MAP-Elites for Constrained Optimization

Stefano Fioravanzo and Giovanni Iacca

Abstract Constrained optimization problems are often characterized by multiple constraints that, in the practice, must be satisfied with different tolerance levels. While some constraints are hard and as such must be satisfied with zero-tolerance, others may be soft, such that non-zero violations are acceptable. Here, we evaluate the applicability of MAP-Elites to "illuminate" constrained search spaces by mapping them into feature spaces where each feature corresponds to a different constraint. On the one hand, MAP-Elites implicitly preserves diversity, thus allowing a good exploration of the search space. On the other hand, it provides an effective visualization that facilitates a better understanding of how constraint violations correlate with the objective function. We demonstrate the feasibility of this approach on a large set of benchmark problems, in various dimensionalities, and with different algorithmic configurations. As expected, numerical results show that a basic version of MAP-Elites cannot compete on all problems (especially those with equality constraints) with state-of-the-art algorithms that use gradient information or advanced constraint handling techniques. Nevertheless, it has a higher potential at finding constraint violations vs. objectives trade-offs and providing new problem information. As such, it could be used in the future as an effective building-block for designing new constrained optimization algorithms.

1 Introduction

Several real-world applications, for instance in engineering design, control systems and healthcare, can be described in the form of constrained continuous optimization

---

Stefano Fioravanzo
Arrikto Inc. 60 E 3rd Ave, Ste 262, San Mateo, CA 9441 e-mail: `stefano@arrikto.com`

Giovanni Iacca
University of Trento, Trento, 38122, Italy e-mail: `giovanni.iacca@unitn.it`

problems, i.e. problems where a certain objective/cost function must be optimized within a certain search space, subject to some problem-dependent constraints. Without loss of generality, these problems can be formulated as:

$$\begin{aligned}
\underset{\mathbf{x}\in\mathbf{D}}{\text{minimize}} \quad & f(\mathbf{x}) \\
\text{subject to:} \quad & g_i(\mathbf{x}) \le 0, \quad i = 1, 2, \ldots, m \\
& h_j(\mathbf{x}) = 0, \quad j = 1, 2, \ldots, p
\end{aligned}$$

where; 1) $\mathbf{x} \in \mathbf{D} \subseteq \mathbb{R}^n$ is a candidate solution to the problem, being n the problem dimensionality, and $\mathbf{D}$ the search space, typically defined in terms of bounding box constraints $lb_k \le x_k \le ub_k \ \forall k \in \{1, 2, \ldots, n\}$, where lb_k and ub_k are the lower and upper bound, respectively, for each k-th variable; 2) $f(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}$ is the objective function; 3) $g_i(\mathbf{x})$ and $h_j(\mathbf{x})$ (both defined as: $\mathbb{R}^n \to \mathbb{R}$) are respectively, inequality and equality constraints. A summary of the aforementioned symbols, as well as the symbols used in the rest of the paper, is reported in Table 1.

Table 1: List of symbols used in the paper (symbols in boldface indicate vectors)

Symbol	Meaning
n	Problem dimensionality (no. of variables)
N	Number of features (i.e., no. of tolerance levels)
lb_k	Lower bound of k-th variable ($k = 1, 2, \ldots n$)
ub_k	Upper bound of k-th variable ($k = 1, 2, \ldots n$)
$\mathbf{D}$	Search space ($\subseteq \mathbb{R}^n$)
$f(\mathbf{x})$	Objective function calculated in $\mathbf{x}$
$g_i(\mathbf{x})$	i-th inequality constraint calculated in $\mathbf{x}$ ($i = 1, 2, \ldots m$)
$h_j(\mathbf{x})$	j-th equality constraint calculated in $\mathbf{x}$ ($j = 1, 2, \ldots p$)
$\mathbf{x}$	Candidate solution ($\in \mathbb{R}^n$)
$\mathbf{x}'$	Candidate solution ($\in \mathbb{R}^n$)
$\mathbf{b}'$	Feature descriptor calculated in $\mathbf{x}'$ ($\in \mathbb{R}^N$), i.e. constraint violations within each of the N predefined tolerance levels
$\mathcal{P}()$	Associative map <feature descriptor, performance>
$\mathcal{X}()$	Associative map <feature descriptor, solution>
$\mathcal{P}(\mathbf{b}')$	Best performance associated to the feature descriptor $\mathbf{b}'$ (it can be empty)
$\mathcal{X}(\mathbf{b}')$	Best solution associated to the feature descriptor $\mathbf{b}'$ (it can be empty)
p'	Performance calculated in $\mathbf{x}'$, i.e. $p' = f(\mathbf{x}')$
g	Current number of iterations ($1, 2, \ldots I$)
I	Number of function evaluations (NFEs)
G	Number of function evaluations (NFEs) for map initialization ($G < I$)

In the past three decades, a large number of computational techniques has been proposed to solve efficiently this class of problems, among which Evolutionary Algorithms (EAs) [17] have shown a great potential due to their general applicability and effectiveness. So far, most of the research in the field has focused on how to improve the *feasible* results obtained by EAs, for instance developing *ad hoc* evolutionary operators, specific constraint repair mechanisms, or constraint handling techniques (CHTs). However, in various real-world applications it could be desirable,

or at least acceptable, to consider also *infeasible* solutions. This could be obtained for instance by defining different *tolerance levels* for each constraint, so to reason on the effect of relaxing a certain constraint (and, if so, how much to do that) in order to obtain an improvement on the objective function, and therefore find different trade-offs in terms of constraint violations vs. objective [25]. Despite these application needs, to date little research effort has been put on how to allow EAs to identify, rather than a single optimal solution, a *diverse* set of solutions characterized by different trade-offs of this kind. In this sense, the most notable exceptions that explicitly addressed this problem –although with contrasting results– have focused on multi-objective approaches, where the constraint violations were considered as additional objectives to be minimized [26, 35, 33, 10, 5], or surrogate methods [2].

In this paper, our goal is to evaluate the applicability of the Multi-dimensional Archive of Phenotypic Elites (MAP-Elites) [3, 19], an EA recently introduced in the literature in the context of robotic tasks, for tackling these problems, specifically to provide trade-off solutions in constrained optimization. Differently from conventional EAs, MAP-Elites conducts the search by mapping the highest-performing solutions found during the search (elites) into another multi-dimensional *discretized* space, defined by problem-specific features (the latter space is separate from the original search space, and typically of a lower dimensionality). These features are uncorrelated to the actual objective function, and describe some domain-specific properties of the candidate solutions. By means of this mapping, the algorithm "illuminates" the search space by showing the potential value of each area of the feature space, and the corresponding trade-off between the objective and the features of interest.

The functioning of MAP-Elites is simple and intuitive. First, the multi-dimensional feature space is discretized into a multi-dimensional grid, where each bin (i.e., a cell in the grid, which is in general a hyper-rectangle) represents a different "niche". Then, an EA-like search is performed by means of selection and variation (mutation and crossover), but instead of keeping a population of solutions which may or may not be diverse, MAP-Elites explicitly maintains diversity by keeping in each niche one elite, which identifies the best solution characterized by the corresponding feature values. At the end of the optimization procedure, a full *map* of possible solutions is provided (rather than a single optimal solution, as in conventional single-objective EAs), each characterized by different features. This map is shown in the form of a multi-dimensional heatmap, which allows for an easy visual inspection of how the objective function changes across the feature space.

In order to apply MAP-Elites to constrained optimization, the main idea we propose here is to define the feature space based on a discretization of the constraint violations. It is worth noting that in practical applications the discretization has a concrete, domain-dependent meaning: it can be seen as a set of tolerance levels, which as we mentioned can be different for each constraint. With this approach, we are able to produce a visual representation of the objective values in the feature space (in this case, space of constraint violations), thus uncovering possible correlations between the constraints and the objective. Thanks to this visualization, it is indeed easy to understand "where", with respect to the constraints boundaries, the best

solutions lie. It also is easy to inspect the best overall solution, and check if the algorithm was able to produce particularly interesting solutions violating some of the constraints. As said, this insight can be helpful in cases where the violation of some constraints (within a certain tolerance level) can be an acceptable trade-off for a better overall performance.

