2023 IEEE Symposium Series on Computational Intelligence (SSCI)
Mexico City, Mexico. December 5-8, 2023

Effects of Optimal Genetic Material in the Initial Population of Evolutionary Algorithms

Tobias Benecke
Otto-von-Guericke-University
Magdeburg, Germany
tobias.benecke@ovgu.de

Sanaz Mostaghim
Otto-von-Guericke-University
Magdeburg, Germany
sanaz.mostaghim@ovgu.de

***Abstract*—The quality of individuals in evolutionary algorithms (EAs) is usually measured in terms of their fitness. If an individual has a good fitness, a good genome is assumed. However, a good fitness value does not guarantee that the individual can produce good offspring and guide the algorithm towards the global optimum. Answering the question of what makes a genome good is not trivial, especially when considering different types of crossover operators, copying or combining genome values. This work aims towards answering this question by evaluating the influence of optimal gene values in the initial population of EAs. In computational experiments, a random population is seeded with generated individuals of different fitness qualities and containing different amounts of optimal genetic material. Tests are done for multiple dimensions and with crossover operators copying or combining the parents genes to the offspring. Data is evaluated both in terms of algorithmic performance and population dynamics, clearly showing the influence of optimal gene values.**

***Index Terms*—evolutionary algorithm, initial population, seeding, optimal gene values, population dynamics**

I. Introduction

Individuals in evolutionary algorithms (EAs) are represented in two spaces, the solution space represented by the fitness value, and the search space, represented by the genome. The quality of an individual is usually only measured in the form of its fitness. This makes sense for measuring the final performance, as the fitness function is designed to measure the quality of the phenotype for a problem. However, individuals in EAs do not only represent the solutions found by the algorithm at any given time, but also build the base for the ongoing search with the goal of finding the global optimum. This is especially true for the initial population, which is the starting point for each EA. It contains the genetic material which will be recombined through crossover and mutated until the termination criterion is met.

The question ”what makes a genome good” is not trivial to answer. In previous works, the traceable evolutionary algorithm (T-EA) was developed to evaluate the population dynamics of EAs [1]. A first evaluation on the combinatorial Knapsack problem showed the best fitness individuals to not always feature the highest influence on the result [1]. Later extensions to the T-EA showed the same for continuous [2] and multi-objective cases [3]. This raises the question of what is a good genome at the beginning of the evolutionary process.

This paper starts towards this by evaluating the effect of seeding the initial population in EAs with different types of genome configurations. Individuals featuring optimal genetic material but a few ”bad” genes far off their optimal value will be compared to individuals where each gene is roughly the same distance to their respective optimum. What configuration will be better is not trivial, especially when considering different crossover operators. For operators which are designed to (partially) copy the genome values directly from the parents to the offspring, such as uniform crossover (UX) [4], it is intuitively better to have the already optimal genetic material. However, for crossover operators that combine two genome values, like the popular simulated binary crossover [5], this is different. As the operator combines the genome of both parents, it might be more desirable for all genes to be close to their respective optimal value. Furthermore, if an individual already has a good fitness, a good genome is assumed. However, it is unclear if the algorithm can take advantage of good genome values seeded into a bad fitness individual.

To answer these questions, this paper studies the effects of seeding the initial population with individuals featuring different genome distributions in computational experiments. To achieve this, individuals with different amounts of optimal genetic material need to be generated by solving the benchmarking function for a specific target fitness. The experiments are performed on a genetic algorithm (GA) featuring both the genome copying UX [4] and the genome combining SBX [5] operators in multiple dimensions. Evaluation of the results is done both in terms of performance gains of the algorithm, and in terms of population dynamics in the search space. With this, we can directly link performance enhancements to the seeded individuals. The results clearly show the differences in performance and population dynamics when including optimal genes in the initial population.

II. Theoretical Background

Initialization techniques for EAs have been studied in the past, for example in terms of seeding the algorithm with specific solutions [6], [7]. Other, more general approaches use gap search [8] or quasi-random sequences [9] to sample

This work is part of the Research Initiative ”SmartProSys: Intelligent Process Systems for the Sustainable Production of Chemicals” funded by the Ministry for Science, Energy, Climate Protection and the Environment of the State of Saxony-Anhalt.

higher dimensional spaces more evenly. This includes the well-known Latin hypercube sampling [10]. However, the results are often evaluated only in terms of performance gains, like accuracy and convergence speed, and not in terms of population dynamics. This leaves a knowledge gap for why some genome values produced for the initial population work better than others.

