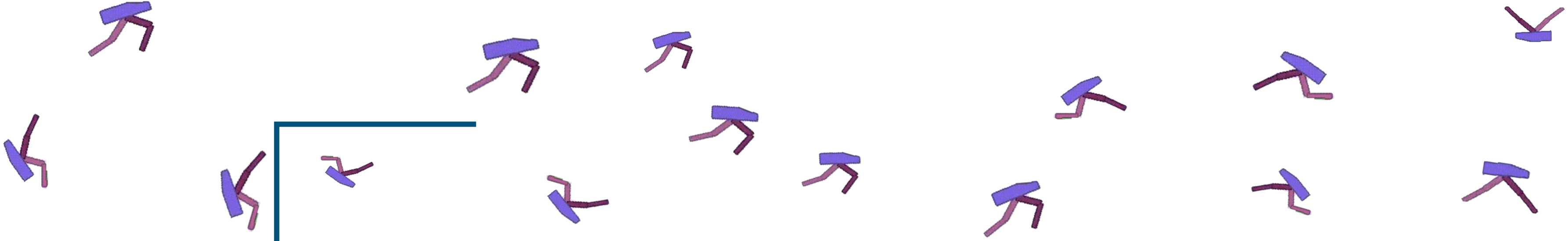


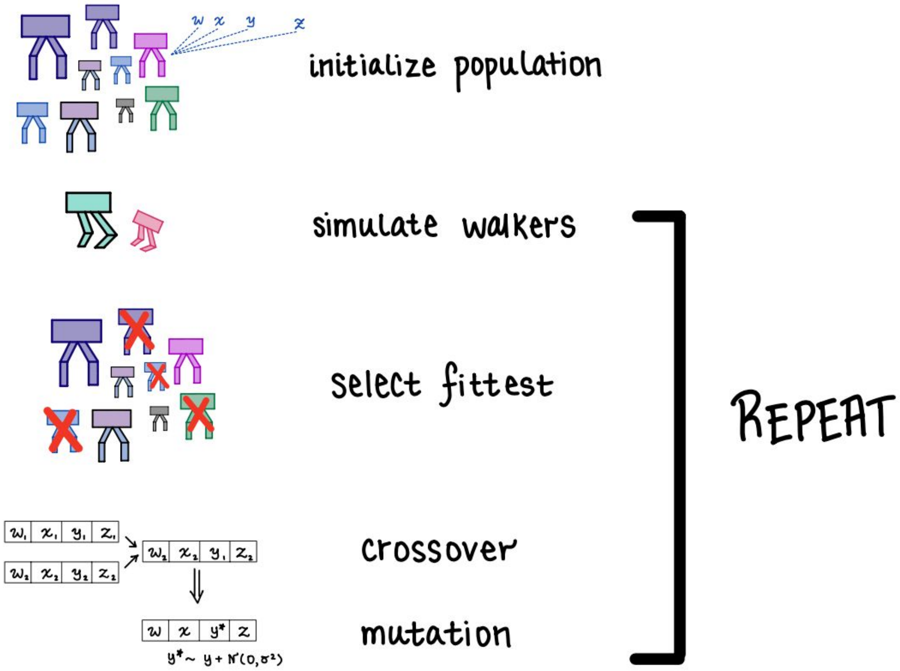
Teaching an Agent How to Walk 1.0

May Soshi, Ricky Williams, Omar Abdel Haq, Elie Eshoa

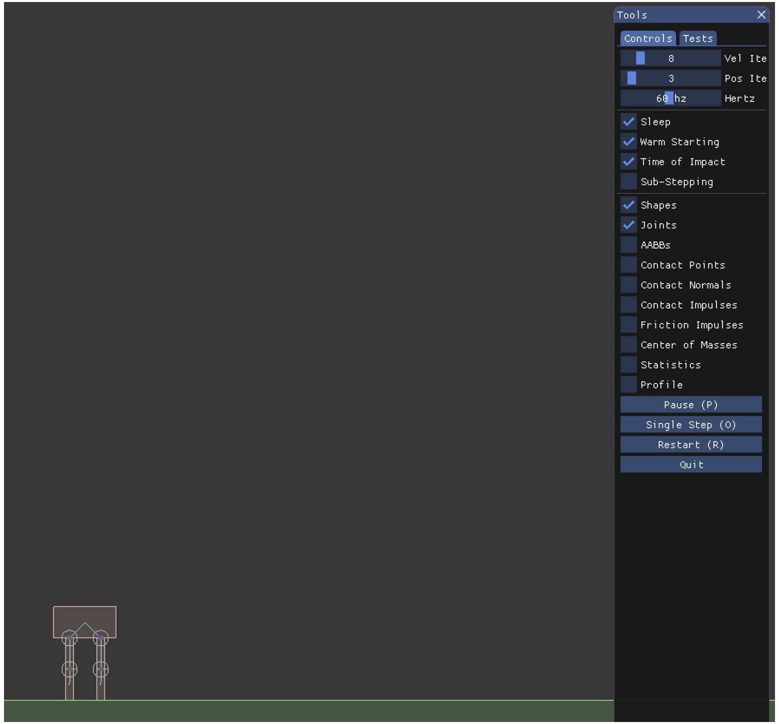
Sequential Baseline

Algorithm

1. Create a population
2. Simulate walkers
3. Select the fittest walkers
4. Cross and mutation of the fittest walkers
5. Repeat steps 2-4



Initial Visualization



Tools

Controls | Tests

8 Vel Ite

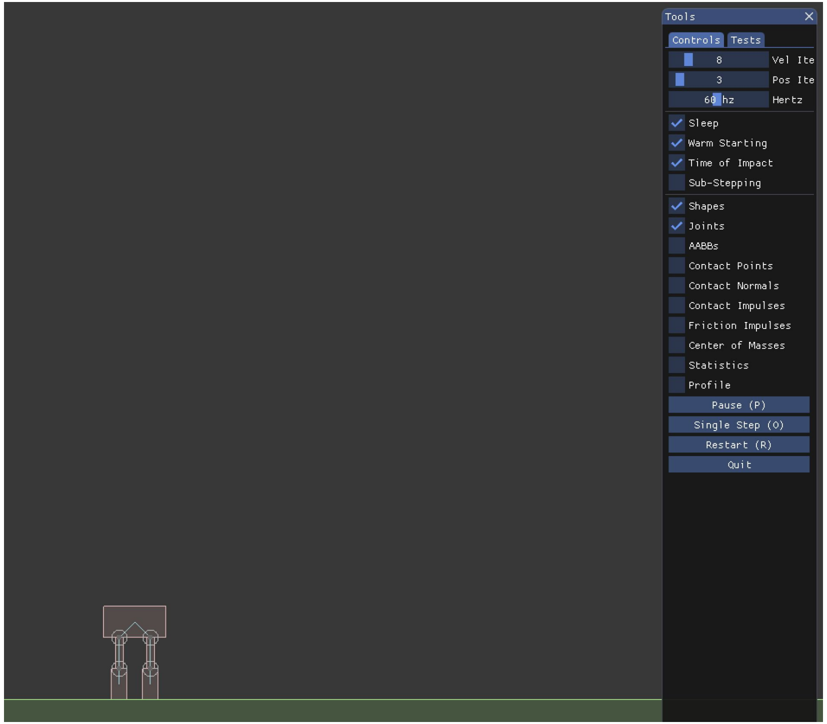
3 Pos Ite

60 hz Hertz

☑ Sleep
☑ Warm Starting
☑ Time of Impact
□ Sub-Stepping

☑ Shapes
☑ Joints
□ AABBs
□ Contact Points
□ Contact Normals
□ Contact Impulses
□ Friction Impulses
□ Center of Masses
□ Statistics
□ Profile

Pause (P)
Single Step (O)
Restart (R)
Quit



Tools

Controls | Tests

8 Vel Ite

3 Pos Ite

60 hz Hertz

☑ Sleep
☑ Warm Starting
☑ Time of Impact
□ Sub-Stepping

☑ Shapes
☑ Joints
□ AABBs
□ Contact Points
□ Contact Normals
□ Contact Impulses
□ Friction Impulses
□ Center of Masses
□ Statistics
□ Profile

