A comparison of illumination algorithms in unbounded spaces

Vassilis Vassiliades, Konstantinos Chatzilygeroudis, Jean-Baptiste Mouret

A comparison of illumination algorithms in unbounded spaces

Vassilis Vassiliades
Inria Nancy - Grand-Est
CNRS UMR 7503
Université de Lorraine, UMR 7503
vassilis.vassiliades@inria.fr

Konstantinos Chatzilygeroudis
Inria Nancy - Grand-Est
CNRS UMR 7503
Université de Lorraine, UMR 7503
konstantinos.chatzilygeroudis@inria.fr

Jean-Baptiste Mouret
Inria Nancy - Grand-Est
CNRS UMR 7503
Université de Lorraine, UMR 7503
jean-baptiste.mouret@inria.fr

ABSTRACT

Illumination algorithms are a new class of evolutionary algorithms capable of producing large archives of diverse and high-performing solutions. Examples of such algorithms include Novelty Search with Local Competition (NSLC), the Multi-dimensional Archive of Phenotypic Elites (MAP-Elites) and the newly introduced Centroidal Voronoi Tessellation (CVT) MAP-Elites. While NSLC can be used in unbounded behavioral spaces, MAP-Elites and CVT-MAP-Elites require the user to manually specify the bounds. In this study, we introduce variants of these algorithms that expand their bounds based on the discovered solutions. In addition, we introduce a novel algorithm called "Cluster-Elites" that can adapt its bounds to non-convex spaces. We compare all algorithms in a maze navigation problem and illustrate that Cluster-Elites and the expansive variants of MAP-Elites and CVT-MAP-Elites have comparable or better performance than NSLC, MAP-Elites and CVT-MAP-Elites.

CCS CONCEPTS

•Computing methodologies → Evolutionary robotics; •Theory of computation → Evolutionary algorithms;

KEYWORDS

illumination algorithms, MAP-Elites, quality diversity, behavioral diversity, novelty search

ACM Reference format:
Vassilis Vassiliades, Konstantinos Chatzilygeroudis, and Jean-Baptiste Mouret. 2017. A comparison of illumination algorithms in unbounded spaces. In *Proceedings of The Genetic and Evolutionary Computation Conference, Berlin, Germany, July 2017 (GECCO'17),* 4 pages.
DOI: 10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Evolutionary illumination algorithms are a new class of algorithms capable of producing large archives of diverse and high-performing solutions [9, 10]. Inspired by the phenomenon of species diversification in nature (e.g., see [7]), these algorithms have been introduced in the field of *evolutionary robotics* with the purpose of encouraging diversity in what is known as the *behavior space* [4, 8, 11]. This space describes the possible behaviors of individuals over their lifetimes: for example, a point in this space, i.e., a behavior characterization/signature, could be the final positions of simulated robots whose controllers are evolved [4]. In contrast, the genotype space is the space in which the evolutionary algorithm operates (e.g., a space of bit strings) and the phenotype space encodes the possible controllers (e.g., neural networks) that are derived from the genotype space.

Novelty Search (NS) [4] is the first algorithm that suggested to abandon any fitness objective and continually explore for novel behaviors by defining novelty as sparseness, i.e., the average distance to the n nearest neighbors, in behavior space. NS with Local Competition (NSLC) [6] improved upon NS based on the observation that it is more beneficial to explore globally and optimize locally: this local optimization is achieved using a secondary objective.

The Multi-dimensional Archive of Phenotypic Elites (MAP-Elites) [1, 9] algorithm proposed a conceptually simpler approach: it discretizes the behavior space into a grid of k cells, storing in each cell the *elite* solution over the evolutionary generations. This algorithm has recently been extended using a Centroidal Voronoi Tessellation (CVT) by the CVT-MAP-Elites algorithm to deal with high-dimensional behavior spaces [12]. Intuitively, CVT-MAP-Elites partitions the behavior space by uniformly distributing k centroids: these centroids correspond to the centers of the cells in MAP-Elites if both algorithms use the same number of cells.

