Better Fixed-Arity Unbiased Black-Box Algorithms

Nina Bulanova
ITMO University
Saint-Petersburg, Russia
ninasbulanova@gmail.com

Maxim Buzdalov
ITMO University
Saint-Petersburg, Russia
mbuzdalov@gmail.com

ABSTRACT

In their GECCO'12 paper, Doerr and Doerr proved that the k-ary unbiased black-box complexity of OneMax on n bits is $O(n/k)$ for $2 \le k \le \log_2 n$. We propose an alternative strategy for achieving this unbiased black-box complexity when $3 \le k \le \log_2 n$. While it is based on the same idea of block-wise optimization, it uses k-ary unbiased operators in a different way.

For each block of size $2^{k-1} - 1$ we set up, in $O(k)$ queries, a virtual coordinate system, which enables us to use an arbitrary unrestricted algorithm to optimize this block. This is possible because this coordinate system introduces a bijection between unrestricted queries and a subset of k-ary unbiased operators. We note that this technique does not depend on OneMax being solved and can be used in more general contexts.

This together constitutes an algorithm which is conceptually simpler than the one by Doerr and Doerr, and in the same time achieves better constant multiples in the asymptotic notation. Our algorithm works in $(2 + o(1)) \cdot n/(k - 1)$, where $o(1)$ relates to k. Our experimental evaluation of this algorithm shows its efficiency already for $3 \le k \le 6$.

CCS CONCEPTS

• **Theory of computation → Theory of randomized search heuristics**;

KEYWORDS

Black-box complexity, unbiased variation, OneMax.

ACM Reference Format:

Nina Bulanova and Maxim Buzdalov. 2018. Better Fixed-Arity Unbiased Black-Box Algorithms. In *GECCO '18 Companion: Genetic and Evolutionary Computation Conference Companion, July 15–19, 2018, Kyoto, Japan.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3205651.3205762

1 MAIN IDEA

The *unbiased black-box complexity* of a problem is one of the measures for how complex the problem is for solving it by evolutionary algorithms and other randomized heuristics. Often, complexities of rather simple problems are studied, such as the famous OneMax problem, defined on bit strings of length n as follows:

$$\textsc{OneMax}_z : \{0, 1\}^n \to \mathbb{R}; x \mapsto |\{i \in [1..n] \mid x_i = z_i\}|.$$

GECCO '18 Companion, July 15–19, 2018, Kyoto, Japan

ACM ISBN 978-1-4503-5764-7/18/07.
https://doi.org/10.1145/3205651.3205762

One of possible restrictions to the unbiased black-box search model is the use of unbiased operators with restricted arity. The original paper [5] studied mostly unary unbiased black-box complexity, e.g. the class of algorithms allowing only unbiased operators taking one individual and producing another one, or mutation-based algorithms. The paper [2] (see also the journal version [3]) presents the current state of the art for the k-ary unbiased black-box complexities of OneMax for $k \ge 2$.

The result presented in [2, 3] is rather complicated and non-trivial to use for two reasons. First, it has to use, as a building block, a derandomized unrestricted algorithm to solve OneMax, proposed in [4], which runs in $(1 + \delta) \log_2(9) \cdot n/\log_2 n$ for δ decreasing with n. The corresponding sequence is only proven to exist, but no way to construct it has ever been proposed. Second, this algorithm needs 4ℓ bits to optimize a piece of ℓ bits and encodes the fitness values of several queried strings into some of these bits to make the final choice. This not only complicates the algorithm, but also increases the arity needed to find the optimum within the given number of fitness queries.