As we will see in detail in the paper, despite its simplicity the proposed approach has various advantages: 1) it can be easily adapted/extended to include custom evolutionary operators; 2) it does not necessarily need explicit CHTs, but it can also include them; 3) it implicitly preserves diversity; 4) it allows the user to easily define custom tolerance levels, different for each constraint; 5) it "illuminates" the search space as it provides additional information on the correlation between constraints and objective, which might be of interest in practical applications; 6) it facilitates the interpretation of results through an intuitive visualization.

The rest of the paper is structured as follows. In Section 2, we will briefly summarize the most recent works on MAP-Elites and constrained optimization. Then, Section 3 describes the basic MAP-Elites algorithm and how it can be applied to constrained optimization. In Section 4, we describe the experimental setup (benchmark and algorithmic settings), followed by the analysis of the numerical results, reported in Section 5. Finally, Section 6 concludes this work and suggests possible future developments.

2 Related work

The study of MAP-Elites and, more in general, EAs explicitly driven by *novelty* [12, 18, 24] or *diversity* [20, 4], rather than the objective alone, is a relatively new area of research in the Evolutionary Computation community. Among these algorithms, MAP-Elites [3, 19] has attracted quite some attention in the field, due to its simplicity and general applicability. Since its introduction in 2015, MAP-Elites has been mostly used as a means to identify *repertoires* of different agent behaviors e.g. in evolutionary robotics setups. Various examples of applications to maze navigation, legged robot gait optimization, and anthropomorphic robot trajectory optimization can be found in [1, 3, 19, 30, 32, 31, 23, 4]. More recently, MAP-Elites has been applied also to Workforce Scheduling and Routing Problem (WSRP) [29] and Genetic Programming [6]. To the best of our knowledge, no prior work exists on the explicit use of MAP-Elites for solving constrained optimization problems.

Evolutionary constrained optimization is, on the other hand, a much more mature area of research: hundreds of papers have shown in the past three decades various algorithmic solutions and real-world problems where EAs were successfully applied to constrained optimization. Summarizing all the recent advances in this area would be impossible, and is obviously outside the scope of this paper. A thorough survey of the literature is performed, for instance, in [13], to which we refer the interested reader for a comprehensive analysis of the state-of-the-art updated to 2016. Another

interesting study, published at the end of 2018 by Hellwig and Beyer [9], covers all the aspects related to benchmarking EAs for constrained optimization, including a thorough analysis of the most important benchmark suites available in the literature. Among these, the CEC 2010 benchmark [15] has attracted in the past few years a large body of works that showed how to solve its functions efficiently, and is often used nowadays for benchmarking new algorithms. Currently, the state-of-the-art results on this benchmark have been obtained by εDEag, an ε constrained Differential Evolution algorithm with an archive and gradient-based mutation proposed by Takahama and Sakai [28], followed by ECHT-DE, another variant of Differential Evolution that includes an ensemble of four constraint handling techniques, proposed by Mallipeddi and Suganthan [14]. These two works are also good examples of two of the most successful recent trends in the field, that are the use of gradient-based information (if available, or at least approximable), and the combination of multiple CHTs into a single evolutionary algorithm.

3 Methodology

The basic version of MAP-Elites, as introduced in [3, 19], is shown in Algorithm 1. In the pseudo-code, $\mathbf{x}$ and $\mathbf{x}'$ are candidate solutions (i.e., n-dimensional vectors defined in the search space $\mathbf{D}$); $\mathbf{b}'$ is a *feature descriptor*, that is a location in a user-defined *discretized* feature space, corresponding to the candidate solution $\mathbf{x}'$, (i.e., an N-dimensional vector of user-defined features that characterize $\mathbf{x}'$, typically with $N < n$); p' is the performance of the candidate solution $\mathbf{x}'$ (i.e., the scalar value returned by the objective function $f(\mathbf{x}')$; the function itself is assumed to be a black-box, that is its mathematical formulation, if any, is unknown to the algorithm); $\mathcal{P}$ is a <feature descriptor, performance> map (i.e. an associative table that stores the best performance associated to each feature descriptor encountered by the algorithm); $\mathcal{X}$ is a <feature descriptor, solution> map (i.e. an associative table that stores the best solution associated to each feature descriptor encountered by the algorithm); $\mathcal{P}(\mathbf{b}')$ is the best performance associated to the feature descriptor $\mathbf{b}'$ (it can be empty); $\mathcal{X}(\mathbf{b}')$ is the best solution associated to the feature descriptor $\mathbf{b}'$ (it can be empty).

Following the pseudo-code, the algorithm first creates the two maps $\mathcal{P}$ and $\mathcal{X}$, which are initially empty. Then, a loop of I iterations (i.e., function evaluations) is executed. For each of the first G iterations, G solutions are randomly sampled in the search space $\mathbf{D}$, which are used for initializing the two maps $\mathcal{P}$ and $\mathcal{X}$. Then, starting from the iteration $G + 1$, a solution $\mathbf{x}$ is randomly selected from the current map $\mathcal{X}$, and a randomly modified copy of it, $\mathbf{x}'$, is generated. The feature descriptor $\mathbf{b}'$ and performance p' associated to this new, perturbed solution are then evaluated. At this point, the two maps $\mathcal{P}$ and $\mathcal{X}$ are updated: if the performance associated to $\mathbf{b}'$, $\mathcal{P}(\mathbf{b}')$, is empty (which can happen if this is the first time that the algorithm generates a solution with that feature descriptor), or if it contains a value that is worse than the performance p' of the newly generated solution (in Algorithm 1, we assume a minimization problem, therefore we check the condition $\mathcal{P}(\mathbf{b}') > p'$), the

---

Algorithm 1 MAP-Elites algorithm, taken from [19]

$\mathcal{P} \leftarrow \emptyset, \mathcal{X} \leftarrow \emptyset$
for $g = 1 \rightarrow I$ **do**
    if $g < G$ **then**
        $\mathbf{x}' \leftarrow$ random_solution()
    else
        $\mathbf{x} \leftarrow$ random_selection($\mathcal{X}$)
        $\mathbf{x}' \leftarrow$ random_variation($\mathbf{x}$)
    $\mathbf{b}' \leftarrow$ feature_descriptor($\mathbf{x}'$)   ▷ for constrained optimization, $\mathbf{b}'$ is a vector of constraint violations
    $p' \leftarrow$ performance($\mathbf{x}'$)
    if $\mathcal{P}(\mathbf{b}') = \emptyset \vee \mathcal{P}(\mathbf{b}') > p'$ **then**
        $\mathcal{P}(\mathbf{b}') \leftarrow p'$
        $\mathcal{X}(\mathbf{b}') \leftarrow \mathbf{x}'$
return $\mathcal{P}$ and $\mathcal{X}$

---

new solution $\mathbf{x}'$ and its performance p' are assigned to the elements of the maps corresponding to its feature descriptor $\mathbf{b}'$, namely $\mathcal{P}(\mathbf{b}')$ and $\mathcal{X}(\mathbf{b}')$. Once the loop terminates, the algorithm returns the two maps $\mathcal{P}$ and $\mathcal{X}$, which can be later analyzed for further inspection and post-processing.

It can be immediately noted how simple the algorithm is. With reference to the pseudo-code, in order to apply MAP-Elites to a specific problem the following methods must be defined:

- random_solution(): returns a randomly generated solution;
- random_selection($\mathcal{X}$): randomly selects a solution from $\mathcal{X}$;
- random_variation($\mathbf{x}$): returns a modified copy of $\mathbf{x}$;
- feature_descriptor($\mathbf{x}$): maps a candidate solution $\mathbf{x}$ to its representation in the feature space, $\mathbf{b}$;
- performance($\mathbf{x}$): evaluates the objective function corresponding to the candidate solution $\mathbf{x}$.

The first three methods are rather standard, i.e., they can be based on general-purpose operators typically used in EAs. However, it is possible to customize them according to the specific need. For instance, the basic version MAP-Elites randomly selects at each iteration one solution, and applies only Gaussian mutation operator; on the other hand, the algorithm can be easily configured to use a different selection mechanism (e.g. an informed operator that introduces some selection pressure/bias) or select multiple solutions at each iteration so to apply a recombination operator (crossover) or some other search mechanism such as a local search. We will see in Section 4 the details of three different algorithm configurations that we have used in our experimentation.

As for what concerns feature_descriptor($\mathbf{x}$) and performance($\mathbf{x}$), these are obviously problem-dependent: the first one, being dependent on how the user defines the features of interest and the corresponding feature space; the latter, being dependent on the specific objective function at hand.

