Enhancing and Utilizing Diversity in Neuroevolution

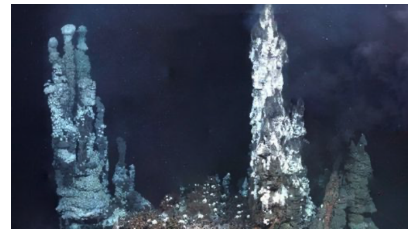
Risto Miikkulainen

September 18, 2024

Diversity in Biological Evolution

A remarkable outcome of biological evolution: Organisms thrive in a large variety of environments as highly diverse solutions.

- Life in extreme heat and cold, thin atmosphere, deep ocean pressure
- Variety of energy sources and chemical building blocks

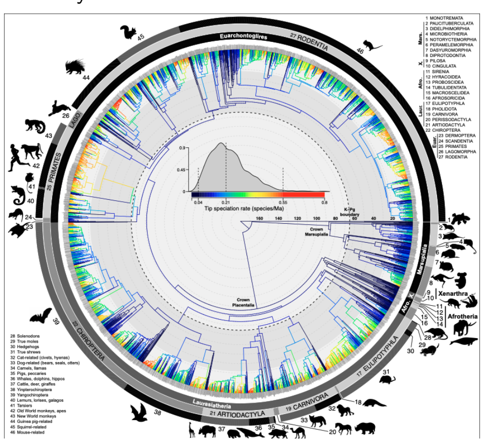

Deep-sea hydrothermal vent


Tardigrade (Water Bear)

Diversity as an Evolutionary Mechanism

Diversity makes it possible to construct complex solutions over time and to adapt to changing environments.

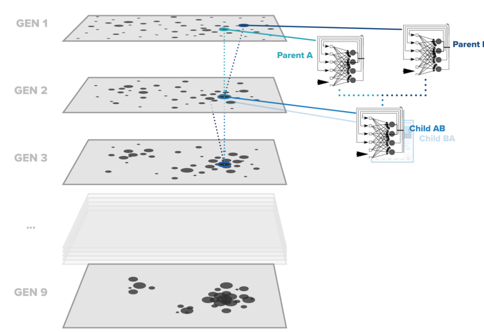
- Evolution is selection upon variation.
- Evolution is a "tinkerer" (Jacob 1977): Small modifications to existing solutions eventually lead to discoveries.


Mammals

Diversity in Computational Evolution

Generating and maintaining diversity is crucial in evolutionary computation as well.

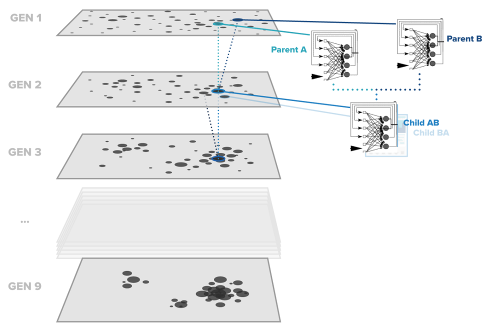
- Diversity does not arise naturally in most algorithms.
- Special mechanisms are needed to maintain it.



Genetic Diversity

Evolutionary computation seeks an optimum in a fitness landscape.

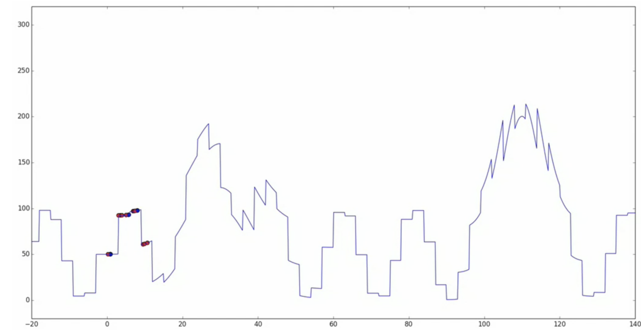
- Initial population is widespread, gradually converging around the peaks.
- Convergence helps refine solutions to find the best ones.



Premature Convergence

Premature convergence can occur before promising areas of the fitness landscape are explored.

- Best solutions may have narrow basins, leading to missed discoveries.
- Dynamic problems require diversity to adapt to changing fitness landscapes.

