

A Linear Constrained Optimization Benchmark for Probabilistic Search Algorithms: The Rotated Klee-Minty Problem

Michael Hellwig^[0000-0002-6731-8166] and Hans-Georg Beyer^[0000-0002-7455-8686]

Vorarlberg University of Applied Sciences, Research Centre PPE,
Campus V, Hochschulstraße 1, 6850 Dornbirn, Austria
{[michael.hellwig](mailto:michael.hellwig@fhv.at),[hans-georg.beyer](mailto:hans-georg.beyer@fhv.at)}@fhv.at
<https://homepages.fhv.at/hemi/>
<https://homepages.fhv.at/hgb/>

Abstract. The development, assessment, and comparison of randomized search algorithms heavily rely on benchmarking. Regarding the domain of constrained optimization, the small number of currently available benchmark environments bears no relation to the vast number of distinct problem features. The present paper advances a proposal of a scalable linear constrained optimization problem that is suitable for benchmarking Evolutionary Algorithms. By comparing two recent Evolutionary Algorithm variants, the linear benchmarking environment is demonstrated.

Keywords: Probabilistic Search Algorithms · Benchmarking · Evolutionary Algorithms · Linear Problems · Constrained Optimization · Klee-Minty Cube

1 Introduction

Benchmark environments establish well-defined experimental settings that aim at providing reproducible and comparable algorithmic results. They are essential for the assessment and the comparison of contemporary algorithms. Benchmarking also is important for the development of new algorithmic ideas. This is particularly true in the field of Evolutionary Algorithms (EA) for real-valued constrained optimization, where the theoretical background is comparably scarce.

Regarding EA benchmarks, the CEC competitions on constrained real-parameter optimization [9,10,19] introduced specific constrained test environments (usually referred to as constrained CEC benchmarks). The corresponding benchmark definitions supply a collection of mainly nonlinear objective functions that are constrained by various numbers of equality, inequality, and box-constraints. When considering real-valued constrained optimization problems, the CEC function sets represent the most frequently used benchmarking environment for contemporary EA. Recently, a COCO branch for constrained black-box optimization benchmark problems [4] (BBOB-constrained¹) is developing. The BBOB-

¹ The code related to the BBOB-constrained suite under development is available in the `development` branch on <http://github.com/numbbbo/coco/development>.

constrained test suite is a progression of the unconstrained COCO framework towards constrained benchmarks. However, it currently takes into account only a limited number of objective functions as well as almost linear inequality constraints of scalable quantity.

Compared to the number of test problems currently used in constrained benchmark environments, the domain of real-valued constrained optimization problems is tremendous. Constrained real-valued optimization problems may differ with respect to a multitude of features (and their combinations), including but not restricted to the number and type of constraints, the analytical structure of objective function, and the characteristics of the feasible region in the search space. Although first investigations exist [12], it is not conclusively determined which features are making a constrained optimization problem hard. Among the collections of constrained test problems available [14], the CEC and COCO benchmarks basically represent the two most elaborated benchmarking environments for EA [6]. Considering that the EA development for constraint optimization tasks will further rely on suitable benchmarks and remembering the *no free lunch* theorem [18], the need for benchmark definitions that take into account consistent subgroups of conceivable problems is evident.

This paper presents a supplementary benchmark proposal. By providing hard but strictly linear constrained optimization problems that are scalable with respect to the problem dimension, it differs from the CEC and COCO environments. The benchmark is constructed on the basis of the Klee-Minty polytope. It represents a unit hypercube of variable dimension with perturbed vertices [8]. The inside of the cube represents the feasible region of the constrained problem. The corresponding objective function is constructed in such a way that the classical Simplex algorithm yields an exponential worst-case running time, i.e. it can be considered hard with respect to computational complexity.

Considering the number of sophisticated deterministic approaches available, taking into account linear optimization problems for EA benchmarking might appear questionable. However, many purpose-built linear optimization algorithms show poor performance on the Klee-Minty problem. For example, other basis-exchange pivoting algorithms and even interior-point algorithms exhibit severe problems in this environment [1,11]. Contrary, some EA variants suited for generally constrained problems are able to obtain a similarly good or even better precision (cf. Tables 2 and 3). In this regard, the Klee-Minty problem serves for demonstrating the applicability of EA to linear constrained black-box optimization problems. In [16], a Klee-Minty problem representation has already been used to assess the suitability of a custom-built Covariance Matrix Self-Adaptation Evolution Strategy for linearly constrained problems. The study substantiated a certain need for benchmarking functions suitable for testing EA that solely deal with linear constraints.

