

Population network structure impacts genetic algorithm optimisation performance

Aymeric Vié

Mathematical Institute, University of Oxford
Institute of New Economic Thinking, University of Oxford
Oxford, United Kingdom
vie@maths.ox.ac.uk

ABSTRACT

A genetic algorithm (GA) is a search method that optimises a population of solutions by simulating natural evolution. Good solutions reproduce together to create better candidates. The standard GA assumes that any two solutions can mate. However, in nature and social contexts, social networks can condition the likelihood that two individuals mate. This impact of population network structure over GAs performance is unknown. Here we introduce the Networked Genetic Algorithm (NGA) to evaluate how various random and scale-free population networks influence the optimisation performance of GAs on benchmark functions. We show evidence of significant variations in performance of the NGA as the network varies. In addition, we find that the best-performing population networks, characterised by intermediate density and low average shortest path length, significantly outperform the standard complete network GA. These results may constitute a starting point for network tuning and network control: seeing the network structure of the population as a parameter that can be tuned to improve the performance of evolutionary algorithms, and offer more realistic modelling of social learning.¹

CCS CONCEPTS

• **Networks** → *Network architectures*; • **Computing methodologies** → **Search methodologies**;

KEYWORDS

Genetic Algorithms, Learning, Optimisation, Population Network

ACM Reference Format:

Aymeric Vié. 2021. Population network structure impacts genetic algorithm optimisation performance. In *2021 Genetic and Evolutionary Computation Conference Companion (GECCO '21 Companion)*, July 10–14, 2021, Lille, France. ACM, New York, NY, USA, 4 pages. <https://doi.org/10.1145/3449726.3463134>

¹All source code for the NGA, the figures and the results are available at <https://github.com/aymericvie/networked-genetic-algorithms>.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

GECCO '21 Companion, July 10–14, 2021, Lille, France

© 2021 Copyright held by the owner/author(s). Publication rights licensed to the Association for Computing Machinery.

ACM ISBN 978-1-4503-8351-6/21/07...\$15.00

<https://doi.org/10.1145/3449726.3463134>

1 INTRODUCTION

A genetic algorithm (GA) is a search method that optimises a population of individuals by simulating natural evolution [4, 12] and reproduction of the fittest. Individuals in the population are assumed to be able to mate with any other. Such a connection between all individuals can be seen in a network science perspective as a constant, fully connected network in the population. However, in social learning as in nature, populations are not fully connected, as individuals only interact with a finite subset of the whole [10]. The network structure of these interactions have different structures, that can be an important determinant of population-level dynamics [11], and accounts in nature for changes in population genetic diversity and the emergence of different species [2]. While network design in operations research [8] or neural architecture search [6] are active areas of research, there have been no attempts so far to investigate alternative GA network architectures.

It is unknown whether and to what extent this assumption of completeness and the population network structure impact GAs performance. In this article, we introduce *Networked GAs* (NGA) that use alternative population network structures, and measure their performance over benchmark optimisation tasks to identify the impact of network architectures. We consider various population structures generated by Erdos-Rényi random networks [3] and Albert-Barabasi scale-free networks [1].

We show evidence of significant variations in performance of the NGA as the network structure varies. We compare optimisation performance to various network metrics such as *density*, which measures the ratio of realised links to the number of possible links; and *average shortest path length*, which describes the average minimum number of links to be crossed to connect any pair of individuals. NGA performance is higher with intermediate levels of density and low average shortest path length, settings that favor circulation of fittest genomes while preventing those fittest solutions from becoming dominant too fast. In addition, we discover that the optimal network NGA outperforms the standard GA with a fixed complete network structure.

The results open the possibility of *network tuning*: identifying the optimal population network architecture to maximise the GA optimisation performance. It is also possible that different network structure could be optimal during the run. *Network control*, i.e. allowing this population network to change over time, could further improve the performance of GAs, model social learning and make

Table 1: Domains of test functions

Function	Name	Domain
f_1	Rastrigin function	$-5.12 \leq x_1 \leq d \leq 5.12$
f_2	Sphere function	$-5.12 \leq x_1 \leq d \leq 5.12$
f_3	Ackley function	$-32.768 \leq x_1 \leq d \leq 32.768$

one step towards a closer implementation of biological features in evolutionary algorithms [7].