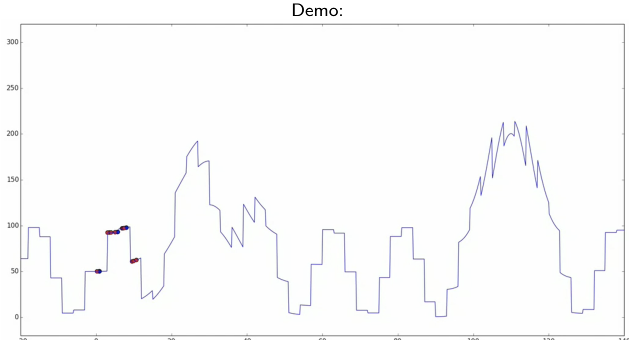


Premature Convergence

When populations converge, recombination loses its power.

- The process reduces to (parallel) random search through mutation.

Demo:



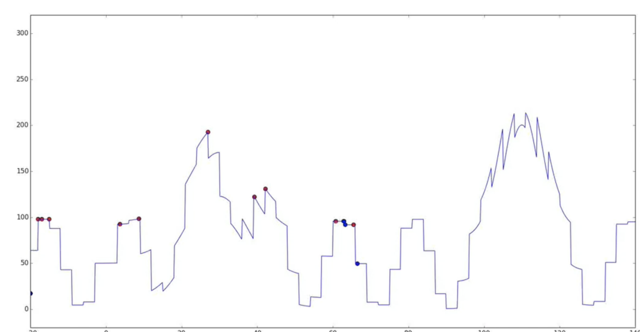
Maintaining Genetic Diversity Indirectly

Increased mutation rates help maintain diversity.

- However they undermine search, making it more random.

Archives of past individuals help maintain diversity.

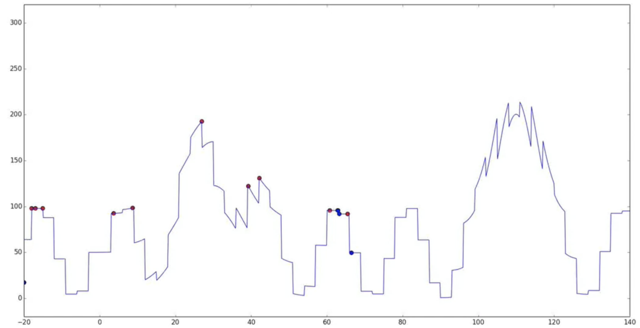
- Only accept individuals into the population if they are new.
- However, decisions about which individuals to archive is challenging.



Measuring Genetic Diversity

Diversity is often measured as the distance between genomes.

- Euclidean, Manhattan, and Hamming distances are commonly used.
- Diversity can be focused on local areas or k-nearest neighbors.



Maintaining Genetic Diversity Directly

Once there is a measure, can develop methods to maintain diversity directly.

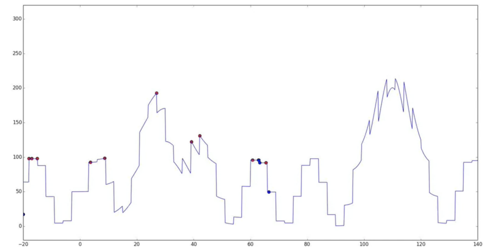
- Crowding: Slows down convergence by replacing similar individuals.
- Fitness Sharing: Adjusts fitness based on how similar an individual is to others in the population:

  Fitness $f(x)$ of individual x is adjusted by

  $$f'(x) = \frac{f(x)}{s(x)}.$$

  where similarity s to other individuals y is based on distance d:

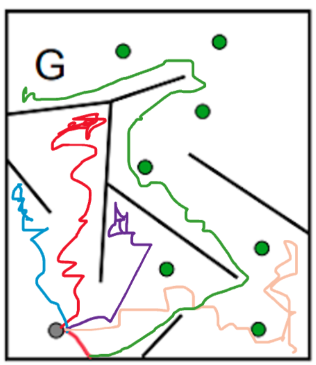
  $$s(x) = \sum_{j=1}^{n} d(x, y_j).$$



Behavioral Diversity

When genotypes result in complex behavior that determines fitness, behavioral diversity becomes more telling than genetic diversity.