In this paper we aim to improve this situation. We propose an algorithm which, like the one from [2], optimizes OneMax by blocks of ℓ bits but does it in a simpler and more explicit manner and performs fewer queries. For k-ary operators, $k \ge 3$, we set $\ell = 2^{k-1} - 1$ and initialize a "virtual coordinate system" in k queries very similar to "storage initialization" in [2]. However, subsequently we use the opportunities offered by this coordinate system in a different manner. We notice that it introduces a bijection between bit strings of length ℓ and 2^ℓ different k-ary unbiased operators. We use this fact to optimize these ℓ bits by simulating an arbitrary algorithm for solving OneMax of length ℓ, regardless of unbiasedness or arity of this algorithm. In particular, we can use the random sampling algorithm from [4] which works in expected $(2 + o(1)) \cdot \ell/\log \ell$ time. Thus, the overall number of queries needed by the proposed algorithm to solve the OneMax problem sums up from expected $(2 + o(1)) \cdot n/(k - 1)$ queries from solving each block and $O(n \cdot k/2^k)$ additional work coming from the initialization of coordinate systems and the aggregation of answers found for each block. Note that the second addend is negligibly small in k compared to the first one, so it hides entirely in $o(1)$ inside the first addend.

2 THE PROPOSED ALGORITHM AND EXPERIMENTS

Similarly to the algorithm from [2], we maintain two bit strings, x and y, to encode which bits are already guessed right. The only difference is that we change the meaning to the opposite one: the bits that differ between x and y are guessed right in x. Initially, x is generated uniformly at random and $y = x$.

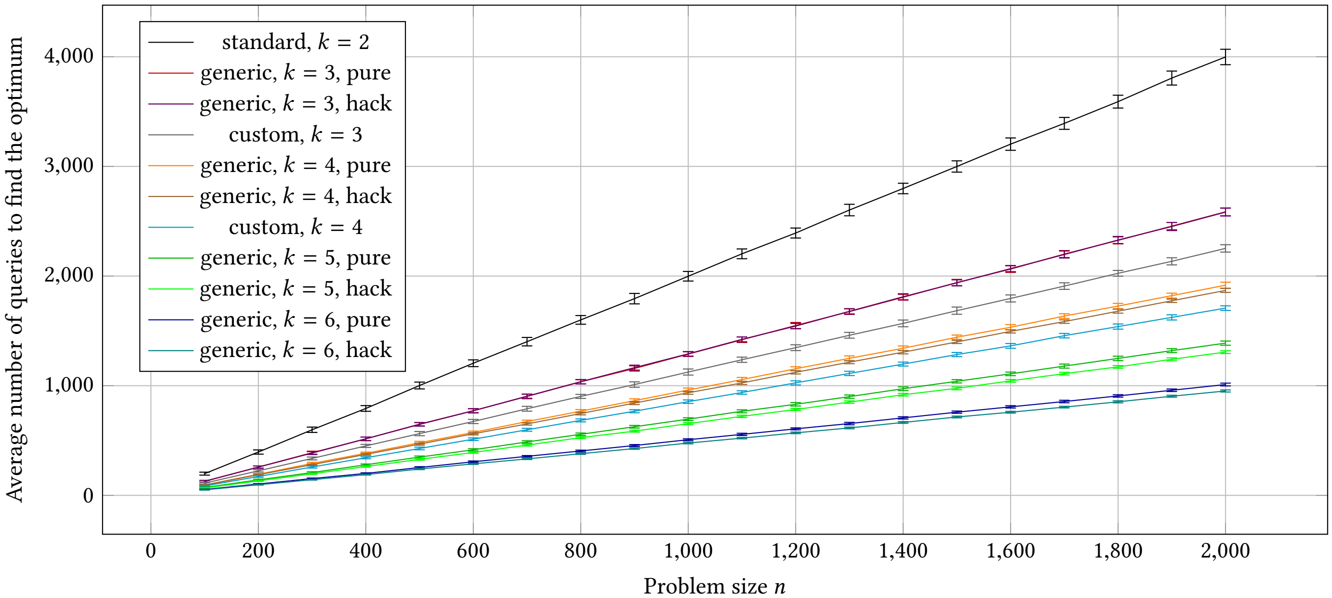

Figure 1: Experiment results for k-ary unbiased algorithms on OneMax. 100 independent runs were made for each point.

