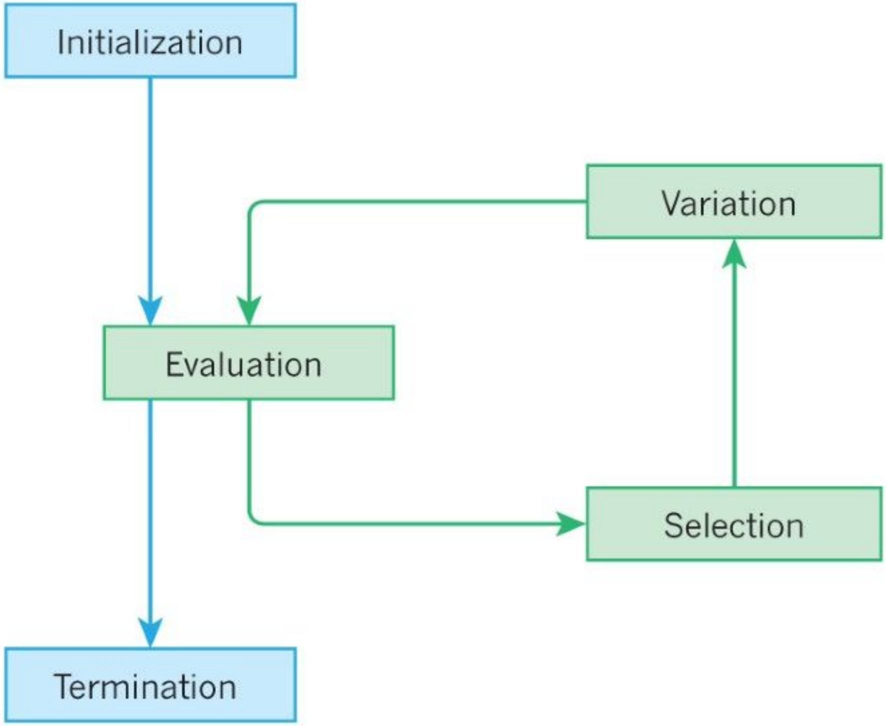
Lexicase Selection and RL

Ryan Boldi

University of Massachusetts Amherst

Evolutionary Computation



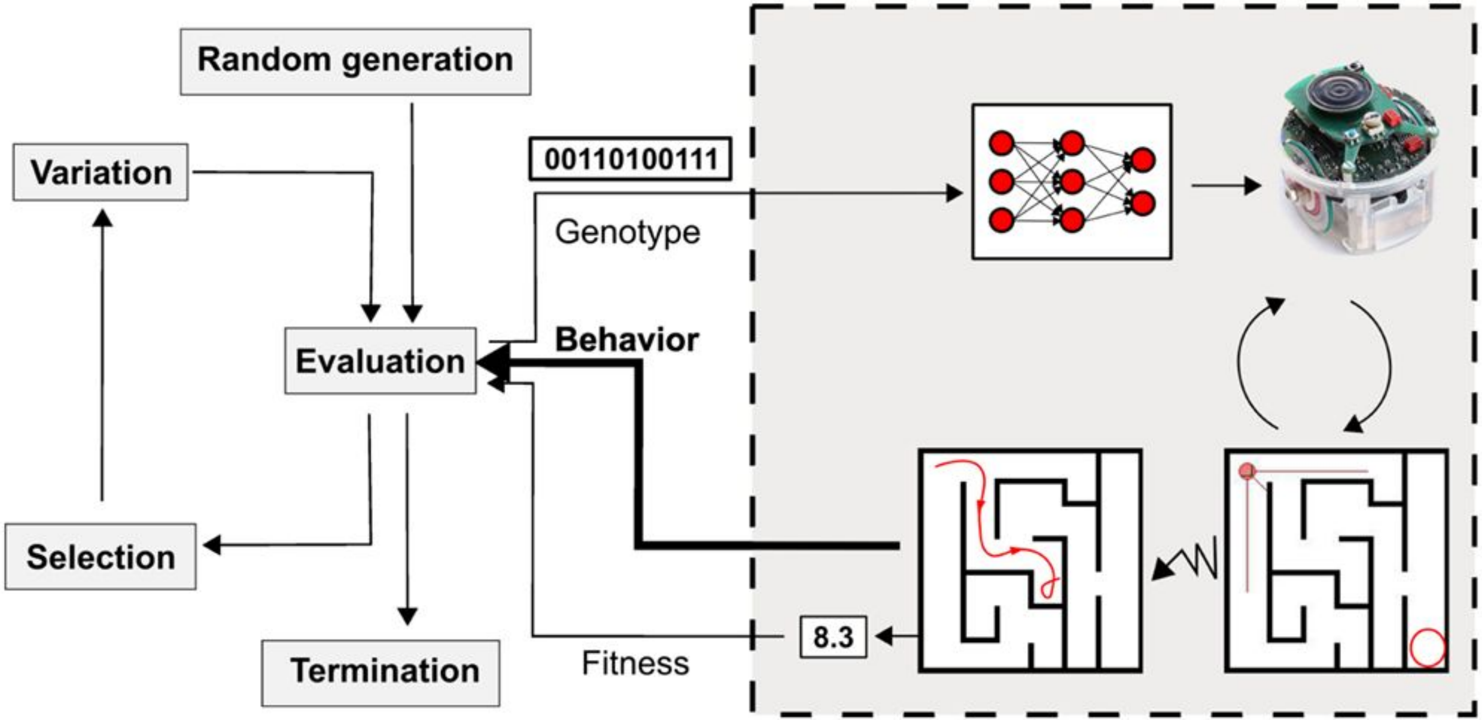
Quick: Evolution vs RL

$$\text{fitness} \approx \text{Reward}$$

Evolutionary methods usually **do not** construct value estimates of state-action pairs.

Makes EC potentially less powerful as observations are ignored and not learned from.

But, EC could help RL cope with **partial observability** and **continuity** in domains where state-action pairs are hard to define



Evolutionary robotics: what, why, and where to
Stephane Doncieux, Nicolas Bredeche, Jean-Baptiste Mouret and Agoston E. (Gusz) Eiben

Evaluation and Selection

How do you take a population of different solutions and select the best ones?

What if there are many things you care about?

- Performance
- Energy efficiency
- Safety
- Reliability

⋮

Evaluation

First, you should probably be able to assign a score to each individual on each of the different metrics you care about.

Individual i

- Performance
- Energy efficiency
- Safety
- Reliability
- …

$p = 9$

$e = 20\%$

$s = 94\%$

$r = 45\%$

Selection

Most Selection strategies then convert this into a single *fitness* value by either:

simply adding them together:

$$F = p + e + s + r = 9 + 0.2 + 0.94 + 0.45$$

or with a weighted sum:

$$\begin{aligned} F &= Ap + Be + Cs + Dr \\ &= A \times 9 + B \times 0.2 + C \times 0.94 + D \times 0.45 \end{aligned}$$

- Adding them together might result in scaling issues: some objectives will be weighted higher than others.
- This problem is NOT resolved with the weighted sum, as we still need to decide how important each objective is, and this requires human input (and possible bias)

Different Selection Strategies

$$F = p + e + s + r = 9 + 0.2 + 0.94 + 0.45$$

After this F value is found, we need to pick the individuals that are the best based on it. How?

- Tournament selection:
  - Select k individuals at random, and then pick the one of these with the highest F value.
- Fitness Proportionate Selection:
  - Select individuals at a probability proportional to their F value