Both MAP-Elites and CVT-MAP-Elites assume knowledge of the bounds of the behavior space. More specifically, they enclose the behavior space inside a bounding hyperrectangle and make the assumption that the user knows the ranges of this rectangle. In contrast, NSLC does not make such an assumption. As the original spirit of illumination algorithms is constant exploration and diversification, such user-defined knowledge is a limitation of the MAP-Elites family of algorithms. Thus, in this study we ask two questions:

(1) Can expansive versions of MAP-Elites and CVT-Map-Elites be as effective as their non-expansive counterparts, in spite of the fact that they do not know the bounds?
(2) Would an algorithm that allocates centroids using the actual shape of the behavior space, instead of the bounding rectangle like expansive MAP-Elites, be more effective?

2 NEW ALGORITHMS

2.1 Expansive MAP-Elites

In the "expansive" variant of MAP-Elites (Appendix A, Alg.1), the behavior characterizations of the offspring at every generation define the bounds of the space. As the newly calculated bounds change the width of the cells, solutions that already exist in the

GECCO'17, Berlin, Germany

DOI: 10.1145/nnnnnnn.nnnnnnn

archive are taken out and treated as new solutions along with the offspring. A side-effect of this procedure could be an initial increase in the archive size (due to filling a restricted space), with some subsequent decrease, due to the expansion of the bounds and the mapping of more than one solutions to a single cell. In contrast, in the case of standard MAP-Elites, the archive can only grow.

2.2 Expansive CVT-MAP-Elites

For the "expansive" version of CVT-MAP-Elites (Appendix A, Alg.2), we perform a similar procedure to the above. We (re)compute the CVT based on the newly-found bounds, taking out the existing solutions and treating them as new ones (because the centroids will fill a larger bounding hyperrectangle). It is worth noting that we perform this procedure periodically and not at every generation (as in expansive MAP-Elites), in order to reduce the computational load induced by the repeated CVT construction.

2.3 Cluster-Elites

In this paper, we introduce a variant of MAP-Elites and CVT-MAP-Elites called "Cluster-Elites" (Appendix A, Alg.3) that attempts to maximally spread a number of centroids on a potentially non-convex manifold on which the behavioral descriptors reside, rather than inside a hyperrectangle defined by the ranges of the sampled behavioral descriptors. Since the shape of this manifold is unknown and potentially high-dimensional, we cannot use approaches like alpha-shapes [2] (which are generalizations of convex-hulls), but we instead resort to methods that use nearest-neighbor calculations. Clustering algorithms such as the "k-medoids" [3] aim to partition a dataset into k clusters by choosing the "centroid" of each cluster to be the point that minimizes the within-cluster sum of squares. Such algorithms, however, cannot be used in our case because of the problem of sampling bias: denser regions place more emphasis on allocating "resources" (centroids) there, whereas we would like to have a set of uniformly-spread points that is not highly affected by the density of sampled solutions. In addition, sparser regions could provide the stepping stones needed for discovering better solutions.

Cluster-Elites attempts to address these issues by continually sampling the behavior space and maximally spreading its available resources in the space spanned by the sampled solutions, while keeping in each centroid the locally fittest solution. More specifically, in each generation, Cluster-Elites first creates a set that contains a copy of the offspring and the current centroids. It then *iteratively* computes and removes the densest solution from the resulting set, until the size of this set reaches the desired number of well-spread centroids k, which is progressively increased over the generations. Finally, it stores at each centroid position the fittest solution among its local neighborhood by considering all initial solutions of the current generation (offspring and previous centroids).

3 EXPERIMENTAL SETUP

3.1 Task

We compare the original MAP-Elites and CVT-MAP-Elites, with their expansive variants, NSLC and Cluster-Elites. We use a maze navigation task (see Fig. 1 upper left) where a simulated mobile robot (radius: 10 units) is controlled by an artificial neural network, whose structure and weights are evolved [8]. The robot starts from the bottom of the arena (size: 1000×1000 sq. units) and needs to reach the goal at the center. This arena permits 16 families of trajectories towards the goal due to the openings (thus, at least 16 behaviorally distinct optima). The fitness function is the smallest Euclidean distance between the center of the robot and the goal over the robot's lifetime [8], which is set to 3000 simulation steps. The behavior characterization of each individual is the final (x,y) location of the robot [5, 8].

3.2 Evaluating the quality of the archives

We evaluate the quality of the archives produced by the algorithms by measuring the performance of their solutions in 16 modified versions of the environment used during evolution, each of which corresponds to a different family of trajectories (see Fig. 1). If an archive is made of diverse and high-performing individuals, then it should contain individuals with every type of trajectory, including some that work in the modified environments; in the extreme opposite, if all the individuals of an archive have the same behavior, none of them will work in the modified environments.