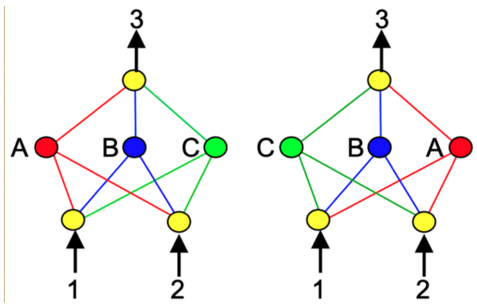
- E.g. neuroevolution constructs computational structures (neural networks) rather than static solutions.
- Diverse behaviors are needed to explore deceptive or flat fitness landscapes.



Challenges in Mapping Genotype to Behavior

Genetic diversity does not always lead to behavioral diversity.

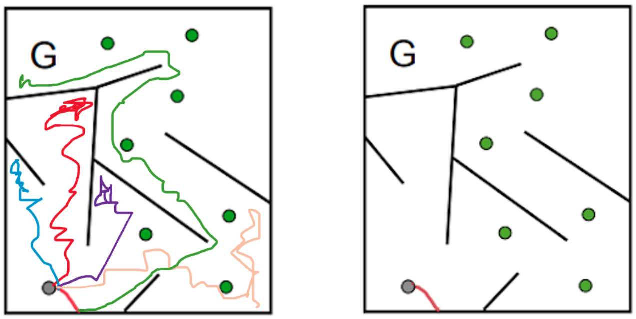
- The mapping between genotype and behavior is complex and unpredictable.
- Example: Competing conventions – different neural networks may produce the same behavior despite genetic differences.



Measuring Behavioral Diversity

To construct mechanisms for generating and maintaining behavioral diversity, we need to measure it.

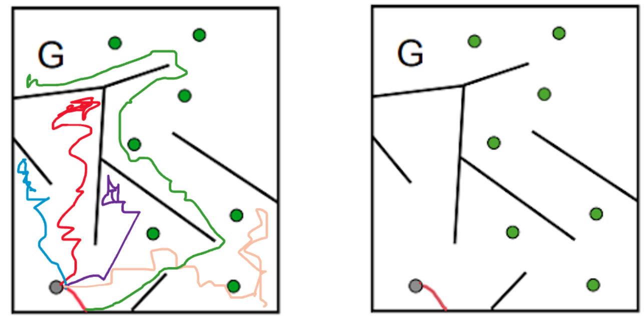
- Need a formal way for behavior characterization (BC)
- BC can be a histogram of sensory inputs, actions, and locations for a mobile robot.
- Alternatively, BC can be a vector of responses to a set of representative inputs (i.e. a syllabus, or a questionnaire).



BC for Recurrent Neural Networks

Characterizing behavior in recurrent neural networks is more complex.

- The history of sensory inputs and actions matters, not just current inputs.
- Actions can be represented as distributions, and BCs as mappings from sensory states to action distributions.



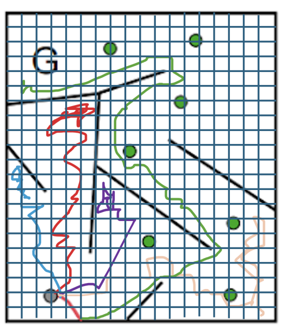
Entropy Measures of Diversity

Entropy can used to measure behavioral diversity of a population.

- Entropy measures the level of uncertainty, surprise, disorder.
- Behavioral diversity can be quantified by dividing the behavioral space into intervals m and counting how many agents visit them as p_m:

$$H = -\sum_{m=1}^{M} p_m \log(p_m)$$

- Entropy is maximized with a uniform distribution.



Empowerment: A Measure of Causal Influence

Empowerment measures how much an agent's actions influence future sensory inputs.

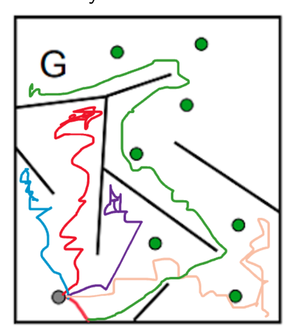
- Defined as the channel capacity between actuators and sensors over time:

$$E = \max_{p(a_t)} I(S_{t+1}; A_t),$$

  where $I(S; A)$ is the mutual information between sensors S and actuators A

$$I(S; A) = H(S) - H(S|A),$$

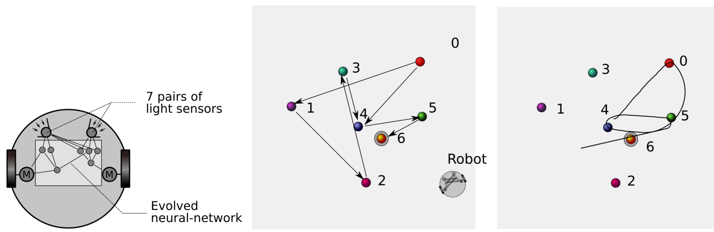
  and $p(a_t)$ is the probability of actuator value a_t at time t.
- Empowerment provides intrinsic motivation for agents to explore and act, promoting behavioral diversity.



