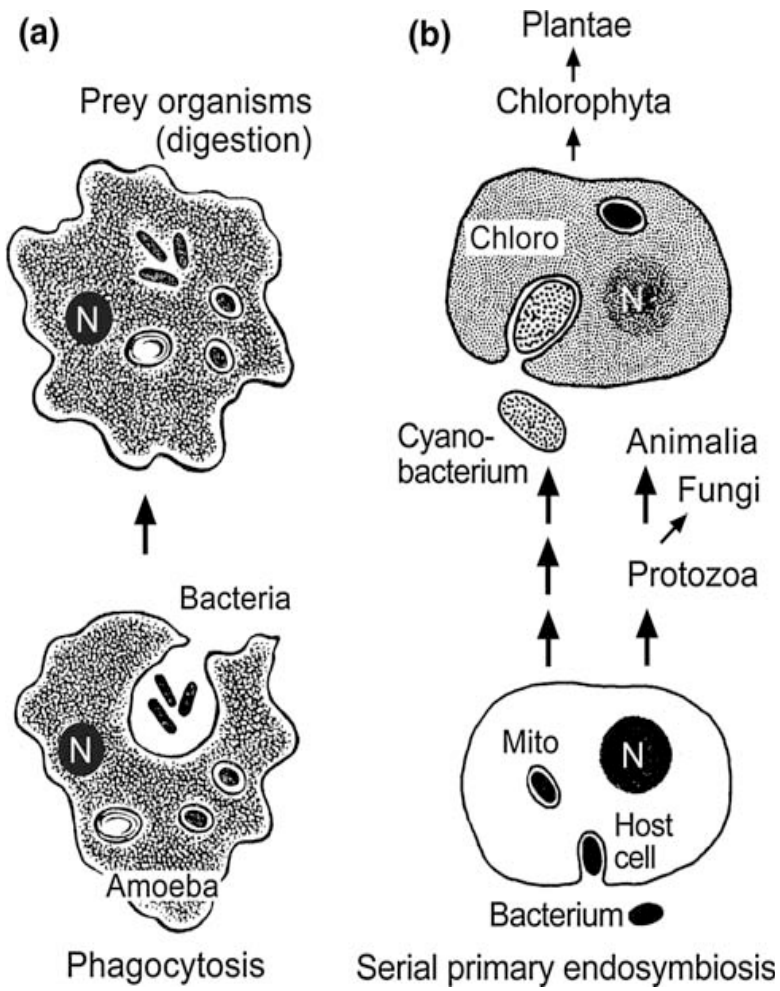
The Rubik Cube and GP Temporal Sequence Learning: An initial study

Peter Lichodzijewski and Malcolm Heywood

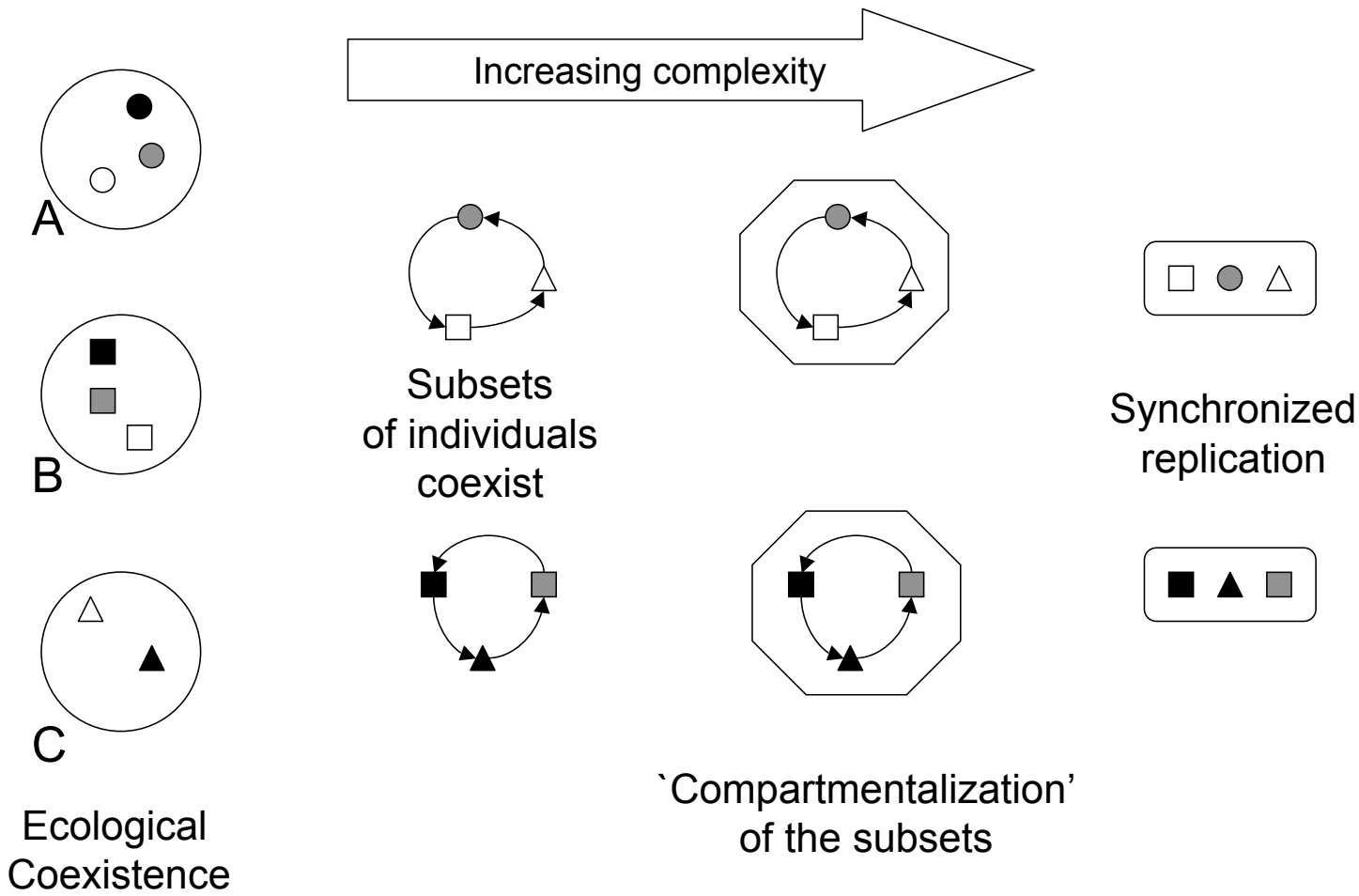
Dalhousie University, Computer Science

Serial Primary Endosymbiosis

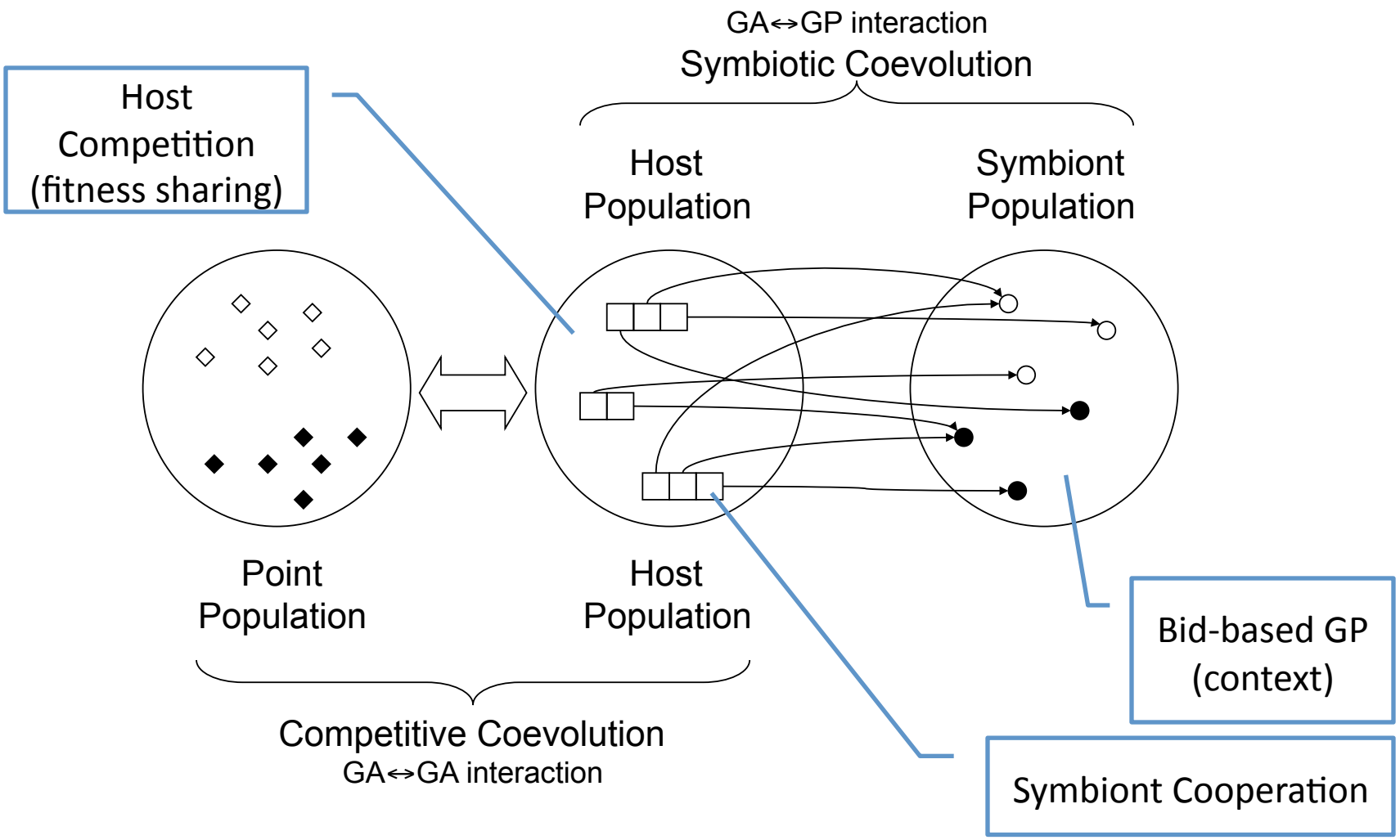
Kutschera (2009)



