

Lecture 2:

Search-based PCG

Procedural Content Generation 2013

Julian Togelius

What are the problems?

- *Speed*
Real-time? Or design-time?
- *Reliability*
Catastrophic failures break gameplay
- *Controllability*
Allow specification of constraints and goals
- *Diversity*
Content looks like variations on a theme
- *Creativity*
Content looks “computer-generated”

A taxonomy of PCG

- Online/Offline
- Necessary/Optional
- Random seeds/Parameter vectors
- Stochastic/Deterministic
- Constructive/Generate-and-test

Online/Offline

- Online: as the game is being played
- Offline: during development of the game

Necessary/Optional

- Necessary content: content the player needs to pass in order to progress
- Optional content: can be discarded, or bypassed, or exchanged for something else

Stochastic/ Deterministic

- Deterministic: given the same starting conditions, always creates the same content
- Stochastic: the above is not the case

Random seeds/ Parameter vectors

- a.k.a. dimensions of control
- Can we specify the shape of the content in some meaningful way?

Constructive/ Generate-and-test

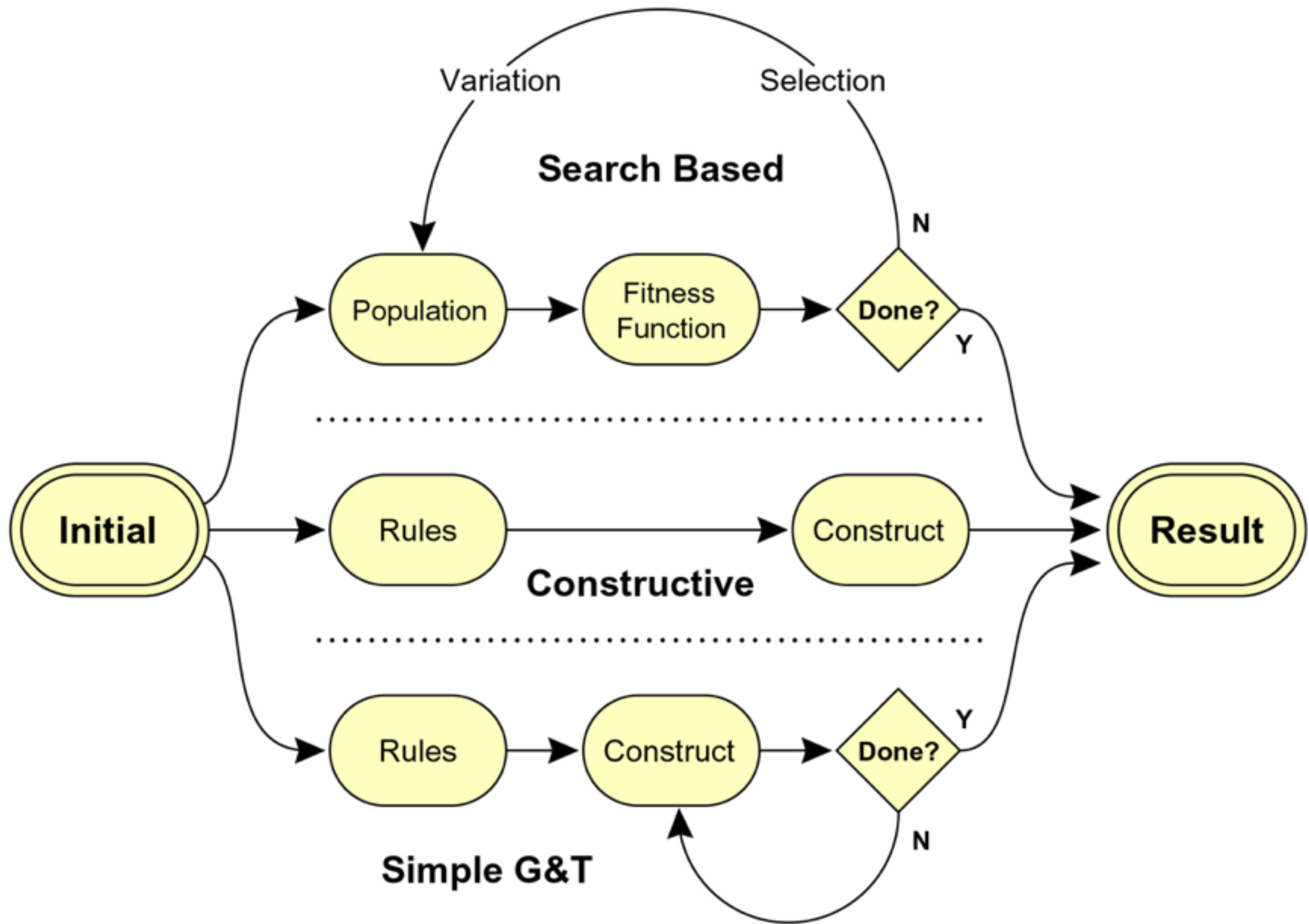
- Constructive: generate the content once and be done with it
- Generate-and-test: generate, test for quality, and re-generate until the content is good enough