The present paper advances the Klee-Minty problem with respect to the following aspects. We introduce an upstream motion in the search space that relocates the optimal solution which usually is placed on the axes. This kind of location may present a bias towards Coordinate Search algorithms or box-

constraint handling approaches. The modified Klee-Minty problem is motivated in detail in Section 2. Moreover, in Section 3 basic benchmarking conventions are proposed in order to provide a thorough basis for reproducible and comparable benchmarking tests. The suggestion of a comprehensive presentation style and a comparison methodology for algorithm assessment are specified in Section 4. For demonstration purposes, two recent EA variants for constrained black-box optimization that proved successful in the context of the CEC competitions [19] are tested. The paper concludes with the discussion of currently unresolved issues and the suggestion of future development directions in Section 5.

2 The rotated Klee-Minty problem

The Klee-Minty cube (named after Victor Klee and George J. Minty) is a unit hypercube of variable dimension with perturbed corners [8]. The inside of the cube represents the feasible region of a linear optimization problem which is referred to as the Klee-Minty problem. The corresponding objective function is constructed in such a way that the Simplex algorithm visits all the corners in the worst case and thus its worst-case runtime is exponential.

Linear optimization test problems are inadequately represented in the context of EA benchmarking, as EAs usually cannot compete with custom-build linear solvers. However, the Klee-minty problem represents a reasonable hard linear problem that is suitable to present the potential of EAs. In order to remove undesired problem characteristics with respect to the location of the optimal solution and the orientation of the feasible region, the Klee-Minty problem is modified by application of a transformation.

The introduction of a set of rotated Klee-Minty problems represents a benchmark proposal for reasonably hard linear optimization problems. The section provides a first suggestion in the style of the CEC benchmarks which is supported by performance profile plots [13]. A modified Klee-Minty cube representation [2]² is considered to build the basis of the proposed linear benchmarks

$$\begin{aligned} \min_{\mathbf{y} \in \mathbb{R}^N} \quad & \mathbf{c}^\top \mathbf{y} \\ \text{s.t.} \quad & \mathbf{A}\mathbf{y} \leq \mathbf{b}, \\ & \check{\mathbf{y}} \leq \mathbf{y} \leq \hat{\mathbf{y}}. \end{aligned} \tag{1}$$

The matrix \mathbf{A} and the right-hand side vector \mathbf{b} are defined as

$$\mathbf{A} = \begin{pmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \end{pmatrix} \in \mathbb{R}^{2N \times N}, \quad \mathbf{b} = \begin{pmatrix} \mathbf{1} \\ \mathbf{0} \end{pmatrix} \in \mathbb{R}^{2N \times 1}. \tag{2}$$

² We omit use of the redundant constraints introduced in [2]. That is, the parameter \mathbf{h} in [2] is considered to be an array of all-zeros for our Klee-Minty representation.

where $\mathbf{1}$ and $\mathbf{0}$ represent vectors of all ones, and all zeros, respectively. The $N \times N$ matrices \mathbf{A}_1 and \mathbf{A}_2 are defined as follows

$$\mathbf{A}_1 = \begin{pmatrix} 1 & 0 & \dots & 0 & 0 \\ \epsilon & 1 & \dots & 0 & 0 \\ \vdots & \epsilon & \ddots & \vdots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ 0 & 0 & \dots & \epsilon & 1 \end{pmatrix} \quad \text{and} \quad \mathbf{A}_2 = \begin{pmatrix} -1 & 0 & \dots & 0 & 0 \\ \epsilon & -1 & \dots & 0 & 0 \\ \vdots & \epsilon & \ddots & \vdots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ 0 & 0 & \dots & \epsilon & -1 \end{pmatrix}. \quad (3)$$

The parameter $0 < \epsilon \leq 1/3$ governs the perturbation of the unit cube. It is set to $\epsilon = 1/10$ to obtain problems of reasonable complexity. Notice, that both matrices only differ with respect to the sign of their diagonal elements.

While the above problem formulation strictly bounds the feasible region, we specify lower bounds $\check{\mathbf{y}}$ and upper bounds $\hat{\mathbf{y}}$ for the parameter vector components

$$\check{\mathbf{y}} = \mathbf{0} \in \mathbb{R}^N \quad \text{and} \quad \hat{\mathbf{y}} = 5N^3 \cdot \mathbf{1} \in \mathbb{R}^N. \quad (4)$$

When considering LP solvers that search exclusively inside the feasible region or on its borders (like the Simplex algorithm or interior point methods) the introduction of box constraints appears redundant. However, having in mind EA variants for constrained black-box optimization that move through infeasible regions of the search space, the box-constraints can be used to represent the domain of eligible input values³. They define a reasonable limitation of high dimensional search spaces and allow for the generation of initial candidate solution populations.