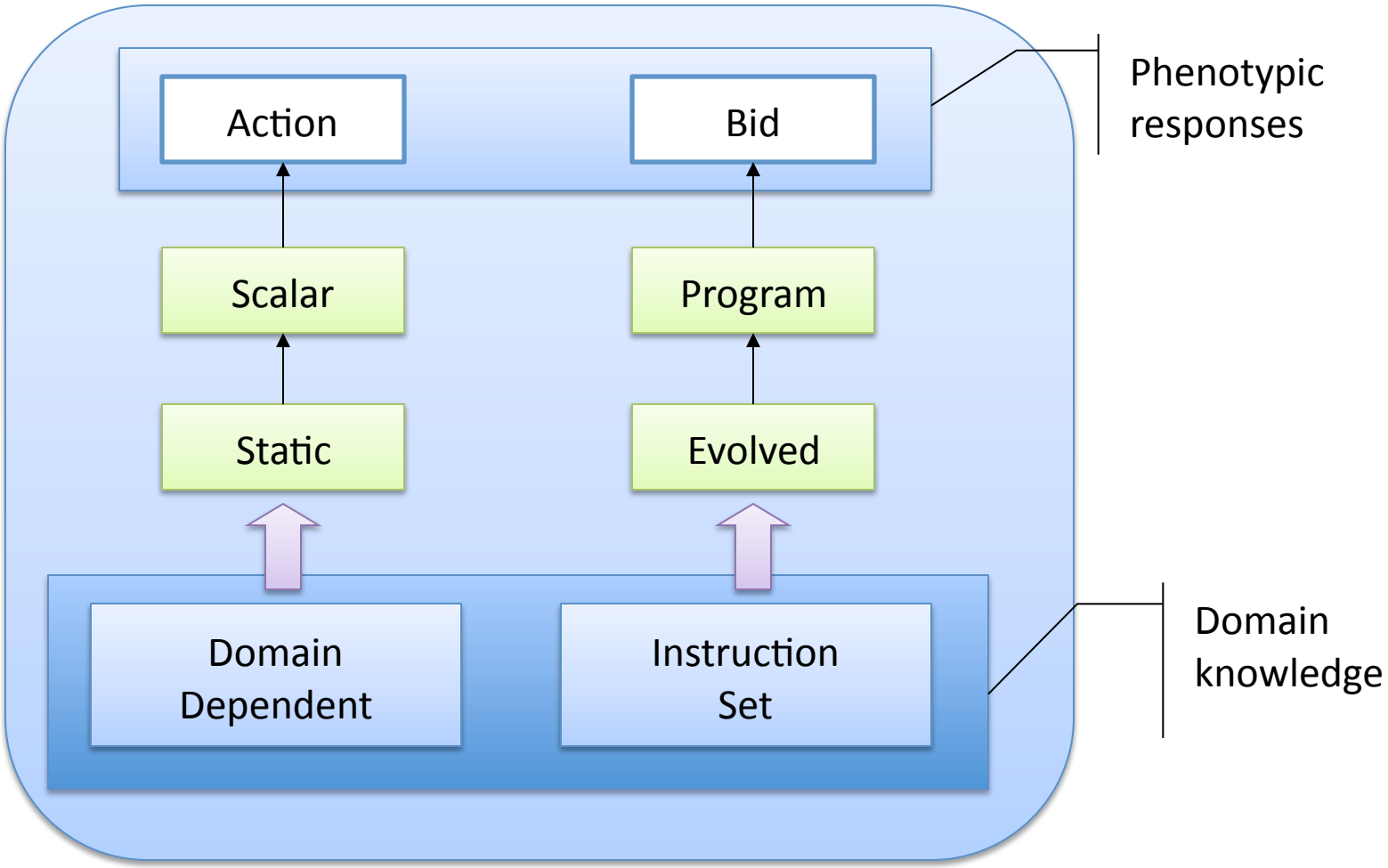
Abstract Model of Symbiosis:
Maynard Smith (1991)



Multi-Population/ Coevolution – GP:
Lichodzijewski and Heywood (2008)



Achieving Context – Bid-based GP:
Lichodzijewski and Heywood (2007)



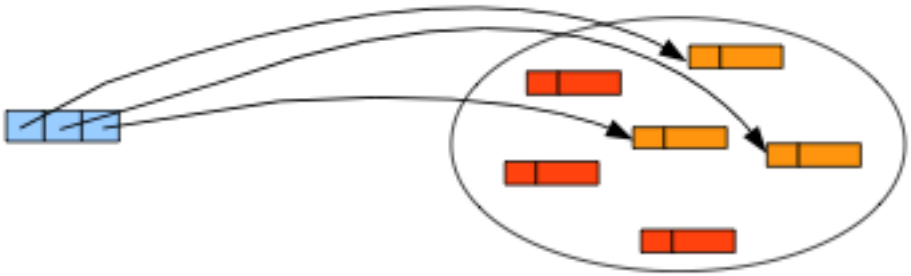
Algorithm 1 The core SBB training algorithm. P^t, H^t, and S^t refer to the point, host, and symbiont populations at time t.

```
procedure TRAIN
    t = 0                                                    ▷ Initialization
    initialize point population P^t
    initialize host population H^t (and symbiont population L^t)
    while t ≤ t_max do                                       ▷ Main loop
        create new points and add to P^t
        create new hosts and add to H^t (add new symbionts to S^t)
        for all h_i ∈ H^t do
            for all p_k ∈ P^t do
                evaluate h_i on p_k
            end for
        end for
        remove points from P^t
        remove hosts from H^t (remove symbionts from S^t)
        t = t + 1
    end while
end procedure
```

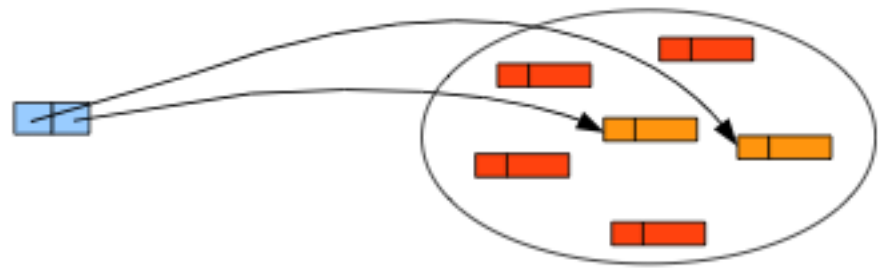
Asexual Variation Operators

Cannot assume genotypic context

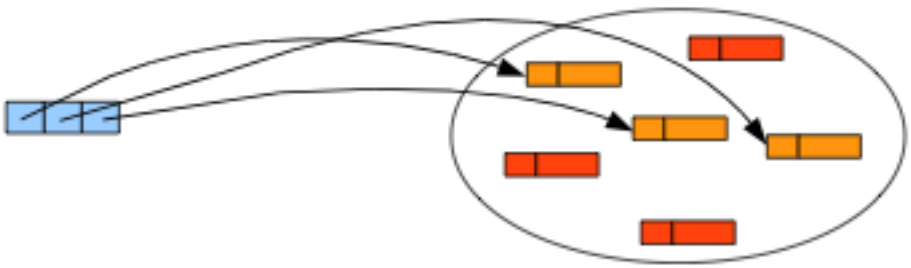
Step 0: Copy parent host



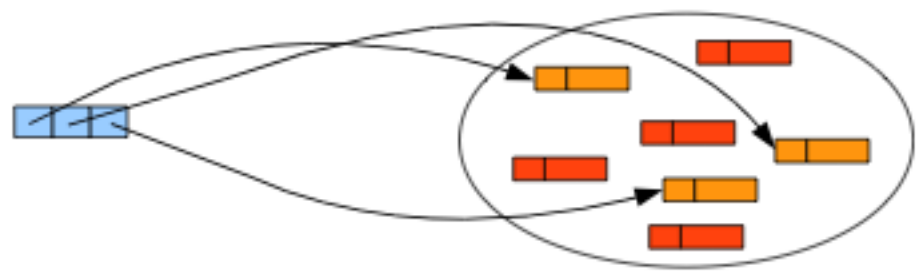
Step 1: Remove index(es)