4 RESULTS

We use 30 independent evolutionary runs of 200k evaluations (990 generations). For MAP-Elites and our expansive variant we use 71 discretization intervals per dimension (thus, $71^2 = 5041$ cells), for CVT-MAP-Elites and the expansive version we use 5041 centroids, and we set the maximum archive size of NSLC to be 5041. For Cluster-Elites we use an initial number of centroids $k_{init} = 50$ and increase it by adding $k_{incr} = 5$ more centroids at every generation, resulting at 5k centroids at the final generation. For the calculation of the densest points in Cluster-Elites, we empirically set the number of nearest neighbors to $d + 2$, where d is the dimensionality of the behavior space (i.e., $d = 2$).

All algorithms return solutions with a median fitness of less than 10 units (radius of the robot) in all evaluation environments. In the 8^{th} and 14^{th} evaluation environments MAP-Elites, CVT-MAP-Elites and NSLC display a large variance, whereas the newly introduced algorithms have a lower median distance to the goal and less variance. The expansive variants of MAP-Elites and CVT-MAP-Elites find "good" bounds in this environment from the 0^{th} generation, which become more refined and stop changing after approximately 100 generations. The bounds do not extend to 0 and 1000, as in the non-expansive variants, due to the outer border; this means that more cells are allocated inside the arena.

We have also calculated the quality-diversity (QD) score [10] by mapping an archive's behavior descriptors to a 32×32 MAP-Elites grid, keeping the best performing one in a cell, and summing the fitness scores from all cells. The QD-scores for a typical archive of all algorithms, calculated in the initial environment are the following (lower is better): NSLC: 79396.1; Cluster-Elites: 79572.5; expansive MAP-Elites: 83770.2; expansive CVT-MAP-Elites: 94492.9; CVT-MAP-Elites: 119168.8; MAP-Elites: 129493.8.

5 CONCLUSION

Overall, our results illustrate that Cluster-Elites and the expansive variants of MAP-Elites and CVT-MAP-Elites have comparable

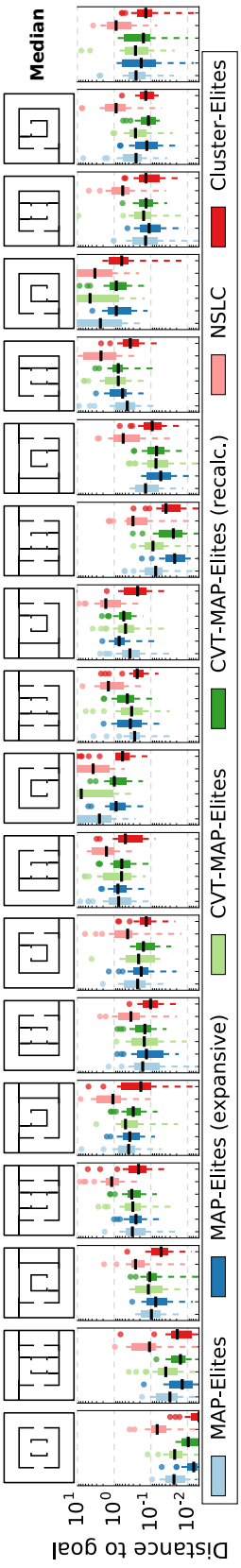

Figure 1: Best performance (distance to the goal) for each algorithm in the maze environment used during evolution (leftmost column) and all 16 evaluation environments each of which permits a single path to the center (goal). The box plots show the median and the interquartile range over 30 solutions, apart from the rightmost column which is calculated from the medians over all 17 environments; the whiskers extend to the most extreme data points not considered outliers, and outliers are plotted individually. Overall, Cluster-Elites and the variants of MAP-Elites and CVT-MAP-Elites that expand their bounds have comparable or better performance with lower variance than NSLC or MAP-Elites and CVT-MAP-Elites that know the bounds of the space.

performance with their "fixed-bounds" counterparts and NSLC, *without* requiring knowledge of the bounds. Moreover, Cluster-Elites is a promising algorithm that demands further investigation. In particular, experiments with complex tasks in which the points in behavior space lie on highly non-convex manifolds could highlight the benefits of Cluster-Elites over the bounding rectangle approach followed by MAP-Elites and CVT-MAP-Elites. In addition, combining NSLC with Cluster-Elites, i.e., by reducing the novelty archive in a manner similar to Cluster-Elites, might have advantages over both algorithms.