Accordingly, the feasible region $M \subset \mathbb{R}^N$ is determined by $2N$ inequality constraints. It forms a perturbed unit hypercube within the subset of the search space that is determined by the box-constraints of $\mathbf{y} \in \mathbb{R}^N$. The objective function $\mathbf{c}^\top \mathbf{y}$ is determined by the vector

$$\mathbf{c} = (0, 0, \dots, 0, 1)^\top \in \mathbb{R}^N. \quad (5)$$

It is designed in such a way that the optimal solution of the Klee-Minty problems is located at $\mathbf{y}^* = \mathbf{0} \in \mathbb{R}^N$. The optimal objective function value is $f_{opt} = 0$.

Considering the design of problem (1), the optimal solution is always located at the origin of the N -dimensional search space. This construction can attribute bias in two different ways. On the one hand, the location of \mathbf{y}^* favors algorithms that predominantly search in direction of the Cartesian axes, e.g. Coordinate Search or certain DE algorithms [17]. On the other hand, the non-negativity requirement $\mathbf{y} \geq 0$ might be handled in a box-constrained approach⁴. Hence, the optimal solution \mathbf{y}^* is located on the boundary of that respective box. In

³ The Klee-minty problem usually assumes the non-negativity of the parameter vector components. The upper bound $\hat{\mathbf{y}}$ is set in accordance with the translation in Eq. (6).

⁴ In the field of EA, several methods to treat box-constraints do exist, e.g. by random reinitialization inside the box or by repair of violated components.

case that a search algorithm is allowed to take into account infeasible candidate solutions in its procreation process, this can have a considerable effect on its performance. That is, the creation of infeasible solutions outside the box may be compensated by the box-constraint handling approach. Depending on the method used, the box-constraint handling might bias the search towards infeasible candidate solutions that are repaired in a beneficial way and thus approach the optimal solution more quickly and/or with higher precision.

Both issues are resolved by introducing a direct motion of the vectors in the parameter space. The direct motion is an isomorphic transformation that preserves the orientation of a parameter vector \mathbf{y}

$$\begin{aligned} T(\mathbf{y}) : \mathbb{R}^N &\rightarrow \mathbb{R}^N \\ \mathbf{y} &\mapsto \tilde{\mathbf{y}} = \mathbf{R}(\mathbf{y} - \mathbf{t}). \end{aligned} \quad (6)$$

The transformation consists of two components. A translation by the vector \mathbf{t} that is followed by a rotation \mathbf{R} in a suitable hyperplane of the search space. The terms \mathbf{R} and \mathbf{t} are arbitrarily chosen in the following way. The N -dimensional rotation matrix \mathbf{R} is composed of two orthonormal vectors $\bar{\mathbf{v}}_1$ and $\bar{\mathbf{v}}_2$ that span a two-dimensional hyperplane of the search space. They read

$$\begin{aligned} \bar{\mathbf{v}}_1 &= (0, \dots, 0, 1)^\top \in \mathbb{R}^N, \quad \text{and} \\ \bar{\mathbf{v}}_2 &= \frac{\mathbf{u}}{\|\mathbf{u}\|} \quad \text{with } \mathbf{u} = (1, \dots, 1, 0)^\top \in \mathbb{R}^N \end{aligned} \quad (7)$$

The matrix is then build as

$$\mathbf{R} = \mathbf{I} + (\cos(\varrho) - 1) (\bar{\mathbf{v}}_1 \bar{\mathbf{v}}_1^\top + \bar{\mathbf{v}}_2 \bar{\mathbf{v}}_2^\top) - \sin(\varrho) (\bar{\mathbf{v}}_1 \bar{\mathbf{v}}_2^\top - \bar{\mathbf{v}}_2 \bar{\mathbf{v}}_1^\top), \quad (8)$$

where \mathbf{I} denotes the N -dimensional identity matrix and the term ϱ refers to the rotation angle. The matrix $\mathbf{R} \in \mathbb{R}^{N \times N}$ is an orthogonal matrix, i.e. $\mathbf{R}\mathbf{R}^\top = \mathbf{I}$, with determinant $\det(\mathbf{R}) = 1$. Hence, it represents a rotation about the origin of the N -dimensional search space. Aiming at a reasonable amount of complexity, the considered rotation angle is preset to $\varrho = \frac{350}{180}\pi$, and the N -dependent translation vector⁵ $\mathbf{t} = (N^3, N^3, N^3, \dots, N^3)^\top$ is chosen.

