Experimental Analyses of Crossover and Diversity on Jump

Timo Kötzing
Hasso-Plattner-Institute, Potsdam University
Potsdam, Germany
Timo.Koetzing@hpi.de

Xiaoyue Li
Hasso-Plattner-Institute, Potsdam University
Potsdam, Germany
Xiaoyue.Li@hpi.de

ABSTRACT

While it is mathematically proven that the $(\mu + 1)$ GA optimizes Jump_k efficiently for low crossover probabilities, theory research still struggles with the analysis of crossover-based optimization for high crossover probabilities on this key test function. Research in this area has improved our understanding of crossover in general, in particular regarding the emergence of diversity, the crucial ingredient for successful optimization with genetic algorithms.

In this paper we study the optimizing process after the $(\mu + 1)$ GA has reached the plateau of Jump_k. We are interested in (a) the stationary distribution of the algorithm on the plateau (when ignoring the optimum) and (b) the dynamics of the stationary distribution. We experimentally show that the $(\mu+1)$ GA achieves 10% complementary pairs if $\mu = 10 \cdot k$, unless n is very small. Regarding the dynamics, we show samples of how bit positions gain and lose individuals with a 0 at that position.

CCS CONCEPTS

• **Theory of computation** → *Design and analysis of algorithms.*

KEYWORDS

genetic algorithm; diversity; crossover

ACM Reference Format:
Timo Kötzing and Xiaoyue Li. 2023. Experimental Analyses of Crossover and Diversity on Jump. In *Genetic and Evolutionary Computation Conference Companion (GECCO '23 Companion), July 15–19, 2023, Lisbon, Portugal.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3583133.3590657

1 INTRODUCTION

In [7], five benefits from populations are named, several pertaining to the benefits of crossover. In evolutionary computing, crossover is a ubiquitous part of algorithms [3, 4, 6, 8].

For years, theory research aims at explaining the effectiveness of crossover, and at suggesting what particular crossover operators are effective for which kind of landscapes. An important milestone was the work of Jansen and Wegener [5], introducing the now-classic test function Jump_k with parameter k. Formally, Jump_k is defined such that

$$\forall x \in \{0,1\}^* : \text{Jump}_k(x) = \begin{cases} k + |x|_1, & \text{if } |x|_1 = n \text{ or } |x|_1 \leq n - k; \\ n - |x|_1, & \text{otherwise.} \end{cases}$$

where $|x|_1$ denotes the number of 1s of a bit string. Essentially, there is a plateau of equal fitness in distance k of the optimum (in fact, all and only the bit strings in distance k of the optimum are on the plateau of equal fitness closest to the optimum). While it is easy for any hill-climbing search heuristic to find the plateau, making the final jump (hence the name) to the optimum is hard for search operators focused on the neighborhood of search points (like in mutation-based search). By contrast, if in a population are two individuals that do not share a 0, a crossover operator might be able to combine the specific features of two such individuals to obtain the optimum.

Concretely, we consider the $(\mu + 1)$ GA as given in Algorithm 1. It uses one variation operator, which picks two parents from the population uniformly at random, performs a uniform crossover (inheriting each bit independently from a uniformly chosen parent) followed by mutation (flipping each bit independently with probability $1/n$).

Algorithm 1: The $(\mu + 1)$ GA on fitness function f.

```
P_0 ← μ individuals from {0,1}* chosen u.a.r.;
t = 0;
while stopping criterion not met do
    x ← uniform random bit string from P_t;
    y ← uniform random bit string from P_t;
    z ← uniform crossover of x and y;
    z' ← flip each bit of z independently with probab. 1/n;
    P_t = P_t ∪ {z'};
    P_{t+1} = P_t \ {an individual x ∈ P_t with least f(x) value};
    t = t + 1;
```

The only way in which crossover can be beneficial is by exploiting diversity in the population. In fact, the key difficulty in the formal analysis of crossover is to give an account of the emergence and progression of diversity. The analysis of Jansen and Wegener [5] considered a very strong implicit diversity mechanism, using crossover very sparingly, only with probability of $O(1/(n \log n))$. It is a long-standing open problem to give insights into settings with less strong diversity mechanisms.

While strong explicit diversity mechanisms quickly lead to good run time guarantees [1], essential progress on the analysis of the $(\mu + 1)$ GA on Jump_k without such mechanisms was made on this question in a work on emergent diversity [2]. However, still a very large gap remains between observed run time and theoretical bounds; see [9] for a survey on theoretical work in this area.