The application of MAP-Elites to constrained optimization is then quite straightforward: here, we map each constraint of a constrained optimization problem to a different feature in the feature space explored by MAP-Elites, such that each candidate solution is associated to a feature descriptor that is basically a vector of constraint violations. In this specific case then, the user does not necessarily have to define any additional feature, but the features themselves are already part of the problem definition. Leaving aside the algorithmic details (selection and variation) and parameters (the only two parameters of the algorithm are the total and initial number of iterations, respectively I and G, which can be easily set by the user based on computing resources and/or time constraints), the only input required from the user is the discretization of the features (constraint violations) space.

An intuitive way of discretizing this space is to define, for each constraint, a certain number of *tolerance levels*, i.e. amounts of constraint violation used as discretization steps. These can be easily expressed in absolute terms (based on the values of $g_i(\mathbf{x})$ and $h_j(\mathbf{x})$ in case of violations), or normalized w.r.t. known minimum and maximum violations. A simple example of discretization steps is $\{0, \varepsilon, 2\varepsilon, \dots\}$, where ε is a user-defined parameter. However, as we will show in Section 4, also non-linear discretization is possible. In general, the discretization strategy should be based on domain knowledge and defined in such a way that solutions whose violations are equivalent, from a practical point of view, are grouped in the same bin. This would allow to "illuminate" the relation between objective function and constraint violations in a significant, meaningful way. Finally, we must note that while in general a different set of *tolerance levels* can be defined for each constraint (especially if these are expressed in absolute terms), if all constraints have the same codomain (or, if they are normalized), the same tolerance levels can be used for all of them.

4 Experimental Setup

We evaluated the performance of the proposed approach on the benchmark functions defined for the CEC 2010 Competition on Constrained Real-Parameter Optimization [15]. This benchmark presents 18 problems with different landscape characteristics, subject to a varying number (up to four) of equality and/or inequality constraints. To assess the scalability of MAP-Elites, we tested these problems in 10 and 30 dimensions. All the details of the experimental setup were set according to the CEC indications [15], with the main parameters listed in Table 2.

As for the discretization steps, the selected values correspond to having each feature discretized into 5 bins, namely: $\{0\}$, $(0, 0.0001]$, $(0.0001, 0.01]$, $(0.01, 1.0]$ and $(1.0, \inf)$. This last aspect deserves attention, since as we have seen in Section 3 this is what allows the application of MAP-Elites to constrained optimization. Here, we have defined as discretization steps the three tolerance levels defined in [15], i.e. 0.0001, 0.01, and 1.0 (see Section 5), in addition to an explicit step corresponding to

Table 2: Parameters used in the experimental setup

Parameter	Value
Number of benchmark problems	18 {C1, C2, . . . , C18}
Number of dimensions (n), for each problem	10D and 30D
Number of runs, for each problem/dimensionality	25
NFEs, per run	I = 2.0e5 for 10D
	I = 6.0e5 for 30D
NFEs for map initialization, per run	G = 2000
Discretization steps for every feature	{0, 0.01, 0.0001, 1}

zero-tolerance (corresponding to solutions with $g_i(\mathbf{x}) \leq 0$ and $h_j(\mathbf{x}) = 0$, respectively for inequality and equality constraints).

The features used in MAP-Elites follow the same order as they appear in the corresponding problem definition, with inequality constraints considered before equality constraints. Since the problems contained in the the CEC 2010 benchmark have a variable number of equality/inequality constraints, we define a variable-sized feature space, where for each problem there are as many MAP-Elites features as constraints. For visualization purposes, we represent the map $\mathcal{P}(\mathbf{b}')$ obtained in each run of MAP-Elites in the form of a multi-dimensional heatmap, as explained in [3, 19]. The color represents the objective value corresponding to the solution contained in each bin. The 1st axis (abscissa) corresponds to the 1st constraint violation, the 2nd axis (ordinate) corresponds to the 2nd constraint violation, and so on for the 3rd and 4th constraints. Feature dimensions are "nested" such that the feature space is first discretized along the 1st and 2nd axes (so to obtain a 2D grid of bins), while the following features (if any) are represented by an "inner" (1D or 2D) discretization inside each bin in the "outer" grid. Obviously, this visualization procedure can be easily extended to handle more than four constraints, although the visual interpretability of the results tends to decrease with the number of features shown in the heatmap.

It should be noted that, according to the CEC 2010 benchmark definition [15], a solution $\mathbf{x}$ is considered feasible iff $g_i(\mathbf{x}) \leq 0$ $\forall i \in \{1, 2, \ldots, m\}$ and $|h_j(\mathbf{x})| - \epsilon \leq 0$ $\forall j \in \{1, 2, \ldots, p\}$, where ϵ is the equality constraint tolerance, set to 0.0001. Otherwise, the solution is considered infeasible. From what we have just discussed, it follows then that feasible solutions in the sense of the CEC 2010 definition can be found: 1) for what concerns inequality constraints, in the first bin ({0}) along each feature dimension; 2) for what concerns equality constraints, in the first two bins ({0}, (0, 0.0001]). In plain terms, this means that we can easily identify feasible solutions found by MAP-Elites by simply looking at the lower-left corner of the heatmap, while solutions with increasing constraint violations are found scanning the heatmap (and each inner bin in case of more than two constraints) towards the upper-right side. Some examples of heatmaps obtained by MAP-Elites are shown in Figure 1.

As for the evolutionary operators (selection and variation, as shown in Algorithm 1), we defined three different algorithmic settings. In all cases, selection is
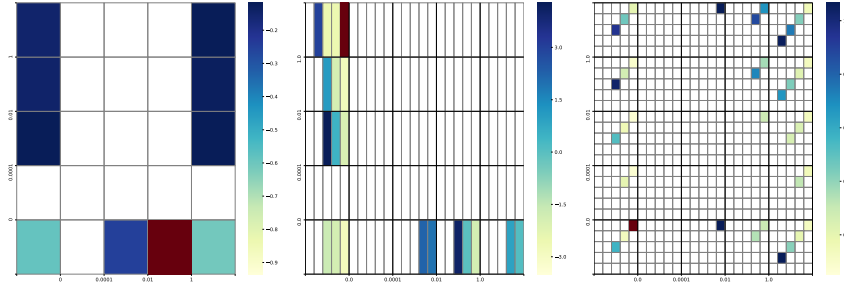
Fig. 1: Final heatmaps found in a single run of MAP-Elites on C1, Configuration 1 (left), C2, Configuration 1 (center), C16, Configuration 3 (right) in 10D. The three benchmark functions are characterized, respectively, by 2I, 2I-1E, and 2I-2E, where 'I' and 'E' stand for inequality and equality constraints respectively. In each heatmap, the color of each bin is proportionate to the objective value corresponding to the solution present in it (assuming minimization, the lower the better), while the red bin indicates the solution with the best objective value (regardless its feasibility). It can be observed that the maps allows to "illuminate" the search space of each problem, identifying various trade-off solutions in terms of objective value vs. constraint violations, such as solutions with a high performance but with some violated constraints. Note that in case of 3 or 4 constraints the discretization along the first two (outer) dimensions of the heatmap is indicated by a thicker black line, while the discretization along the "nested" (inner) dimensions is indicated by a thinner black line.

performed according to a uniform distribution over the current map. Variation is instead applied according to the following configurations:

- Configuration 1: mutation ($\sigma = 0.1$), without crossover
- Configuration 2: mutation ($\sigma = 0.5$), without crossover
- Configuration 3: mutation ($\sigma = 0.1$), with crossover

In all three cases, mutation is implemented by applying to the selected solution (with probability 0.5 for each variable) a Gaussian mutation with $\mu = 0$ and the given value of σ. Boundary constraints are handled according to a toroidal mechanism: given a decision variable x constrained to the interval $[a, b]$, if the corresponding mutated variable x' exceeds the upper bound b (i.e., $x' = b + \zeta$), x' is transformed into $x' = a + \zeta$, $\zeta > 0$. Similarly, if $x' = a - \zeta$, x' is transformed into $x' = b - \zeta$, $\zeta > 0$.

In Configuration 3, at each iteration two solutions are randomly selected from the current map, after which uniform crossover (with probability 0.5 for each variable) is applied by swapping the corresponding variables from the two parents. Then, the first of the two offspring generated by crossover undergoes Gaussian mutation, as in Configurations 1 and 2, and is evaluated in terms of feature descriptor and performance, as shown in Algorithm 1.