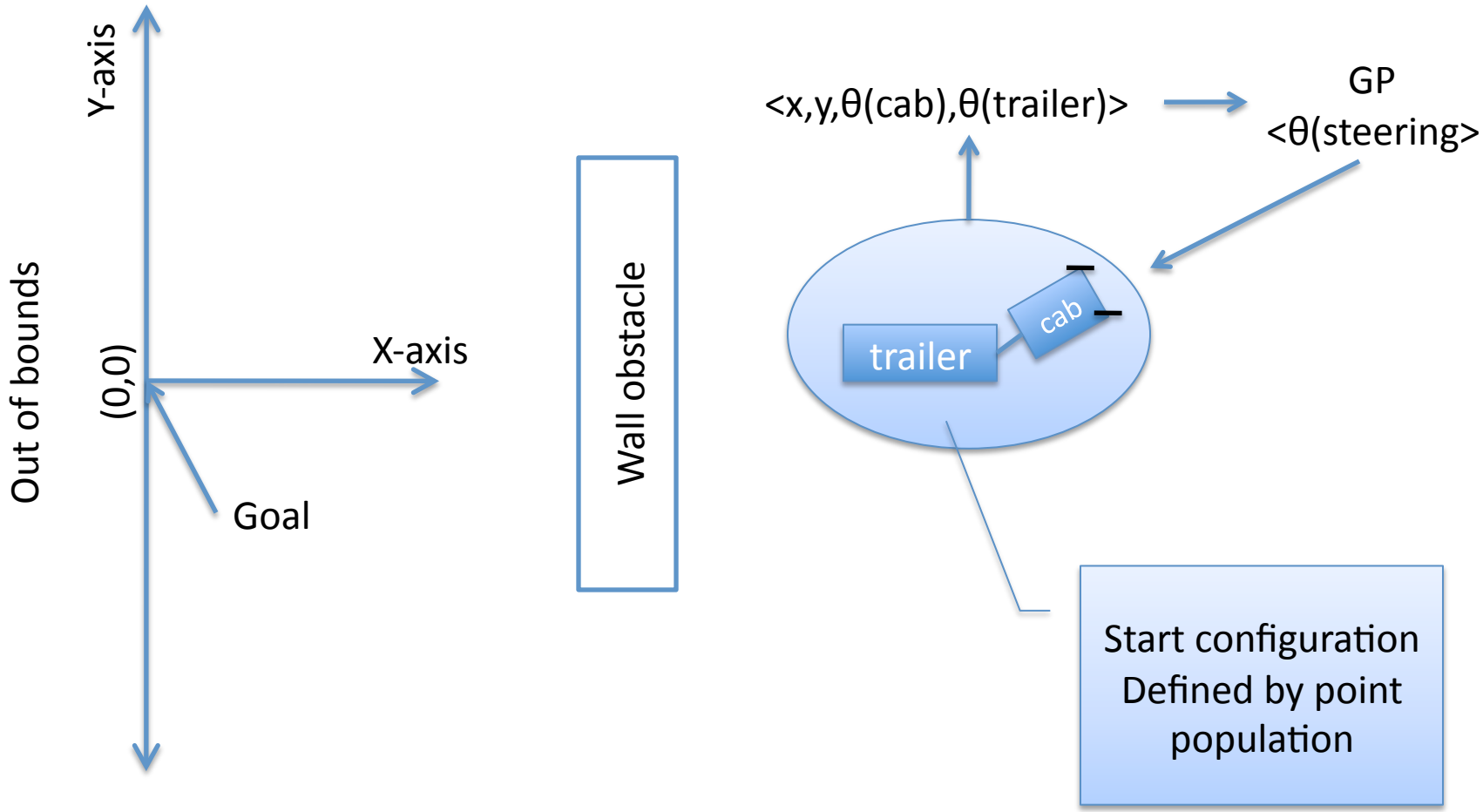
Step 2: Add index(es)



Step 3: Create new symbiont(s)

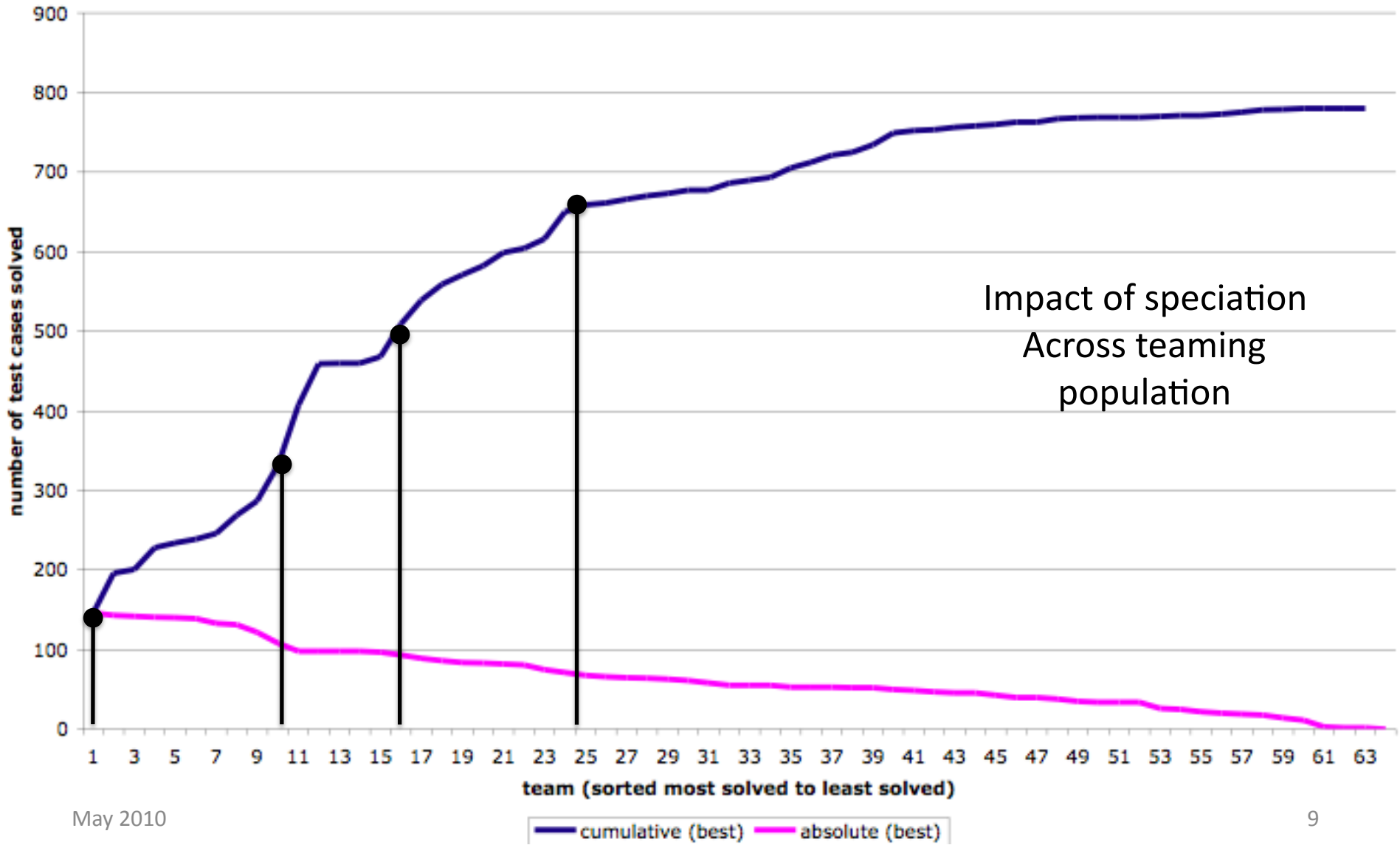


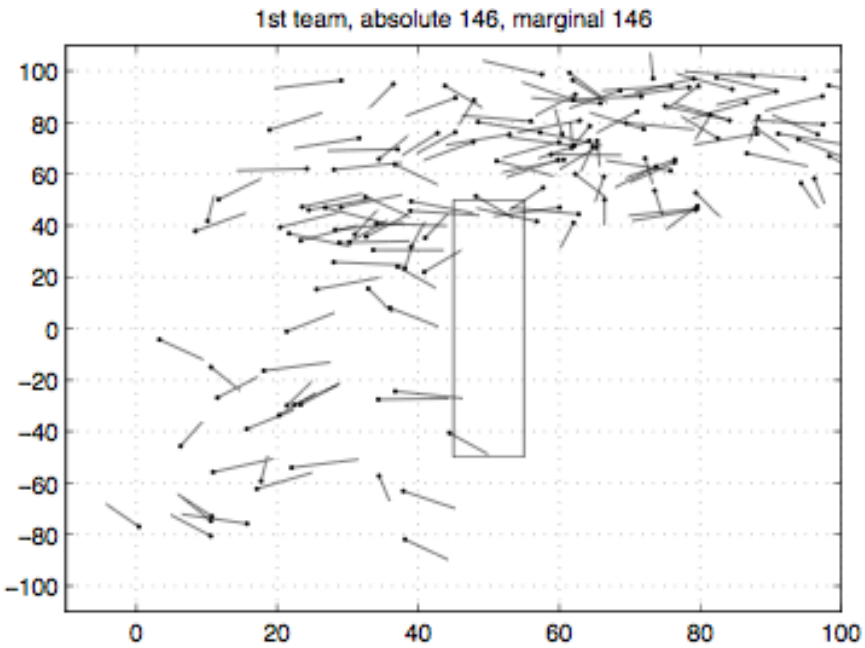
Hidden State Truck Backer-upper:
Lichodzijewski and Heywood (2009)