With this paper we want to give a very careful and in-depth analysis of the behavior of the $(\mu + 1)$ GA at the plateau of Jump_k.

GECCO '23 Companion, July 15–19, 2023, Lisbon, Portugal

ACM ISBN 979-8-4007-0120-7/23/07.
https://doi.org/10.1145/3583133.3590657

Our goal is to inform both practical uses and the theoretical analysis of crossover alike about the inner workings of crossover and about how diversity emerges in populations.

In particular, we let the $(\mu + 1)$ GA run on Jump_k where the optimum is removed; thus, the population will assemble on the plateau (of bit strings with $n - k$ many 1s) and perform a random walk on the set of all possible such populations. This random walk is a Markov chain with a unique stationary distribution which is approached quickly by the process. Thus, it is now interesting to analyze exactly this stationary distribution.

For the optimization of Jump_k, uniform crossover requires two parent individuals which do not share any of their k bit positions where they have a 0. We call such pairs *complementary pairs*. We are particularly interested in the proportion of complementary pairs among all possible pairs of individuals in the population.

In Section 2 we show two ways to sample the stationary distribution. We establish what run times are required for the process to arrive at the stationary distribution.

In Section 3 we show how the proportion of complementary pairs depends on the various parameters of problem and algorithm (n and k for the problem and μ for the population pool size). We see that it is surprisingly easy to find a proportion of 10% in complementary pairs. We see that there is a minimal μ required to achieve such a proportion of 10% complementary pairs and show that the requirement is about $\mu \geq 10k$. We note that very small values of n do not allow for finding μ large enough to achieve 10% complementary pairs; we show that approximately $n \geq 0.6\, k^2 + 4$ is sufficient to be able to find μ which give 10% complementary pairs.

In Section 4, we consider the dynamics of the stationary distribution. That is, we wait until the algorithm has reached the stationary distribution and observe then what happens with the diversity. Specifically, we consider, for each bit position, the number of individuals with a 0 at that position. We find that this number typically stays far away from μ (its potential maximum) for all bit positions. For the positions there are occasional fluctuations between 0 and just a few individuals with a 0 at that position; a few positions gain higher values.

2 STATIONARY DISTRIBUTION

In this section we consider the stationary distribution exhibited by running the algorithm on Jump_k with the optimum removed. We are interested in how the population is spread out on the plateau of equal fitness at distance k to 1^n. In order to experimentally sample the stationary distribution on the plateau, we run the $(\mu+1)$ GA for a sufficient number of iterations. We consider two initialization strategies as follows.

(1) **Uniformly random**: Drawing independently μ individuals u.a.r. from $\{0, 1\}^n$.
(2) **Homogeneous on plateau**: All individuals are identical, starting with k sequential 0s followed by $(n - k)$ many 1s.

In [1] it was proven that the $(\mu + 1)$ GA requires $O(\mu n \log n)$ to reach the plateau. Disregarding log-factors and normalizing by $n \cdot \mu$ (to account for the time it takes to consider any given bit in any given individual), we observe the population once every $n \cdot \mu$ iterations. Since the maximum value of the number of complementary pairs depends on μ, we normalize by $\binom{\mu}{2}$. All results are averaged over 50 independent runs. In Figure 1 we see that the stationary

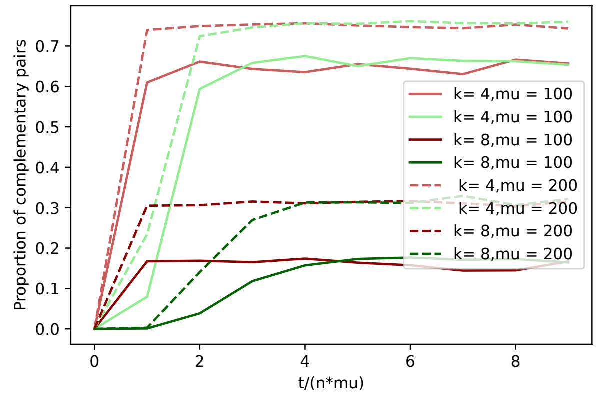

Figure 1: Proportion of complementary pairs. Red plots indicate uniform initialization, green plots homogeneous initialization on plateau.

distribution is approximated after at most $5\mu n$ iterations. For our further investigations, we use this as the time budget and assume the population to be sufficiently mixed.