The entire experimental setup was implemented in Python 3[1], and the experimentation was performed on a Ubuntu 18.10 workstation, with a CPU Intel Core i9-7940X @3.10GHz and 64GB DDR4.

---

[1] Code available at: `https://github.com/StefanoFioravanzo/MAP-Elites`.

5 Numerical results

We present here the results obtained on the experimental setup described in Section 4. In Tables 3-8, we report the results for all the CEC 2010 functions in 10 and 30 dimensions, for the three algorithm settings described above[2]. In the tables, we report the results as suggested in [15], where for each function we show:

1. The objective value corresponding to the best, worst, and median solution[3] (over 25 runs) obtained at the end of the computational budget; next to each objective value, we show in parenthesis the no. of violated constraints corresponding to each of these three solutions.
2. The number of violated constraints at the median solution, $c = (c_1, c_2, c_3)$ (where each element $c_i, i = 1, 2, 3$ represents the number of violations higher than three tolerance levels set to 1, 0.01, and 0.0001, respectively), and the corresponding mean violation $\bar{v}$, calculated as:

$$\bar{v} = \frac{\sum_{i=1}^{m} G_i(x) + \sum_{j=1}^{p} H_j(x)}{m + p} \tag{1}$$

   where $G_i(x) = g_i(x)$ if $g_i(x) > 0$ (otherwise zero), and $H_j(x) = |h_j(x)|$ if $|h_j(x)| - \epsilon > 0$ (otherwise zero), being ϵ is the equality constraint tolerance (as seen earlier, 0.0001).
3. The average objective value (over 25 runs) of the final solutions obtained at the end of the budget, and its std. dev.
4. The Feasibility Rate (FR), that is, for each function, the ratio between the number of runs during which at least one feasible solution was found within the budget, and the total number of runs (in our case, 25).

For reference, we report in Tables 9-10 the results for all the CEC 2010 functions after $I = 2.0e5$ and $I = 6.0e5$ NFEs, respectively for 10D and 30D, obtained by εDEag [28], the best algorithm on the CEC 2010 benchmark.

From the tables, we can observe that in all three configurations, MAP-Elites solves with 100% FR C1, C7, C8, C14, C15 in 10D, i.e. all the functions with inequality constraints only (except C13, that is however the only function with inequality constraints only whose volume of the feasible region is approximately zero); in 30D, it also finds feasible solutions on C18 in 100% of the runs in all configurations (92% in 10D for Configuration 3). The peculiarity of this function is that, despite it has one equality constraint, the volume of its feasible region is non-zero. The only other functions on which a non-zero FR is obtained, although not in all configurations and dimensionalities, are: C2, C9, C10, C16, C17. Except C10 that has 1 "rotated"

---

[2] The complete set of numerical results and the final heatmaps for each problem and dimensionality are available as Supplementary Material at: `https://bit.ly/2BQIR8B`.

[3] The final solutions are sorted according to these three criteria: 1) feasible solutions are sorted in front of infeasible solutions; 2) feasible solutions are sorted according to their objective value; 3) infeasible solutions are sorted according to their mean value of constraint violation, calculated as in Eq. (1).

equality constraint, all other functions have only separable constraints, which could explain why in some cases even by Gaussian mutation only (which acts independently on each variable) it is possible to reach the feasible region. Overall, among the 3 configurations Configuration 3 has the highest FR across all the tests, while it results that an excessively high value of σ in Gaussian mutations (as in Configuration 2) is detrimental.

From these observations, we can conclude that the basic MAP-Elites algorithm we have used in our experimentation is not able to solve efficiently either problems with non-separable equality constraints, or with an approximately zero-volume feasible region. This is not surprising though, as the algorithm is only based on simple genetic operators (Gaussian mutations and uniform crossover in our case) that do not use any information about the constraints. In contrast, εDEag [28] encapsulates highly efficient CHTs and uses gradient information about constraints that allows the algorithm to reach an 100% FR on all functions in 10D and 30D (except, respectively for the two algorithms, C12 in 30D, and C11-C12 in both 10D and 30D), as reported in the original papers.

This comparison encourages though the idea to explore in the future the possibility to include into the MAP-Elites scheme at least one dedicated technique for better handling equality constraints, such as the ε constrained method, initially introduced in [27] and since then used in most of the state-of-the-art algorithms for constrained optimization. Notably, the strength of this method is that it guides the search by allowing ε level comparisons with a progressively shrinking relaxation (defined by the ε parameter) of the constraint boundaries.

Considering the objective values, similar considerations can be drawn: limiting the analysis on the functions with 100% FR, it results that MAP-Elites is less efficient at finding optimal values than εDEag [28]. In all cases MAP-Elites is several orders of magnitude worse than εDEag, except C1 in 10D where instead the configuration with crossover finds a better optimal value[4]. Once again, this conclusion is not surprising and is also in line with what was observed by Runarsson and Yao [22], who identified the reason for the sometimes poor results obtained by multi-objective approaches (such as [26, 35, 33, 10]): in fact, when applied to constrained optimization, the Pareto ranking leads to a "bias-free" search that is not able to properly guide the search towards (and within) the feasible region. In other words, allowing the search to spend too many evaluations in the infeasible region makes it harder to find feasible solutions, but also to find feasible solutions with optimal values of the objective function. This might be the case also of MAP-Elites, where some form of bias (such as the ε constrained method) might be needed.

---

[4] We refer the interested reader to the Supplementary Material online, where we show a detailed report of the MAP-Elites results focused on a fitness-based rank, rather than the rank based on the sorting criteria described in the text. These results are omitted here for brevity.

6 Conclusions

In this paper we have explored the use of MAP-Elites for solving constrained continuous optimization problems. In the proposed approach, each feature in the feature space explored by MAP-Elites corresponds, quite straightforwardly, to the violation of each constraint, discretized according to user-defined steps (tolerance levels). In this way, the algorithm allows to "illuminate" the search space and thus uncover possible correlations between the constraints and the objective. The visualization of MAP-Elites also gives users the possibility to focus on different solutions characterized by different values of constraint violations. We have tested this approach on a large number of benchmark problems in 10 and 30 dimensions, characterized by up to four equality/inequality constraints. Our numerical results showed that while MAP-Elites obtains results that are not particularly competitive with the state-of-the-art on all problems (especially those with equality constraints), it is still able to provide new valuable, easy-to-understand information that can be of great interest for practitioners. Additionally, the algorithm can be easily implemented and applied without any specific tuning to various real-world problems, for instance in engineering design, where different tolerance levels can be defined depending on the specific constraints.

Since our goal was to evaluate the performance of the basic MAP-Elites on constrained optimization, the proposed approach is purposely quite simplistic, but clearly it can be extended in various ways. First of all, the basic MAP-Elites algorithm we used in this work (as shown in Algorithm 1 can be replaced with some more advanced variants recently proposed in the literature. In particular, the version of MAP-Elites based on centroidal Voronoi tessellation (CVT-MAP-Elites) [31] can be used instead of the basic one in order to scale the algorithm to a larger number of constraints. In order to better handle the unbounded feature spaces (thus avoiding the need for an explicit "upper" bin, $(1.0, \inf)$ in our case), the "expansive" MAP-Elites variants introduced in [30] can be employed instead, which are able to expand their bounds (in the feature space) based on the discovered solutions. Other possibilities will be to use the "directional variation" operator introduced in [32], that exploits inter-species (or inter-elites) correlations to accelerate the search, add, most of all, specific constraint handling techniques (especially for handling equality constraints, which as we have seen is the main weakness of this approach) [14, 7]. It is also worth considering the use of surrogate models, such as in [34], in order to further speed up the the search and guide it towards the feasible region, still allowing the algorithm to keep infeasible solutions as part of the map. Another improvement can be obtained by using an explicit Quality Diversity measure [1, 21, 20, 4], so to enforce at the same time a better coverage of the feature space and a further improvement in terms of optimization results. It is also possible to hybridize the basic MAP-Elites algorithm with local search techniques (such that MAP-Elites explores the feature space and local search is applied within one bin to further refine the search), or to devise memetic computing approaches based on combination of MAP-Elites and other metaheuristics, such as CMA-ES, that has been recently applied successfully also to constrained optimization [16, 11, 8].

Finally, on the application side it will be interesting to evaluate the applicability of this approach on combinatorial constrained optimization problems, which can be obtained by simply modifying the variation operators, or multi-objective constrained optimization, which can be obtained by adding a Pareto-dominance check, as recently shown in the context of robotic experiments [23].