The Search-based Paradigm

- A special case of generate-and-test:
 - The test function returns a numeric fitness value (not just accept/reject)
 - The fitness value guides the generation of new candidate content items
- Usually implemented through evolutionary computation

Evolutionary computation?

- Keep a population of candidates
- Measure the fitness of each candidate
- Remove the worst candidates
- Replace with copies of the best (least bad) candidates
- Mutate/crossover the copies



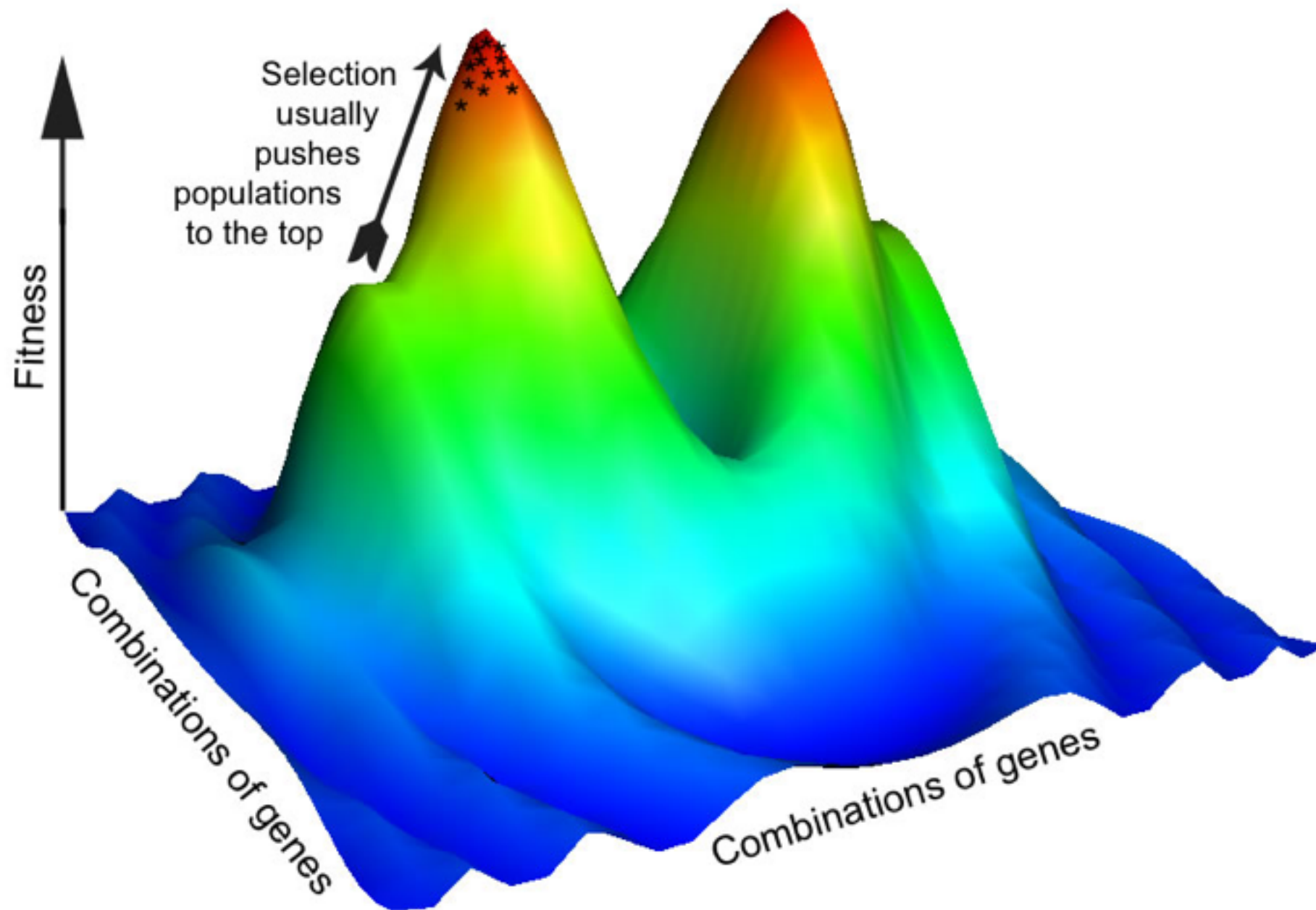
And of course, the algorithm!