3 PROPORTION OF COMPLEMENTARY PAIRS

Next we regard the impact of the population size μ on the proportion of complementary pairs in the stationary distribution. Following Section 2, we sample the stationary distribution by running the $(\mu + 1)$ GA for $5\mu n$ iterations and average over 20 samples.

We consider different values for n (400 and 800) and different values for k (4 to 20 in steps of 4). In Figure 2 we see the resulting proportion of complementary pairs for a range of population sizes μ, from $\mu = 20$ to $\mu = 200$ with a step size of 10.

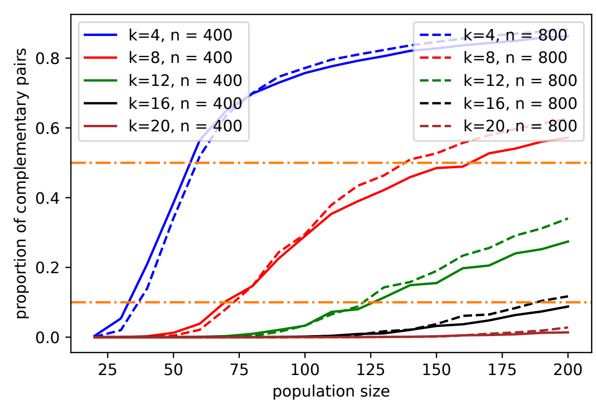

Figure 2: $n = 800$. Complementary pairs increasing with population size. Horizontal lines in blue are at 50% and at 10%.

We can see from Figure 2 that the value of n has only a small impact on the proportion of complementary pairs, while the values

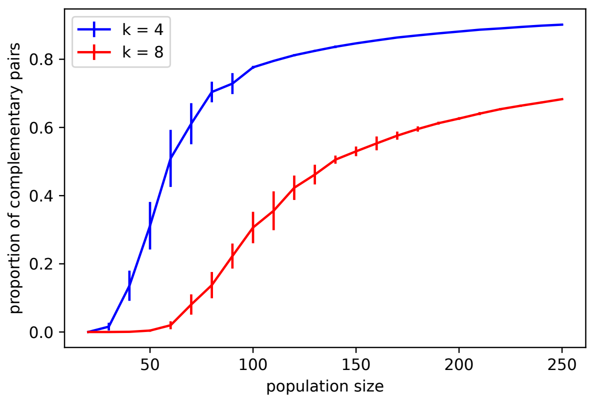

Figure 3: Complementary pairs increasing with population size. Different color represents different k value Plots are averages and bars are ± standard deviation.

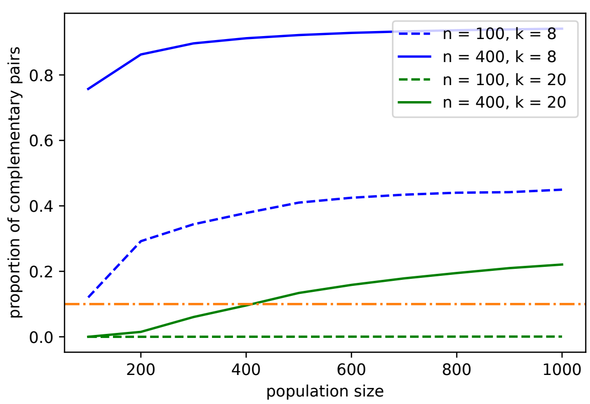

Figure 4: Complementary pairs leveling out for large pool sizes. Horizontal line in blue is at 10%. Same color represents same k value. The red is $k = 8$ and the green is $k = 20$. Same linestyle represents same n value. Dotted line is $n = 100$ and the other is $n = 400$

of k and μ have a strong impact. Large values of μ give a higher proportion of complementary pairs while larger value of k decrease the proportion of complementary pairs. Note that we consider a value of $k = 16$ already large, and also in this setting proportions of 10% of complementary pairs are easily reached.

We estimate sampling bias with Figure 3. Here we see that our samples do have some variance for mid-range population sizes, but it diminishes quickly for larger sizes.

In Figure 2 the proportion of complementary pairs seems to keep increasing. As shown in Figure 4, we can see that also large values of μ do not significantly increase the proportion of complementary pairs. For example, for $k = 20$, it seems impossible that μ could be large enough to reach a proportion of 10% complementary pairs.