In general, the field of evaluating population dynamics in EAs has been gaining interest in the past years. Early attempts use dimensionality reduction techniques to visualize the behavior of EAs in the otherwise large dimensional search space in a humanly understandable way [11], [12]. More recently, Ochoa et. al. introduced the concept of search trajectory networks [13], visualizing the path of a population in the search space as a network-based model. This allows to evaluate how well the algorithms traverse through the search space, showing the frequently visited areas and deceptive points where the algorithm gets stuck.

In this paper, the population dynamics are evaluated with the genome heritage tracking approach of the T-EA [2]. The T-EA was developed for combinatorial and integer representations in [1] and later extended to real-valued problems in [2]. In the T-EA, each gene in the initial population is initialized with a traceID. In practical terms, the traceID is a marker for a specific gene, which points back to the individual in the initial population, where the gene was originated. Updating the traceIDs alongside the genome in the crossover operation allows for tracking the origin of a gene throughout the EA. The influence of the mutation operator can also be tracked by assigning a dedicated traceID for mutations.

With this heritage information, the influence of individuals from the initial population can be calculated. In [2], the counting impact metric is used to calculate the impact $I(x)$ of an individual x on the result. The impact is calculated by counting the amount of genetic material found for a specific individual. If an individual has a high impact on the result, a high amount of genetic material was found in the final population. If the individual had no impact on the result ($I(x) = 0$), no genetic material of it was found.

III. Generating the Seed Individuals

To investigate the effect of optimal genes in individuals on the population dynamics and performance of EAs, seed individuals of the same fitness need to be generated to compare their performance. For this, the benchmarking function needs to be solved for a given genome. In this paper, the sphere function (Equation 1) is used, as it allows for an easy calculation of the genome from a given fitness. Furthermore, it does not have locally optimal solutions, meaning the genome of two individuals with the same fitness will also have the same distance to the optimum in the search space. The genome of an individual with the genome size n is notated as $x = [x_1, x_2, ..., n_n]$.

$$f(x) = \sum_{i=1}^{n} x_i^2 \tag{1}$$

The optimal solution for the sphere function is known as $x_i = 0 \ \forall i \in n$. This means the number of optimal genes in a genome can be altered by setting them to zero. Assuming a genome size of n, of which o genes are optimal, the value of the r remaining genes for a given target fitness f_{target} can be calculated systematically with Equation 2.

$$x_i = \sqrt{\frac{f_{target}}{r}} \ , \forall i \in r \tag{2}$$

The individual of a given target fitness can then be constructed by calculating the first r genes with Equation 2 and setting the remaining o genes to zero. As an example, the value of the not optimal gene for an individual with the target fitness of $f_{target} = 2$ and two not optimal genes ($r = 2$) would be $\sqrt{2/2} = 1$, resulting in the genome $x = [1, 1, 0, 0]$. The same example with three optimal genes would result in the genome $x = [1.414, 0, 0, 0]$.

This means that individuals which have a high number of optimal genes feature a few genes that are very far away from their optimal value, dragging down the fitness of the otherwise good individual. This might be desirable for crossover operators which copy the genome values directly from the parent to the offspring, as the already optimal value is kept. If no genes are optimal, the distance of each gene to their respective optimal value is lower. In theory, this might be more desirable for crossover operators that combine genome values instead of copying them. However, the evaluation in section V shows a different result.

IV. Experimental Setup

To test the differences in the amount of optimal genes in an individual, the performance when seeding the algorithm with selected individuals is compared. For this, four types of seed individuals of the same fitness are used, with 75%, 50%, 25%, and 0% of optimal genes. In this way, it can be examined if it is more desirable to have a few bad genes in an otherwise optimal genome, or if it is better for no genes to be optimal but each gene individually being closer to its optimal value.

Tests are repeated in four different dimensions $d = [4, 8, 16, 32]$. For each dimension, an initial population with the size of 19 is generated, resulting in a size of 20 when adding the seed individual. The same initial population is used for each dimension to keep the impact data comparable.