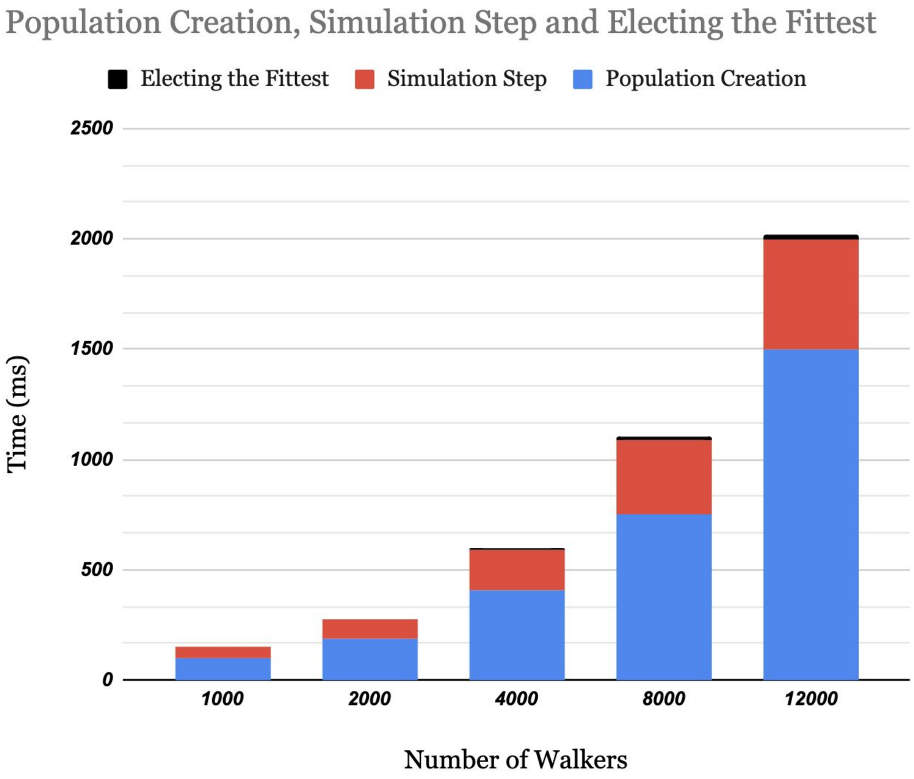
Pause (P)
Single Step (O)
Restart (R)
Quit

Bottlenecks

Bottlenecks after profiling:

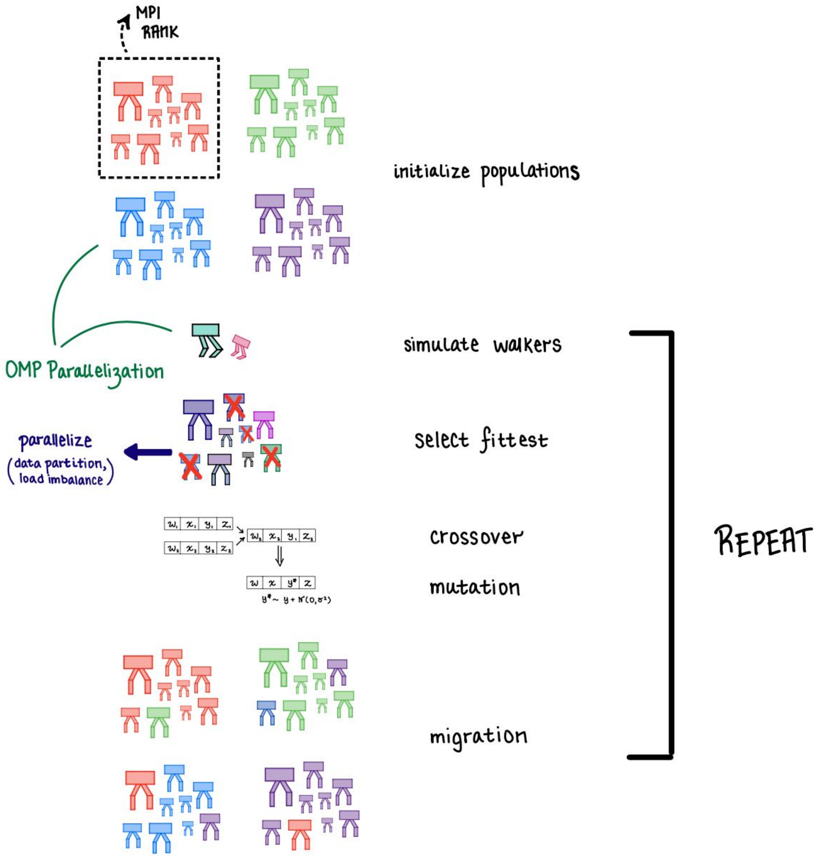
1. **Creating a new population** is the most expensive step
2. **Simulating a step** for each of the walkers
3. **Memory constraints** on sequential implementation. Opportunities for scaling and parallelizing (Cache locality and exploiting multiple processes can be explored).

The algorithm is synchronization- and not compartmentalized computation-heavy



Parallelism

- Create population with MPI
  - Each rank takes a subset of the population to be created
  - Every rank simulates the walkers of the subpopulation
  - Reports the fitnesses and the torque
- Simulation
  - Pass walkers to Box2D environment, calculate distance travelled
- Select the fittest walkers
  - Sorting algorithm
- Migration
  - Randomly select 5% of the walkers from population A → population B
  - Communication between ranks



Migration – Student Visas