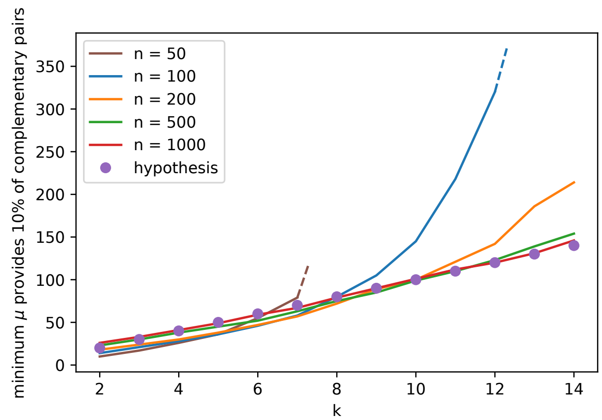

Figure 5: Minimum pool size for 10% complementary pairs. Given as purple dots is the hypothesis of $10k$ as minimum population size. Dashed lines indicate that, for high values of k, even $\mu = 50k$ is not sufficient to achieve 10% complementary pairs.

We conduct an experiment to find the minimal μ which is sufficient to arrive at a value of 10% complementary pairs. For various n, we see in Figure 5 that roughly $\mu = 10k$ is necessary and sufficient, as long as n is large enough. As an aside we note that n needs to be large enough; experimentally, this is at about $n \geq 0.6\ k^2 + 4$.

4 DYNAMICS OF THE STATIONARY DISTRIBUTION

In the previous section, we saw snapshot values of the stationary distribution. In this section, we are interested in how the stationary distribution behaves from step to step. Since we are interested in diversity, and all individuals on the plateau have 1s in most positions, we are interested in the distribution of 0s in the population.

For any $i \leq n$ and a given population P, we say that P *has a* 0 *at* i, if some individual of P has a 0-bit at position i. Similarly, we talk of "many 0s at i" and so forth. We are interested in how long the population maintains a 0 at a certain bit position, as well as how many 0s are presented at any given position. In order to understand this, we focus on one specific parameter setting and observe the stationary distribution over a number of time steps.

Specifically, we let $n = 500$, $\mu = 50$ and $k = 4$. As argued in Section 3, this is a setup with at least 10% complementary pairs in the stationary distribution. We let the algorithm run for 10^5 iterations and only record the zero positions for the last 10^4 generations.

In Figure 6 we see the result for all bit positions, while Figure 7 focuses on the eleven bit positions 400 to 410. As we see, many bit positions appear in the population, some disappear quickly, some take some time to disappear without ever gaining prominence (in the sense of a lot of individuals having a 0 here) and some almost take over the entire population (note that no position ever has more than 40 0s in the population, at a population size of 50).

In our example, out of the 500 bit positions, in the last 10^4 iterations, there are 463 bit positions in which a 0 appears. This justifies

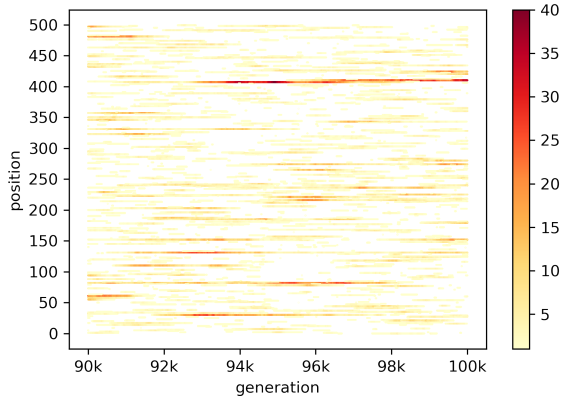

Figure 6: The x-axis shows the number of generations, and the y-axis shows the bit positions. A white spot for a generation and a bit position i means that i is not in the population at that time step. The color codes the number of individuals in the population that have a 0 in that position.

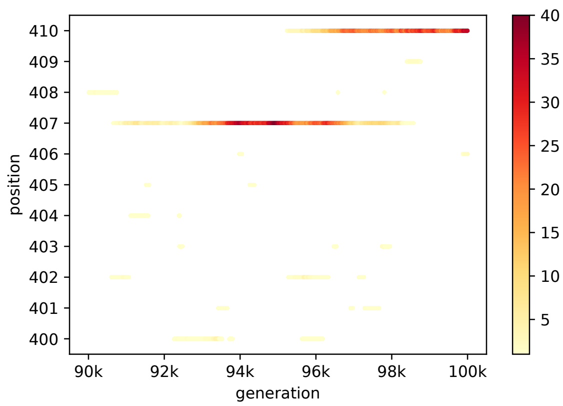

Figure 7: Time is on the x-axis, and the bit positions are on the y-axis. Bit positions range from 400 till 410

our previous argument that with this parameter setup, there are complementary pairs through all generations.