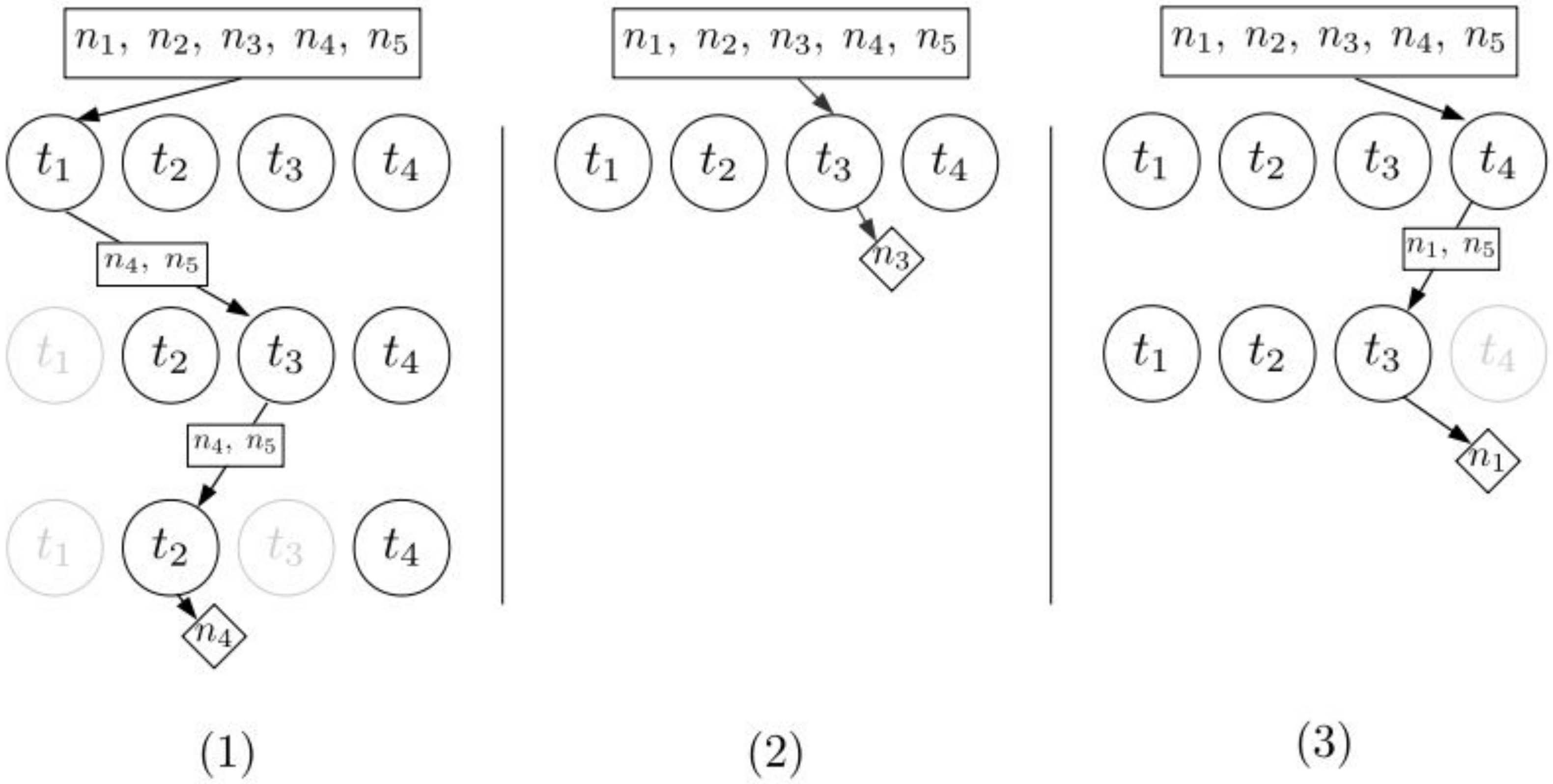
Lexicase Selection

- Avoids aggregation issues.
- Considers each objective in its own right .
- Does not compromise between objectives.
  - A really good model does not get any extra wiggle room to be unreliable

Put simply:

- Do not aggregate your objective scores, but instead consider them in a random order, and only keep the best individuals on the metrics in the order they come.