ACKNOWLEDGMENTS

This work was supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (Project: ResiBots, grant agreement No 637972).

A PSEUDOCODES

Algorithm 1 Expansive MAP-Elites algorithm

procedure EXPANSIVE-MAP-ELITES($[n_1, ..., n_d]$)
  $(\mathcal{X}, \mathcal{P}, \mathcal{B}) \longleftarrow$ create_empty_archive($[n_1, ..., n_d]$)
  $(\mathbf{b}_{min}, \mathbf{b}_{max}) = ([inf, ..., inf], [-inf, ..., -inf])$
  $X \longleftarrow$ random_solutions() ▷ *Initialization: I random* **x**
  $(P, B) \longleftarrow$ evaluate(X) ▷ *I evaluations*
  $(\mathbf{b}_{min}, \mathbf{b}_{max}) \longleftarrow$ update_ranges($B, \mathbf{b}_{min}, \mathbf{b}_{max}$)
  $B \longleftarrow$ normalize($B, \mathbf{b}_{min}, \mathbf{b}_{max}$)
  ADD_TO_ARCHIVE($\mathcal{X}, \mathcal{P}, \mathcal{B}, X, P, B$)
  for $g = 1 \rightarrow G$ **do** ▷ *Main loop, G generations*
    X = selection($\mathcal{X}$)
    X' = variation(X)
    $(P', B') \longleftarrow$ evaluate(X') ▷ *m evaluations*
    $(\mathbf{b}_{min}, \mathbf{b}_{max}) \longleftarrow$ update_ranges($B', \mathbf{b}_{min}, \mathbf{b}_{max}$)
    $\mathcal{B} \longleftarrow$ normalize($\mathcal{B}, \mathbf{b}_{min}, \mathbf{b}_{max}$)
    $B' \longleftarrow$ normalize($B', \mathbf{b}_{min}, \mathbf{b}_{max}$)
    $(X, P, B) \longleftarrow (\mathcal{X} \cup X', \mathcal{P} \cup P', \mathcal{B} \cup B')$
    $(\mathcal{X}, \mathcal{P}, \mathcal{B}) \longleftarrow (\{\}, \{\}, \{\})$ ▷ Clear the archive
    ADD_TO_ARCHIVE($\mathcal{X}, \mathcal{P}, \mathcal{B}, X, P, B$)
  return archive $(\mathcal{X}, \mathcal{P}, \mathcal{B})$
procedure ADD_TO_ARCHIVE($\mathcal{X}, \mathcal{P}, \mathcal{B}, X, P, B$)
  for $i = 0 \rightarrow |X|$ **do**
    $c \longleftarrow$ get_cell_index($B[i]$)
    if $\mathcal{P}[c] = null$ or $\mathcal{P}[c] < P[i]$ **then**
      $(\mathcal{P}[c], \mathcal{X}[c], \mathcal{B}[c]) \longleftarrow (P[i], X[i], B[i])$

REFERENCES

[1] A. Cully, J. Clune, D. Tarapore, and J.-B. Mouret. 2015. Robots that can adapt like animals. *Nature* 521, 7553 (2015), 503–507.
[2] Herbert Edelsbrunner, David Kirkpatrick, and Raimund Seidel. 1983. On the shape of a set of points in the plane. *IEEE Transactions on Information Theory* 29, 4 (1983), 551–559.
[3] Leonard Kauffman and Peter J. Rousseeuw. 1987. Clustering by means of Medoids. In *Statistical Data Analysis Based on the L_1–Norm and Related Methods*, Y. Dodge (Ed.). North-Holland, 405–416.
[4] J. Lehman and K. O. Stanley. 2008. Exploiting Open-Endedness to Solve Problems Through the Search for Novelty. In *ALIFE*. 329–336.
[5] J. Lehman and K. O Stanley. 2011. Abandoning objectives: Evolution through the search for novelty alone. *Evol. Comp.* 19 (2011), 189–223.