Behavioral Diversity in Action: Robot Lights Example

When the robot moves to a light, one or two other lights turn on; it needs to find a sequence to turn on light 6.

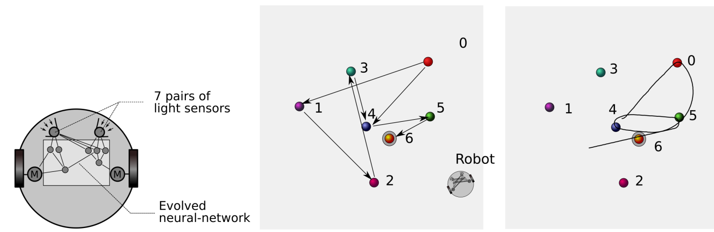
- Fitness is deceptive – no clear feedback on progress until the task is solved.
- Behavioral diversity encourages exploration and helps the robot find the correct sequence.



Impact of Behavioral Diversity

Behavioral diversity helps search processes cope with deceptive or flat fitness landscapes.

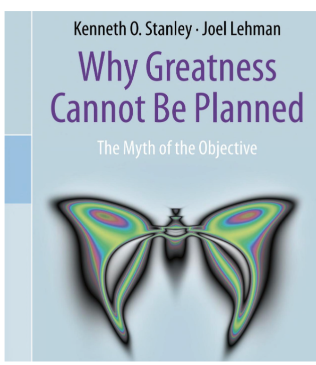
- Encouraging diverse behaviors leads to exploration of broader areas of the search space.
- Eventually, this increases the likelihood of discovering successful solutions.
- Does it lead to innovation?



Encouraging Innovation: Novelty Search

Novelty search focuses entirely on generating new variations without converging on a single solution.

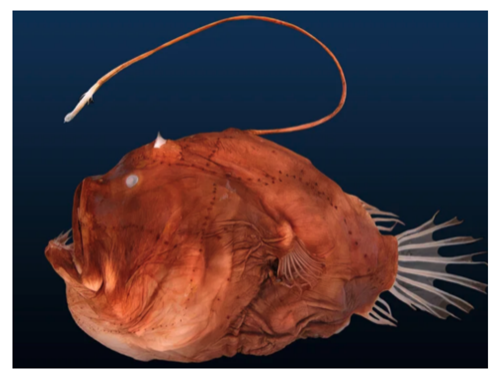
- Unlike traditional evolutionary search, novelty search rewards diversity rather than performance.
- It is a divergent search process instead of convergent.
- A way to achieve greatness?



Biological Inspiration for Divergent Evolution

In nature, evolution does not have a specific goal. It continuously generates variation, allowing adaptation to new niches.

- Divergent search can result in creative solutions like the anglerfish's lure or E.coli evolving to utilize citric acid.



Formalizing Novelty Search

Novelty search is formalized by replacing performance metrics with a novelty metric.

- Calculated as the average distance to an individual's k nearest neighbors in behavior space.

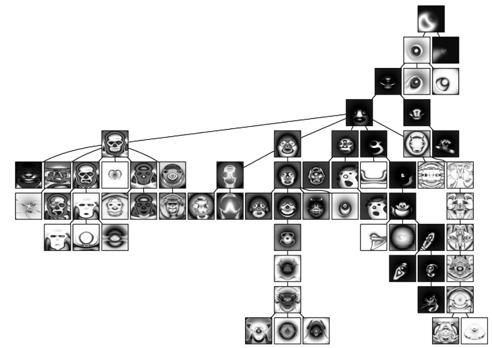
$$\rho(x) = \frac{1}{k} \sum_{j=1}^{k} d(x, y_j), \tag{1}$$

  where $\rho(x)$ is the novelty of individual x, and $d(x, y_j)$ is the distance to neighbor y_j.
- The novelty metric measures how different a candidate solution is from those generated before.