The seed individuals are generated for four different fitness levels, depending on the initial population used for each dimension. First, the seed is generated as the best individual in the population, having half the fitness value of the otherwise best individual. The other seed individuals are also generated based on the fitness of the initial population, with the fitness being the upper quartile, median and lower quartile. This way, we can see the influence of optimal genes when they are found in different fitness levels of the population.

The sphere function (Equation 1) is used as a test function, with the bounds of $x_{upper} = 5$ and $x_{lower} = -5$.

All experiments are done using the pymoo [14] framework. The standard implementation of the genetic algorithm (GA)

is used. Tournament selection with a tournament size of 2 selects parents for recombination. Survival is done with a fitness selection, which in practical terms selects the best performing individuals to survive to the next generation. Two crossover operators are used, uniform crossover (UX) [4] for gene copying crossover and simulated-binary crossover (SBX) [5] ($\eta = 20$) as the gene combining crossover operator. Polynomial mutation (PM) [15] is used as the mutation operator with $\eta = 20$ and a mutation probability of $1/dimension$. Each test was done over 50 generations and each configuration was repeated 31 times.

V. Results

Evaluating the results, first a single test case is shown in more detail, followed by the accumulated results over all dimensions and the different seed fitness levels.

A. Single Configuration in more Detail

Starting the evaluation, the results of a single configuration are discussed in more detail as a case study for a better understanding of the underlying population dynamics for the later plots. The data shown was generated from a test in dimension 8 using the UX operator. The fitness of the seed individual was chosen as the median fitness of the population. Four seed individuals were used, featuring 0 %, 25 %, 50 %, and 75 % of optimal genes in their genome.

First, Figure 1 shows a comparison of the median of the best result found in the 31 test runs for the four seeds. It can be seen that with a larger amount of optimal genetic material (50 % and 75 %), a faster convergence and better average fitness is achieved. However, overall the results between the 0 % and 25 % seeds as well as the results for the 50 % and 75 % seeds are fairly close and do not always meet these expectations, for example between the generations 30 and 40.

Figure 2 shows the impact for the traceID of each individual from the initial population, the seed individual and the mutation operator. It is clearly visible that increased amounts of optimal genetic material also lead to a larger influence on the result. The seed individual with no optimal genetic material did not show any influence on the result, even though its fitness was the same. While the performance data was still fairly close, the population dynamics changed more dramatically when introducing optimal genetic material.

Finally, Figure 3 shows the fitness of each individual in the initial population used for the test in this section. Furthermore, the median fitness of the seed individual is marked as a line. Comparing this data to the final impact data in Figure 2, no real correlation between the fitness of an initial individual and their influence on the result can be found. Individual 13, which has the best fitness of the population, does not show a particularly high influence on the result. On the other hand, the individuals 16 and 18 both show the highest influences (besides the seed individual), with a considerably worse fitness than the median of the population. This is in line with the before mentioned evaluations in [1] and [2] and again shows the importance of studying the effects of the genome in more detail to better understand the search behavior of EAs.

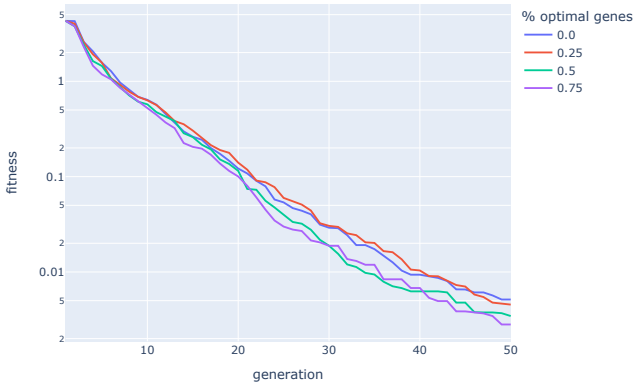

Fig. 1. Median fitness values of test in dimension 8 with a seed of median quality.

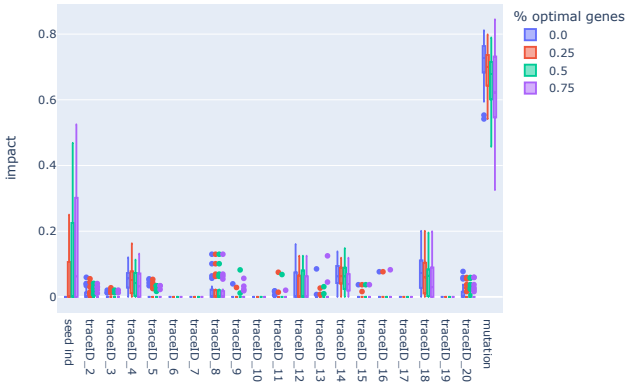

Fig. 2. Impact values in the final generation for the median quality seed individual in dimension 8.