References

1. Auerbach, J.E., Iacca, G., Floreano, D.: Gaining insight into quality diversity. In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO) Companion, pp. 1061–1064. ACM, New York, NY, USA (2016)
2. Bagheri, S., Konen, W., Bäck, T.: How to solve the dilemma of margin-based equality handling methods. In: F. Hoffmann, E. Hüllermeier, R. Mikut (eds.) Proceedings of the Workshop Computational Intelligence, pp. 257–270. KIT Scientific Publishing, Karlsruhe (2018)
3. Cully, A., Clune, J., Tarapore, D., Mouret, J.B.: Robots that can adapt like animals. Nature **521**(7553), 503 (2015)
4. Cully, A., Demiris, Y.: Quality and diversity optimization: A unifying modular framework. IEEE Transactions on Evolutionary Computation **22**(2), 245–259 (2018)
5. Deb, K., Datta, R.: A hybrid bi-objective evolutionary-penalty approach for computationally fast and accurate constrained optimization. Tech. rep., KanGAL (2010)
6. Dolson, E., Lalejini, A., Ofria, C.: Exploring genetic programming systems with map-elites. PeerJ Preprints **6**, 1–18 (2018)
7. Gong, W., Cai, Z., Liang, D.: Adaptive ranking mutation operator based differential evolution for constrained optimization. IEEE Transactions on Cybernetics **45**(4), 716–727 (2015)
8. Hellwig, M., Beyer, H.G.: A matrix adaptation evolution strategy for constrained real-parameter optimization. In: Congress on Evolutionary Computation (CEC), pp. 1–8. IEEE, New York (2018)
9. Hellwig, M., Beyer, H.G.: Benchmarking evolutionary algorithms for single objective real-valued constrained optimization–a critical review. Swarm and evolutionary computation **44**, 927–944 (2019)
10. Hernáandez Aguirre, A., Botello Rionda, S., Lizáarraga Lizáarraga, G., Coello Coello, C.: Is-paes: Multiobjective optimization with efficient constraint handling. In: T. Burczyński, A. Osyczka (eds.) IUTAM Symposium on Evolutionary Methods in Mechanics, pp. 111–120. Springer, Dordrecht (2004)
11. Jamal, M.B., Ming, F., Zhengang, J.: Solving constrained optimization problems by using covariance matrix adaptation evolutionary strategy with constraint handling methods. In: Proceedings of the International Conference on Innovation in Artificial Intelligence, pp. 6–15. ACM, New York, NY, USA (2018)
12. Lehman, J., Stanley, K.O.: Exploiting open-endedness to solve problems through the search for novelty. In: International conference on Artificial Life (Alife XI), pp. 329–336. MIT Press, Cambridge, MA (2008)
13. Maesani, A., Iacca, G., Floreano, D.: Memetic viability evolution for constrained optimization. IEEE Transactions on Evolutionary Computation **20**(1), 125–144 (2016)
14. Mallipeddi, R., Suganthan, P.N.: Differential evolution with ensemble of constraint handling techniques for solving cec 2010 benchmark problems. In: Congress on Evolutionary Computation (CEC), pp. 1–8. IEEE, New York (2010)
15. Mallipeddi, R., Suganthan, P.N.: Problem definitions and evaluation criteria for the cec 2010 competition on constrained real-parameter optimization. Tech. rep., NTU, Singapore (2010)
16. de Melo, V.V., Iacca, G.: A modified covariance matrix adaptation evolution strategy with adaptive penalty function and restart for constrained optimization. Expert Systems with Applications **41**(16), 7077–7094 (2014)

17. Michalewicz, Z., Schoenauer, M.: Evolutionary algorithms for constrained parameter optimization problems. Evolutionary Computation **4**(1), 1–32 (1996)
18. Mouret, J.B.: Novelty-based multiobjectivization. In: S. Doncieux, N. Bredèche, J.B. Mouret (eds.) New Horizons in Evolutionary Robotics, pp. 139–154. Springer, Berlin, Heidelberg (2011)
19. Mouret, J.B., Clune, J.: Illuminating search spaces by mapping elites. CoRR **abs/1504.04909**, 1–15 (2015)
20. Pugh, J.K., Soros, L.B., Stanley, K.O.: Quality diversity: A new frontier for evolutionary computation. Frontiers in Robotics&AI **3**, 40 (2016)
21. Pugh, J.K., Soros, L.B., Szerlip, P.A., Stanley, K.O.: Confronting the challenge of quality diversity. In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO), pp. 967–974. ACM, New York, NY, USA (2015)
22. Runarsson, T.P., Yao, X.: Search biases in constrained evolutionary optimization. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) **35**(2), 233–243 (2005)
23. Samuelsen, E., Glette, K.: Multi-objective analysis of map-elites performance. CoRR **abs/1803.05174**, 1–8 (2018)
24. Shahrzad, H., Fink, D., Miikkulainen, R.: Enhanced optimization with composite objectives and novelty selection. CoRR **abs/1803.03744**, 1–7 (2018)
25. Singh, H.K., Asafuddoula, M., Ray, T.: Solving problems with a mix of hard and soft constraints using modified infeasibility driven evolutionary algorithm (IDEA-M). In: Congress on Evolutionary Computation (CEC), pp. 983–990. IEEE, New York (2014)
26. Surry, P.D., Radcliffe, N.J.: The comoga method: constrained optimisation by multi-objective genetic algorithms. Control and Cybernetics **26**, 391–412 (1997)
27. Takahama, T., Sakai, S.: Constrained optimization by ε constrained particle swarm optimizer with ε-level control. In: Soft computing as transdisciplinary science and technology, pp. 1019–1029. Springer, Berlin, Heidelberg (2005)
28. Takahama, T., Sakai, S.: Constrained optimization by the ε constrained differential evolution with an archive and gradient-based mutation. In: Congress on Evolutionary Computation (CEC), pp. 1–9. IEEE, New York (2010)
29. Urquhart, N., Hart, E.: Optimisation and illumination of a real-world workforce scheduling and routing application via map-elites. CoRR **abs/1805.11555**, 1–13 (2018)
30. Vassiliades, V., Chatzilygeroudis, K., Mouret, J.B.: A comparison of illumination algorithms in unbounded spaces. In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO) Companion, pp. 1578–1581. ACM, New York, NY, USA (2017)
31. Vassiliades, V., Chatzilygeroudis, K., Mouret, J.B.: Using centroidal voronoi tessellations to scale up the multidimensional archive of phenotypic elites algorithm. IEEE Transactions on Evolutionary Computation **22**(4), 623–630 (2018)
32. Vassiliades, V., Mouret, J.B.: Discovering the elite hypervolume by leveraging interspecies correlation. In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO), pp. 149–156. ACM, New York, NY, USA (2018)
33. Venkatraman, S., Yen, G.G.: A generic framework for constrained optimization using genetic algorithms. IEEE Transactions on Evolutionary Computation **9**(4), 424–435 (2005)
34. Wang, Y., Yin, D.Q., Yang, S., Sun, G.: Global and local surrogate-assisted differential evolution for expensive constrained optimization problems with inequality constraints. IEEE Transactions on Cybernetics **1**(1), 1–15 (2018)
35. Zhou, Y., Li, Y., He, J., Kang, L.: Multi-objective and mgg evolutionary algorithm for constrained optimization. In: Congress on Evolutionary Computation (CEC), pp. 1–5. IEEE, New York (2003)

Table 3: Numerical results (rank based on CEC rules) obtained from 25 independent of MAP-Elites (Configuration 1) on 10D problems