By application of $T(\mathbf{y})$, the vertices of the Klee-Minty cube are relocated. Hence, the optimal solution of (1) is displaced as well. As the origin is not affected by the rotation, the optimal solution is transferred into $\tilde{\mathbf{y}}^* := T^{-1}(\mathbf{y}^*) = \mathbf{t}$. Refer to Fig. 1 for an illustration of the two-dimensional scenario.

While the search is carried out, constraint violation is evaluated after transformation with respect to $T(\mathbf{y})$. The objective function is left unchanged. As a consequence, the Klee-Minty problem (1) transforms into

$$\begin{aligned} \min_{\mathbf{y} \in \mathbb{R}^N} f(\mathbf{y}) &= \mathbf{c}^\top \mathbf{y} \\ \text{s.t. } \mathbf{A}\mathbf{R}(\mathbf{y} - \mathbf{t}) &\leq \mathbf{b}, \\ \tilde{\mathbf{y}} &\leq \mathbf{y} \leq \hat{\mathbf{y}}. \end{aligned} \quad (9)$$

⁵ The choice of the translation \mathbf{t} represents an empirically motivated compromise between complexity and numerical stability for a wide range of search space dimensions.

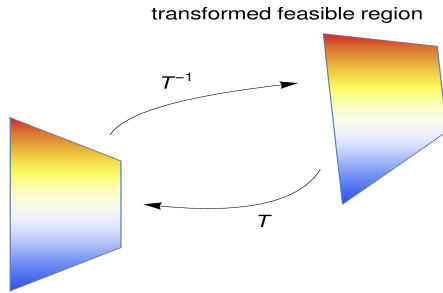


Fig. 1. Translation and rotation of the 2-dimensional Klee-Minty polytope. The contour lines of the corresponding objective function are indicated by the color (or grey-scale) gradient.

with $\mathbf{c} = (0, 0, \dots, 0, 1)^\top$. Problem (9) is referred to as *the rotated Klee-Minty problem*. It represents our proposal of a linear constrained benchmark environment that scales the number of linear inequality constraints with the dimension. By construction, the optimal objective function value is $F_{opt} = N^3$ for the relocated optimal solution.

While the rotation angle, the rotation plane, as well as the translation vector can essentially be determined randomly, this section considers fixed values as a first step. Note that the positive orthant is still used to generate an initial population of candidate solutions. The relocated optimal solution $\tilde{\mathbf{y}}^*$ may conceivably still be placed in the positive orthant of \mathbb{R}^N . To ensure the optimality of $\tilde{\mathbf{y}}^*$ rotations about angles $\varrho \in [\frac{3}{2}\pi, 2\pi]$ in the constructed hyperplane are admissible. It must be noticed that rotations of the feasible region may simplify the rotated Klee-Minty problem for LP solvers considerably. This is due to the definition of the objective function which is geared to slow progress of the LP solvers when iterating through the Klee-Minty cube. However, the focus of this paper is on the comparison of EA variants for constrained optimization. Hence, the rotations appear reasonable to address the rotational invariance of EA strategies.

3 Benchmarking conventions

In order to use the rotated Klee-Minty problem (9) as a constrained benchmark for algorithm comparison, some benchmarking principles need to be specified. These aim at providing a comprehensive benchmarking environment that allows for generating reproducible and comparable test results. In any case, algorithm developers intending to use the specified benchmark environment are prompted to carefully report on their complete algorithmic details.

Considering the original Klee-Minty cube representation [8], problem dimensionalities $N \geq 16$ would have to be excluded due to numerical instabilities.

Instead, the perturbed unit cube [2] presented in Sec. 2 allows for the consideration of larger N values. The proposed linear constrained benchmark problem takes into account search space dimensions $N \in \{2, 3, 5, 10, 20, 40\}$ ⁶. Accordingly, 6 distinct constrained functions are considered as benchmark set.

While the analytical formulation of problem (9) is available, tested algorithms are expected to treat the problem like a black-box. Each evaluation of the whole constrained function is accounted one function evaluation. The predefined budget of function evaluations is $2 \cdot 10^4 N$. However, this represents a first recommendation and may be changed according to choice.

Besides the maximal number of function evaluations, two optional termination criteria are proposed. The search may successfully stop after an algorithm approaches the known optimal function value of N^3 by a factor of 10^{-8} . Further, algorithms might stagnate in suboptimal edges of the feasible hypercube. That is, the search is also terminated after the best-so-far solution is not improved for a predefined number of generations, e.g. 1% of the number of function evaluations ($100N$). This can save considerable amounts of experimentation time.⁷

Taking into account the definition of problem (9), a box-constrained handling technique is dispensable. However, it may be applied to enforce searching in the non-negative orthant of \mathbb{R}^N . For initialization, a starting point (or population) is supposed to be randomly sampled inside the non-negative orthant of \mathbb{R}^N . Each algorithm should execute at least 15 independent runs on each constrained function, i.e. in each dimension $N \in \{2, 3, 5, 10, 20, 40\}$.