- Lots of different types of evolutionary algorithms: Genetic Algorithms, Evolution Strategies, Evolutionary Programming
- And evolution-like algorithms: Particle Swarm Optimisation, Differential Evolution
- Keep It Simple, Stupid!
 - Often, simple $\mu+\lambda$ ES with no crossover and no self-adaptation works well enough

Simple $\mu+\lambda$ ES

- Create a population of $\mu+\lambda$ individuals
- Each generation
 - Evaluate all individuals in the population
 - Sort by fitness
 - Remove the worst λ individuals
 - Replace with mutated copies of the μ best

The fitness landscape



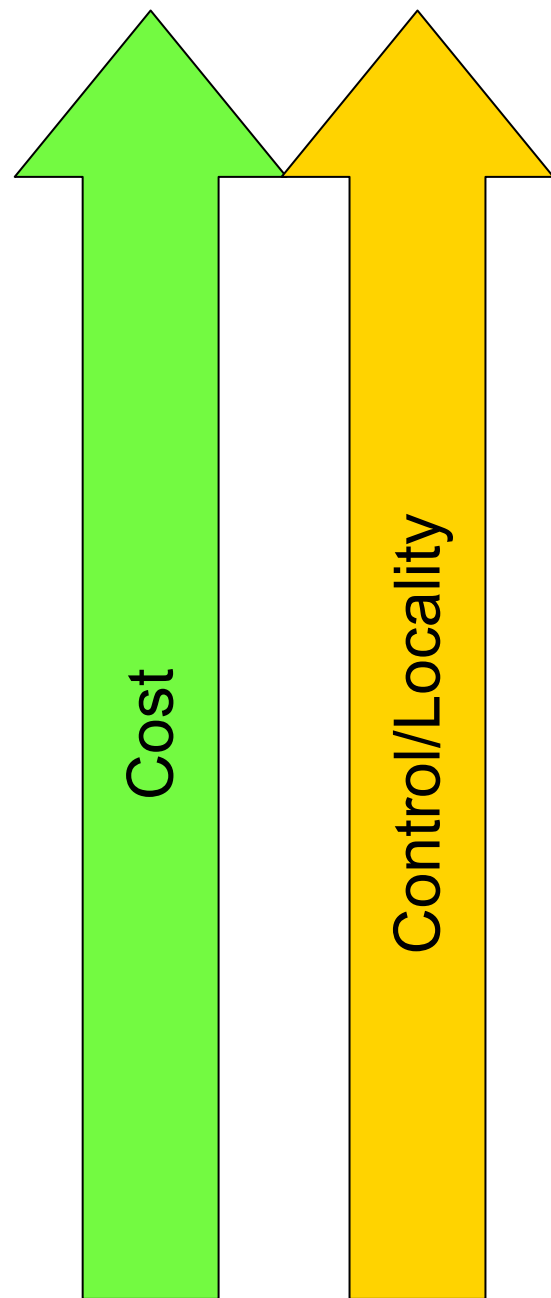
Locality

- The extent to which magnitude of changes in fitness correlate with magnitude of changes in genotype
- A good thing!
- Avoid: all small changes cause catastrophic fitness drops
- Depends on representation and fitness function

Issues in search-based PCG

- Content representation and search space
 - Direct or indirect?
- Fitness function
 - Direct, simulation-based, interactive?