Lexicase Selection with 5 Individuals and 4 tests



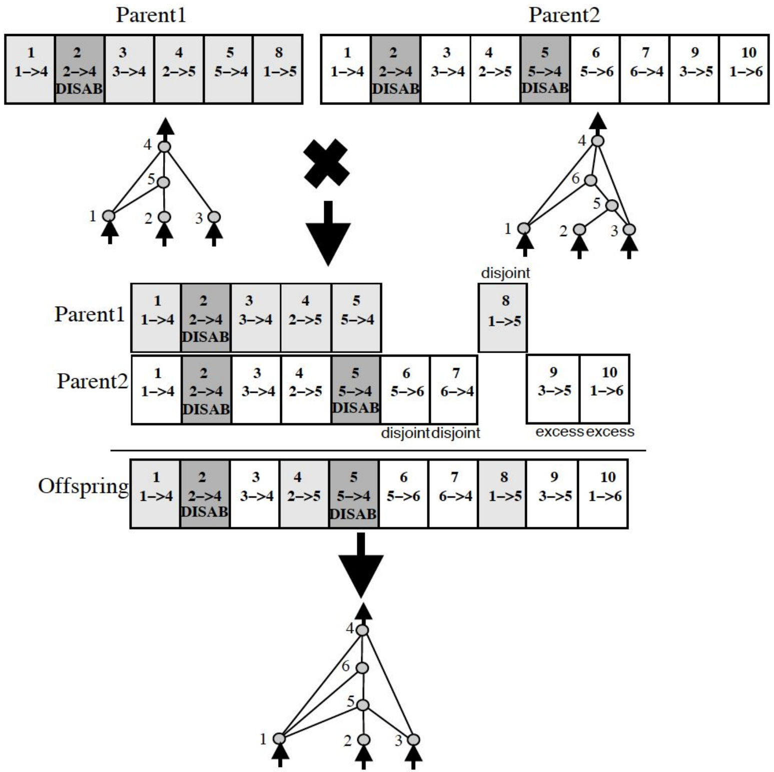
Neuroevolution

"Neural Evolution" = Evolution of Neural Networks

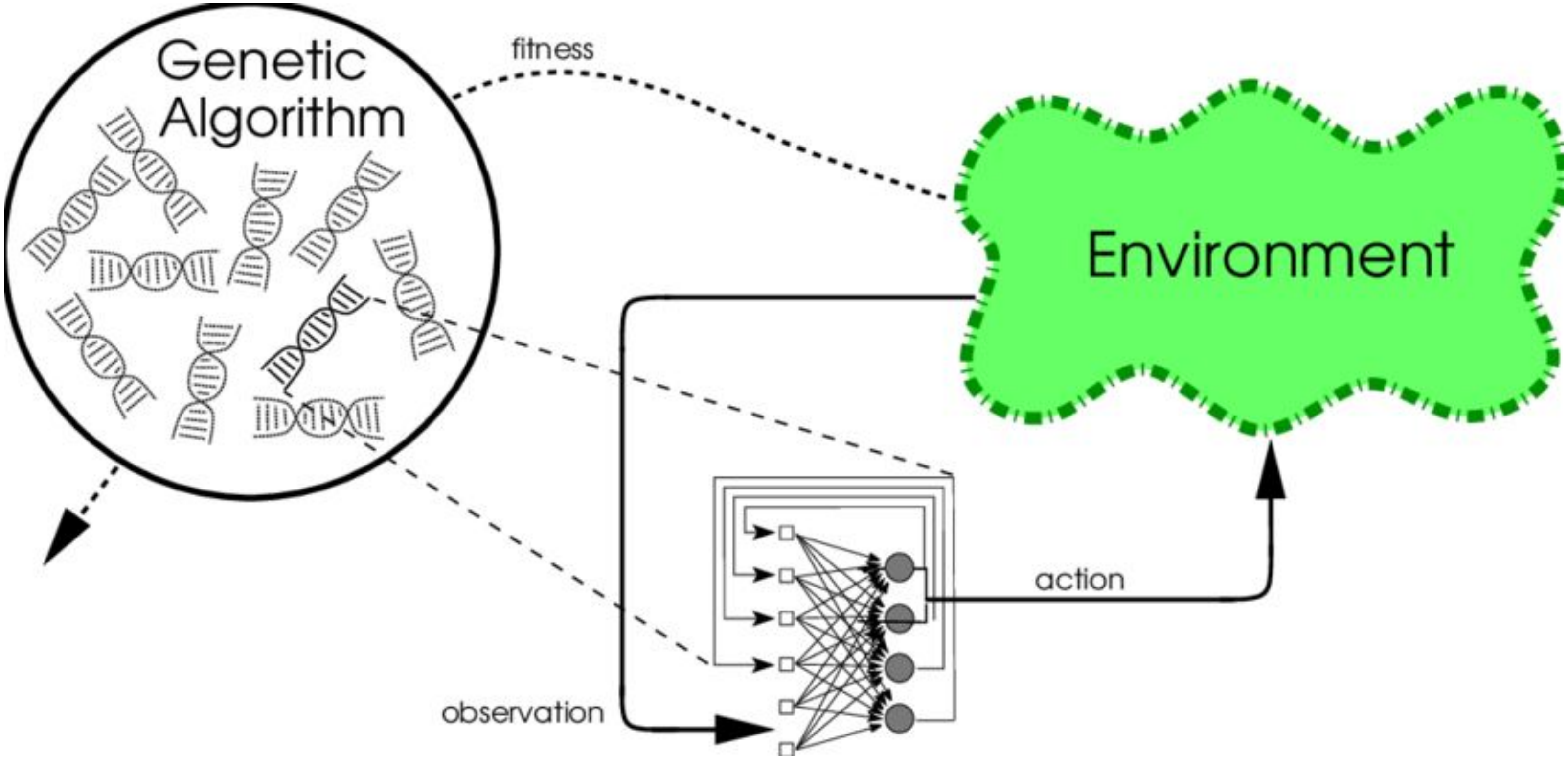
Seminal Paper:

