Towards Novel Migration Topologies for Parallel Evolutionary Algorithms

Sylwia Biełaszek[1][0000−0001−8947−6272], Laszlo Gulyas[2][0000−0002−6367−6695], and
Aleksander Byrski[1][0000−0001−6317−7012]

[1] AGH University of Krakow, Al. Mickiewicza 30, 30-059 Krakow, Poland
`{bielsyl,doroh,olekb}@agh.edu.pl`
[2] ELTE Eötvös Loránd University, Budapest, Egyetem tér 1-3, 1053, Hungary
`lgulyas@inf.elte.hu`

Abstract. Metaheuristics are excellent tools for solving difficult optimization problems. Parallel metaheuristic models serve to increase the diversity of the search and to avoid premature convergence. Usually, the topologies of the island neighborhood are simple (fully connected, torus, ring); however, network science informs us that there are certain families of graphs (e.g. the Erdős-Rényi random networks) that have interesting features (e.g., limited average path length due to the "small world" property) that could be used as novel neighborhood models. In this paper, we take the first step in exploring the use of topologies informed by network science and study Erdős-Rényi networks in the case of Parallel Evolutionary Algorithms. The benefits of such an approach are presented and discussed.

Keywords: parallel evolutionary algorithm · migration topology · random networks · Erdős-Rényi graphs

1 Introduction

Metaheuristics are well-known methods of last resort, when there are no other possibilities to solve optimization problems [19]. In addition to the experimental results, many metaheuristics, e.g. evolutionary algorithms, were proven to be actual global optimizers [21], [17]. Many real-world problems like scheduling, vehicle routing, and multi-modal optimization cannot be solved in a reasonable time without using metaheuristics.

The decomposition of metaheuristics [4] (construction of, e.g., a parallel evolutionary algorithm) has two goals; the first and most important is to increase the exploration possibilities (decrease the possibility of early convergence). The second goal may be taken for granted, as many subpopulations may be much easier to implement in such a way that utilizing concurrent, parallel, distributed, or GPGPU [14], [13] ,[20], infrastructure becomes fairly easy [12,18]. Very often, the decomposition leads to the creation of agent-based computing systems, where all the substantial parts of the computation are distinct and autonomous (see, e.g. [9]).

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97554-7_20

Evolutionary algorithms (EAs) are powerful, inspired by natural evolution, widely described in the literature [7], [10], [21] probabilistic optimization methods to solve a large scale of problems that can be described in an appropriate form. It works on population of individuals (an exemplary solution of the problem), randomly created, placed in a certain environment to which he may be better or worse adapted. The solution is coded (genotype) in a way that allows for quality comparison in the sense of the problem that is being solved. The work of evolutionary algorithms 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 compare individuals.

The scheme of the classical genetic algorithm includes the creation of an initial population and a series of operations (in a loop) performed until a termination condition is met. These loop operations include the use of genetic operators (i.e. certain transformations of the genetic code of individuals), calculating the value of the objective function of individuals, and selection. The final population in each cycle becomes the current for the 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 genetic operators: mutation, crossover, and selection can be used in different variants. 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 that we will get it in a time that satisfies us.

When we divide population in several parts, working separately, we can talk about Parallel Evolutionary Algorithm (PEA).

The subpopulations or islands have means for migration of individuals (to have the whole computing system work together on the problem). Of course, there must be some kind of topology of the island connections, the relation of neighborhood, etc. Usually, the islands are ordered in fully connected graphs, rings, torus [15], [3]. We should consider that the migration implemented actually supports the exchange of information, so the selected topology greatly affects this exchange. Network science studies graphs occurring in the real world, both in natural and human-made systems. [2] Since many of these systems are effective in solving optimization tasks, the results of network science may be informative in creating island connections. In particular, most models of network science generate randomized graphs with low (limited) average path lengths. The simplest of these are Erdős-Rényi graphs [5], which have the property that the expected number of steps needed to move between the two randomly chosen vertices is proportional to the natural logarithm of the number of vertices.

In this paper, we focus on checking the influence of using Erdős-Rényi (ER) graphs and compare them with original deterministically generated topologies. The next section gives an insight into related work, later we present our idea formally, then the experiments are shown, and finally the paper is concluded.