Representing a dungeon



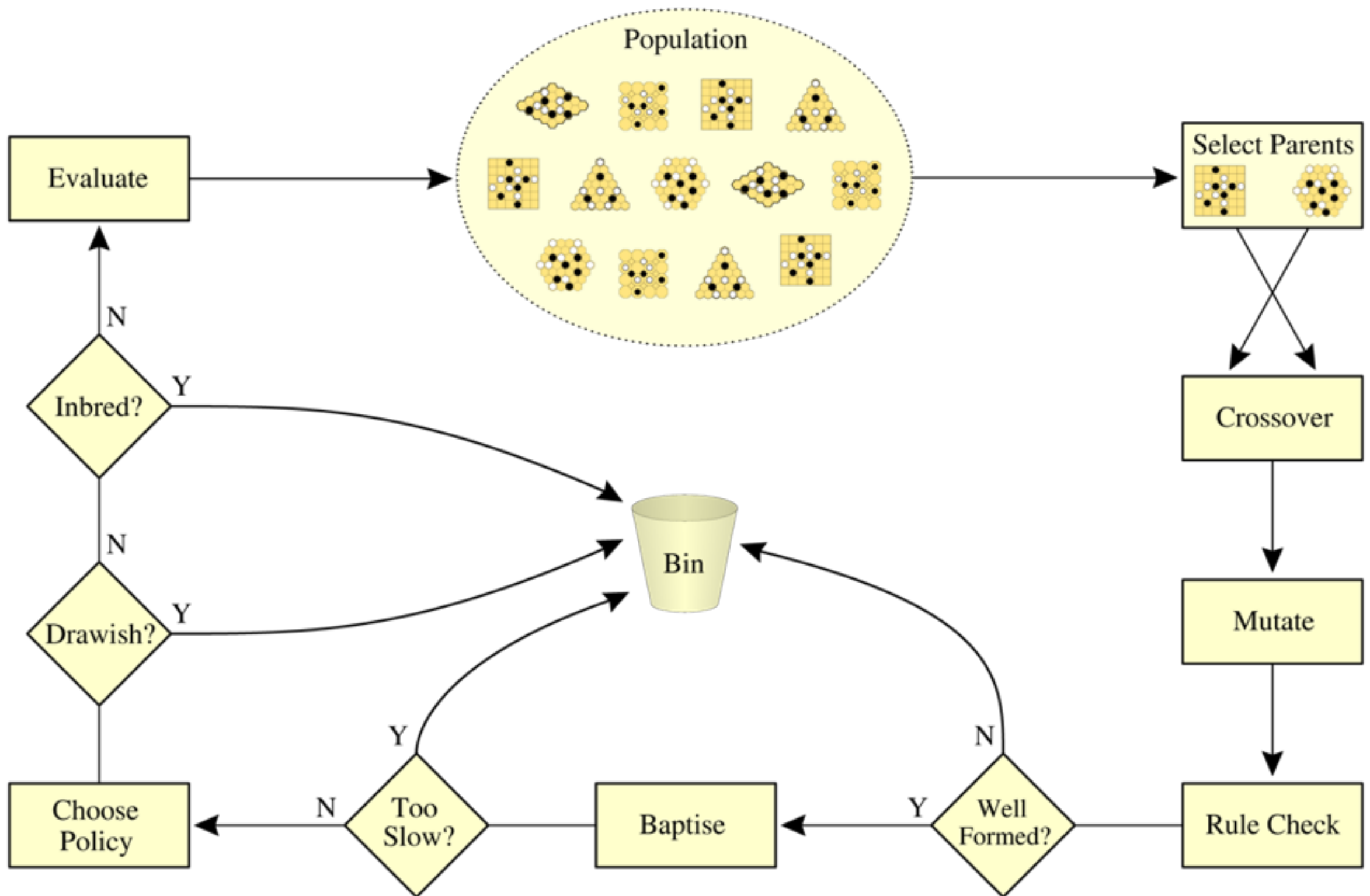
- Directly: grid
- More indirectly: position and orientation of walls
- Even more indirectly: patterns of walls and floor
- Very Indirectly: number of rooms and doors
- Indirectly: random seed