Function	Best	Worst	Median	c	$\bar{v}$	Mean	Std	FR
C01	-5.900e-01(0)	-3.500e-01(0)	-4.400e-01(0)	(0, 0, 0)	0	-4.600e-01	7.000e-02	1.0
C02	8.900e-01(0)	2.860e+00(2)	1.290e+00(0)	(0, 0, 0)	0	1.780e+00	1.870e+00	0.52
C03	1.254e+14(1)	9.654e+14(1)	5.118e+14(1)	(0, 0, 1)	2.477e+06	5.208e+14	2.730e+14	0.0
C04	-1.512e+01(4)	3.930e+01(4)	1.646e+01(3)	(0, 0, 3)	7.993e+03	1.487e+01	1.249e+01	0.0
C05	-1.795e+02(2)	3.301e+02(2)	8.052e+01(2)	(1, 0, 1)	-8.064e+01	6.719e+01	1.716e+02	0.0
C06	-1.222e+02(2)	5.043e+02(1)	1.705e+02(1)	(0, 0, 1)	-1.300e+02	1.885e+02	1.669e+02	0.0
C07	1.761e+09(0)	3.347e+11(0)	7.117e+09(0)	(0, 0, 0)	0	6.854e+10	9.030e+10	1.0
C08	2.610e+04(0)	1.504e+08(0)	3.490e+06(0)	(0, 0, 0)	0	1.690e+07	3.667e+07	1.0
C09	2.752e+13(0)	2.327e+15(1)	4.526e+13(1)	(0, 1, 0)	-2.950e+01	3.607e+14	6.492e+14	0.12
C10	7.187e+12(1)	8.964e+14(1)	3.631e+13(1)	(0, 0, 1)	-6.560e+02	1.805e+14	3.008e+14	0.0
C11	-5.486e+01(1)	-3.766e+01(1)	-4.441e+01(1)	(0, 0, 1)	2.183e+12	-4.561e+01	4.870e+00	0.0
C12	-5.327e+03(2)	1.704e+03(2)	-3.904e+03(2)	(0, 0, 2)	1.501e+12	-3.008e+03	2.173e+03	0.0
C13	-3.702e+02(2)	-1.721e+02(2)	-2.804e+02(2)	(0, 0, 1)	5.853e+02	-2.823e+02	5.116e+01	0.0
C14	2.141e+14(0)	4.448e+15(0)	1.304e+15(0)	(0, 0, 0)	0	1.555e+15	1.018e+15	1.0
C15	9.018e+12(0)	1.041e+15(0)	4.145e+14(0)	(0, 0, 0)	0	4.449e+14	2.936e+14	1.0
C16	4.900e-01(0)	1.100e+00(2)	6.500e-01(4)	(0, 1, 3)	-4.585e+06	6.200e-01	3.000e-01	0.16
C17	1.310e+00(0)	4.958e+02(1)	1.784e+01(0)	(0, 1, 0)	-8.500e+05	6.456e+01	1.025e+02	0.4
C18	3.365e+01(0)	3.542e+03(1)	5.894e+02(0)	(0, 0, 0)	0	1.478e+03	2.594e+03	0.88

Table 4: Numerical results (rank based on CEC rules) obtained from 25 independent of MAP-Elites (Configuration 2) on 10D problems

Function	Best	Worst	Median	c	$\bar{v}$	Mean	Std	FR
C01	-6.700e-01(0)	-4.300e-01(0)	-4.800e-01(0)	(0, 0, 0)	0	-5.100e-01	6.000e-02	1.0
C02	1.700e+00(0)	4.980e+00(1)	2.500e+00(2)	(1, 1, 0)	-1.046e+01	2.230e+00	1.930e+00	0.36
C03	4.516e+13(1)	7.934e+14(1)	4.101e+14(1)	(0, 0, 1)	2.642e+06	3.827e+14	1.928e+14	0.0
C04	-1.654e+01(4)	3.941e+01(4)	1.958e+01(4)	(1, 0, 3)	3.074e+03	1.224e+01	1.789e+01	0.0
C05	-3.516e+02(2)	5.255e+02(2)	-9.369e+01(2)	(1, 0, 1)	-6.720e+01	-3.351e+01	2.121e+02	0.0
C06	-2.519e+02(2)	5.891e+02(2)	-1.068e+02(2)	(0, 1, 1)	1.434e+01	-2.237e+01	2.178e+02	0.0
C07	4.191e+08(0)	2.371e+11(0)	4.222e+09(0)	(0, 0, 0)	0	2.836e+10	5.798e+10	1.0
C08	8.302e+05(0)	2.420e+08(0)	1.664e+07(0)	(0, 0, 0)	0	4.818e+07	6.160e+07	1.0
C09	8.585e+11(1)	7.835e+13(1)	1.933e+13(1)	(1, 0, 0)	-1.388e+02	2.804e+13	2.302e+13	0.0
C10	5.666e+12(0)	3.540e+14(1)	1.854e+13(1)	(1, 0, 0)	-2.489e+02	5.046e+13	7.657e+13	0.16
C11	-5.366e+01(1)	-3.885e+01(1)	-4.573e+01(1)	(0, 0, 1)	1.894e+12	-4.589e+01	3.750e+00	0.0
C12	-6.849e+03(2)	5.433e+02(2)	-4.272e+03(2)	(0, 0, 2)	7.650e+11	-3.843e+03	1.779e+03	0.0
C13	-3.849e+02(2)	-4.087e+01(2)	-3.048e+02(2)	(1, 0, 1)	4.854e+02	-2.916e+02	7.552e+01	0.0
C14	1.004e+07(0)	9.089e+13(0)	8.365e+08(0)	(0, 0, 0)	0	3.681e+12	1.780e+13	1.0
C15	3.640e+06(0)	1.594e+14(0)	4.011e+10(0)	(0, 0, 0)	0	7.150e+12	3.112e+13	1.0
C16	1.060e+00(0)	1.080e+00(4)	9.900e-01(4)	(1, 1, 2)	-5.208e+07	8.700e-01	2.300e-01	0.08
C17	9.251e+01(0)	6.959e+02(3)	3.041e+01(3)	(0, 2, 1)	-3.098e+08	2.108e+02	2.661e+02	0.36
C18	2.253e+02(0)	2.242e+04(1)	8.008e+03(0)	(0, 0, 0)	0	9.665e+03	8.152e+03	0.8

Table 5: Numerical results (rank based on CEC rules) obtained from 25 independent of MAP-Elites (Configuration 3) on 10D problems

Function	Best	Worst	Median	c	$\bar{v}$	Mean	Std	FR
C01	-8.300e-01(0)	-4.500e-01(0)	-5.500e-01(0)	(0, 0, 0)	0	-5.700e-01	1.000e-01	1.0
C02	8.000e-01(0)	3.050e+00(2)	2.870e+00(2)	(1, 1, 0)	-8.750e+00	1.850e+00	2.230e+00	0.44
C03	1.154e+14(1)	9.250e+14(1)	4.362e+14(1)	(0, 0, 1)	1.310e+06	5.005e+14	2.213e+14	0.0
C04	-4.441e+01(4)	5.003e+01(4)	3.800e-01(4)	(0, 2, 2)	9.570e+01	1.270e+00	2.920e+01	0.0
C05	-4.801e+02(2)	1.527e+02(1)	-2.316e+02(2)	(0, 1, 1)	1.188e+01	-2.181e+02	1.804e+02	0.0
C06	-4.275e+02(2)	4.210e+02(2)	-1.927e+02(1)	(0, 0, 1)	-1.397e+02	-1.363e+02	2.086e+02	0.0
C07	6.152e+03(0)	5.908e+07(0)	3.171e+05(0)	(0, 0, 0)	0	4.070e+06	1.251e+07	1.0
C08	4.550e+03(0)	1.391e+06(0)	1.518e+05(0)	(0, 0, 0)	0	4.183e+05	4.949e+05	1.0
C09	1.690e+07(0)	1.600e+15(1)	8.227e+07(1)	(0, 1, 0)	-6.704e+01	1.612e+14	4.344e+14	0.2
C10	9.235e+06(0)	7.494e+14(1)	7.169e+07(1)	(1, 0, 0)	-3.527e+01	3.833e+13	1.490e+14	0.44
C11	-5.242e+01(1)	-3.835e+01(1)	-4.532e+01(1)	(0, 0, 1)	3.883e+12	-4.474e+01	3.120e+00	0.0
C12	-7.742e+03(2)	3.236e+02(2)	-5.923e+03(2)	(1, 0, 1)	1.573e+12	-5.795e+03	1.603e+03	0.0
C13	-4.474e+02(2)	-2.925e+02(2)	-3.952e+02(2)	(0, 1, 1)	8.816e+02	-3.900e+02	3.742e+01	0.0
C14	1.033e+05(0)	3.280e+14(0)	1.724e+09(0)	(0, 0, 0)	0	3.958e+13	8.132e+13	1.0
C15	4.980e+03(0)	4.089e+11(0)	9.060e+07(0)	(0, 0, 0)	0	1.863e+10	8.010e+10	1.0
C16	1.020e+00(0)	1.100e+00(4)	2.900e-01(3)	(2, 1, 0)	-3.723e+05	5.100e-01	3.800e-01	0.12
C17	3.610e+00(0)	1.560e+01(3)	1.123e+01(3)	(2, 1, 0)	-8.454e+05	6.054e+01	1.196e+02	0.52
C18	7.190e+00(0)	1.480e+01(2)	5.091e+01(0)	(0, 0, 0)	0	1.078e+02	1.165e+02	0.92

