

Supplemental Materials for

On the use of feature-maps for improved quality-diversity meta-evolution

DAVID M. BOSSENS, University of Southampton

DANESH TARAPORE, University of Southampton

S1 k -BEST DATABASE

In Meta-evolution with CMA-ES, the database \mathcal{D} stores a large number of previously found solutions to enable rapidly generating new behaviour-performance maps. Each such solution is a tuple $\langle \mathbf{g}, \mathbf{b}, f \rangle$, where \mathbf{g} is the bottom-level genotype (e.g., the parameters of a controller), \mathbf{b} is an extended behavioural description of the solution according to a large number of N_b user-defined behavioural base-features, and f is the fitness (e.g., the performance of a controller).

In previous work [1], a circular buffer was used as a datastructure for the database. This meant that once the database is full, previously found solutions will be removed regardless of their contribution to fitness or behavioural diversity. To solve this problem, we use what we call a k -best database, which gives each bin in the base-behavioural space a capacity to store the k best solutions. Since there is no need to remove solutions when the number of solutions is far from total database capacity and initially few behavioural bins are filled, behavioural bin capacity k is initialised large. When more behavioural bins are being filled and the total database is reaching capacity, this number will shrink progressively.

S1.1 Base-behavioural bins

To retain behavioural diversity, the database \mathcal{D} divides the base-behavioural space $[0, 1]^{N_b}$ into coarse-grained bins of equal width δ , in which it stores up to a number of k solutions. The hypercube partitioning is geometrically similar to the behaviour-performance maps in MAP-Elites except that (a) only a small number, such as 2 or 3, bins per dimension (corresponding to a width of $\delta = 1/2$ or $\delta = 1/3$) are allowed to limit the maximal capacity of the database to 2^{N_b} or 3^{N_b} given the many base-features; and (b) for each hypercube, an array containing up to at most k solutions is stored within a single bin.

S1.2 Adding solutions

When a new solution $\langle \mathbf{g}, \mathbf{b}, f \rangle$ is presented to the database, its corresponding coarse-grained bin is looked up, yielding the array of solutions $\mathcal{C} = \mathcal{D}[\mathbf{b}]$. Then an additional check for fitness and diversity is performed: if there is another solution in \mathcal{C} that is in the same base-behavioural hypercube of width δ/k , then the solution is only added if it has higher fitness; if there is no such similar solution, then the solution is always added.¹ If \mathcal{C} now has $k + 1$ solutions, then its lowest-fitness solution is removed. If the database's capacity is exceeded, the number of allowed solutions per bins is decremented, $k \leftarrow k - 1$, and for each coarse-grained bin the lowest-fitness solution is removed. While k is initially large, e.g., $k = 1000$, k shrinks progressively as the run continues.

¹Since there are k^{N_b} possible hypercubes of width δ/k within each coarse-grained bin of width δ and up to k solutions are allowed per coarse-grained bin, the check for diversity and fitness is not too restrictive.

S2 EXPERIMENTAL PARAMETERS

For convenience, Table S1 includes all the parameter settings for the experimental setup.

Table S1. Parameter settings for evolution. Top half shows settings common to all conditions while bottom half shows settings for Meta-MAP-Elites conditions.

Parameter	Setting
Genotype (g)	discretised in $[0, 1]^8$
Mutation rate	0.125
Mutation type	random increment/decrement with step of 0.025
Maximal map coverage	4,096 solutions
Function evaluations	100,000,000
Batch size per generation	400 bottom-level individuals
Initial population (p)	2,000 bottom-level individuals
Meta-population size (λ)	5
Meta-genotype (w)	$[-1, 1]^{182}$ for non-linear feature-map $[0, 1]^{56}$ otherwise
Number of base-features (N_b)	14
Number of target-features (D)	4
Normalisation range ($[m, M]$)	$[0.20, 0.80]$ (linear feature-maps)
Number of hidden units (N_h)	10 (non-linear feature-maps)
Sigmoid scaling factor (α_s)	30 (non-linear feature-maps)
Database settings	initial $k = 5000$; bin width $\delta = 1/3$; capacity 3^{14} (just below 5 million)

S3 SOURCE CODE

Source code for the experiments is publicly available at <https://github.com/resilient-swarms/meta-cmaes>.

REFERENCES

- [1] David M Bossens, Jean-Baptiste Mouret, and Danesh Tarapore. 2020. Learning behaviour-performance maps with meta-evolution. In *GECCO'20 - Genetic and Evolutionary Computation Conference*. Cancun, Mexico, 49–57.