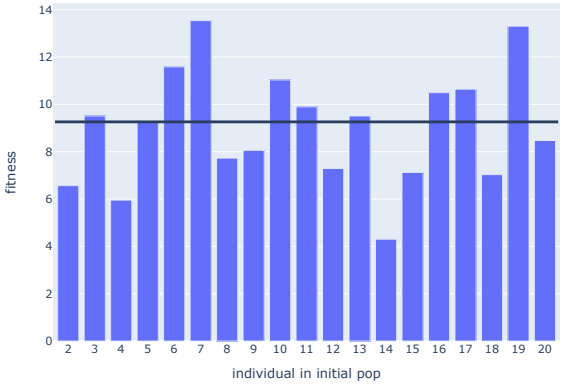

Fig. 3. Fitness values of the initial population for the dimension 8. The line is showing the median fitness.

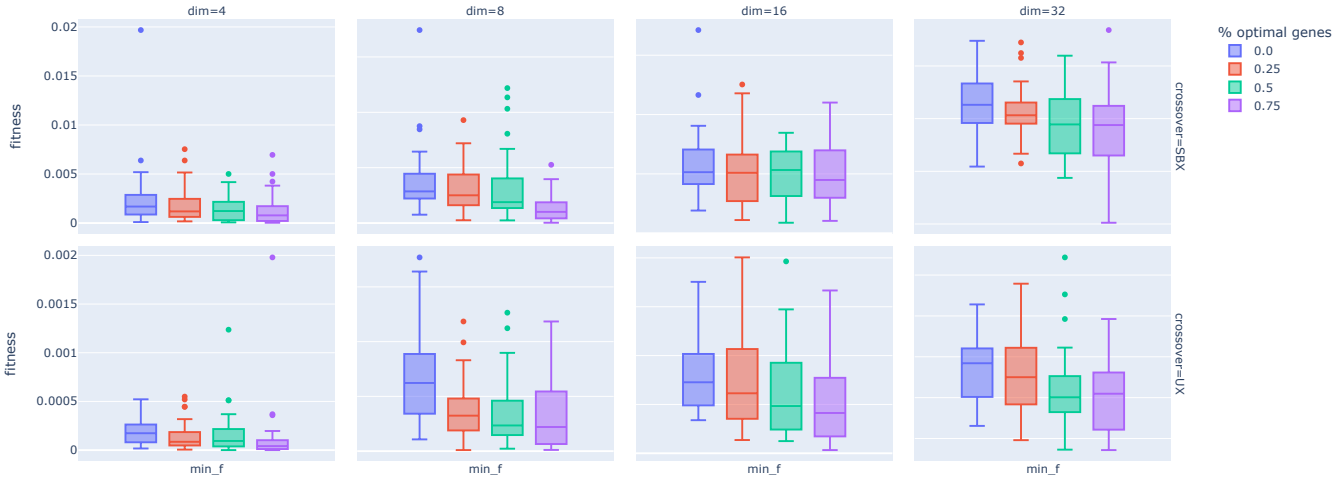

Fig. 4. Fitness data of the upper quartile seed individuals over multiple dimensions. The top row shoes the results using the genome combining SBX, the bottom row the results for the gene copying UX.