Novelty Search: A Path to Diverse Solutions

Novelty search leads to a variety of diverse solutions. Remarkably, these solutions can also be useful, even without a direct fitness reward.

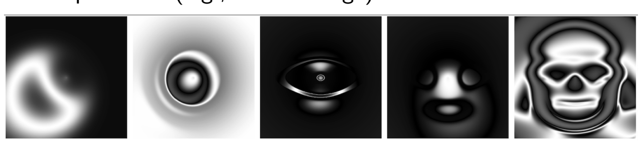
- In order to be different, need to make large changes.
- In order to make large changes, need to take advantage of structure.
- Structured changes can result in useful solutions.



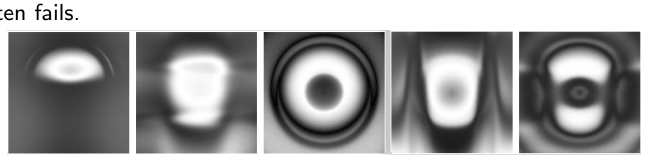
Stepping Stones: The Picbreeder Example

In particular, novelty search often discovers stepping stones: Partial solutions that later combine into more complex ones.

- E.g. in the Picbreeder game, players evolve images through novelty search, often without a clear goal in mind.
- Images that don't resemble the target initially serve as stepping stones to more complex ones (e.g., a skull image).



- In contrast, direct evolution towards a specific goal (e.g., a skull image) often fails.



Bipedal Walker Example: Fitness vs. Novelty Search

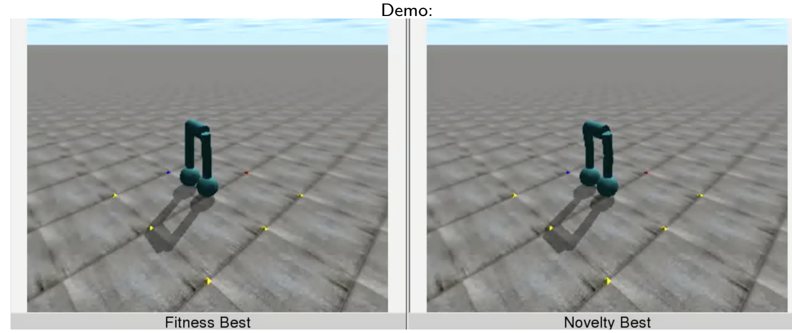
Novelty search often discovers useful behaviors by exploring failures.

- Falling, jumping, and stumbling all contribute stepping stones to better walking behavior.

Convergent search (fitness-based) often fails to find these creative solutions.

- Incremental improvement doesn't allow discovering dynamic moves.

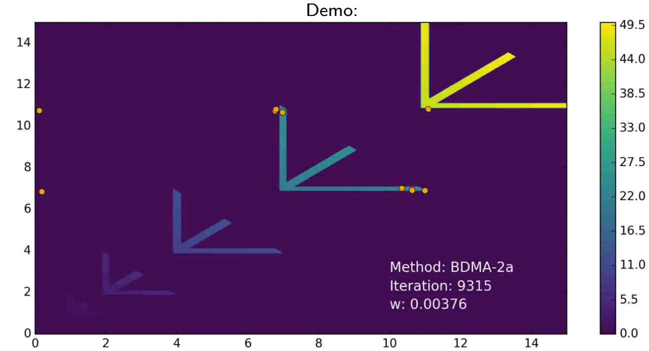
Demo:



Stepping Stones in Fitness Landscapes

Novelty search can uncover stepping stones for fitness-based search.

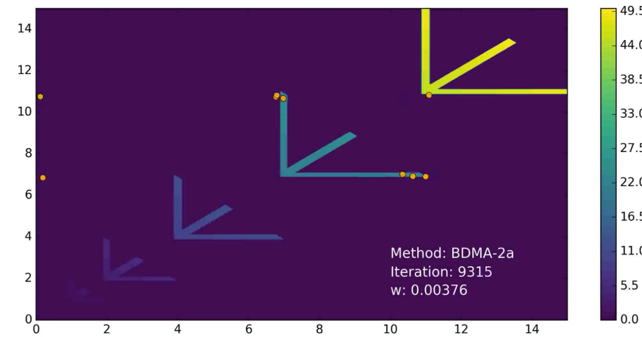
- E.g. flat fitness in the background and each structure.
- Novelty drives exploration that fitness immediately utilizes.
- Is there a way to combine novelty and fitness systematically?