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97554-7_20

2 Topologies in parallel metaheuristics

In traditional parallelization of metaheuristics, the basic master-slave version is something to start with. This technically inspired idea serves as a means of delegating the computing of (usually) fitness to concurrent or distributed processes to speed up the execution of the entire algorithm [8]. The next popular version of parallel metaheuristics is the most interesting to us, at least in this paper—island model of evolution—it consists in decomposition of the population into islands (usually concurrent or distributed processes, but not always), running the algorithms individually with some migration methods present [6] in order to exchange the genetic information in the computing population. Finally, cellular genetic algorithms put the invididuals on a strictly defined structure, usually a lattice [1], which restricts their interaction, improving the diversity of the search. As mentioned, the island model is something we focus on here, in particular thinking about topologies of island connections.

In our previous research [3] we have used those popular island topologies (as other researchers did, see, e.g. [11]):

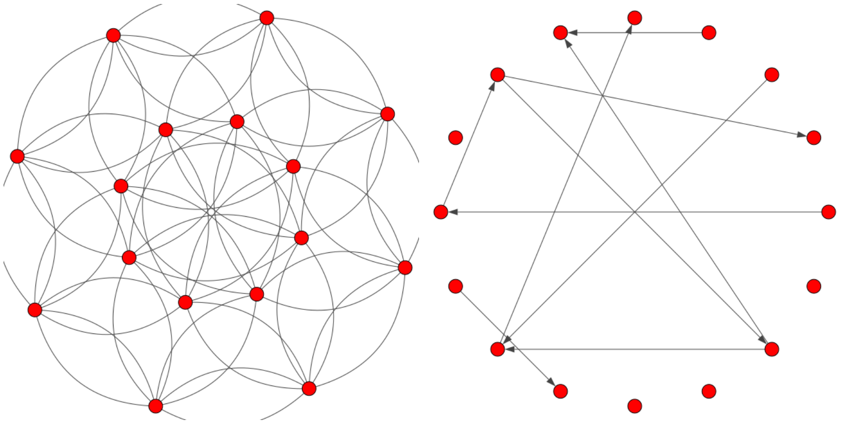
- ring, where each island sends migrants to only one neighbor and receives migrants only from one other neighbor, has the smallest number of connections; many islands work alone most of the time (asynchronous model), while their neighbors finish before them. We should keep in mind that in the worst case, the information (migrants) might have to travel the whole ring between two neighboring islands.
- torus, where islands are arranged in a matrix, islands at the right edge of the matrix connect to those on its left edge to form a cylinder, and then the islands at the bottom of the cylinder connect to those at the top—smaller number of connections between the islands is put together. Every island is connected with four other islands. Each island has a better chance of cooperation than in the ring topology.
- fully connected, where each island is connected to each other and can send or receive migrants to/from them, the information spreads quickly but we can clearly see that there is no such order which is brought by the above-cited ring and torus. This case of course generates very dense traffic and is difficult to scale when implemented concurrently,
- mesh, that is actually a striped-down torus, with a border islands having connections only towards the center of the mesh not to the opposite border (as it happens in the case of tori). In fact, we have not used this topology yet.

Other topologies are used, although the topic is not broadly addressed in the metaheuristic-related literature. Rucinski [16] adds several variations of the basic topologies, identifying the cartwheel, hypercube, several variations of rings, and chains, and compares them in detail regarding different parameters of migration, checking parallel versions of differential evolution, and simulated annealing—those algorithms prefer apparently simpler topologies (like chains).

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97554-7_20

The information spread in the network apparently affects the work of the whole algorithm; therefore, we strongly believe that researching topologies having certain features (such as e.g. ‘small world’) will bring interesting conclusions and help in developing novel reliable metaheuristics. Therefore, we would like to work on testing such topologies as Albert-Barabasi, Erdős-Rényi or Watts-Strogatz [2,5,22], in this paper focusing on the middle one.

3 Erdős-Rényi network topology in parallel evolutionary algorithm



(a) Torus (b) ER graph with probability 0.05

Fig. 1: Examples of the topologies used—in case of 16 nodes

The first topology that we have chosen to test as a means of managing the flow of genetic information in parallel metaheuristics is the Erdős-Rényi (ER) model, also known as random networks [5]. The concept of random graphs is based on the idea that networks have some random influence on their structures. They are defined by generating stochastic procedures (models) and their structural properties are analyzed statistically. The earliest, and arguably the simplest, such model is the Erdos-Renyi (ER) model.