Final candidate solutions \mathbf{y} , and \mathbf{z} , realized in different algorithm runs are compared by use of a *lexicographic ordering* \preceq_{lex} that takes into account the objective function value $f(\mathbf{y})$ as well as the corresponding amount of constraint violation $\nu(\mathbf{y})$. The respective order relation is defined by

$$\mathbf{y} \preceq_{\text{lex}} \mathbf{z} \Leftrightarrow \begin{cases} f(\mathbf{y}) \leq f(\mathbf{z}), & \text{if } \nu(\mathbf{y}) = \nu(\mathbf{z}), \\ \nu(\mathbf{y}) < \nu(\mathbf{z}), & \text{else.} \end{cases} \quad (10)$$

Hence, in the context of the rotated Klee-Minty problem, the objective function value corresponding to \mathbf{y} is $f(\mathbf{y}) := \mathbf{c}^\top \mathbf{y}$. The constraint violation value $\nu(\mathbf{y})$ can be measured as the sum of the deviations over all inequality constraints⁸

$$\nu(\mathbf{y}) := \sum_{i=1}^N \max \{0, (\mathbf{A}\mathbf{R}\mathbf{y} - \mathbf{A}\mathbf{R}\mathbf{t} - \mathbf{b})_i\}. \quad (11)$$

The lexicographic order relation permits to define a number of quality indicators that can be used to assess and compare algorithm performance on problem (9).

⁶ This a first suggestion; larger search space dimensions can easily be included.

⁷ All three termination criteria were considered for both EA variants to realize the experimental results displayed in Sec. 4. Instead, Random Search (RS) omits the third termination criterion as stagnations are likely.

⁸ It is recommended to use the mentioned constraint violation definition. As there exist multiple different ways to define the constraint violation, algorithm developers may use their definition of choice. However, a detailed explanation is obligatory to ensure the comparability of algorithm test results.

Table 1. Quality indicators for algorithm assessment and comparison on the rotated Klee-Minty problem. The indicators refer to the results obtained from 15 independent algorithm runs and make use of the ordering relation according to \preceq_{lex} in Eq. (10).

f_{best}	fitness of the best found solution
$f_{\text{med}}, \nu_{\text{med}}$	fitness and constraint violation of the median solution
$ f_{\text{med}} - f_{\text{opt}} $	absolute error of the median solution
FR	feasibility rate $FR = \frac{\#\text{feasible runs}}{\#\text{runs}}$
$\ \mathbf{y} - \tilde{\mathbf{y}}^*\ $	mean deviation of all final and feasible algorithm realizations \mathbf{y} from the known optimal solution $\tilde{\mathbf{y}}^*$ over 15 algorithm runs
meanFevals	mean number of function evaluations until termination

4 Algorithm assessment and comparison

This section is concerned with the evaluation and presentation of the algorithm results obtained in 15 independent runs on the rotated Klee-Minty problem (9). To this end, two EA variants are exemplarily tested and compared to Random Search (RS). In particular, we consider the Differential Evolution (DE) variant LSHADE44 [15] (CEC2017 competition winner) and the Evolution Strategy (ES) for constrained optimization which is called $\epsilon\text{MAG-ES}^9$ [5].

In order to make a statement about algorithm performance, different quality indicators are derived from the 15 final algorithm realizations. Table 1 specifies these indicators. Accordingly, taking into account different search space dimensions, the final results are presented in the form of Table 2. This presentation style is inspired by the CEC competitions on constrained real-parameter optimization [19]. It allows comparing different algorithms with respect to realized median objective function values. Further, algorithm performance in the search space is measured by taking into account the mean deviation of the best found candidate solution from the optimal solution. As lengthy and hardly comparable tables should ideally be supported with easily interpretable figures [7], we provide an illustration of these information in Figure 2. It can be observed that both EA variants approach the optimal objective function value with the requested precision up to dimension $N = 40$. Compared to the results of the Interior Point LP solver `glpk` 3, both EA variants obtain solutions of improved quality in terms of objective function values and parameter vector accuracy.