2 ALGORITHM

2.1 Test functions

To compare the performance of the GA with respect to the population network structure, we choose three popular test functions as benchmarks: the Rastrigin function (1), the Sphere function (2), and the Ackley function (3), with respective domains in Table 1 and dimensionality $d = 2$. Their global optimum is at $f(\mathbf{0}) = 0$.

$$f_1(\mathbf{x}) = 10d + \sum_{i=1}^d (x_i^2 - 10 \cos 2\pi x_i) \quad (1)$$

$$f_2(\mathbf{x}) = \sum_{i=1}^d x_i^2 \quad (2)$$

$$f_3(\mathbf{x}) = -20 \exp \left(-0.2 \sqrt{\frac{1}{d} \sum_{i=1}^d x_i^2} \right) - \exp \left(\frac{1}{d} \sum_{i=1}^d \cos 2\pi x_i \right) + e + 20 \quad (3)$$

2.2 The networked-population genetic algorithm

2.2.1 Genetic representation and sampling. To optimise the test functions with dimensionality d , each individual is a string of d real numbers sampled uniformly in the corresponding search domain of Table 1. A total of n individuals are generated.

2.2.2 Genetic operators. For each individual i , the resulting value $f(i)$ of the test function is computed. As the global minima of our test functions are equal to 0, we strive to minimise $f(i)^2$. To create two offspring, the first individual i is chosen with a fitness-proportionate method, with selection probability:

$$\pi_i = \frac{\hat{f}(i)}{\sum_{j=1}^n \hat{f}(j)^2} \quad \text{with } \hat{f}(i) = \frac{1}{1 + f(i)} \quad (4)$$

The networked-population GA (NGA) differs from the standard GA in the selection of the second individual to mate. If individual k has been selected in the population, the second will be selected among the $N(k)$ individuals sharing a link with individual k , instead of performing again selection on the full population as in the standard GA. Probabilities of selection are identical to equation 4 applied to the subset $N(k)$ rather than the full population size n .

Once the two parents have been selected, uniform crossover is performed with probability ρ . A crossover point in $[0, d]$ will be randomly and uniformly determined, and chromosomes of parents

Table 2: GA parameter configuration

Parameter	Meaning	Value
n	Population size	50
ρ	Crossover rate	0.7
μ	Mutation rate	0.05
τ	Number of iterations	100

will be exchanged after the crossover point to generate two offspring. With probability $1 - \rho$, the individuals instead create exact copies of themselves. Finally, random Gaussian mutations occur with probability μ , and change the value of the mutated element by $\epsilon \sim \mathcal{N}(0, 1)$. The algorithm runs for τ time periods. Standard parameter values are used for the NGA, and presented in Table 2.

2.2.3 The population network structure. In the canonical genetic algorithm, the crossover operator can involve any combination of individuals in the population, corresponding to complete network NGA. In the NGA, we consider Erdos-Rényi (ER) random networks and Albert-Barabasi (AB) scale-free networks. Population network structure is kept static during search. ER graphs are described by the link formation probability p , from 0 (empty network, or island GA model without migration) to 1 (complete network, standard GA). AB graphs are described by the intensity of the preferential attachment m , generating tree networks for low values of m , then graphs with multiple hubs, up to star networks for highest values of m . Figure 2 shows examples of the generated networks. As these two network parameters change, so do the corresponding network features. ER networks become connected (i.e. there exists a path between any pair of nodes) at $p = \frac{\ln n}{n}$ [3], and increase monotonically in density with p . In AB graphs, m determines the level of density, and the average shortest distance between any pair of nodes [1]². This population network structure is drawn once at the start of the run, and left constant for its duration.