Quality Diversity (QD)

Quality diversity methods combine novelty search with fitness-based search.

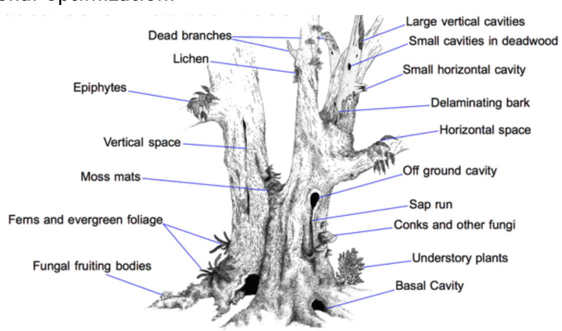
- A shift from finding the best solution to discovering a wide array of viable solutions.
- This combination allows finding better solutions faster by leveraging stepping stones.
- It mimics natural ecosystems where species thrive in different environments.



Core Idea of Quality Diversity

The core idea of QD is to balance diversity and quality, covering all regions of the behavior space.

- QD methods aim to find the best performance in each region, not just overall.
- This leads to a broader exploration of the solution space, beyond traditional optimization.



Novelty Search with Local Competition (NSLC)

NSLC addresses the problem of limited diversity in traditional evolutionary algorithms by introducing competition within local niches.

- Novelty search rewards uniqueness to prevent premature convergence.
- Local competition encourages high-performance solutions within specific niches.

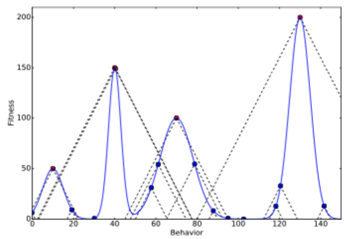
(a) Novelty Only (b) Fitness Only (c) Global Competition (d) Local Competition

NSLC Algorithm

The NSLC algorithm uses both novelty and local competition to drive diversity and performance.

- Individuals compete against their nearest neighbors within niches.
- Novelty is measured by how different an individual is from the rest.

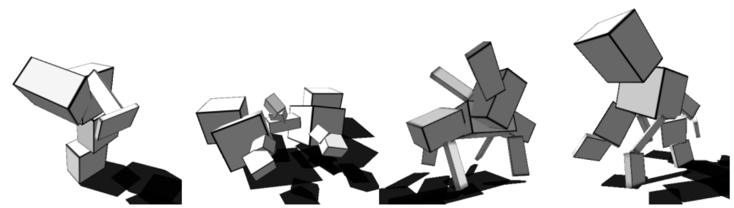
```
While generation < max_generations do:

  Evaluate Population:
    For each individual i in P do:
      - Compute fitness f_i = F(i)
      - Compute behavior characterization b_i = BC(i)

  Compute Scores:
    For each individual i in P do:
      - Compute Novelty Score n_i:
        - neighbors = FindKNearestNeighbors(b_i, P ∪ A, k)
        - n_i = AverageDistance(b_i, neighbors)
      - Compute Local Competition Score lc_i:
        - neighbors_P = neighbors ∩ P
        - lc_i = Count of neighbors_P where f_i > f_j for each neighbor j
      - Compute Selection Score s_i:
        - s_i = n_i + alpha * lc_i

  Update Novelty Archive:
    For each individual i in P do:
      - If n_i > novelty_threshold then:
        - Add b_i to archive A
```



NSLC: Results

NSLC leads to a much higher level of diversity compared to fitness-only approaches.

- E.g. evolving virtual creatures for locomotion.
- NSLC discovers a wide range of functional behaviors and physical forms.
- Utilizing hopping, crawling, momentum, tail,...