The ER model has two generating parameters: the number of nodes ($N \in \mathbb{N}$) and the probability of edges ($p \in [0, 1]$). The defining assumption of the ER model is that *all possible edges* of the network are present with the exact same probability p. Therefore, $p = 0$ is expected to yield an empty network, while $p = 1$ will almost always result in a complete graph.

The networks generated by this stochastic model are attractive for several reasons. In addition to being the first randomized graph family proposed in history, they are also the simplest, least assuming. These networks remove the

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97554-7_20

spatial embedding of nodes present in both the ring and torus topologies and regard all nodes or parts of the network as equal, avoiding complex assumptions or generating procedures. Consequently, the expected number of links per node remains equal across the nodes, just as in the traditional regular topologies.

Moreover, ER networks have well-understood structural properties, two of which are of special importance to our studies. These networks become quasi-connected at low densities (with only a few links added). That is, a giant connected component starts to emerge as soon as having more than 1 edge per node on average. With about 2 edges per node, almost all nodes are expected to be in the large connected component. This corresponds to the lean, minimalistic structure of the ring.

The other important structural property of ER networks is that the average length of shortest paths between pairs of nodes in these graphs is proportional to the logarithm of the number of nodes. These short routes are in contrast to the path lengths found in ring or torus topologies, scaling linearly or with the square root of the number of nodes, respectively. By selecting ER networks as our topology, we can explore whether short path lengths have a positive impact on the performance of parallel evolutionary algorithms.

Fig. 1 illustrates such random graphs, comparing it to the classic torus structure. Ring topology, another regular topology considered in this paper, was omitted due to its triviality.

To construct our topologies, we create Erdos-Renyi graphs similar to the Ring and Torus with 150 nodes used previously. By “similar”, we mean the same number of nodes, or close to it. If the generated random graph is not fully connected, we consider only the largest fully connected component. Given the known emergence of a large connected component in ER networks, this simple procedure is expected to yield connected components with sizes comparable to the targeted number of nodes.

We also set the probability of edges (p) to a value that matches (close to) the density of our previously used graphs (Ring and Torus). This yields networks with about the same number of edges as in the traditional topologies. We experimented with 3 slightly different values p in each case and created 10 instances of ER graphs for each parameter setting.

Thus, based on the classic topologies and ER graphs, we are seeking to answer the following questions :

- How do randomized networks (from network science) perform compared to more traditional topologies?
- What structural properties of networks help performance?
- In particular, how does performance depend on the average path length (expected length of shortest paths)?

4 Experiments

A dedicated computing framework based on Python and Ray 6 technologies was deployed to the Ares HPC cluster located at ACC Cyfronet in Krakow, Poland.

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97554-7_20

The machine is equipped with 532 nodes of 48 core 2.90GHz Intel CPUs. The number of CPU cores required for the efficient running of the framework doubled the number of islands.

In this research, we have chosen one of the most popular benchmarks (De Jong's Sphere) in 200 dimensions. The parameters of our computing system were as follows: island model of the evolutionary algorithm, 150 islands, number of emigrants 5, migration interval 5, migration topology: ring, torus ($5x30$). The 16 individual population and the 4 individual offspring population were used. We performed each experiment ten times and averaged the results. The stop condition was the maximum 8000 of the evaluations.

We created Erdős-Rényi graphs similar to ring and torus with 150 nodes used previously. By "similar" we mean the same number of nodes and edges – or close in numbers. If the probabilistically generated network was not fully connected, we only used the largest connected component. In particular, we calibrated the generating parameter (p) of the ER graphs so that the expected density (i.e., number of edges) of the resulting ER networks be close to that of our previously used graphs (ring and torus). When generating ER graphs similar for ring or torus we used the Erdős-Rényi function from the *igraph* python library with paramerets: number of nodes 150 and probability coefficient p with a value close to the density of the ring or the torus, respectively (details below). We choose directed and self-loops allowed graphs.