While the CEC2017 competition does not include a notion of runtime into the algorithm assessment, we address this issue in two ways. On the one hand, the actual number of constrained function evaluations need to be reported, see Table 2. On the other hand, performance profiles or empirical cumulative distribution function (ECDF) plots are introduced. The runtime of meta-heuristic algorithms can be directly identified with the number of function evaluations needed to satisfy a number of predefined targets. This runtime definition can be traced back to [13] and is widely used in the context of the COCO BBOB bench-

⁹ For performance improvements, the ϵ threshold is initially set to zero in all algorithm runs, i.e. the ϵ -level ordering is replaced with the lexicographic ordering (10).

Table 2. Results of ϵ MAG-ES and LSHADE44 on problem (9) in dimension 2 to 40.

ϵ MAG-ES									
N	f_{opt}	f_{best}	f_{med}	ν_{med}	$ f_{\text{med}} - f_{\text{opt}} $	FR	$\ \mathbf{y} - \tilde{\mathbf{y}}^*\ $	meanFevals	
2	2^3	8.0000e+00	8.0000e+00	0	9.7490e-09	1.00	2.0549e-08	1.5506e+03	
3	3^3	2.7000e+01	2.7000e+01	0	7.5230e-09	1.00	1.0970e-08	4.2336e+03	
5	5^3	1.2500e+02	1.2500e+02	0	8.7761e-09	1.00	3.5589e-08	1.631e+04	
10	10^3	1.0000e+03	1.0000e+03	0	8.8155e-09	1.00	4.6960e-08	2.6747e+04	
20	20^3	8.0000e+03	8.0000e+03	0	9.8480e-09	1.00	5.7747e-08	8.5109e+05	
40	40^3	6.4000e+04	6.4000e+04	0	9.8225e-09	1.00	8.3878e-08	3.4478e+05	

LSHADE44									
N	f_{opt}	f_{best}	f_{med}	ν_{med}	$ f_{\text{med}} - f_{\text{opt}} $	FR	$\ \mathbf{y} - \tilde{\mathbf{y}}^*\ $	meanFevals	
2	2^3	8.0000e+00	8.0000e+00	0	7.6762e-09	1.00	1.8423e-08	3.9534e+03	
3	3^3	2.7000e+01	2.7000e+01	0	8.9507e-09	1.00	2.4902e-08	8.4597e+03	
5	5^3	1.2500e+02	1.2500e+02	0	9.4049e-09	1.00	4.5918e-08	2.3171e+04	
10	10^3	1.0000e+03	1.0000e+03	0	9.4270e-09	1.00	5.0702e-08	7.9120e+04	
20	20^3	8.0000e+03	8.0000e+03	0	9.7224e-09	1.00	1.1357e-07	2.1813e+05	
40	40^3	6.4000e+04	6.4000e+04	0	2.8513e-09	1.00	1.1861e-07	5.7090e+05	

marks¹⁰. However, the target definition used in this paper is different. Instead of defining targets only for the feasible region of the search space, we allocate 50% of the targets to the infeasible region. This supports the illustration of algorithm behavior within the infeasible region. Refer to Figure 3 for a demonstration. In total 103 target values are defined. The 51 targets in the infeasible region are uniformly distributed between 10^4 and 10^{-6} as well as 0.

Realizing a candidate solution with constrained violation below a target definition for the first time, a target value is considered to be hit. The targets in the feasible region range from 10^0 to 10^{-8} . They are reached after having realized a feasible candidate solution with an objective function value smaller than a certain target value. The ECDF plots display the ratio of reached targets for any number of function evaluations. This way they provide a notion of algorithm performance: Smaller upper left areas indicate faster algorithm running times [13]. Both runtime illustrations (in Table 2 and in Figure 3) display the

¹⁰ A more detailed explanation of the ECDF construction and interpretation is provided in [3]. Notice, this version of the rotated Klee-Minty benchmark omits the use of the bootstrapping approach mentioned in that respective paper.

Table 3. Results obtained by the deterministic Octave interior point LP-solver `gplk`.

N	f_{opt}	f	ν	$ f - f_{\text{opt}} $	FR	$\ \mathbf{y} - \tilde{\mathbf{y}}^*\ $	meanFevals	
2	2^3	8.0000e+00	0	9.7600e-09	1.00	2.7862e-08	-	
3	3^3	2.7000e+01	0	9.2953e-09	1.00	3.7072e-08	-	
5	5^3	1.2500e+02	0	1.0744e-07	1.00	5.2181e-07	-	
10	10^3	1.0000e+03	0	1.9369e-06	1.00	1.0368e-05	-	
20	20^3	8.0000e+03	0	4.1322e-06	1.00	2.3340e-05	-	
40	40^3	6.4000e+04	0	9.5593e-05	1.00	5.3734e-04	-	