Evolving Neural Networks through Augmenting Topologies -->

- Ken Stanley and Risto Miikkulainen (2002)

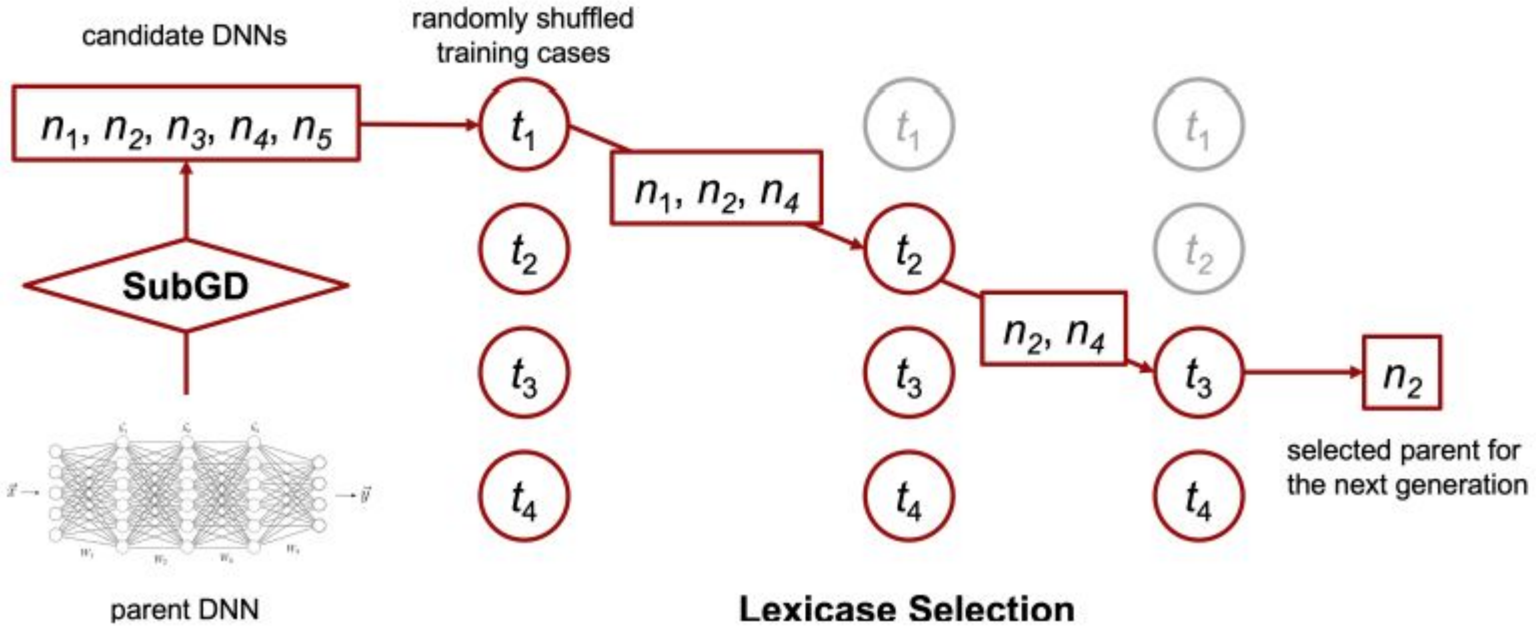


Neuroevolution for Sparsely Supervised Learning



Rewards are usually much sparser than those for RL. Usually the only “reward” signal is at the end of an episode.