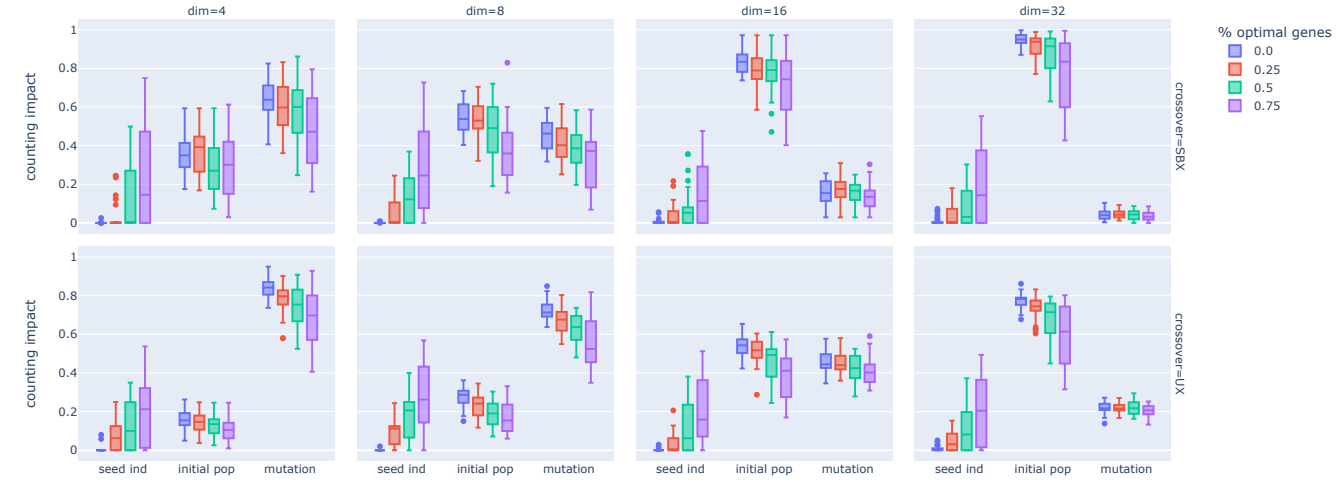

Fig. 5. Impact data of the upper quartile seed individuals over multiple dimensions. Impact values for the individuals in the initial population are accumulated. The top row shoes the results using the genome combining SBX, the bottom row the results for the gene copying UX.

B. Percentage of Optimal Genes

Evaluating the effects of different amounts of optimal genes in a seed individual on a broader scale, test data for multiple dimensions and different crossover operators are compared.

Figure 4 shows the minimum (best) fitness values in the final generation of the lower quartile seed as a box plot. The top row shows the results of the SBX (combining) and the bottom row the results for the UX (copying) crossover operators. For the higher dimensions, it can clearly be seen that the performance of the algorithm improves with more optimal genes. This is true both for the combining and copying crossover operators, and can also be seen for the lower dimensions. However, in lower dimensions the performance is more equal as the algorithm has already converged more to the optimum value. The positive influence of the optimal genetic material can clearly be seen, as the fitness of the seed individuals as well as the general distance to the optimal genome is the same.

Evaluating the impact data clearly attributes this effect to the seed individual. Figure 5 shows the impact of the seed, initial population and mutation operator. Again, the data is shown as a box plot with the rows representing the different dimensions and the columns the crossover operator used. Similar to the performance plot, we can see that seed individuals with a higher percentage of optimal genes contribute more to the final result than with a lower percentage. As all seeds in this test have the same fitness, their selection for offspring creation is equal, so the higher influence can be linked to the optimal genes performing better in crossover. This is to be expected for the gene copying crossover, as it keeps the already optimal genome value. However, this is also true for the gene combining SBX. When the genomes of two individuals values are combined, one could assume that each