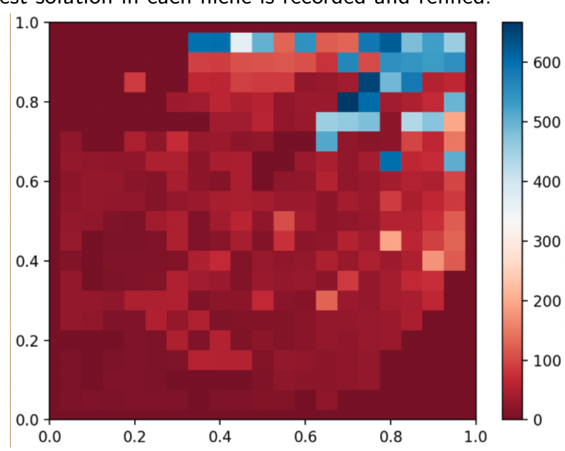


(a) Hopper (b) Crab (c) Quadruped (d) Tailed Quadruped

MAP-Elites: Structured Niching

MAP-Elites partitions the search space into a grid of niches based on behavior characterization.

- Each cell in the grid represents a unique behavior niche.
- The best solution in each niche is recorded and refined.



MAP-Elites Algorithm

The MAP-Elites algorithm explicitly defines niches and searches for the best solutions in each.

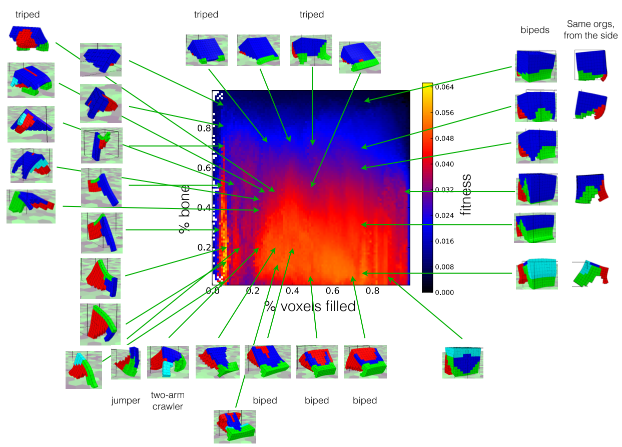
- If a niche is empty, the new candidate becomes the elite.
- If the niche already has an elite, the candidate replaces it only if it has higher fitness.

procedure MAP-ELITES ALGORITHM (SIMPLE, DEFAULT VERSION)
$(\mathcal{P} \leftarrow \emptyset, \mathcal{X} \leftarrow \emptyset)$ ▷ *Create an empty, N-dimensional map of elites: $\{$solutions $\mathcal{X}$ and their performances $\mathcal{P}\}$*
for iter $= 1 \rightarrow I$ **do** ▷ *Repeat for I iterations.*
  if iter $< G$ **then** ▷ *Initialize by generating G random solutions*
    $\mathbf{x}' \leftarrow$ random_solution()
  else ▷ *All subsequent solutions are generated from elites in the map*
    $\mathbf{x} \leftarrow$ random_selection($\mathcal{X}$) ▷ *Randomly select an elite x from the map $\mathcal{X}$*
    $\mathbf{x}' \leftarrow$ random_variation($\mathbf{x}$) ▷ *Create x', a randomly modified copy of x (via mutation and/or crossover)*
  $\mathbf{b}' \leftarrow$feature_descriptor($\mathbf{x}'$) ▷ *Simulate the candidate solution x' and record its feature descriptor $\mathbf{b}'$*
  $p' \leftarrow$performance($\mathbf{x}'$) ▷ *Record the performance p' of x'*
  if $\mathcal{P}(\mathbf{b}') = \emptyset$ or $\mathcal{P}(\mathbf{b}') < p'$ **then** ▷ *If the appropriate cell is empty or its occupants's performance is $\leq p'$, then*
    $\mathcal{P}(\mathbf{b}') \leftarrow p'$ ▷ *store the performance of x' in the map of elites according to its feature descriptor $\mathbf{b}'$*
    $\mathcal{X}(\mathbf{b}') \leftarrow \mathbf{x}'$ ▷ *store the solution x' in the map of elites according to its feature descriptor $\mathbf{b}'$*
return feature-performance map ($\mathcal{P}$ and $\mathcal{X}$)

MAP-Elites Example: Evolving Soft Robots

MAP-Elites preserves a diverse set of solutions that excel in different regions of the design space.

- E.g. number of bone elements and total number of elements.
- The method "illuminates" the search space by highlighting how different features contribute to success.
- Diversity promotes creativity and adaptability to changing conditions.



MAP-Elites Example: Evolving Multilegged Walking

Many designs optimizing behavior under constraints.

- E.g. how many legs touch the ground.
- Adaptation after injury.
- Results transfer to physical robots.

Demo: