Using Multi-Objective Quality Diversity to Evolve Complex Machines in Minecraft

SOUTHWESTERN UNIVERSITY

Joanna Lewis & Travis Rafferty

Introduction

Recent work on generating flying machines in Minecraft using Evolutionary Computation succeeded at producing small flying machines [1]. Larger flying machines and more complex machines require more sophisticated techniques, such as multiobjective optimization via Multi-Objective MAP Elites (MOME [3]). MOME encourages more diverse solutions like the quality diversity algorithm MAP Elites [2], but also prevents the loss of solutions that may only excel in one objective rather than several. This technique has generated promising results in Minecraft.

MAP-Elites and MOME

Both use an archive based on “behavioral” niches (block count or piston counts along various axes) which distinguish solutions.

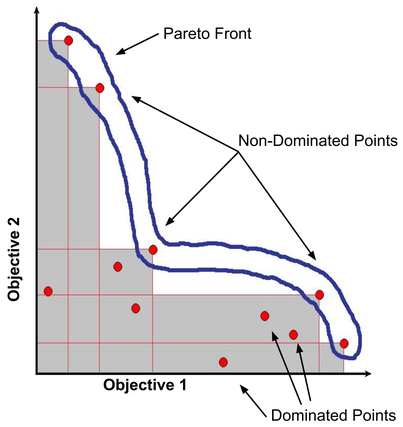
MAP-Elites (Multidimensional Archive of Phenotypic Elites) [2]

- uses a single calculation to measure fitness/quality
- only one individual per bin (only keep the best)

MOME (Multi-Objective MAP Elites) [3]

- uses multiple objectives to measure quality
- more than one individual in a bin: non-dominated solutions

Pareto Optimality

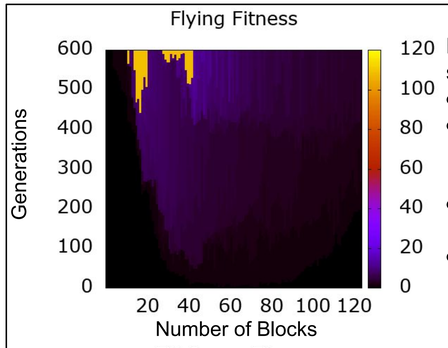


A point dominates another if it is greater in one objective and at least as good in others. Dominated points are in rectangles projected from points that dominate them.

Pareto Front: points not dominated by any others (non-dominated). Circled in blue in figure to left.

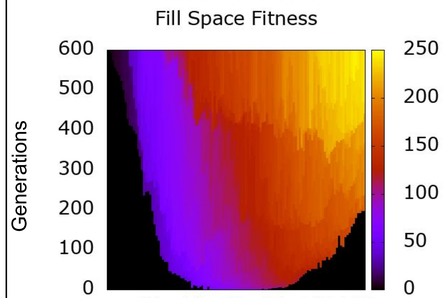
Hypervolume: size of dominated area under non-dominated points, which is the gray area to the left.

Large Flying Machines



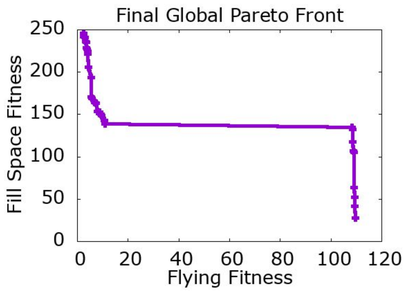
Heatmap of maximum flying score for shapes in each bin.

- Flying results in bright yellow
- Purple means the shape cannot fly, though brighter shades indicate oscillation
- x-axis: number of blocks in shape (behavioral niche)
- y-axis: 100 shapes per generation, bottom to top



Heatmap of maximum space score for shapes in each bin.

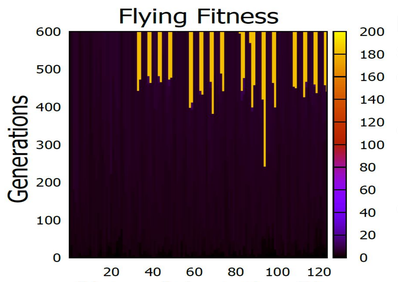
- Shapes whose blocks occupy more locations have brighter colors
- x-axis is still the number of blocks, which naturally fill space more easily
- y-axis: as above. Note that large shapes emerge early.



Non-dominated Pareto front for evolving large flying machines.

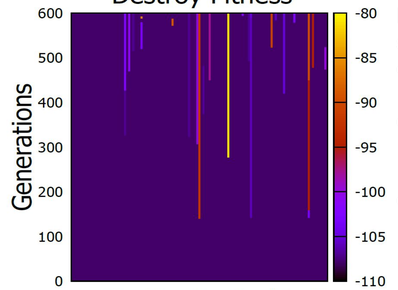
- x-axis: Flying fitness, with a large jump differentiating flying machines (over 100) from those that do not.
- y-axis: Space filling fitness.
- Top left: large shapes that expand across a lot of space, but do not fly.
- Bottom right: Smaller shapes that fly

Flying Missiles



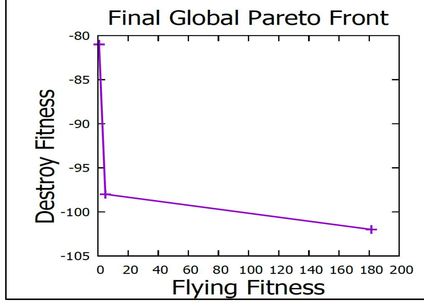
Heatmap of maximum flying score for shapes in each bin.

- Flying results in bright yellow, black and dark purple cannot fly.
- x-axis: one bin for each possible combination of pistons oriented along different directions
- y-axis: 100 shapes per generation, bottom to top



Heatmap of destroyed blocks fitness for shapes in each bin.

- Shapes that destroyed more blocks have brighter colors
- x-axis: one bin for each possible combination of pistons oriented along different directions
- y-axis: As above. Shapes that damage targets are launching TNT blocks, not flying.



Non-dominated Pareto front for evolving missiles.

- x-axis: Flying fitness, with a large jump differentiating flying machines (over 100) from those that do not.
- y-axis: Destroy fitness.
- Top left: shapes that destroy more blocks, but do not fly.
- Bottom right: shapes that fly, but do not destroy many, if any, blocks
- No shape both flies and destroys a significant number of target blocks.

Quality and Fitness Functions

Three fitness functions are utilized.

Flying Fitness:

- Encourages movement
- Extra reward if moves far enough

Space Filling Fitness:

- Encourages blocks to visit many locations

Destroy Fitness:

- Rewards based on the amount of target destroyed

Each experiment uses flying fitness and either space filling fitness or destroy fitness (see above).

References

[1] A. Medina, M. Richey, M. Mueller, and J. Schrum. Evolving Flying Machines in Minecraft Using Quality Diversity. Genetic and Evolutionary Computation Conference. 2023

[2] J. Mouret, J. Clune: Illuminating search spaces by mapping elites. CoRR abs/1504.04909. 2015

[3] T. Pierrot, G. Richard, K. Beguir, and A. Cully. Multi-objective quality diversity optimization. Genetic and Evolutionary Computation Conference. 2022

Conclusion

Successes:

- Produced slightly larger flying machines
- One large shape launches two small flying machines in different directions
- Flying machines can carry TNT

Current issues:

- Flying machines would leave TNT bombs behind
- Machines would "launch" TNT at a target to destroy it
- No explosions on impact.