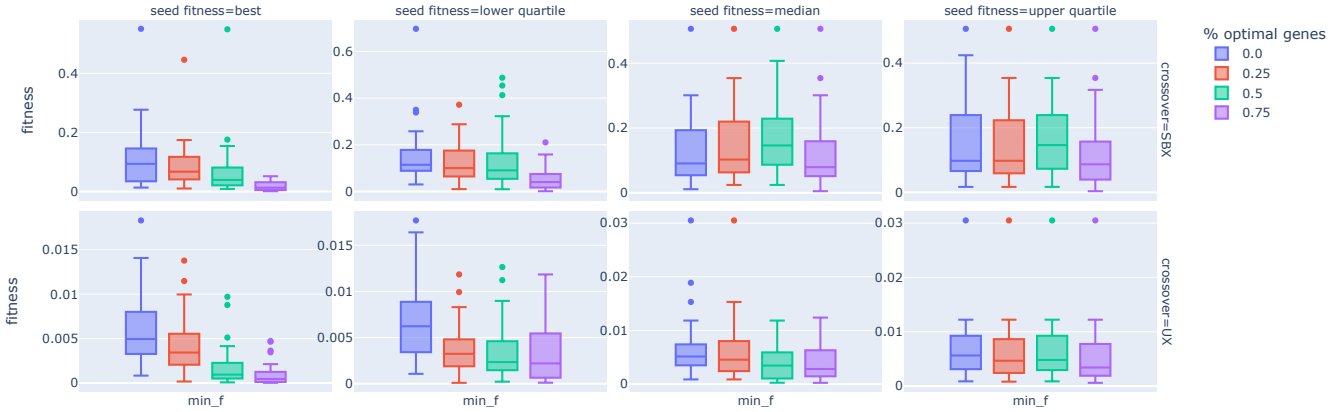

Fig. 6. Fitness data of the different seed individual configurations for dimension 8. Each column shows a different seed fitness. The top row shoes the results using the genome combining SBX, the bottom row the results for the gene copying UX.

Fig. 7. Impact data of the different seed individual configurations for dimension 8. Each column shows a different seed fitness. Impact values for the individuals in the initial population are accumulated. The top row shoes the results using the genome combining SBX, the bottom row the results for the gene copying UX.

gene individually being closer to the optimum would lead to a better performance, but this is not the case.

The results overall clearly show optimal genome values to be beneficial to the performance of an algorithm. This is supported by the evaluation of the heritage information, as more genetic material of seeds with more optimal genes were found in the result. Furthermore, this holds true for both the gene combining and copying crossover operators and can be seen across all different dimensions.

C. Optimal Genes and Fitness

While the previous evaluation already shows that a higher percentage of optimal genes is beneficial, the fitness of the seed individual was kept as the median of the initial population used. The following experiments are varying the fitness to evaluate its effect on the performance gains and population dynamics for better and worse individuals.

The fitness of the seed individuals is, as described in Section IV, derived from the initial population. First the seed is constructed as the best individual of the population (half of the otherwise best individual in the population). Furthermore, the lower quartile, median, and upper quartile are used as fitness points. This way, we hope to capture a very good, good, medium and bad individual. As the previous tests already showed very similar results across different dimensions, this section focuses on the results dimension 8.

Figure 6 shows the performance results of the tests in terms of the best fitness found in the last generation. The columns again show the different crossover operators used, the rows this time show the different qualities of seed individuals. The corresponding impact data of the different seed individuals can be seen in Figure 7.

The effects observed in the previous Section V-C are again found here. The performance data in Figure 6 clearly shows the performance gains for including optimal fitness individuals. However, these gains shrink with a worse fitness from the seed individual. There are two possible reasons for this. On the one hand, as the algorithm uses tournament selection for offspring creation, meaning the optimal genetic material is less likely to be chosen. On the other hand, it is also possible that the very bad genes do not produce good offspring, even if some parts

of the genome are already optimal. This is supported by the impact data in Figure 7. It can clearly be observed that the influence of the seed individual is going down with a worse fitness.

Interestingly, for the median and upper quartile seeds with 25 % and 50 % of optimal genes, performance for SBX is worse than no optimal genetic material (Figure 6). However, the impact data (Figure 7) shows that the influence of individuals is still increasing with the amount of optimal genetic material. Performance for the UX operator, on the other hand, still improves with the higher amount of optimal genetic material. This also indicates other influences on the population dynamics, as already observed in the case study of the single test configuration.

VI. Conclusion and Future Work

This paper investigates the effects of optimal genetic material in the initial population of EAs, aiming towards better understanding the role of the genome in the evolutionary process. For this, experiments were designed seeding a randomly generated population with pre-computed individuals, containing different amounts of optimal genetic material. This results in the comparison of two types of genomes. First, genomes containing optimal genetic material but also "bad" genes dragging down the fitness. Secondly, genomes in which all genes are mediocre. The evaluation was done using the sphere function for multiple dimensions. Furthermore, the effects of different crossover operators, which combine or copy the genome values, were assessed. Results were evaluated both in terms of performance and population dynamics, using the T-EA [2] to compute the influence of the seed individual on the result.

The evaluation clearly shows the benefits of including optimal genetic material. Increasing the number of optimal genes in a genome resulted in a better performance of the algorithm in most tests. Analyzing the heritage data clearly links these performance gains to the optimal genetic material. This is an interesting consideration when solving real-world problems, seeding the initial population with problem knowledge. However, it could also be shown that optimal genetic material in individuals with a bad fitness has little effect, as the individuals are less likely to be chosen for reproduction. Finally, both the gene copying crossover operator (UX [4]) and the gene combining crossover operator (SBX [5]) did show the mentioned effects. While this may seem trivial for operators only copying the gene values, it is interesting to see that the same effects are also found when combining the genome values.