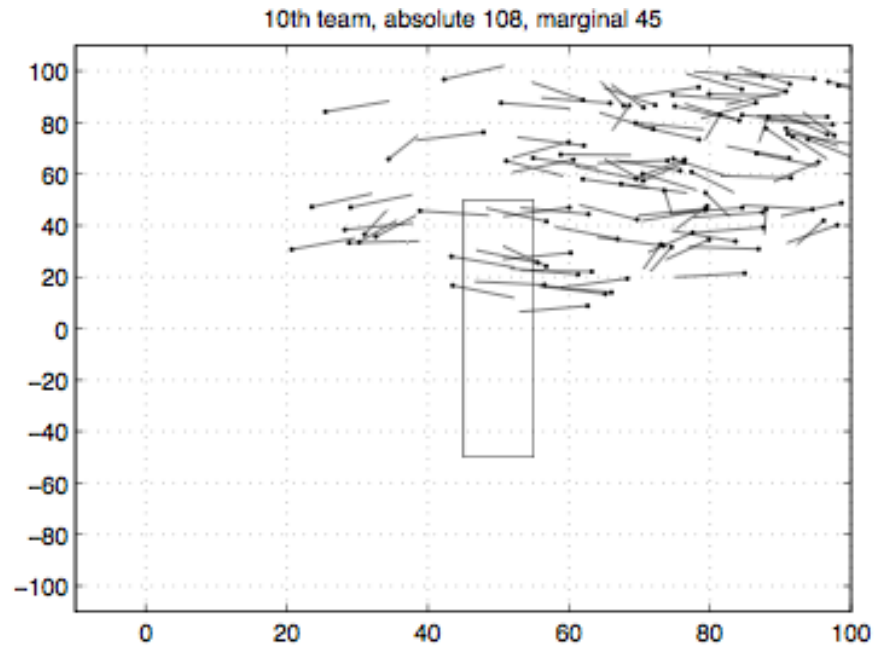


Hidden State Truck Backer-upper:

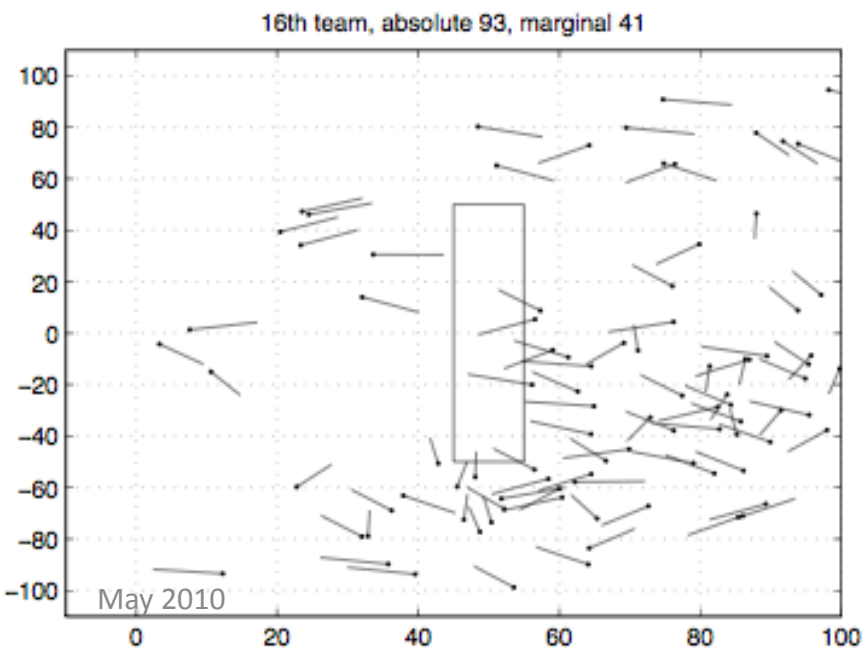
Case for layered learning

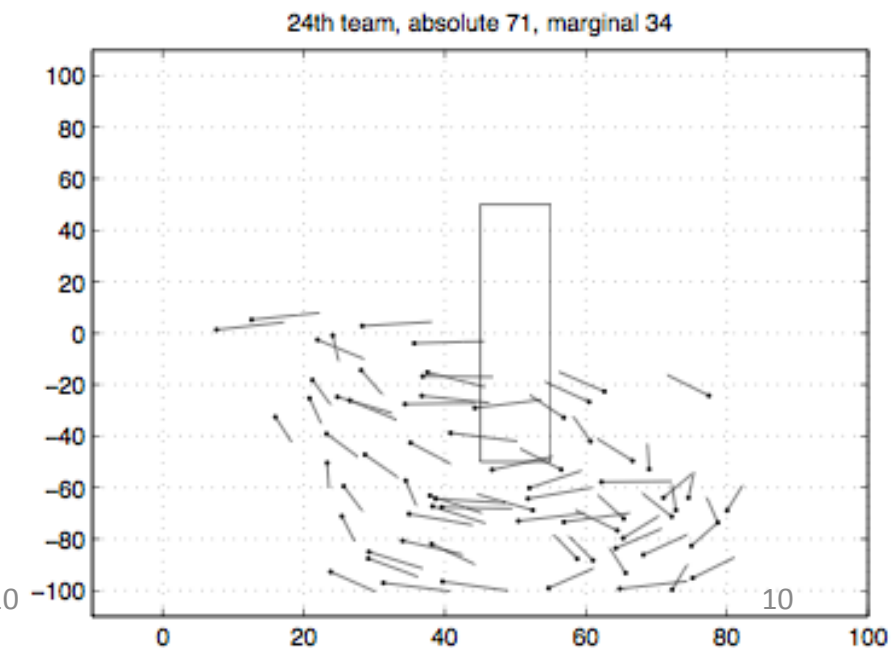






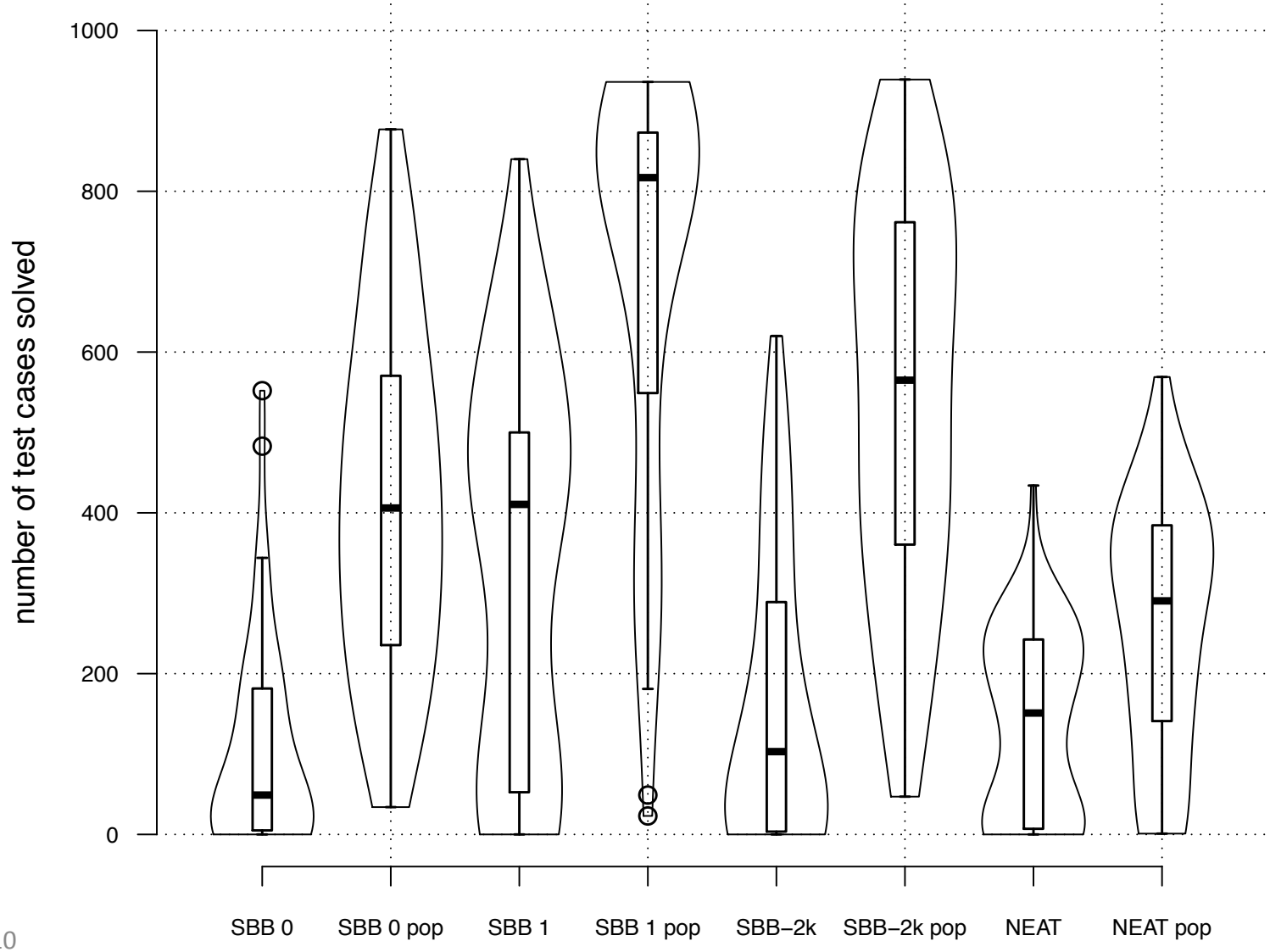
Case for layered Learning: Layer (i – 1) as actions for layer(i)