3 RESULTS

3.1 NGA performance over iterations

Various network structures visibly affect the performance of the GA in Figure 3. Average fitness at the final iteration $\tau = 100$ and convergence speed vary with the network type (random or scale-free) and parameters (p and m). In particular, the empty network (top left of Figure 2) and the star network (bottom right) achieve a higher fitness score than their counterparts, showing evidence for weaker performance in all benchmark tasks, with differences attributed to the respective task difficulty. Comparing the performance of the complete network (top right) commonly used in GAs with other structures suggests that alternative network structures may offer better optimisation performance. Notably, optimising the Ackley function appears to be more successful in an ER graph with $p = 0.5$ and with the AB graph with $m = 25$. These particular results invite us to analyse more comprehensively how benchmark performance varies with the link formation probability p in ER networks, and with the preferential attachment factor m in AB graphs.

²We refer the reader to the GitHub repository for a graphical representation of the impact of network parameters p and m over network features.

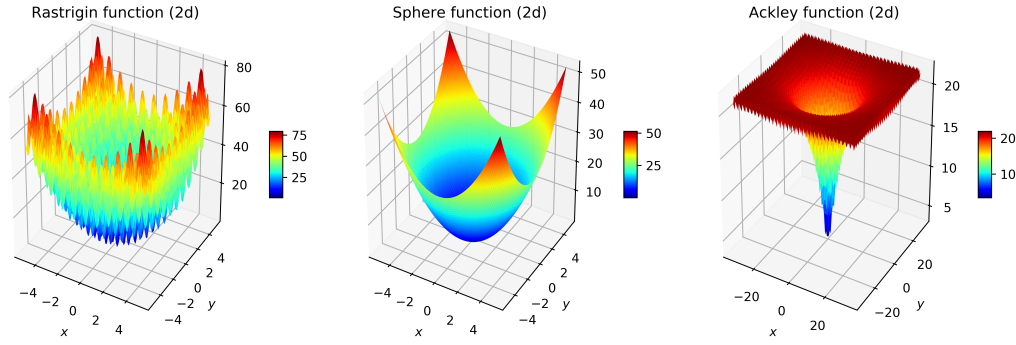


Figure 1: Fitness landscapes of the three test functions (Rastrigin, Sphere, Ackley) with $d = 2$.

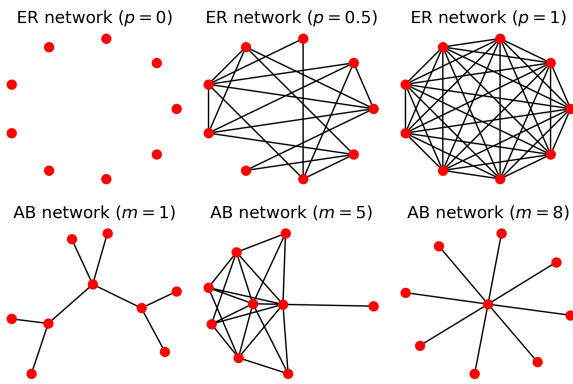


Figure 2: 6 examples of Erdos-Rényi (ER, [3]) and Albert-Barabasi (AR, [1]) networks with 10 nodes

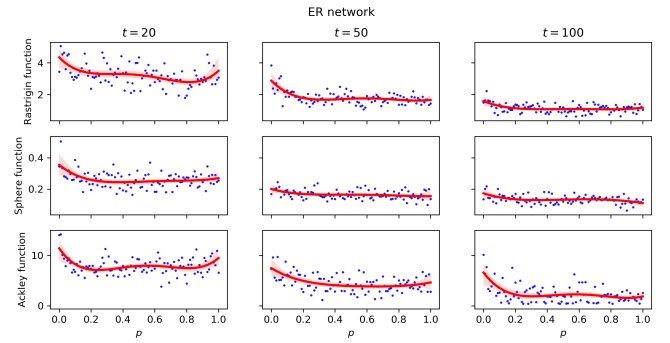


Figure 4: Average fitness for the test functions optimisation in ER networks with link probability p , with confidences intervals over 10 repetitions for each p

