8.

[8] Z. Michalewicz, Genetic Algorithms + Data Structures = Evolution Programs, Springer Science & Business Media, 1996, google-Books-ID: vlhLAobsK68C.

[9] E. Cantú-Paz, Master-Slave Parallel Genetic Algorithms, Genetic Algorithms and Evolutionary Computation, Boston, MA, Springer, 2001, doi: 10.1007/978-1-4615-4369-5-3.

[10] E. Cantú-Paz, Fine-Grained and Hierarchical Parallel Genetic Algorithms, Genetic Algorithms and Evolutionary Computation, Boston, MA, Springer, 2001, doi: 10.1007/978-1-4615-4369-5-8.

[11] E. Cantú-Paz, D. E. Goldberg, "Efficient parallel genetic algorithms: theory and practice", Computer Methods in Applied Mechanics and Engineering, vol. 186, no. 2–4, pp. 221–238, 2000, doi: 10.1016/S0045-7825(99)00385-0.

[12] Y. Sato, Y. Takai, M. Munetomo, "An efficient migration scheme for subpopulation-based asynchronously parallel genetic algorithms.", Material: Proceedings of the fifth international conference on genetic algorithms, S.F. Ed., San Mateo, CA: Morgan Kaufmann, 1993, pp. 649.

[13] H. Braun, "On solving travelling salesman problems by genetic algorithms", Parallel Problem Solving from Nature, Lecture Notes in Computer Science, H.P. Schwefel and R. Manner, Eds. Berlin, Heidelberg: Springer, 1991, pp. 129–133, doi: 10 .1007/BFb0029743.

[14] E. Cantú-Paz, D. E. Goldberg, On the Scalability of Parallel Genetic Algorithms, vol. 7, 1999, pp. 429–449, doi: 10.1162/evco.1999.7.4.429.

[15] M. Nowostawski, R. Poli, "Parallel genetic algorithm taxonomy", 1999 Third International Conference on Knowledge-Based Intelligent Information Engineering Systems. Proceedings (Cat. No.99TH8410), 1999, pp. 88-92, doi: 10.1109/KES.1999.820127.

[16] M. Ruciński, D. Izzo, F. Biscani, "On the impact of the migration topology on the island model", Parallel Computing, Issues, vol. 36, no. 10–11, pp. 555–571, 1993, doi: 10.1016/j.parco.2010.04.002.

[17] D. Whitley, S. Rana, R. Heckendorn, "The Island Model Genetic Algorithm: On Separability, Population Size and Convergence", Journal of Computing and Information Technology, vol. 7, Dec. 1998.

[18] Z. Skolicki, K. De Jong, "The importance of a two-level perspective for island model design", Conference: Evolutionary Computation, 2007. CEC 2007. IEEE Congress on, IEEE Xplore, 2007, doi: 10.1109/CEC.2007.4425078.