We optimize bits in blocks of size at most $\ell = 2^{k-1} - 1$, where all blocks except for possibly the last one have the maximum possible size ℓ. A k-ary unbiased operator is able to distinguish 2^{k-1} groups of bits. One of these groups has to be dedicated to "unrelated" bits, i.e. the $n - \ell$ bits outside of the interesting region. For simplicity, we allocate the 0-th group for this purpose, that is, the group of bits which are equal throughout all arguments. Then we build the virtual coordinate system for every block by $k - 1$ auxiliary queries in such a way that the remaining $2^{k-1} - 1$ groups cover this block and each group consists of at most a single bit, which enables subsequent fine manipulations with them. After that we use this coordinate system to forward queries of an unrestricted algorithm solving OneMax of length ℓ to the ℓ bits of the original OneMax problem of length n using only k-ary unbiased operators.

We have implemented a small programming platform for experimenting with unbiased algorithms of fixed arity. In this platform, the bit strings of individuals are encapsulated in a so-called "unbiased processor", and the individuals are accessed through handles, which reveal only their fitness. Unbiased operators are naturally encoded as functions which map one integer array, $\langle n_0, n_1, \ldots \rangle$, to another one, $\langle d_0, d_1, \ldots \rangle$. This ensures that one implements only unbiased algorithms. The source code is available on GitHub[1].

By the means of this platform, we have implemented the well-known binary algorithm of optimization of OneMax [1], the new custom algorithms for $k = 3$ and 4, as well as the *pure* and *hack* flavors of the generic algorithm for arbitrary fixed arities. The *pure* flavor is the original algorithm, and the *hack* introduces a modification in the random sampling unrestricted algorithm to solve OneMax, which samples a new bit string not uniformly at random among all possible bit strings, but uniformly at random among all *potential optima*, that is, among strings consistent with the previous samples and measurements. The latter appears to be generally more efficient in terms of queries, however, a formal proof for this is yet to be given.

We could handle only arities $3 \le k \le 6$, since with $k = 7$ the block size becomes $\ell = 63$, the size which currently cannot be solved by neither *pure* nor *hack* random sampling in reasonable time. The experimental results are presented in Figure 1.

For $k = 3$, both generic algorithms showed the performance of $\approx 1.29n$, which is somewhat greater than $9n/8 = 1.125n$ by the custom algorithm. However, for $k = 4$ the runtimes of $\approx 0.958n$ was shown by the pure algorithm, and $\approx 0.934n$ by the hack, both smaller than n. The custom algorithm for $k = 4$ is still slightly faster with $765/896n \approx 0.854n$. For $k = 5$ the pure and hack versions were $\approx 0.694n$ and $\approx 0.653n$ correspondingly. Finally, for $k = 6$ the results were close to half the size, with the pure algorithm being slightly above, $\approx 0.505n$, while the hack is slightly below, $\approx 0.476n$.

This research was supported by the Russian Scientific Foundation, agreement No. 17-71-20178. The full text of this paper that features the missing proofs and more thorough explanations is available at arXiv[2].

REFERENCES

[1] Benjamin Doerr, Daniel Johannsen, Timo Kötzing, Per Kristian Lehre, Markus Wagner, and Carola Winzen. 2011. Faster black-box algorithms through higher arity operators. In *Proceedings of Foundations of Genetic Algorithms*. 163–172.

[2] Benjamin Doerr and Carola Winzen. 2012. Reducing the arity in unbiased black-box complexity. In *Proceedings Genetic and Evolutionary Computation*. 1309–1316.

[3] Benjamin Doerr and Carola Winzen. 2014. Reducing the arity in unbiased black-box complexity. *Theoretical Computer Science* 545 (2014), 108–121.

[4] Paul Erdős and Alfréd Rényi. 1963. On two problems of information theory. *Magyar Tudományos Akadémia Matematikai Kutató Intézet Közleményei* 8 (1963), 229–243.

[5] Per Kristian Lehre and Carsten Witt. 2010. Black-box Search by Unbiased Variation. In *Proceedings of Genetic and Evolutionary Computation Conference*. ACM, 1441–1448.

[1] https://github.com/mbuzdalov/unbiased-bbc

[2] https://arxiv.org/abs/1804.05443