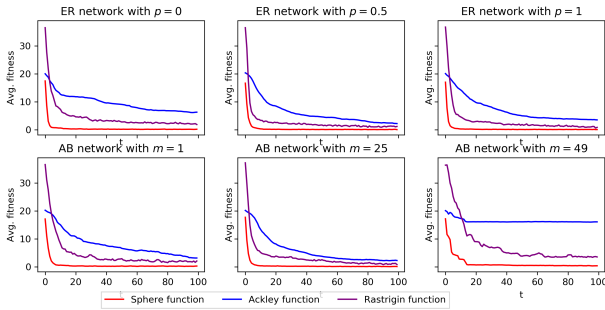


Figure 3: Average fitness for the three test functions as a function of time, for various network types and parameters, with 10 repetitions

3.2 The impact of network structure over performance

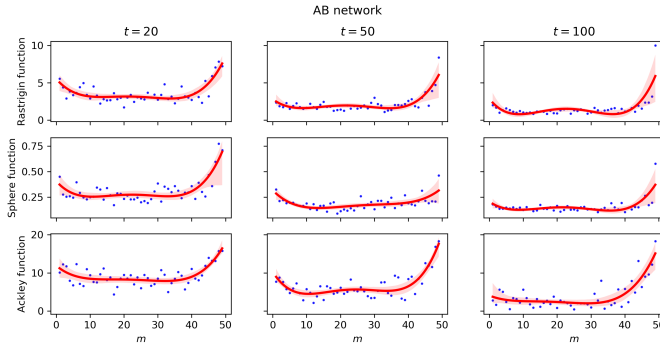
We first evaluate how the link formation probability p in Erdos-Rényi networks changes NGA performance, shown in Figure 4. For each value of $p \in [0, 1]$ with an increment of 0.01, we run the algorithm 10 times and record the average fitness performance at

$t = 20$, $t = 50$ and $t = 100$. As the population network becomes connected at $p = 0.046$, a significant improvement of performance is observed. Below this threshold, as crossover is heavily limited by the disconnected network, most of the evolution happens through the mutation operator only, limiting the performance of the NGA. Above the connectedness threshold, the average fitness moderately decreases, exhibiting peaks of lower average fitness in the range $p \in [0.2, 0.3]$ and $p \in [0.8, 0.9]$. The polynomial fits (order 4) suggest that such intermediate levels of p allow a better performance than the complete network.

We then study the influence of the preferential attachment factor m in Albert-Barabasi scale-free networks. m varies from 1 to 49 with an increment of 1. Tree networks (low m), and star networks (highest m) perform significantly worse in all benchmark tasks, as shown by Figure 5. Tree networks are indeed connected, but exhibit high shortest path lengths, limiting the circulation of good solutions in the population. Star networks are vulnerable to errors or premature local dominance of the central hub. Polynomial fits suggest again the existence of two performance peaks for values of m close to 10, and to 30, that coincide with intermediate levels of network density but low average shortest paths lengths, similar to biological networks' characteristics [5].

Table 3: Average fitness in the standard GA (GA) and the best average fitness of the NGA across all networks (ER* & AB*)

Function & Time	$\tau = 20$			$\tau = 50$			$\tau = 100$		
Rastrigin	3.067	1.796	1.705	1.445	1.072	1.093	1.043	0.625	0.82
Sphere	0.257	0.164	0.172	0.138	0.1	0.09	0.135	0.064	0.103
Ackley	6.597	3.859	4.328	2.809	1.171	2.157	2.568	0.412	0.451
Network	GA	ER*	AB*	GA	ER*	AB*	GA	ER*	AB*

**Figure 5: Average fitness for the test functions optimisation in AB networks with preferential attachment factor m , with confidences intervals over 10 repetitions for each m**