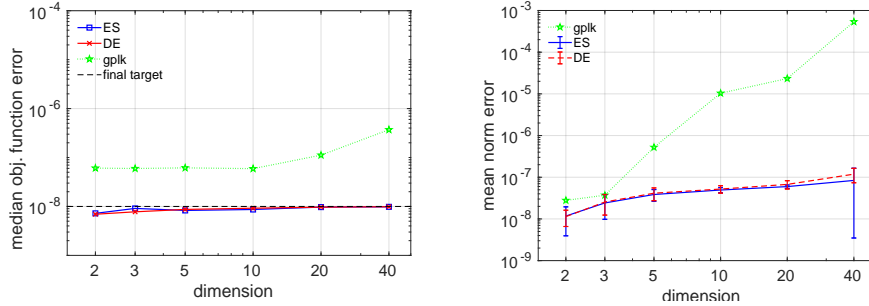


Fig. 2. Illustration of the algorithm performance with respect to the quality indicators $|f - f_{\text{opt}}|$ and $\|\mathbf{y} - \tilde{\mathbf{y}}^*\|$ reported in Table 2. All reports are plotted against the search space dimension.

runtime advantage of the ϵ MAG-ES for search space dimensionalities $N \leq 40$. Its advantage on the rotated Klee-Minty problem appears to grow with the search space dimension. Note that this runtime definition aims at the comparison of probabilistic search algorithms. Hence, it is not compatible with a comparison to the LP solver.

The algorithms are compared in each individual dimension. We propose to rank two algorithms according to the quality indicators displayed in Figure 2 as well as their run times illustrated in Figure 3. The use of three distinct equally weighted ranking factors avoids ties. In this respect, the ES receives a better rank than LSHADE44 as it basically realizes solutions of similar quality but revealing faster running times in terms of the function evaluations needed to reach the given targets. However, the final design of the ranking procedure is not ultimately determined. After having ranked all algorithms in every dimension, an overall winner can be determined by aggregating over all dimensions if considered necessary.

The developmental stage of a Matlab implementation of the rotated Klee-Minty benchmark environment is publicly available in a Github repository¹¹.

5 Conclusion

This paper suggests a novel set of linear constrained optimization problems that are suitable for benchmarking probabilistic search algorithms in a black-box setting. To this end, the Klee-Minty problem known from linear programming was modified. The emerging optimization problem is referred to as rotated Klee-minty problem. Further, corresponding reporting and presentation rules are specified to ensure reproducible and comparable benchmarking results. Still, the benchmark problems are rather statically designed. A first recommendation for the ranking of competing algorithms is provided. However, additional investigations are necessary to decide whether the proposed consensus ranking can be

¹¹ <https://github.com/hellwigm/RotatedKleeMintyProblem>

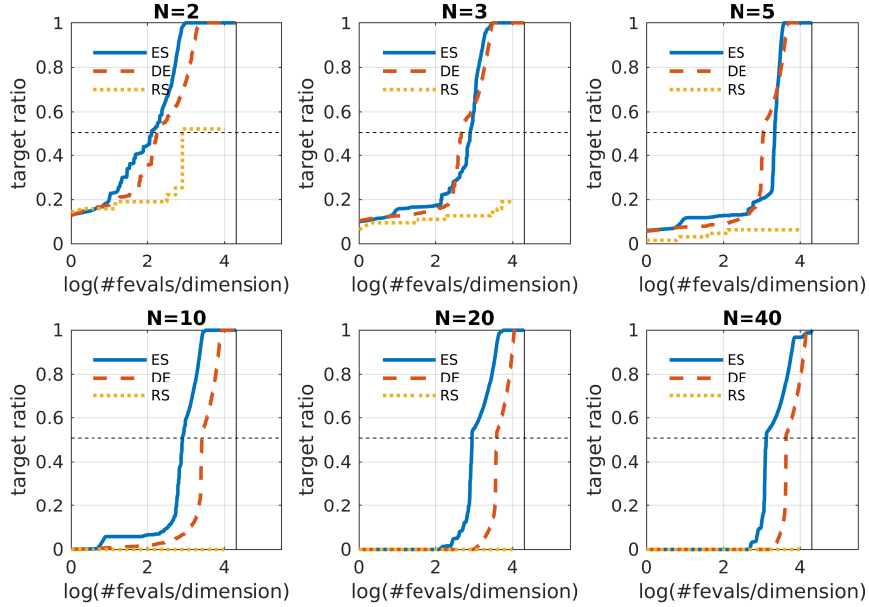


Fig. 3. ECDF results of the ϵ MAg-ES (solid lines) and the LSHADE44 (dashed lines) on problem (9). The dotted lines show the baseline results of Random Search (RS), respectively.

improved. Furthermore, the introduction of redundant constraints reduces the performance of interior point methods (without preprocessing steps) considerably [2]. As the EA performance is considered independent of such constraints, their incorporation into the proposed black-box benchmark definition may be a task for future research. These tasks and other advancements of the benchmark environment will be addressed in future research.