Table 6: Numerical results (rank based on CEC rules) obtained from 25 independent of MAP-Elites (Configuration 1) on 30D problems

Function	Best	Worst	Median	c	$\bar{v}$	Mean	Std	FR
C01	-2.800e-01(0)	-2.000e-01(0)	-2.200e-01(0)	(0, 0, 0)	0	-2.300e-01	2.000e-02	1.0
C02	3.880e+00(0)	4.320e+00(2)	3.760e+00(1)	(1, 0, 0)	-2.570e+00	3.210e+00	1.480e+00	0.36
C03	6.664e+15(1)	1.887e+16(1)	1.472e+16(1)	(0, 0, 1)	1.798e+07	1.463e+16	3.100e+15	0.0
C04	1.869e+01(4)	4.507e+01(4)	3.239e+01(4)	(1, 1, 2)	1.527e+06	3.235e+01	5.850e+00	0.0
C05	1.739e+02(2)	4.920e+02(1)	2.784e+02(1)	(0, 1, 0)	-1.224e+01	3.036e+02	8.686e+01	0.0
C06	2.237e+02(2)	5.782e+02(1)	3.076e+02(2)	(1, 1, 1)	5.920e+00	3.545e+02	1.043e+02	0.0
C07	3.226e+10(0)	6.284e+12(0)	9.281e+11(0)	(0, 0, 0)	0	1.328e+12	1.386e+12	1.0
C08	6.346e+07(0)	6.746e+10(0)	1.811e+09(0)	(0, 0, 0)	0	7.092e+09	1.399e+10	1.0
C09	7.159e+14(0)	3.818e+15(1)	8.511e+14(1)	(0, 0, 1)	-1.028e+03	1.323e+15	9.556e+14	0.2
C10	6.189e+14(0)	3.359e+15(1)	7.702e+14(1)	(1, 0, 0)	-1.984e+02	9.456e+14	7.034e+14	0.16
C11	-3.289e+01(1)	-2.454e+01(1)	-2.802e+01(1)	(0, 0, 1)	5.839e+12	-2.833e+01	2.180e+00	0.0
C12	-1.363e+04(2)	2.778e+03(2)	-8.056e+03(2)	(0, 1, 1)	3.562e+12	-7.862e+03	3.439e+03	0.0
C13	-3.129e+02(2)	-1.248e+01(2)	-2.373e+02(2)	(0, 1, 1)	5.074e+02	-2.253e+02	5.712e+01	0.0
C14	6.971e+15(0)	2.152e+16(0)	1.160e+16(0)	(0, 0, 0)	0	1.203e+16	3.312e+15	1.0
C15	1.754e+15(0)	2.898e+16(0)	6.434e+15(0)	(0, 0, 0)	0	7.380e+15	5.067e+15	1.0
C16	9.100e-01(0)	1.050e+00(3)	8.500e-01(4)	(1, 2, 1)	-1.920e+19	8.200e-01	1.700e-01	0.08
C17	6.822e+01(0)	1.316e+03(2)	8.772e+02(0)	(0, 0, 0)	0	9.761e+02	8.256e+02	0.64
C18	1.228e+03(0)	1.276e+04(0)	3.849e+03(0)	(0, 0, 0)	0	4.380e+03	2.752e+03	1.0

Table 7: Numerical results (rank based on CEC rules) obtained from 25 independent of MAP-Elites (Configuration 2) on 30D problems

Function	Best	Worst	Median	c	$\bar{v}$	Mean	Std	FR
C01	-2.600e-01(0)	-2.100e-01(0)	-2.300e-01(0)	(0, 0, 0)	0	-2.400e-01	1.000e-02	1.0
C02	3.510e+00(0)	4.940e+00(2)	3.760e+00(2)	(0, 2, 0)	-5.310e+00	3.440e+00	1.030e+00	0.24
C03	6.155e+15(1)	1.970e+16(1)	1.396e+16(1)	(0, 0, 1)	1.316e+07	1.378e+16	3.582e+15	0.0
C04	1.892e+01(4)	4.564e+01(4)	3.525e+01(4)	(0, 1, 3)	1.317e+06	3.429e+01	6.280e+00	0.0
C05	-2.140e+00(2)	5.772e+02(1)	2.484e+02(2)	(0, 1, 1)	-2.131e+01	2.570e+02	1.174e+02	0.0
C06	1.318e+02(2)	5.961e+02(1)	2.776e+02(1)	(0, 0, 1)	-1.296e+02	3.380e+02	1.436e+02	0.0
C07	6.860e+10(0)	6.957e+12(0)	2.374e+12(0)	(0, 0, 0)	0	2.560e+12	1.740e+12	1.0
C08	4.992e+08(0)	1.523e+12(0)	6.357e+10(0)	(0, 0, 0)	0	1.370e+11	2.923e+11	1.0
C09	1.715e+15(0)	1.520e+15(1)	7.103e+14(1)	(1, 0, 0)	4.026e+02	7.634e+14	4.300e+14	0.08
C10	1.375e+14(1)	1.511e+15(1)	7.229e+14(1)	(0, 0, 1)	-5.762e+02	7.349e+14	3.095e+14	0.0
C11	-3.339e+01(1)	-2.650e+01(1)	-2.925e+01(1)	(0, 0, 1)	5.776e+12	-2.991e+01	2.060e+00	0.0
C12	-1.146e+04(2)	-4.734e+01(2)	-7.854e+03(2)	(0, 1, 3)	3.687e+12	-6.797e+03	3.248e+03	0.0
C13	-3.302e+02(2)	2.444e+01(2)	-2.143e+02(2)	(1, 0, 1)	5.036e+02	-1.960e+02	8.269e+01	0.0
C14	2.752e+10(0)	4.214e+13(0)	3.568e+12(0)	(0, 0, 0)	0	7.098e+12	1.001e+13	1.0
C15	1.921e+09(0)	2.006e+15(0)	1.110e+13(0)	(0, 0, 0)	0	1.083e+14	3.901e+14	1.0
C16	1.030e+00(4)	1.300e+00(2)	1.110e+00(4)	(0, 2, 2)	-6.197e+17	1.130e+00	8.000e-02	0.0
C17	1.110e+03(0)	2.864e+03(3)	1.417e+03(0)	(0, 0, 0)	0	1.539e+03	7.542e+02	0.44
C18	5.037e+03(0)	4.064e+04(0)	3.043e+04(0)	(0, 0, 0)	0	2.847e+04	9.539e+03	1.0

Table 8: Numerical results (rank based on CEC rules) obtained from 25 independent of MAP-Elites (Configuration 3) on 30D problems