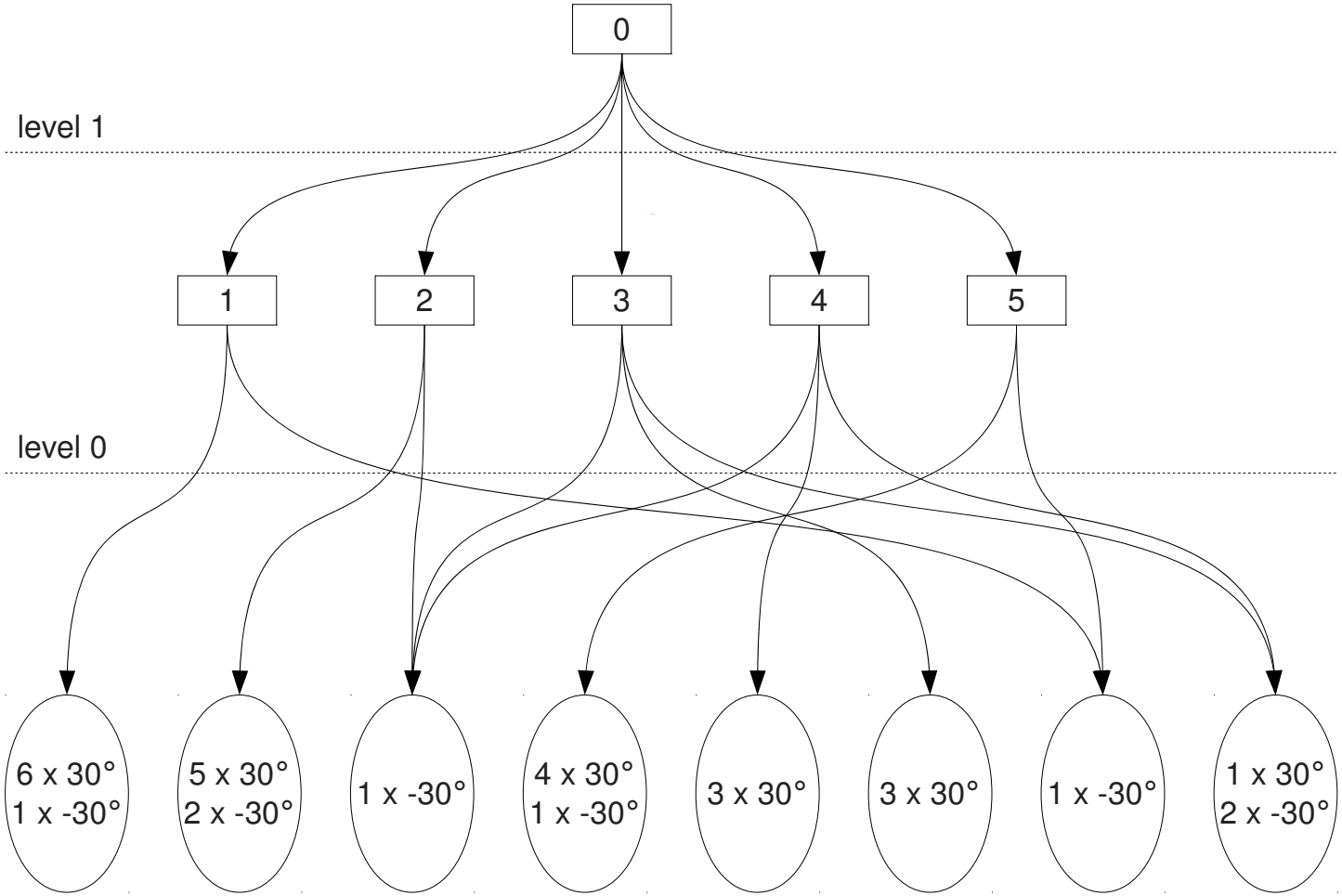


Truck backer-upper: Layered SBB

Lichodzijewski and Heywood (2010)

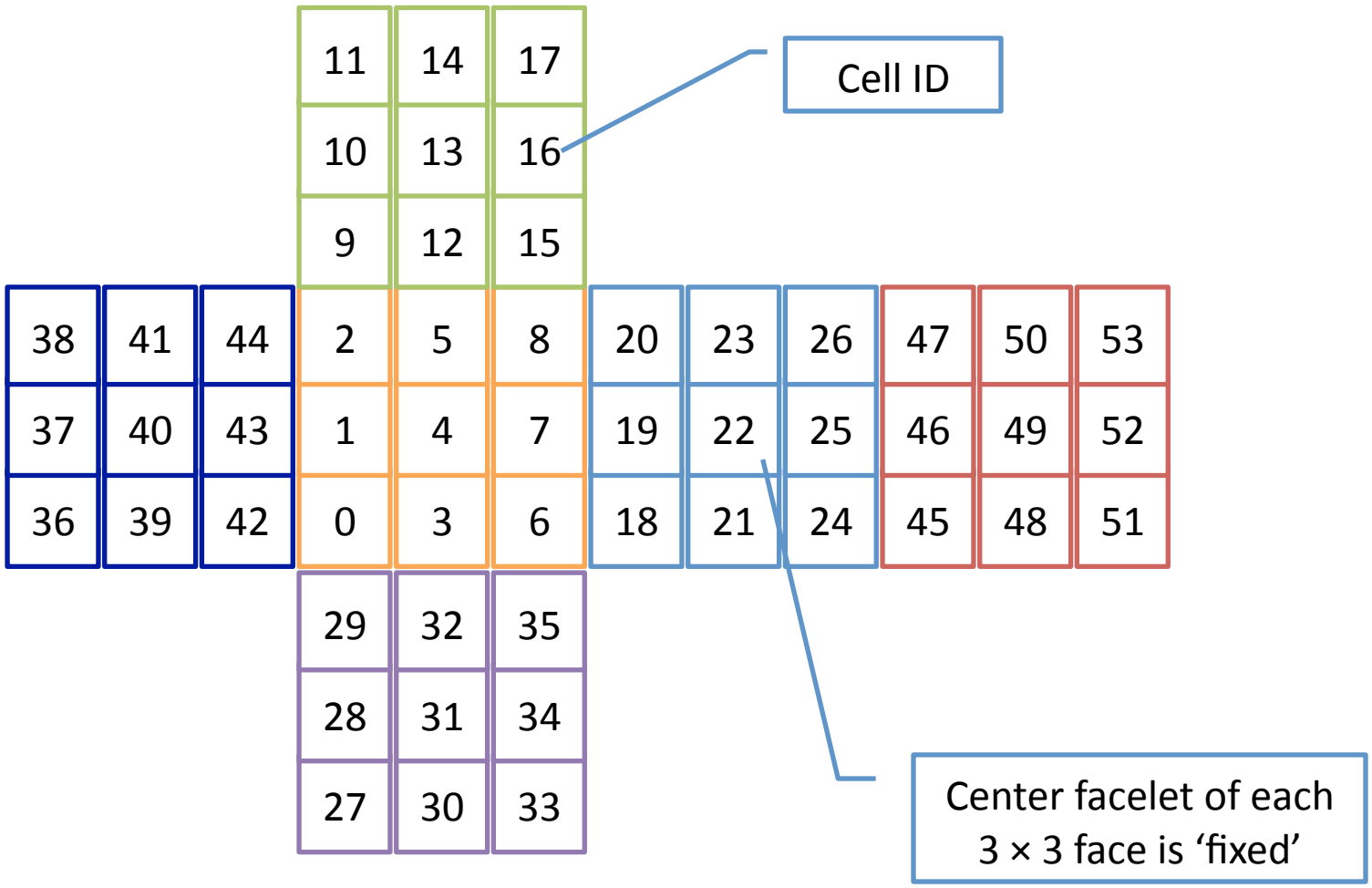


Truck backer-upper: Layered SBB



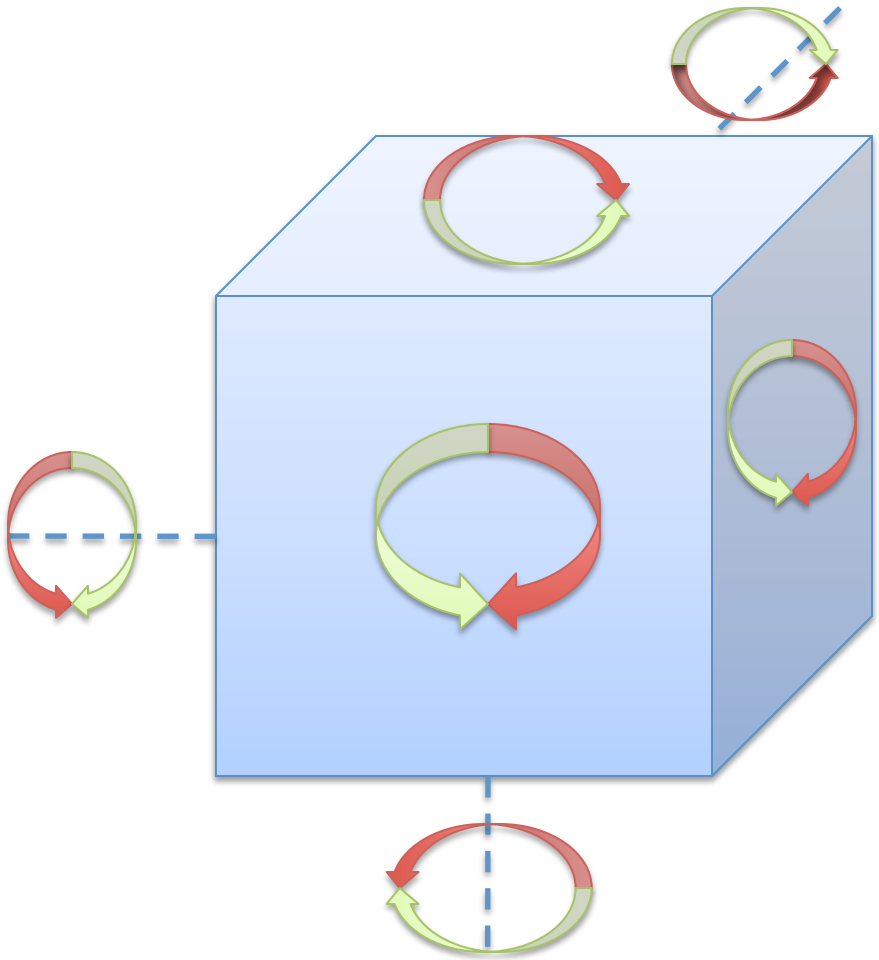
Rubik Cube:

Base Representation (54 cell vector)



Rubik Cube: Action Set

(12 actions: 6 clk-wise, 6 counter clk-wise)



Reward Functions:
Host

Host

- Competitive Fitness Sharing: $$s_i = \sum_{p_k} \left(\frac{G(h_i, p_k)}{\sum_{h_j} G(h_j, p_k)} \right)^3$$

- Reward function: $$G(h_i, p_k) = \frac{1}{(1 + D(s_f, s^*))^2}$$

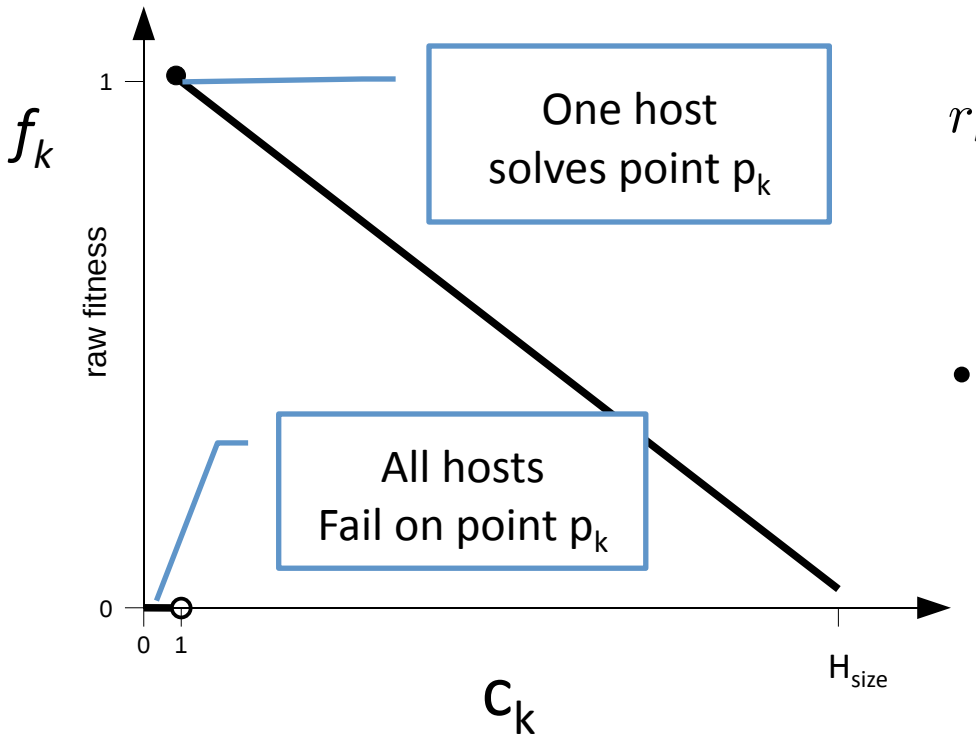
- Distance function: $D(s_2, s_1)$

$$\begin{cases} 0, & \text{when } 0 \text{ quarter twists match state } s_2 \text{ with } s_1 \\ 1, & \text{when } 1 \text{ quarter twists match state } s_2 \text{ with } s_1 \\ 4, & \text{when } 2 \text{ quarter twists match state } s_2 \text{ with } s_1 \\ 16, & \text{when } > 2 \text{ quarter twists match state } s_2 \text{ with } s_1 \end{cases}$$

Reward Functions:
Point

Global reward function

- 'Raw' fitness (f_k):



Local Reward function

- Point Reward factor:

$$r_k = \left(\frac{\sum_{p_l} (D(p_k, p_l))^2}{K} \right)^2$$

- Resulting Point fitness:

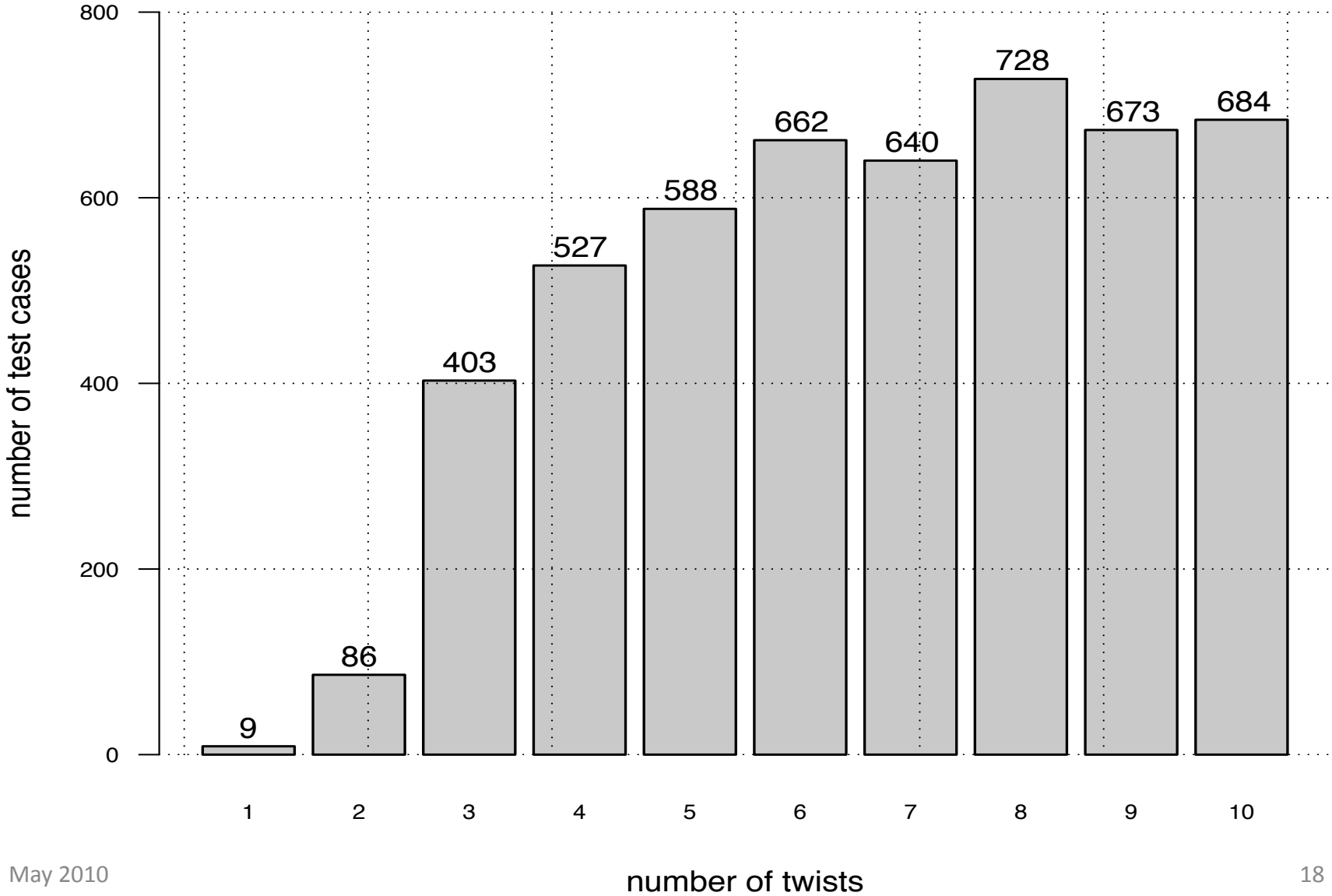
$$f'_k = f_k \cdot r_k$$

Evaluation Methodology

- Layered SBB
  - 2 layers
  - Max Generations
    - 1,000
  - `Default` Parameters
    - Max 24 symbiont/ host
    - Max 24 Instructions
    - Point / Host Pop Size:
      - 120
- Single Layer SBB
  - Max Generations: 2000
- `Big Prog`
  - Instruction limit: 36
- `Big Team`
  - Sym./ host Limit: 36