As we see in Figure 7, position 407 has a high frequency of zeros appearing, while in bit position 404 zeros appear and disappear relatively fast. Next, we study in more detail how the number of individuals with a 0 in several representative positions evolves.

In Figure 8, we see three conditions of 0 behavior: 0 appear and disappear quickly but also no other 0 share the same bit position (bit position 404); 0s appears at a generation and stay through some generations but with low frequencies (bit position 400); in some period of generations, 0s vividly increase and decrease frequency (bit position 407).

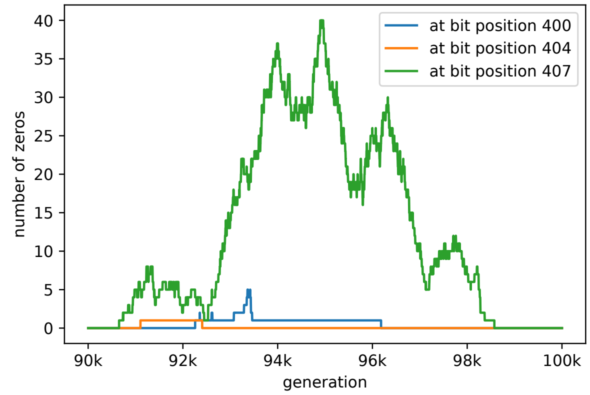

Figure 8: Number of 0s in the population at position 404, 402 and 407.Notice at the bit position 404, there is only one 0 appears within a brief period.

5 CONCLUSION

We performed a number of analyses detailing the behavior of the $(\mu + 1)$ GA on Jump_k for a wide range of parameters. We find it surprising that complementary pairs emerge quickly, yet it is so hard to prove rigorous statements about this emergence of diversity. Especially Figure 6 might give a strong hint: the behavior is very varied, gaining and losing 0 at bis positions over and again.

Overall, we believe that this work contributes to a dialectic process between proofs and experiments to understand better the emergence of diversity and the effectiveness of crossover.

ACKNOWLEDGMENTS

This work is supported by grant FR 2988/17-1 by the German Research Foundation (DFG).

REFERENCES

[1] Duc-Cuong Dang, Tobias Friedrich, Martin S. Krejca, Timo Kötzing, Per Kristian Lehre, Pietro S. Oliveto, Dirk Sudholt, and Andrew Michael Sutton. 2016. Escaping Local Optima with Diversity Mechanisms and Crossover. In *Proc. of GECCO'16*. ACM Press, 645–652.
[2] Duc-Cuong Dang, Tobias Friedrich, Timo Kötzing, Martin S. Krejca, Per Kristian Lehre, Pietro S. Oliveto, Dirk Sudholt, and Andrew M. Sutton. 2018. Escaping Local Optima Using Crossover With Emergent Diversity. *IEEE Transactions on Evolutionary Computation* 22, 3 (2018), 484–497.
[3] Grasiele R. Duarte and Beatriz S. L. P. de Lima. 2021. An Operation to Promote Diversity in Evolutionary Algorithms in a Dynamic Hybrid Island Model. In *Proc. of GECCO'21 Companion*. ACM Press, 1779–1787.
[4] Thomas Gabor, Lenz Belzner, and Claudia Linnhoff-Popien. 2018. Inheritance-Based Diversity Measures for Explicit Convergence Control in Evolutionary Algorithms. In *Proc. of GECCO'18*. ACM Press, 841–848.
[5] Thomas Jansen and Ingo Wegener. 2002. The Analysis of Evolutionary Algorithms - A Proof That Crossover Really Can Help. *Algorithmica* 34 (2002), 47–66.
[6] Valentín Osuna-Enciso, Erik Cuevas, and Bernardo Morales Castañeda. 2022. A diversity metric for population-based metaheuristic algorithms. *Information Sciences* 586 (2022), 192–208.
[7] Adam Prügel-Bennett. 2010. Benefits of a Population: Five Mechanisms That Advantage Population-Based Algorithms. *IEEE TEvC* 14 (2010), 500–517.
[8] Giovanni Squillero and Alberto Tonda. 2016. Divergence of character and premature convergence: A survey of methodologies for promoting diversity in evolutionary optimization. *Information Sciences* 329 (2016), 782–799.
[9] Dirk Sudholt. 2020. *The Benefits of Population Diversity in Evolutionary Algorithms: A Survey of Rigorous Runtime Analyses.* Springer, 359–404.