3.3 Network tuning outperforms the standard GA

The optimal-network NGA significantly outperforms the standard GA in all benchmark tasks. The best alternative NGA (ER or AB) exhibits average fitness scores that are 53% lower than those obtained by the standard GA (Table 3). Intermediate ER networks are often the best performing structure. The best solutions' potential for dominance and premature convergence to local optima is limited by the network incompleteness, while the dense network structure still allows fittest individuals to circulate. This translates into particularly high performance gains in multimodal landscapes, e.g. the Ackley function. Like optimisation of neural network architectures [9], the GA population networks may be tuned to improve GAs performance. Though robust across our three test functions, repeated runs and various time horizons, future investigation with higher dimensionality, alternative configurations to Table 2 and test functions could support these findings. Efficiency gains in the network selection operator are also required to increase repetitions. *Network tuning* (architecture optimisation before the run), or *network control* (during the run) can unlock new performance in population-based search algorithms for optimisation and social learning tasks. Further research could explore what evolution mechanisms could converge to optimal population network structures.

4 CONCLUSIONS

We introduced Networked GAs, that constrain crossover in a network. We considered varied population network structure in random and scale-free graphs. Using NGAs to optimise the Ackley, Rastrigin and Sphere test functions, we found evidence of significant changes in the average population fitness at given times as the population network structure varies. The best-performing networks significantly outperform the standard complete network GA in all benchmark tasks, suggesting that the population network structure, as other GA parameters, could be tuned to improve performance.

ACKNOWLEDGMENTS

The author thanks Doyne J. Farmer, Alissa M. Kleinnijenhuis, Renaud Lambiotte, José Moran, Manpreet Singh and two anonymous reviewers for precious support and advice.

REFERENCES

- [1] Albert-László Barabási and Réka Albert. 1999. Emergence of scaling in random networks. *science* 286, 5439 (1999), 509–512.
- [2] Thomas Broquet, Sonia Angelone, Julie Jaquier, Pierre Joly, JEAN-PAUL LENA, Thierry Lengagne, Sandrine Plenet, Emilien Luquet, and Nicolas Perrin. 2010. Genetic bottlenecks driven by population disconnection. *Conservation Biology* 24, 6 (2010), 1596–1605.
- [3] Paul Erdos, Alfréd Rényi, et al. 1960. On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci.* 5, 1 (1960), 17–60.
- [4] John H Holland. 1992. Genetic algorithms. *Scientific american* 267, 1 (1992), 66–73.
- [5] Joost Huizinga, Jean-Baptiste Mouret, and Jeff Clune. 2016. Does aligning phenotypic and genotypic modularity improve the evolution of neural networks?. In *Proceedings of the Genetic and Evolutionary Computation Conference 2016*. 125–132.
- [6] Zhichao Lu, Ian Whalen, Vishnu Boddeti, Yashesh Dhebar, Kalyanmoy Deb, Erik Goodman, and Wolfgang Banzhaf. 2018. Nsga-net: a multi-objective genetic algorithm for neural architecture search. (2018).
- [7] Risto Miikkulainen and Stephanie Forrest. 2021. A biological perspective on evolutionary computation. *Nature Machine Intelligence* 3, 1 (2021), 9–15.
- [8] Mahmoud Owais and Mostafa K Osman. 2018. Complete hierarchical multi-objective genetic algorithm for transit network design problem. *Expert Systems with Applications* 114 (2018), 143–154.
- [9] Kenneth O Stanley and Risto Miikkulainen. 2002. Evolving neural networks through augmenting topologies. *Evolutionary computation* 10, 2 (2002), 99–127.
- [10] Demival Vasques Filho and Dion RJ O’Neale. 2020. Transitivity and degree assortativity explained: The bipartite structure of social networks. *Physical Review E* 101, 5 (2020), 052305.
- [11] Aymeric Vié. 2019. Information Selection Efficiency in Networks: A Neurocognitive-Founded Agent-Based Model. In *Network Theory and Agent-Based Modeling in Economics and Finance*. Springer, 11–34.
- [12] Aymeric Vié. 2020. Qualities, challenges and future of genetic algorithms: a literature review. *arXiv preprint arXiv:2011.05277* (2020).