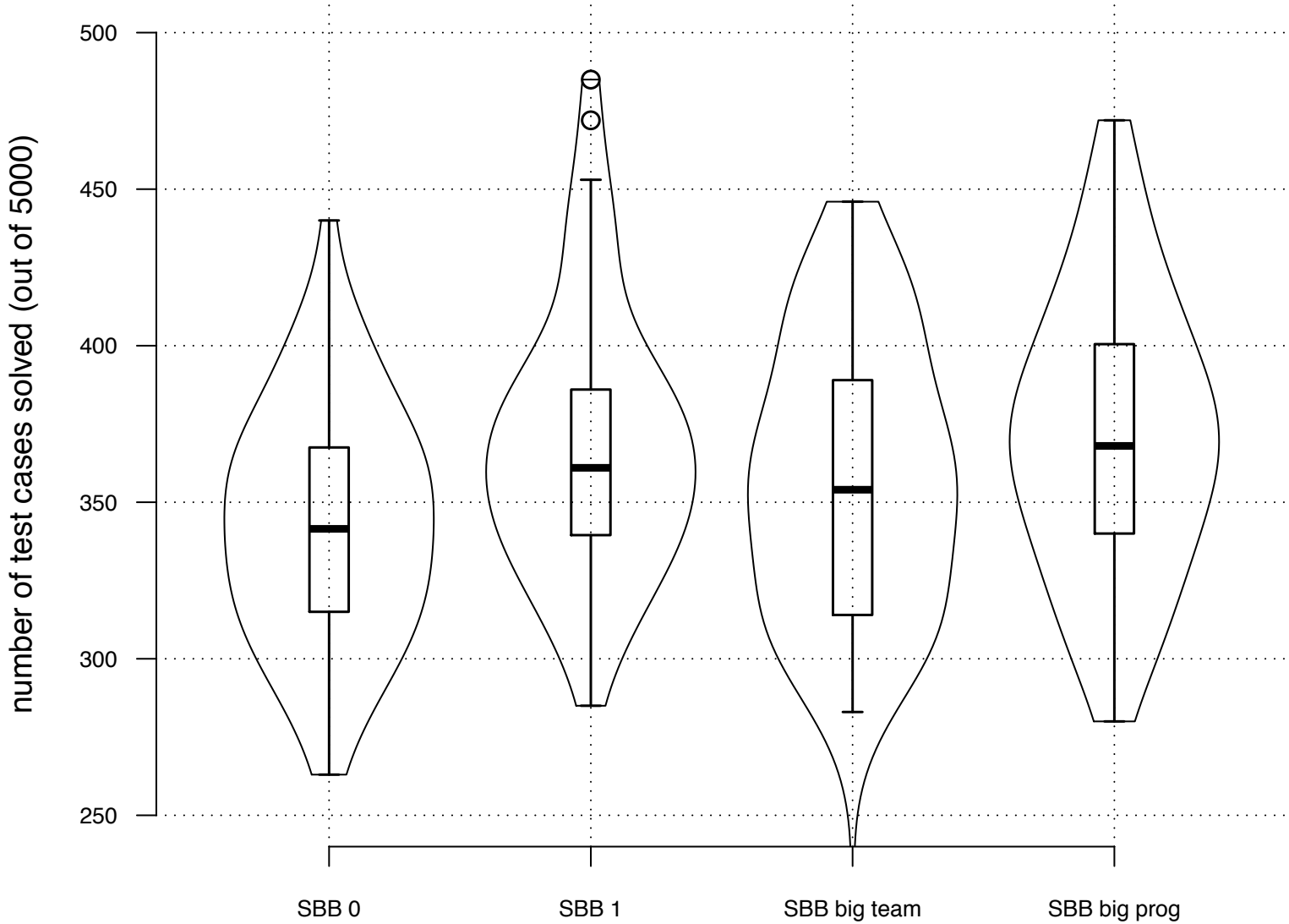
Post Training Test Cases: Test Set A

Sampled



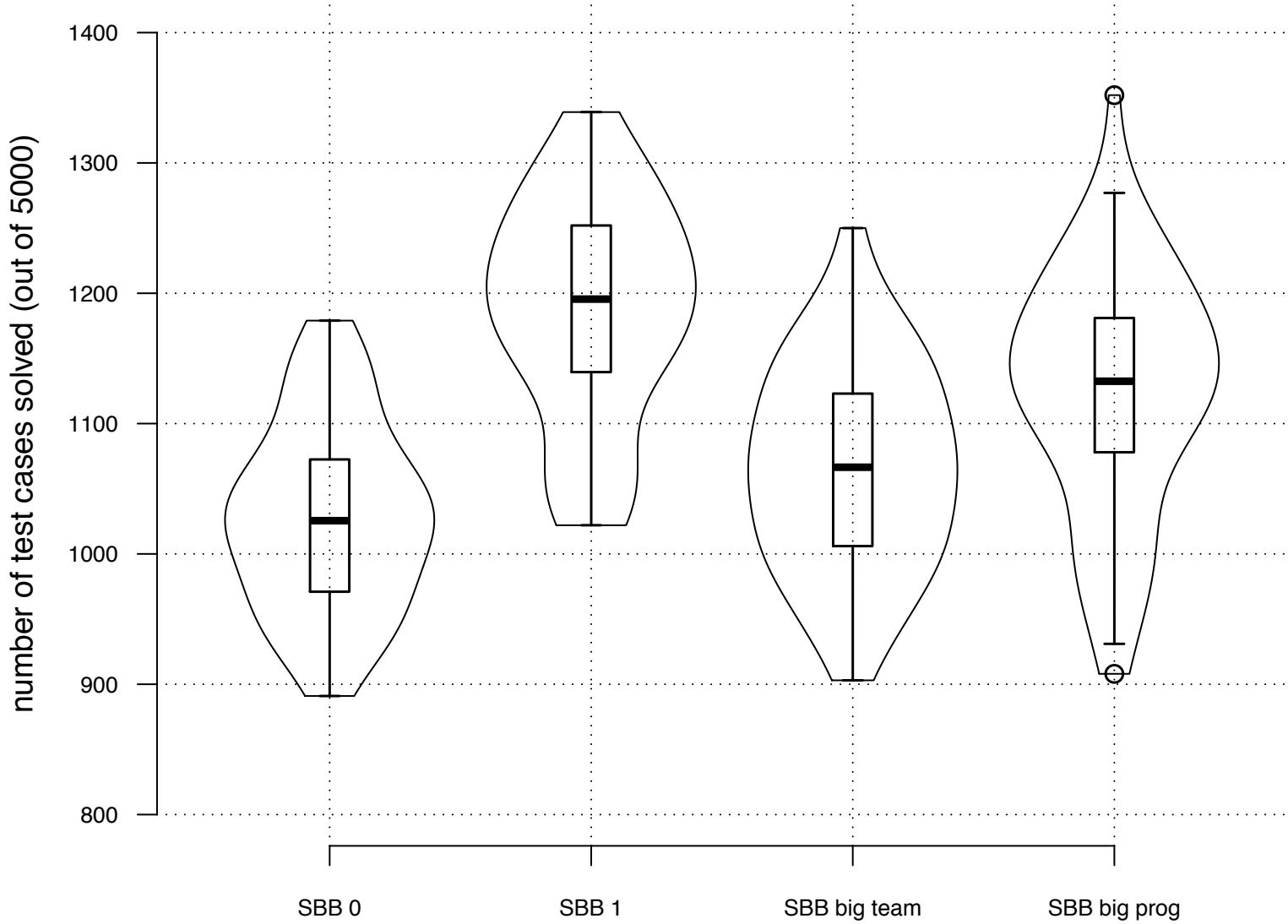
Test Performance: Test Set A

Single Champion



Test Performance: Test Set A

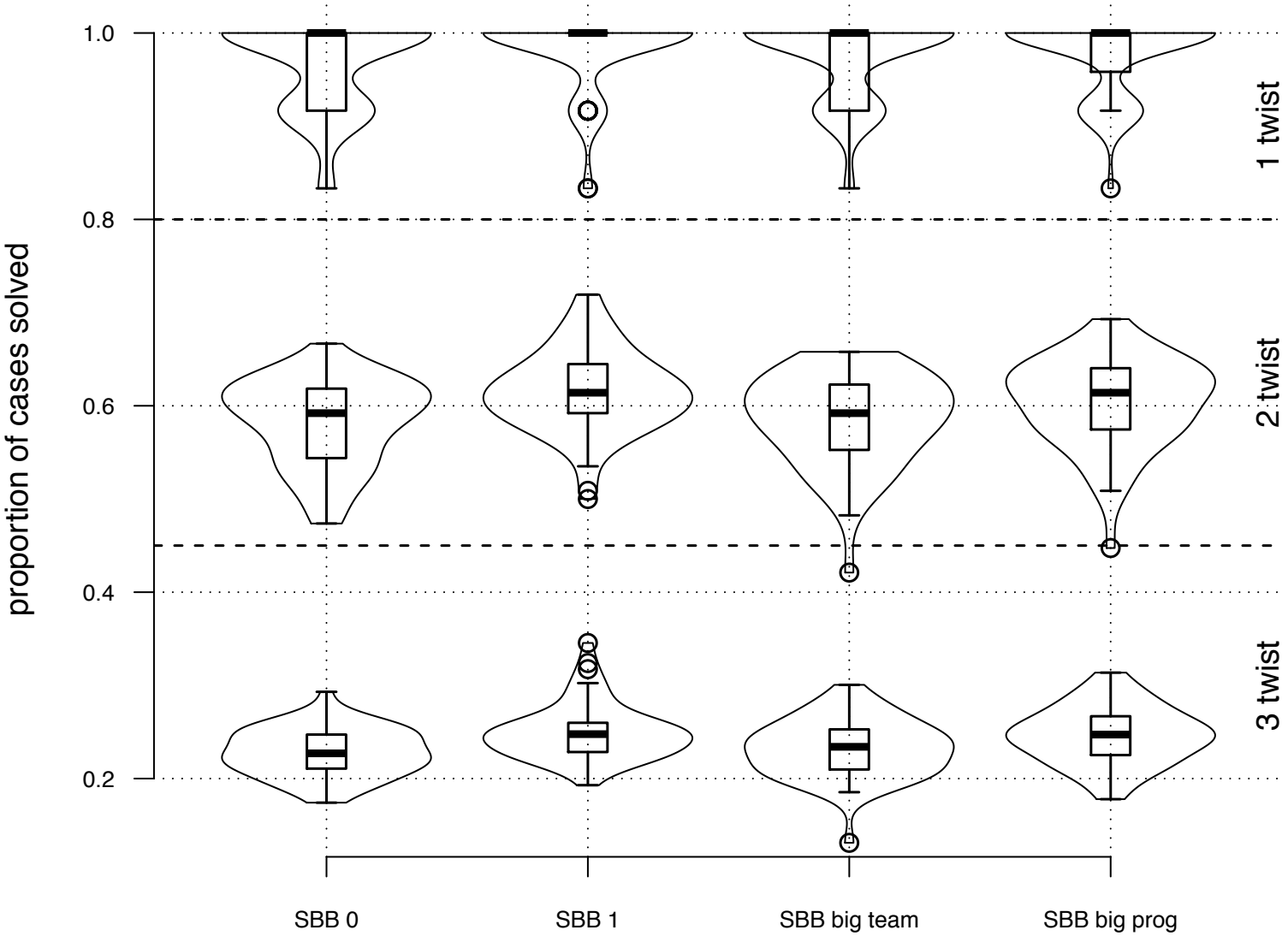
Population wide



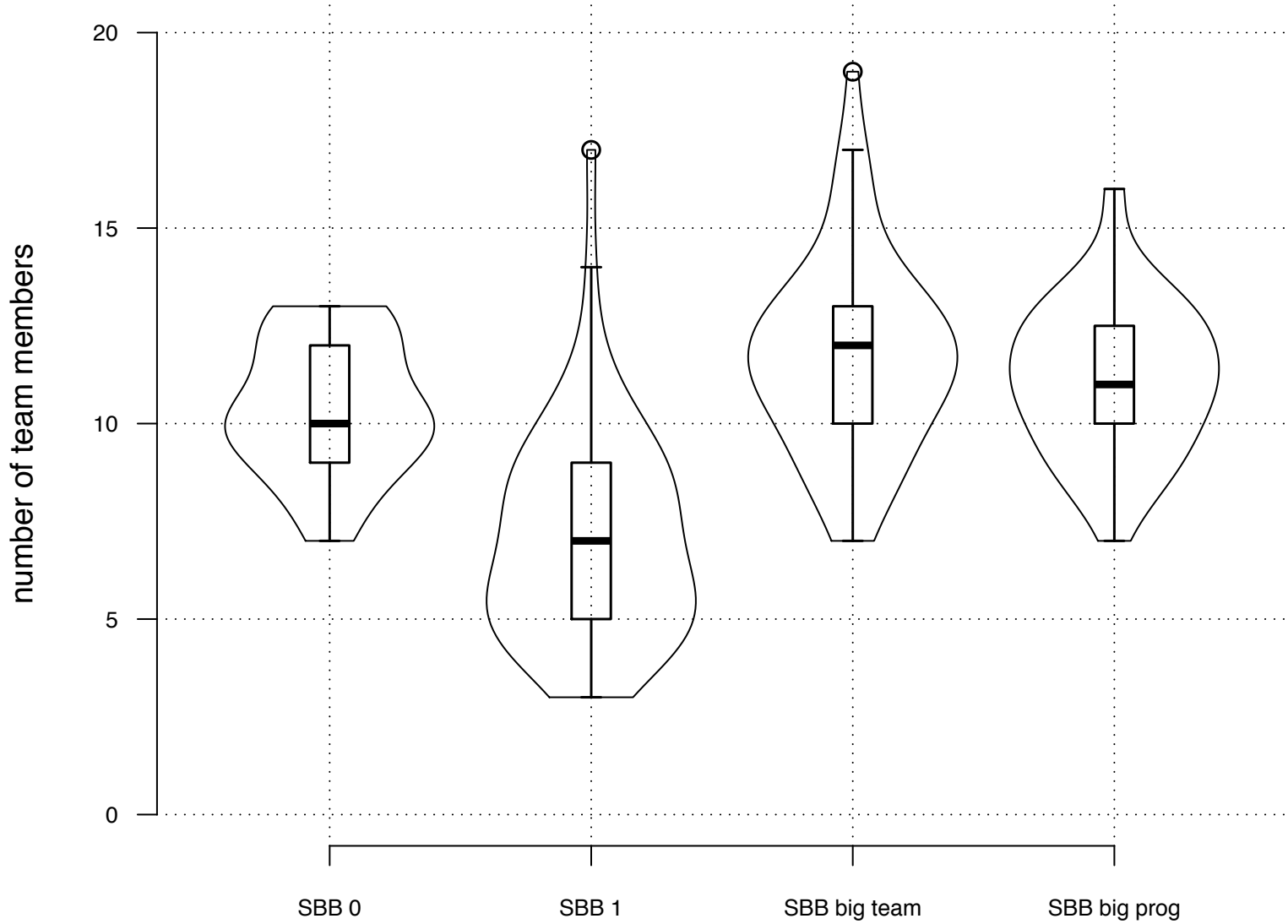
Test Performance: Test Set B
(Complete enumeration)

Number of Twists	Number of unique cube configurations
1	12
2	114
3	1 068

Test Performance: Test Set B

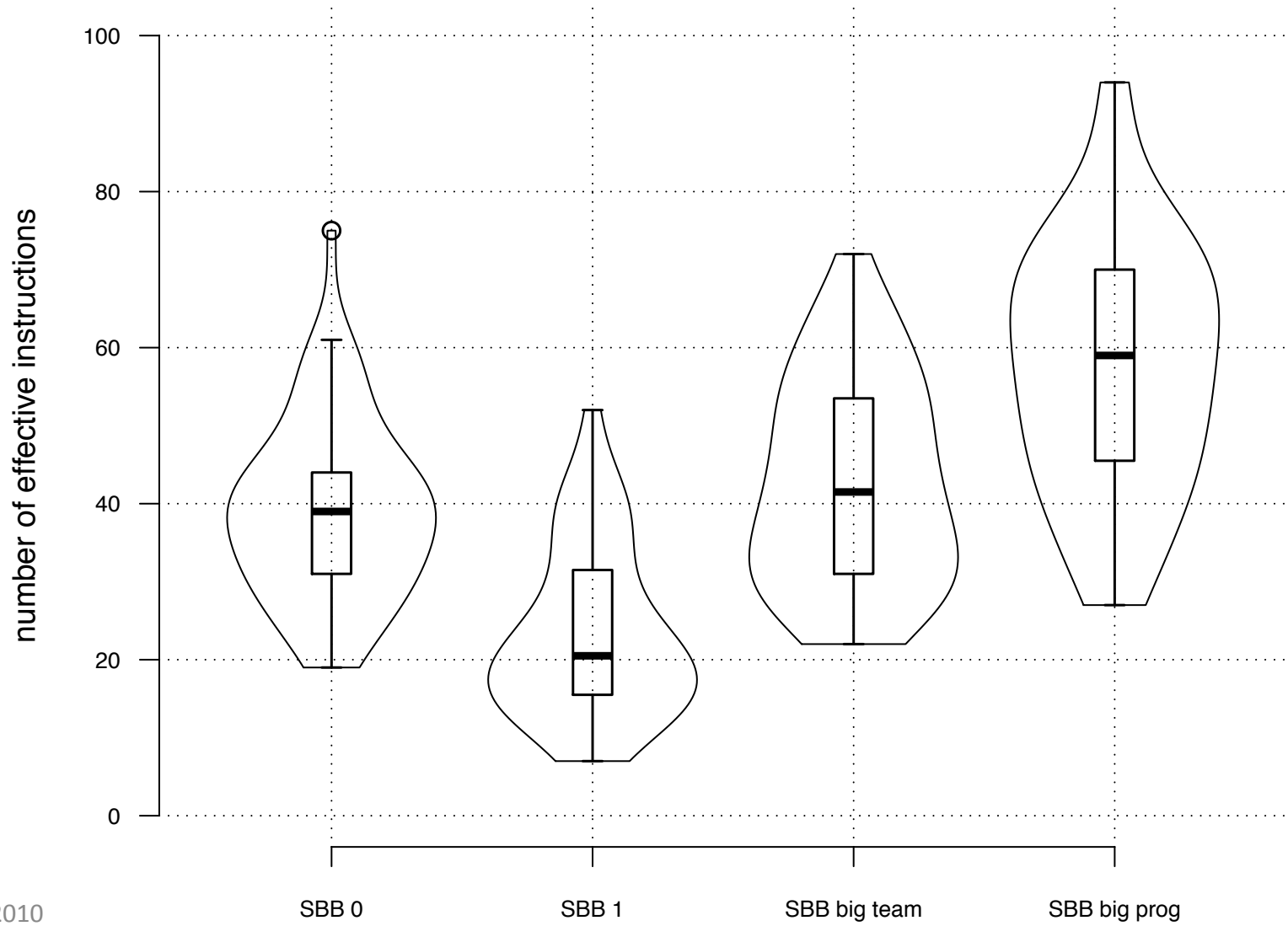


Solution Complexity:
Symbiont Count



Solution Complexity:

Instruction Count



Discussion (1)

- Rubik Cube domain
  - ‘Deceptive’ fitness function
  - Face independent representation
  - Layers potentially solve for different
    - Cube faces
    - Objectives
- Truck backer-upper
  - Informative fitness function
  - Point to Host interaction
    - Random sampling sufficient

Discussion (2)

- SBB implies
  - Hosts:
    - Combinatorial search for relevant symbionts
    - Enforce Inter host competition: fitness sharing
  - Symbionts:
    - No direct fitness evaluation
    - Population size `floats`
    - Bid-based GP
      - Context and communication
  - Points:
    - Diversity of training scenarios
      - Major source of evolutionary innovation
    - Point fitness biases replacement
- Layered Learning
  - Hosts.layer(m – 1) ➔ Symbiont actions.layer(m)

Acknowledgements

- NSERC
- MITACS
- Killam Scholarship program