Function	Best	Worst	Median	c	$\bar{v}$	Mean	Std	FR
C01	-3.100e-01(0)	-2.300e-01(0)	-2.600e-01(0)	(0, 0, 0)	0	-2.600e-01	2.000e-02	1.0
C02	4.590e+00(0)	5.160e+00(2)	3.150e+00(1)	(1, 0, 0)	-1.940e+00	2.610e+00	2.260e+00	0.32
C03	1.033e+16(1)	2.039e+16(1)	1.521e+16(1)	(0, 0, 1)	9.516e+06	1.534e+16	2.809e+15	0.0
C04	-1.859e+01(4)	4.163e+01(4)	1.518e+01(4)	(0, 2, 2)	3.534e+05	1.215e+01	1.744e+01	0.0
C05	-2.304e+02(1)	5.599e+02(2)	-7.310e+01(1)	(0, 0, 1)	-6.470e+00	9.960e+00	1.769e+02	0.0
C06	-1.678e+02(1)	5.736e+02(1)	-1.477e+01(1)	(0, 0, 1)	-8.221e+01	7.093e+01	2.096e+02	0.0
C07	7.226e+05(0)	2.288e+09(0)	2.058e+07(0)	(0, 0, 0)	0	1.503e+08	4.498e+08	1.0
C08	6.173e+05(0)	3.216e+07(0)	2.674e+06(0)	(0, 0, 0)	0	6.223e+06	7.284e+06	1.0
C09	5.770e+09(0)	6.548e+10(1)	1.463e+10(0)	(0, 0, 0)	0	2.184e+13	1.050e+14	0.48
C10	1.301e+08(0)	1.153e+11(1)	4.820e+09(1)	(1, 0, 0)	-2.706e+01	4.189e+13	1.970e+14	0.48
C11	-3.475e+01(1)	-2.439e+01(1)	-2.780e+01(1)	(0, 0, 1)	5.744e+12	-2.825e+01	2.340e+00	0.0
C12	-1.818e+04(2)	2.716e+03(2)	-1.583e+04(2)	(0, 0, 2)	3.737e+12	-1.369e+04	5.767e+03	0.0
C13	-3.659e+02(2)	-2.891e+02(2)	-3.336e+02(2)	(1, 0, 1)	6.148e+02	-3.291e+02	2.011e+01	0.0
C14	2.949e+06(0)	9.116e+15(0)	3.783e+08(0)	(0, 0, 0)	0	6.363e+14	2.134e+15	1.0
C15	9.592e+05(0)	4.604e+09(0)	1.375e+07(0)	(0, 0, 0)	0	2.319e+08	8.981e+08	1.0
C16	9.600e-01(0)	1.040e+00(4)	8.100e-01(4)	(1, 0, 3)	-1.020e+20	8.300e-01	1.400e-01	0.04
C17	6.458e+01(0)	1.234e+02(3)	8.320e+01(3)	(1, 1, 1)	-1.291e+20	5.433e+02	7.307e+02	0.48
C18	1.570e+02(0)	2.793e+04(0)	4.208e+02(0)	(0, 0, 0)	0	1.896e+03	5.424e+03	1.0

Table 9: Numerical results (rank based on CEC rules) obtained from 25 independent of εDEag on 10D problems (taken from [28])

Function	Best	Worst	Median	c	$\bar{v}$	Mean	Std	FR
C01	-7.473104e-01(0)	-7.473104e-01(0)	-7.405572e-01(0)	0, 0, 0	0.00e+00	-7.47e-01	1.32e-03	1.0
C02	-2.277702e+00(0)	-2.269502e+00(0)	-2.174499e+00(0)	0, 0, 0	0.00e+00	-2.26e+00	2.39e-02	1.0
C03	0.000000e+00(0)	0.000000e+00(0)	0.000000e+00(0)	0, 0, 0	0.00e+00	0.00e+00	0.00e+00	1.0
C04	-9.992345e-06(0)	-9.977276e-06(0)	-9.282295e-06(0)	0, 0, 0	0.00e+00	-9.92e-06	1.55e-07	1.0
C05	-4.836106e+02(0)	-4.836106e+02(0)	-4.836106e+02(0)	0, 0, 0	0.00e+00	-4.84e+02	3.89e-13	1.0
C06	-5.786581e+02(0)	-5.786533e+02(0)	-5.786448e+02(0)	0, 0, 0	0.00e+00	-5.79e+02	3.63e-03	1.0
C07	0.000000e+00(0)	0.000000e+00(0)	0.000000e+00(0)	0, 0, 0	0.00e+00	0.00e+00	0.00e+00	1.0
C08	0.000000e+00(0)	1.094154e+01(0)	1.537535e+01(0)	0, 0, 0	0.00e+00	6.73e+00	5.56e+00	1.0
C09	0.000000e+00(0)	0.000000e+00(0)	0.000000e+00(0)	0, 0, 0	0.00e+00	0.00e+00	0.00e+00	1.0
C10	0.000000e+00(0)	0.000000e+00(0)	0.000000e+00(0)	0, 0, 0	0.00e+00	0.00e+00	0.00e+00	1.0
C11	-1.522713e-03(0)	-1.522713e-03(0)	-1.522713e-03(0)	0, 0, 0	0.00e+00	-1.52e-03	6.34e-11	1.0
C12	-5.700899e+02(0)	-4.231332e+02(0)	-1.989129e-01(0)	0, 0, 0	0.00e+00	-3.37e+02	1.78e+02	1.0
C13	-6.842937e+01(0)	-6.842936e+01(0)	-6.842936e+01(0)	0, 0, 0	0.00e+00	-6.84e+01	1.03e-06	1.0
C14	0.000000e+00(0)	0.000000e+00(0)	0.000000e+00(0)	0, 0, 0	0.00e+00	0.00e+00	0.00e+00	1.0
C15	0.000000e+00(0)	0.000000e+00(0)	4.497445e+00(0)	0, 0, 0	0.00e+00	1.80e-01	8.81e-01	1.0
C16	0.000000e+00(0)	2.819841e-01(0)	1.018265e+00(0)	0, 0, 0	0.00e+00	3.70e-01	3.71e-01	1.0
C17	1.463180e-17(0)	5.653326e-03(0)	7.301765e-01(0)	0, 0, 0	0.00e+00	1.25e-01	1.94e-01	1.0
C18	3.731439e-20(0)	4.097909e-19(0)	9.227027e-18(0)	0, 0, 0	0.00e+00	9.68e-19	1.81e-18	1.0

Table 10: Numerical results (rank based on CEC rules) obtained from 25 independent of εDEag on 30D problems (taken from [28])

Function	Best	Worst	Median	c	$\bar{v}$	Mean	Std	FR
C01	-8.218255e-01(0)	-8.206172e-01(0)	-8.195466e-01(0)	0, 0, 0	0.00e+00	-8.21e-01	7.10e-04	1.0
C02	-2.169248e+00(0)	-2.152145e+00(0)	-2.117096e+00(0)	0, 0, 0	0.00e+00	-2.15e+00	1.20e-02	1.0
C03	2.867347e+01(0)	2.867347e+01(0)	3.278014e+01(0)	0, 0, 0	0.00e+00	2.88e+01	8.05e-01	1.0
C04	4.698111e-03(0)	6.947614e-03(0)	1.777889e-02(0)	0, 0, 0	0.00e+00	8.16e-03	3.07e-03	1.0
C05	-4.531307e+02(0)	-4.500404e+02(0)	-4.421590e+02(0)	0, 0, 0	0.00e+00	-4.50e+02	2.90e+00	1.0
C06	-5.285750e+02(0)	-5.280407e+02(0)	-5.264539e+02(0)	0, 0, 0	0.00e+00	-5.28e+02	4.75e-01	1.0
C07	1.147112e-15(0)	2.114429e-15(0)	5.481915e-15(0)	0, 0, 0	0.00e+00	2.60e-15	1.23e-15	1.0
C08	2.518693e-14(0)	6.511508e-14(0)	2.578112e-13(0)	0, 0, 0	0.00e+00	7.83e-14	4.86e-14	1.0
C09	2.770665e-16(0)	1.124608e-08(0)	1.052759e+02(0)	0, 0, 0	0.00e+00	1.07e+01	2.82e+01	1.0
C10	3.252002e+01(0)	3.328903e+01(0)	3.463243e+01(0)	0, 0, 0	0.00e+00	3.33e+01	4.55e-01	1.0
C11	-3.268462e-04(0)	-2.843296e-04(0)	-2.236338e-04(0)	0, 0, 0	0.00e+00	-2.86e-04	2.71e-05	1.0
C12	-1.991453e-01(0)	5.337125e+02(1)	5.461723e+02(1)	0, 1, 0	3.24e-01	3.56e+02	2.89e+02	0.12
C13	-6.642473e+01(0)	-6.531507e+01(0)	-6.429690e+01(0)	0, 0, 0	0.00e+00	-6.54e+01	5.73e-01	1.0
C14	5.015863e-14(0)	1.359306e-13(0)	2.923513e-12(0)	0, 0, 0	0.00e+00	3.09e-13	5.61e-13	1.0
C15	2.160345e+01(0)	2.160375e+01(0)	2.160403e+01(0)	0, 0, 0	0.00e+00	2.16e+01	1.10e-04	1.0
C16	0.000000e+00(0)	0.000000e+00(0)	5.421011e-20(0)	0, 0, 0	0.00e+00	2.17e-21	1.06e-20	1.0
C17	2.165719e-01(0)	5.315949e+00(0)	1.889064e+01(0)	0, 0, 0	0.00e+00	6.33e+00	4.99e+00	1.0
C18	1.226054e+00(0)	2.679497e+01(0)	7.375363e+02(0)	0, 0, 0	0.00e+00	8.75e+01	1.66e+02	1.0