Algorithm 2 Expansive CVT-MAP-Elites algorithm

procedure EXPANSIVE-CVT-MAP-ELITES(k)
  $(\mathcal{X}, \mathcal{P}, \mathcal{B}) \longleftarrow$ create_empty_archive(k)
  $(\mathbf{b}_{min}, \mathbf{b}_{max}) = ([inf, ..., inf], [-inf, ..., -inf])$
  $X \longleftarrow$ random_solutions() ▷ *Initialization: I random* **x**
  $(P, B) \longleftarrow$ evaluate(X) ▷ I evaluations
  $(\mathbf{b}_{min}, \mathbf{b}_{max}) \longleftarrow$ update_ranges($B, \mathbf{b}_{min}, \mathbf{b}_{max}$)
  $C \longleftarrow$ CVT($k, \mathbf{b}_{min}, \mathbf{b}_{max}$) ▷ Get centroids from CVT
  ADD_TO_ARCHIVE($\mathcal{X}, \mathcal{P}, \mathcal{B}, X, P, B, C$)
  for $g = 1 \rightarrow G$ **do** ▷ *Main loop, G generations*
    X = selection($\mathcal{X}$)
    X' = variation(X)
    $(P', B') \longleftarrow$ evaluate(X') ▷ m evaluations
    if recomputeCVT(g) **then**
      $(\mathbf{b}_{min}, \mathbf{b}_{max}) \longleftarrow$ update_ranges($B', \mathbf{b}_{min}, \mathbf{b}_{max}$)
      $C \longleftarrow$ CVT($k, \mathbf{b}_{min}, \mathbf{b}_{max}$)
      $(X, P, B) \longleftarrow (\mathcal{X}, \mathcal{P}, \mathcal{B})$
      $(\mathcal{X}, \mathcal{P}, \mathcal{B}) \longleftarrow (\{\}, \{\}, \{\})$ ▷ Clear the archive
      ADD_TO_ARCHIVE($\mathcal{X}, \mathcal{P}, \mathcal{B}, X, P, B, C$)
    ADD_TO_ARCHIVE($\mathcal{X}, \mathcal{P}, \mathcal{B}, X', P', B', C$)
  return archive $(\mathcal{X}, \mathcal{P}, \mathcal{B})$
procedure ADD_TO_ARCHIVE($\mathcal{X}, \mathcal{P}, \mathcal{B}, X, P, B, C$)
  for $i = 0 \rightarrow |X|$ **do**
    $c \longleftarrow$ get_index_of_closest_centroid($B[i], C$)
    if $\mathcal{P}[c] = null$ or $\mathcal{P}[c] < P[i]$ **then**
      $(\mathcal{P}[c], \mathcal{X}[c], \mathcal{B}[c]) \longleftarrow (P[i], X[i], B[i])$
procedure CVT($k, \mathbf{b}_{min}, \mathbf{b}_{max}$)
  $C \longleftarrow$ sample_points($k, \mathbf{b}_{min}, \mathbf{b}_{max}$) ▷ k random centroids
  $S \longleftarrow$ sample_points($K, \mathbf{b}_{min}, \mathbf{b}_{max}$) ▷ K random samples
  for i = 0 $\longrightarrow$ *max_iter* **do**
    $\mathcal{I} \longleftarrow$ get_closest_centroid_indices(S, C)
    $C \longleftarrow$ update_centroids($\mathcal{I}$)
  return centroids C