Gradient Lexicase Selection



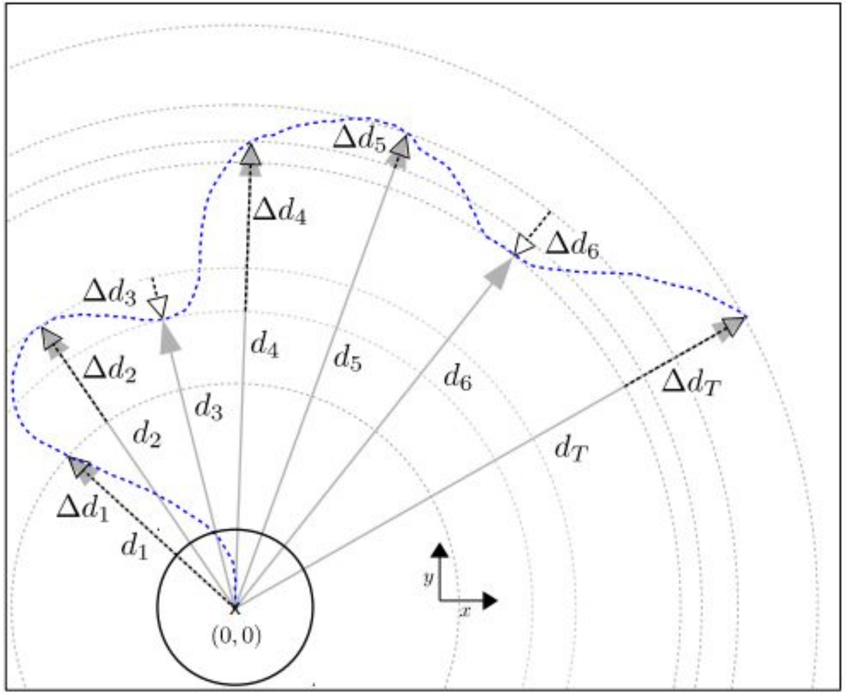
Things that might be Interesting (Advice?)

- Lexicase selection in RL
    - Policy Gradient Lexicase Selection?
        - Take different policies and place them in different starting states (or other ways to get a subset of the “training data”)
        - Find policy gradient for each
        - Follow each of these gradients to generate the children
        - Use lexicase selection to find which policy was the “best”
    - Use to balance different objectives (safety, quality, etc)

Things that might be Interesting

- Lexicase selection in RL
    - Deaggregate reward across time

$$R(\tau) = \cancel{\sum_{t=0}^{T}} r_t.$$



Things that might be Interesting

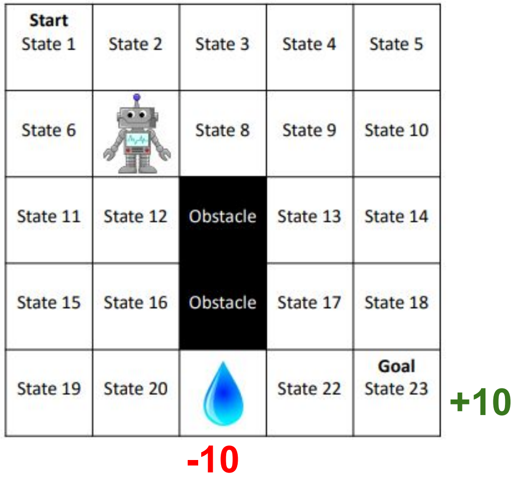
- Lexicase selection in RL

  How do we decide what reward different things should receive in an MDP?

  What scaling factor should we use for each thing?

  Solution: Don’t

2.3 687-Gridworld: A Simple Environment



Things that might be Interesting

- Lexicase-like stuff in RL
    - Hierarchical Preference Learning Project