When creating ER graphs for the ring (ERR), we first chose three coefficients p close to the density of the ring graph. The density of our ring was 0.016, so we set $p_1 = 0.017$, $p_2 = 0.0167$, $p_3 = 0.016715$. For each of them, we generated 10 graphs $ER(150, p)$.

We did the same for torus (ERT). The torus density was 0.03 so we set p coefficients: $p_1 = 0.03$, $p_2 = 0.031$, $p_3 = 0.032$. and for each of them we generated 10 graphs $ER(150, p)$.

Creating fully connected ER graphs with 150 vertices and probability coefficient p as established, is very unlikely—especially for rings where p is very small. Focusing on the fact that we want to check the feature of the "small world", we accept the possibility that some of the islands may not be connected to the large component of the graph. Since disconnected graphs are not useful island topologies, we only considered the largest component of these graphs. Therefore, some of our random topologies had fewer vertices—on average 142.7 for ERR and 148.17 for ERT. Tab. 1. summarizes the structural properties of our random topologies.

The nontrivial structural properties discussed are as follows:

- Large component size: Due to randomness in the process of creating ER graphs, there may be missing edges, which may cause the graph to become disconnected. This value measures the size of the largest group of connected islands. The size of the large component in the ER graphs is guaranteed to approximate the number of nodes for the appropriate generating parameters.

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97554-7_20

Table 1: Average parameters of graphs

graph type	large comp size	avg node degree	number of edges	avg path length	transitivity	density
RING	150	4	300	75	0	0.013
ERR	142,733	5,539	415,433	4.9	0.034	0.02
TORUS	150	10	750	6.27	0	0.03
ERT	148,166	9,422	706,666	3.39	0.06	0.03

- Density of the network: the quantity to assess the ratio of the number of existing links to that of possible links. As a ratio, its value is between 0 and 1. The empty network has a density of 0, while $D = 1$ corresponds to the complete network. For example, for directed networks with self-loops, the density is calculated as $D = E/N^2$ where E stands for number of edges.
- Transitivity: measures the ratio of full triangles to the number of triangles with at least two edges present. It measures the expected probability that two neighbors of a node are also connected.

Let us take a look at the results of our experiments. Table 2 shows the final values obtained during the execution of the algorithm for two topologies: ring and ERR. All the final values for all the repetitions were shown. Apparently the dispersity of the results is acceptable; moreover, the observed values clearly show that the final result is significantly lower for the ERR topology than for the original ring, even though the transitivity is non-zero. The density is almost two times higher for ERR than for the original ring.

Fig. 2 offers a different view of the same results. It is easy to see that the ER topologies prevail in all the cases examined. Moreover, shorter average path lengths and higher densities appear to have a positive effect on performance, whereas transitivity appears to have no effect.

We obtain similar results for the torus and ERT topologies, albeit the performance gain by the randomized topology is less. As shown in Table 3 and Fig. 3, the positive effect introduced by applying ER topologies still exist and it is clearly significant, yet a little closer to the original torus topology. The observations of the path length, density, and transitivity are similar to the case of ring and ERR.

Comparing the relation between structural properties such as average path length, density, and transitivity between the two topologies, one can observe that the results for torus have significantly bigger dispersion than the ones for the ring. This may be caused by a significantly larger complexity of the tori. The flow of the information there is also more free.

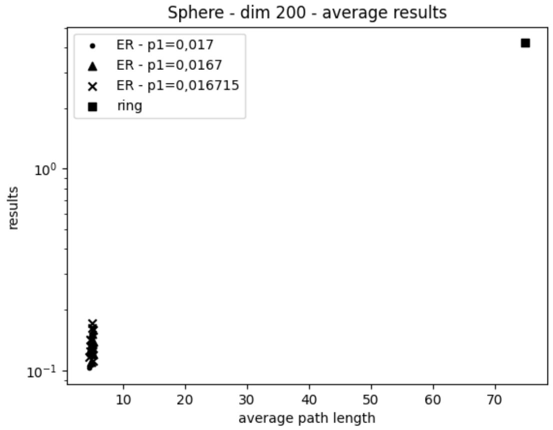
5 Conclusions

This paper proposed randomized graphs informed by network science as topologies for Parallel Evolutionary Algorithms. It showed that these probabilistically

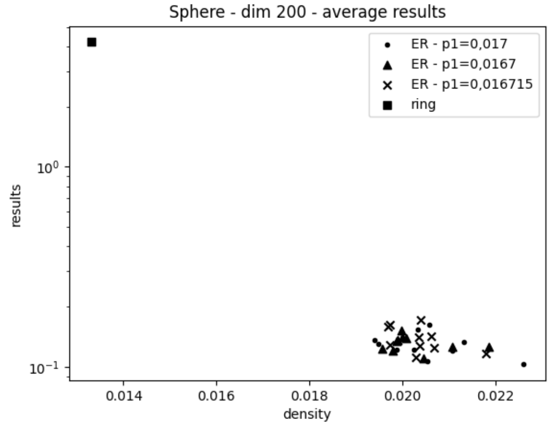
ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97554-7_20

Table 2: Final results obtained for original ring topology and ERR

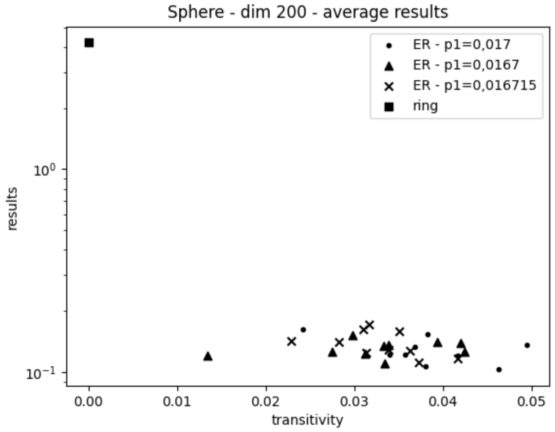
topology	av.result	transitivity	av.path length	density
RING 150	4,20107755	0	75	0,0133
ER(150, 0.017)				
) ERR1.1	0,102931556	0,046278	4,413345	0,022615
ERR1.2	0,121598635	0,033962	4,840688	0,020255
ERR1.3	0,153787762	0,038235	5,204186	0,020333
ERR1.4	0,120908372	0,041631	4,884067	0,021077
ERR1.5	0,122573216	0,035733	5,048267	0,019887
ERR1.6	0,13009082	0,033889	5,345297	0,019490
ERR1.7	0,106908988	0,038043	4,873218	0,020539
ERR1.8	0,133405843	0,036864	4,797705	0,021321
ERR1.9	0,136040905	0,049515	5,011800	0,019405
ERR1.10	0,161723777	0,024217	4,883656	0,020581
AVG	0,128996987	0,0378367	4,9302229	0,0205503
ER(150, 0.167)				
ERR2.1	0,123324736	0,031281	4,869923	0,019561
ERR2.2	0,134954751	0,033366	4,938551	0,019887
ERR2.3	0,138741786	0,041958	5,106787	0,020085
ERR2.4	0,125758136	0,042482	4,682728	0,021077
ERR2.5	0,136865906	0,033840	4,996857	0,019903
ERR2.6	0,141144968	0,039344	4,893255	0,020001
ERR2.7	0,126648094	0,027501	4,572775	0,021859
ERR2.8	0,12092268	0,013413	5,083735	0,019788
ERR2.9	0,152855756	0,029836	4,921682	0,019986
ERR2.10	0,109974475	0,033452	4,871930	0,020448
AVG	0,131119129	0,0326473	4,8938223	0,0202595
ER(150, 0.016715)				
ERR3.1	0,140073426	0,028302	4,737818	0,020351
ERR3.2	0,124639705	0,031392	4,650111	0,020680
ERR3.3	0,111499475	0,037259	5,042127	0,020294
ERR3.4	0,128683646	0,033873	5,127929	0,019724
ERR3.5	0,162834967	0,031004	4,958249	0,019724
ERR3.6	0,158955916	0,035097	5,171728	0,019689
ERR3.7	0,127461692	0,036313	4,891848	0,020383
ERR3.8	0,142400653	0,022890	4,658479	0,020631
ERR3.9	0,171848447	0,031651	5,000000	0,020392
ERR3.10	0,116457149	0,041684	4,485741	0,021810
AVG	0,138485508	0,0329465	4,872403	0,0203678



(a) impact of average path length on the results



(b) impact of density on the results



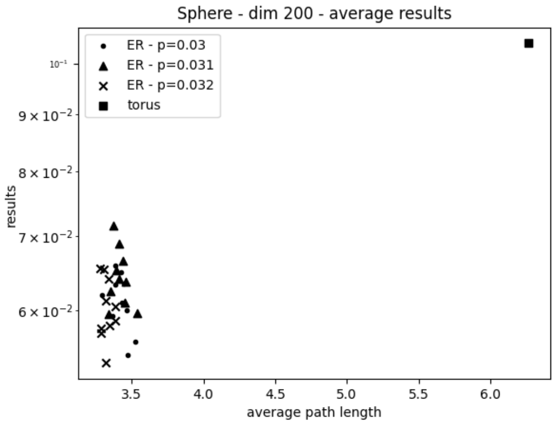
(c) impact of transitivity on the results

Fig. 2: Ring and Erdős-Rényi graphs similar to ring used as a topologies - average results

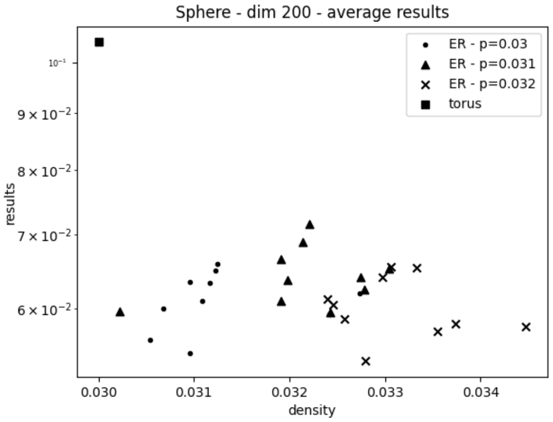
Table 3: Final results obtained for the original torus topology and ERT

topology	av.result	transitivity	av.path length	density
TORUS 150	0,104285714	0	6,27	0,03
ER(150, 0.03)				
ERT1.1	0,059986701	0,060595131	3,467259206	0,030679328
ERT1.2	0,063317426	0,056790123	3,384937615	0,031169767
ERT1.3	0,05476348	0,056201214	3,469006484	0,030953251
ERT1.4	0,059330074	0,067537783	3,364440212	0,032414171
ERT1.5	0,064965388	0,070201276	3,425701719	0,031227173
ERT1.6	0,060941192	0,075305292	3,432004716	0,031090212
ERT1.7	0,065877143	0,058813999	3,384071588	0,031244444
ERT1.8	0,056278588	0,055448997	3,526237557	0,030539165
ERT1.9	0,061959072	0,054401736	3,294803192	0,032733747
ERT1.10	0,063379919	0,054969749	3,444882782	0,030953251
AVG	0,061079898	0,06102653	3,419334507	0,031300451
ER(150, 0.031)				
ERT2.1	0,066473538	0,062208648	3,436357865	0,031291198
ERT2.2	0,071471297	0,065479056	3,37291113	0,032205756
ERT2.3	0,062441863	0,054945055	3,351716581	0,032779401
ERT2.4	0,065229815	0,056759379	3,391326181	0,033041788
ERT2.5	0,059593704	0,068891611	3,336966803	0,032430972
ERT2.6	0,064009313	0,0562954	3,412999414	0,032746273
ERT2.7	0,060925864	0,057517175	3,448737133	0,03191198
ERT2.8	0,068939602	0,067559055	3,412570288	0,032140248
ERT2.9	0,063675096	0,056943057	3,457201552	0,031977417
ERT2.10	0,059609675	0,062231359	3,540561516	0,03021889
AVG	0,064236977	0,06088298	3,416134846	0,032074392
ER(150, 0.032)				
ERT3.1	0,064007437	0,058064516	3,336336201	0,032971488
ERT3.2	0,053815851	0,062325033	3,316922661	0,032791316
ERT3.3	0,061254099	0,067689531	3,317225951	0,0324
ERT3.4	0,060483592	0,065005417	3,382187557	0,032459825
ERT3.5	0,057778086	0,058551395	3,285257727	0,034476376
ERT3.6	0,058201976	0,071449704	3,347395334	0,033735943
ERT3.7	0,057270287	0,068379991	3,284181694	0,033550835
ERT3.8	0,05876257	0,06191459	3,384524081	0,032579018
ERT3.9	0,065395214	0,060202139	3,275708301	0,033061574
ERT3.10	0,065306402	0,06096832	3,306398222	0,033327246
AVG	0,060227551	0,063455064	3,323613773	0,033135362

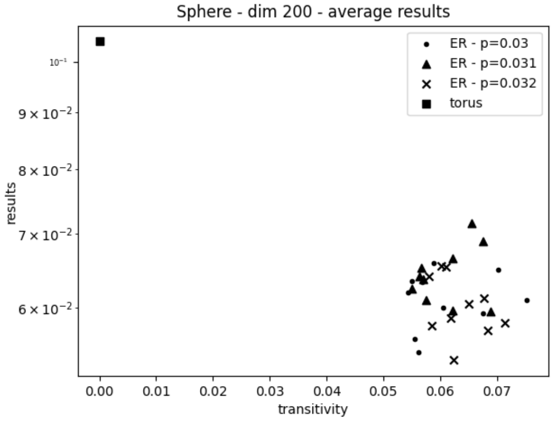
ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97554-7_20



(a) impact of average path length on the results



(b) impact of density on the results



(c) impact of transitivity on the results

Fig. 3: Torus and Erdős-Rényi graphs similar to torus used as a topology - average values

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97554-7_20

generated graphs form an interesting and efficient alternative to standard models. The results show that the flow of genetic information carried by the migrants is heavily affected by the topology of the island connections. It was shown that using Erdős-Rényi graphs introduced a significant change in the quality of the final results, even for the not so hard problem (Sphere), provided that the possibility of having separate islands is properly handled.

Encouraged by the obtained results we plan to continue by studying the use of other random graphs (e.g. Watt-Strogatz, Barabási-Albert, etc.) as topologies and check the influence of the structural properties if these graphs on the efficiency and on the exploration and exploitation capabilities of the examined algorithms.

Acknowledgements

The research presented in this article has been financially supported by: Polish National Science Center Grant No. 2019/35/O/ST6/00571 “Parallelization of metaheuristics with desynchronization.” (SB); NAWA - Polish National Agency for Academic Exchange - Program: NAWA Preludium BIS 1 (SB); Polish Ministry of Science and Higher Education assigned to the AGH University of Science and Technology. We gratefully acknowledge Polish high-performance computing infrastructure PLGrid (HPC Center: ACK Cyfronet AGH) for providing computer facilities and support within computational grant no. PLG/2024/017346. ARTIQ project – Polish National Science Center:DEC-2021/01/-2/ST6/00004, Polish National Center for Research and Development, DWP/ARTIQI/426/-2023.

References

1. Alba, E., Dorronsoro, B.: Cellular Genetic Algorithms. Springer-Verlag (2008)
2. Albert, R., Barabasi, A.: Statistical mechanics of complex networks. Reviews of Modern Physics **74**(47), 47–97 (2002)
3. Biełaszek, S., Nowak, A., Gądek, K., Byrski, A.: Delays in Computing with Parallel Metaheuristics on HPC Infrastructure. In: Nguyen, N.T., Franczyk, B., Ludwig, A., Núñez, M., Treur, J., Vossen, G., Kozierkiewicz, A. (eds.) Computational Collective Intelligence. pp. 163–175. Springer Nature Switzerland, Cham (2024). `https://doi.org/10.1007/978-3-031-70819-0_13`
4. Cantu-Paz, E., Goldberg, D.E.: Efficient parallel genetic algorithms: theory and practice - ScienceDirect (2000), `https://www.sciencedirect.com/science/article/pii/S0045782599003850`
5. Erdos, P., Renyi, A.: On random graphs. i. Publicationes Mathematicae **6**(3–4), 290–297 (1959)
6. Falcón-Cardona, J.G., Hernández Gómez, R., Coello Coello, C.A., Castillo Tapia, M.G.: Parallel multi-objective evolutionary algorithms: A comprehensive survey. Swarm and Evolutionary Computation **67**, 100960 (2021). `https://doi.org/https://doi.org/10.1016/j.swevo.2021.100960`, `https://www.sciencedirect.com/science/article/pii/S221065022100122X`