Representing a first proposal of a Klee-Minty based black-box optimization benchmark environment for EA variants, a constructive discussion of the benchmark design is welcome. Please contact us with suggestions for improvements and/or modifications.

Acknowledgements

This work was supported by the Austrian Science Fund (FWF) under grant P29651-N32. The article is also based upon work from COST Action CA15140 supported by COST (European Cooperation in Science and Technology).

References

1. Deza, A., Nematollahi, E., Terlaky, T.: How good are interior point methods? Klee-Minty cubes tighten iteration-complexity bounds. *Mathematical Program-*

- ming **113**(1), 1–14 (2008)
2. Deza, A., Nematollahi, E., Peyghami, R., Terlaky, T.: The central path visits all the vertices of the klee–minty cube. *Optimisation Methods and Software* **21**(5), 851–865 (2006)
 3. Hansen, N., Auger, A., Brockhoff, D., Tusar, D., Tusar, T.: COCO: performance assessment. *CoRR* **abs/1605.03560** (2016), <http://arxiv.org/abs/1605.03560>
 4. Hansen, N., Auger, A., Mersmann, O., Tusar, T., Brockhoff, D.: COCO code repository. <http://github.com/numbbo/coco>
 5. Hellwig, M., Beyer, H.G.: A matrix adaptation evolution strategy for constrained real-parameter optimization. In: 2018 IEEE Congress on Evolutionary Computation (2018), (accepted)
 6. Hellwig, M., Beyer, H.G.: Benchmarking evolutionary algorithms for single objective real-valued constrained optimization - a critical review. *arXiv preprint* (2018), <https://arxiv.org/abs/1806.04563>
 7. Johnson, D.S.: A theoretician’s guide to the experimental analysis of algorithms. In: *Data structures, near neighbor searches, and methodology: fifth and sixth DIMACS implementation challenges*. vol. 59, pp. 215–250 (2002)
 8. Klee, V., Minty, G.: How good is the Simplex algorithm? Defense Technical Information Center (1970), <https://books.google.at/books?id=R843OAAACAAJ>
 9. Liang, J.J., Runarsson, T.P., Mezura-Montes, E., Clerc, M., Suganthan, P.N., Coello Coello, C.A., Deb, K.: Problem definitions and evaluation criteria for the CEC 2006 special session on constrained real-parameter optimization (2006), online access
 10. Mallipeddi, R., Suganthan, P.N.: Problem definitions and evaluation criteria for the CEC 2010 competition on constrained real-parameter optimization (2010), online access
 11. Megiddo, N., Shub, M.: Boundary behavior of interior point algorithms in linear programming. *Mathematics of Operations Research* **14**(1), 97–146 (1989)
 12. Mezura-Montes, E., Coello, C.A.C.: What makes a constrained problem difficult to solve by an evolutionary algorithm. *Tech. rep.*, Technical Report EVOINV-01-2004, CINVESTAV-IPN, México (2004)
 13. Moré, J.J., Wild, S.M.: Benchmarking derivative-free optimization algorithms. *SIAM Journal on Optimization* **20**(1), 172–191 (2009)
 14. Neumaier, A.: Global optimization test problems, <http://www.mat.univie.ac.at/~neum/glopt.html>, Vienna University
 15. Polakova, R., Tvrdík, J.: L-SHADE with competing strategies applied to constrained optimization. In: 2017 IEEE Congress on Evolutionary Computation, CEC 2017, Donostia, San Sebastián, Spain, June 5-8, 2017. pp. 1683–1689 (2017)
 16. Spettel, P., Beyer, H.G., Hellwig, M.: A covariance matrix self-adaptation evolution strategy for linear constrained optimization. *arXiv preprint* (2018), <https://arxiv.org/abs/1806.05845>
 17. Sutton, A.M., Lunacek, M., Whitley, L.D.: Differential evolution and non-separability: using selective pressure to focus search. In: *Proceedings of the 9th annual conference on Genetic and evolutionary computation*. pp. 1428–1435. ACM (2007)
 18. Wolpert, D.H., Macready, W.G.: No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation* **1**(1), 67–82 (1997)
 19. Wu, G.H., Mallipeddi, R., Suganthan, P.N.: Problem definitions and evaluation criteria for the CEC 2017 competition on constrained real-parameter optimization (2016), online access