Algorithm 3 Cluster-Elites algorithm

procedure CLUSTER-ELITES($k_{init}, k_{incr}, k_{max}$)
  $k = k_{init}, k_f = 0$
  $(\mathcal{X}, \mathcal{P}, \mathcal{B}) \longleftarrow$ create_empty_archive()
  $X \longleftarrow$ random_solutions() ▷ *Initialization: I random* **x**
  $(P, B) \longleftarrow$ evaluate(X) ▷ I evaluations
  ADD_TO_ARCHIVE($\mathcal{X}, \mathcal{P}, \mathcal{B}, X, P, B$)
  CLUSTER($\mathcal{X}, \mathcal{P}, \mathcal{B}, X, P, B$) ▷ Cluster-Elites procedure
  for $g = 1 \rightarrow G$ **do** ▷ *Main loop, G generations*
    if $k >= k_{max}$ **then**
      $k = k_{max}$
    else
      $k = k + k_{incr}$
    X = selection($\mathcal{X}$)
    X' = variation(X)
    $(P', B') \longleftarrow$ evaluate(X') ▷ m evaluations
    ADD_TO_ARCHIVE($\mathcal{X}, \mathcal{P}, \mathcal{B}, X', P', B'$)
    CLUSTER($\mathcal{X}, \mathcal{P}, \mathcal{B}, X, P, B$) ▷ Cluster-Elites procedure
  return archive $(\mathcal{X}, \mathcal{P}, \mathcal{B})$
procedure ADD_TO_ARCHIVE($\mathcal{X}, \mathcal{P}, \mathcal{B}, X, P, B$)
  for $i = 0 \rightarrow |X|$ **do**
    if $X[i] \notin \mathcal{X}$ **then**
      Insert $(X[i], P[i], B[i])$ in $(\mathcal{X}, \mathcal{P}, \mathcal{B})$
procedure CLUSTER($\mathcal{X}, \mathcal{P}, \mathcal{B}, X, P, B$)
  while $k_f < k$ and $k_f < |B|$ **do**
    if $k_f >= |\mathcal{B}|$ **then**
      Insert $(X[k_f], P[k_f], B[k_f])$ in $(\mathcal{X}, \mathcal{P}, \mathcal{B})$
    else
      $(\mathcal{P}[k_f], \mathcal{X}[k_f], \mathcal{B}[k_f]) \longleftarrow (P[k_f], X[k_f], B[k_f])$
    $k_f = k_f + 1$
  C = CENTROIDS($\mathcal{B}, k$) ▷ Get the centroids
  $\mathcal{I} \longleftarrow$ get_closest_centroid_indices($\mathcal{B}, C$)
  $(X_\epsilon, P_\epsilon, B_\epsilon) \longleftarrow$ GET_CLUSTER_ELITES($\mathcal{X}, \mathcal{P}, \mathcal{B}, \mathcal{I}, k$)
  for $i = 0 \rightarrow |\mathcal{X}|$ **do** ▷ Reduce archive
    if $\mathcal{X}[i] \neq X_\epsilon[\mathcal{I}[i]]$ **then**
      Remove $(\mathcal{X}[i], \mathcal{P}[i], \mathcal{B}[i], \mathcal{I}[i])$ from $(\mathcal{X}, \mathcal{P}, \mathcal{B}, \mathcal{I})$
procedure CENTROIDS($\mathcal{B}, k$)
  $C = \mathcal{B}$
  for $i = 0 \rightarrow |\mathcal{B}| - k$ **do**
    D = mean_distances_to_nearest_neighbors(C)
    $d = \arg\min(D)$
    Remove $C[d]$ from C
  return centroids C
procedure GET_CLUSTER_ELITES($\mathcal{X}, \mathcal{P}, \mathcal{B}, \mathcal{I}, k$)
  $(X_\epsilon, P_\epsilon, B_\epsilon) \longleftarrow$ create_empty(k)
  for $i = 0 \rightarrow |\mathcal{X}|$ **do**
    $c = \mathcal{I}[i]$
    if $P_\epsilon[c] = null$ or $P_\epsilon[c] < \mathcal{P}[i]$ **then**
      $(P_\epsilon[c], X_\epsilon[c], B_\epsilon[c]) \longleftarrow (\mathcal{P}[i], \mathcal{X}[i], \mathcal{B}[i])$
  return elites $(X_\epsilon, P_\epsilon, B_\epsilon)$

[6] J. Lehman and K. O. Stanley. 2011. Evolving a Diversity of Virtual Creatures Through Novelty Search and Local Competition. In *GECCO '11*. ACM, 211–218.
[7] E. Lewitus and H. Morlon. 2016. Natural constraints to species diversification. *PLoS Biol* 14, 8 (2016), e1002532.
[8] J.-B. Mouret. 2011. Novelty-based multiobjectivization. In *New Horizons in Evolutionary Robotics*. Springer, 139–154.
[9] J.-B. Mouret and J. Clune. 2015. Illuminating search spaces by mapping elites. *arXiv preprint arXiv:1504.04909* (2015).
[10] J. K. Pugh, L. B. Soros, and K. O. Stanley. 2016. Quality Diversity: A New Frontier for Evolutionary Computation. *Frontiers in Robotics and AI* 3 (2016), 40.
[11] L. Trujillo, G. Olague, E. Lutton, and F. F. De Vega. 2008. Discovering several robot behaviors through speciation. In *Workshops on Applications of Evolutionary Computation*. Springer, 164–174.
[12] V. Vassiliades, K. Chatzilygeroudis, and J.-B. Mouret. 2016. Scaling Up MAP-Elites Using Centroidal Voronoi Tessellations. *arXiv preprint arXiv:1610.05729* (2016).