While this work clearly shows the benefits in performance and effects on the population dynamics when including optimal genetic material, it can only be seen as a starting point towards a more profound understanding of the genome in EAs. Tests are only done using the sphere function. One challenge in this research is creating the seed individuals from a given fitness, which is not trivial for other test problems. However, in future works it will be necessary to evaluate a larger variety of benchmarks with more diverse fitness landscapes. Using the sphere function also means the generated seed individuals all feature the same distance to the optimal genome. While it can be seen as beneficial to remove this as a variable in the evaluation of this paper, future work should study this effect. Finally, this paper only focuses on including optimal genome values. However, the quality of the genome in an EA can not only be measured in the percentage of already optimal genes. In future works, other aspects of what makes a genome good need to be identified and studied, for example the effects of near optimal or local optimal solutions on the result.

References

[1] C. Ramirez-Atencia, T. Benecke, and S. Mostaghim, "T-EA: A Traceable Evolutionary Algorithm," in *2020 IEEE Congress on Evolutionary Computation (CEC)*, Jul. 2020, pp. 1–8.

[2] T. Benecke and S. Mostaghim, "Tracking the Heritage of Genes in Evolutionary Algorithms," in *2021 IEEE Congress on Evolutionary Computation (CEC)*, Jun. 2021, pp. 1800–1807.

[3] ——, "The Impact of Population Size on the Convergence of Multi-objective Evolutionary Algorithms," in *2021 IEEE Symposium Series on Computational Intelligence (SSCI)*, Dec. 2021, pp. 1–8.

[4] G. Syswerda, *Uniform Crossover in Genetic Algorithms*, Jan. 1989.

[5] K. Deb and R. Agrawal, "Simulated Binary Crossover for Continuous Search Space," *Complex Syst.*, 1995.

[6] S. Kazemzadeh Azad, "Seeding the initial population with feasible solutions in metaheuristic optimization of steel trusses," *Engineering Optimization*, vol. 50, no. 1, pp. 89–105, Jan. 2018.

[7] B. İ. Selamoğlu, A. Salhi, and M. Sulaiman, "Strip Algorithms as an Efficient Way to Initialise Population-Based Metaheuristics," in *Recent Developments in Metaheuristics*, ser. Operations Research/Computer Science Interfaces Series, Cham, 2018, pp. 319–331.

[8] M. Kress, S. Mostaghim, H. Schmeck, and D. Seese, "Gap Search in Particle Swarm Optimization," in *9th International Conference on Artificial Evolution*. EA'09, 2009, Oktober.

[9] H. Maaranen, K. Miettinen, and M. M. Mäkelä, "Quasi-random initial population for genetic algorithms," *Computers & Mathematics with Applications*, vol. 47, no. 12, pp. 1885–1895, Jun. 2004.

[10] M. D. McKay, R. J. Beckman, and W. J. Conover, "A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code," *Technometrics*, vol. 21, no. 2, pp. 239–245, 1979.

[11] H. Pohlheim, "Multidimensional scaling for evolutionary algorithms–visualization of the path through search space and solution space using Sammon mapping," *Artificial Life*, vol. 12, no. 2, pp. 203–209, 2006.

[12] A. De Lorenzo, E. Medvet, T. Tušar, and A. Bartoli, "An analysis of dimensionality reduction techniques for visualizing evolution," in *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, ser. GECCO '19. New York, NY, USA: Association for Computing Machinery, Jul. 2019, pp. 1864–1872.

[13] G. Ochoa, K. M. Malan, and C. Blum, "Search trajectory networks: A tool for analysing and visualising the behaviour of metaheuristics," *Applied Soft Computing*, vol. 109, p. 107492, Sep. 2021.

[14] J. Blank and K. Deb, "Pymoo: Multi-Objective Optimization in Python," *IEEE Access*, vol. 8, pp. 89 497–89 509, 2020.

[15] K. Deb and S. Agrawal, "A Niched-Penalty Approach for Constraint Handling in Genetic Algorithms," in *Artificial Neural Nets and Genetic Algorithms*, A. Dobnikar, N. C. Steele, D. W. Pearson, and R. F. Albrecht, Eds. Vienna: Springer, 1999, pp. 235–243.