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97554-7_20

7. Goldberg, D.E.: Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley, Reading, MA (1989)
8. Gong, Y.J., Chen, W.N., Zhan, Z.H., Zhang, J., Li, Y., Zhang, Q., Li, J.J.: Distributed evolutionary algorithms and their models: A survey of the state-of-the-art. Applied Soft Computing **34**, 286–300 (2015). `https://doi.org/https://doi.org/10.1016/j.asoc.2015.04.061`, `https://www.sciencedirect.com/science/article/pii/S1568494615002987`
9. Kisiel-Dorohinicki, M.: Evolutionary multi-agent systems in non-stationary environments. Computer Science **14**(4), 563 (Nov 2013). `https://doi.org/10.7494/csci.2013.14.4.563`, `https://journals.agh.edu.pl/csci/article/view/59`
10. Michalewicz, Z., Schoenauer, M.: Evolutionary algorithms. In: Bidgoli, H. (ed.) Encyclopedia of Information Systems, pp. 259–267. Elsevier, New York (2003). `https://doi.org/https://doi.org/10.1016/B0-12-227240-4/00065-4`, `https://www.sciencedirect.com/science/article/pii/B0122272404000654`
11. Mühlenbein, H., Schomisch, M., Born, J.: The parallel genetic algorithm as function optimizer. Parallel Computing **17**, 619–632 (1991)
12. Najdek, M., Paciorek, M., Turek, W., Byrski, A.: Three new design patterns for scalable agent-based computing and simulation. Informatica (VILNIUS) (2024)
13. Pietron, M., Byrski, A., Kisiel-Dorohinicki, M.: Leveraging heterogeneous parallel platform in solving hard discrete optimization problems with metaheuristics. Journal of Computational Science **18**, 59 – 68 (2017)
14. Pietron, M., Byrski, A., Kisiel-Dorohinicki, M.: Gpgpu for difficult black-box problems. Procedia Computer Science **51**, 1023 – 1032 (2015)
15. Ruciński, M., Izzo, D., Biscani, F.: On the impact of the migration topology on the island model. In: Parallel Computing, Issues 10–11, October–November 2010, Pages 555-571. vol. 36 (1993)
16. Ruciński, M., Izzo, D., Biscani, F.: On the impact of the migration topology on the island model. Parallel Computing **36**(10), 555–571 (2010). `https://doi.org/https://doi.org/10.1016/j.parco.2010.04.002`, `https://www.sciencedirect.com/science/article/pii/S0167819110000487`, parallel Architectures and Bioinspired Algorithms
17. Schaefer, R., Byrski, A., Smolka, M.: The island model as a markov dynamic system. Int. J. Appl. Math. Comput. Sci. **22**(4), 971–984 (2012). `https://doi.org/10.2478/V10006-012-0072-Z`, `https://doi.org/10.2478/v10006-012-0072-z`
18. Stypka, J., Anielski, P., Mentel, S., Krzywicki, D., Turek, W., Byrski, A., Kisiel-Dorohinicki, M.: Parallel patterns for agent-based evolutionary computing. Computer Science **17**(1), 83 (Apr 2016). `https://doi.org/10.7494/csci.2016.17.1.83`, `https://journals.agh.edu.pl/csci/article/view/1395`
19. Talbi, E.: Metaheuristics: From Design to Implementation. WIley (2009)
20. Tsutsui, S., Collet, P.: Massively Parallel Evolutionary Computation on GPGPUs. Springer Publishing Company, Incorporated (2013)
21. Vose, M.D.: What are Genetic Algorithms? A Mathematical Prespective. In: Evolutionary Algorithms, pp. 251–276. The IMA Volumes in Mathematics and its Applications, Springer, New York, NY (1999)
22. Watts, D.J., Strogatz, S.H.: Collective dynamics of 'small-world' networks. Nature **393**(6684), 440–442 (Jun 1998). `https://doi.org/10.1038/30918`, `https://www.nature.com/articles/30918`, publisher: Nature Publishing Group

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97554-7_20