Evaluation functions

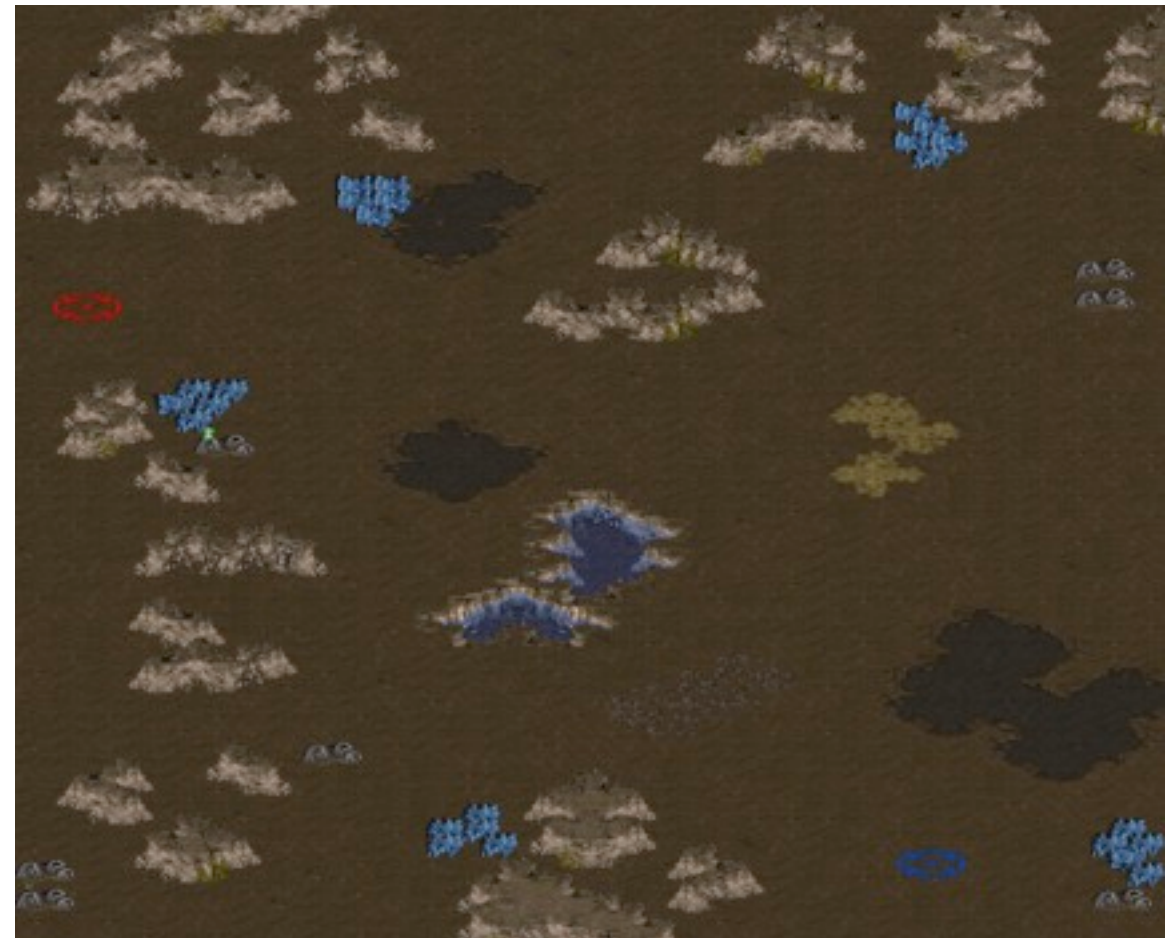
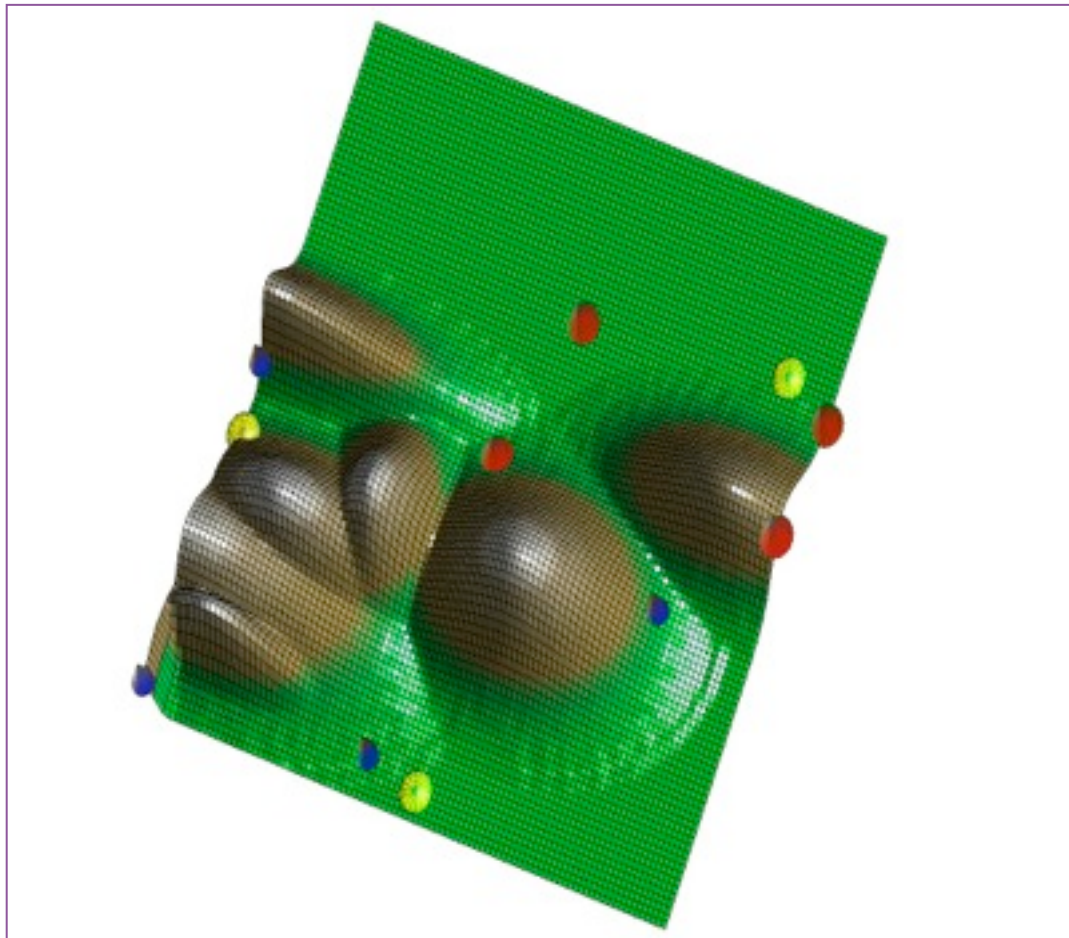
- Direct
- Simulation-based
- Interactive

An example: Ludi

- Evolves complete board games
- Games represented in a game description language
- Simulation-based fitness function (the games are played, various features of gameplay are measured)
- Produced a commercially published game!



Procedural map generation for RTS



J. Togelius, M. Press, N. Beume, S. Wessing, J. Hagelbäck, and G. N. Yannakakis., **Multiojective Exploration of the StarCraft Map Space**, *IEEE CIG 2010*

Procedural map generation for RTS

- Representation:
 - Locations of bases and resources (radial)
 - Locations, sizes of hills
 - Turtle-like procedure for StarCraft walls
- Fitness functions: base space, resource distance, choke points, resource balance...

Towards Automatic Personalized Content Creation for Racing Games

Julian Togelius, Renzo De Nardi and Simon M. Lucas

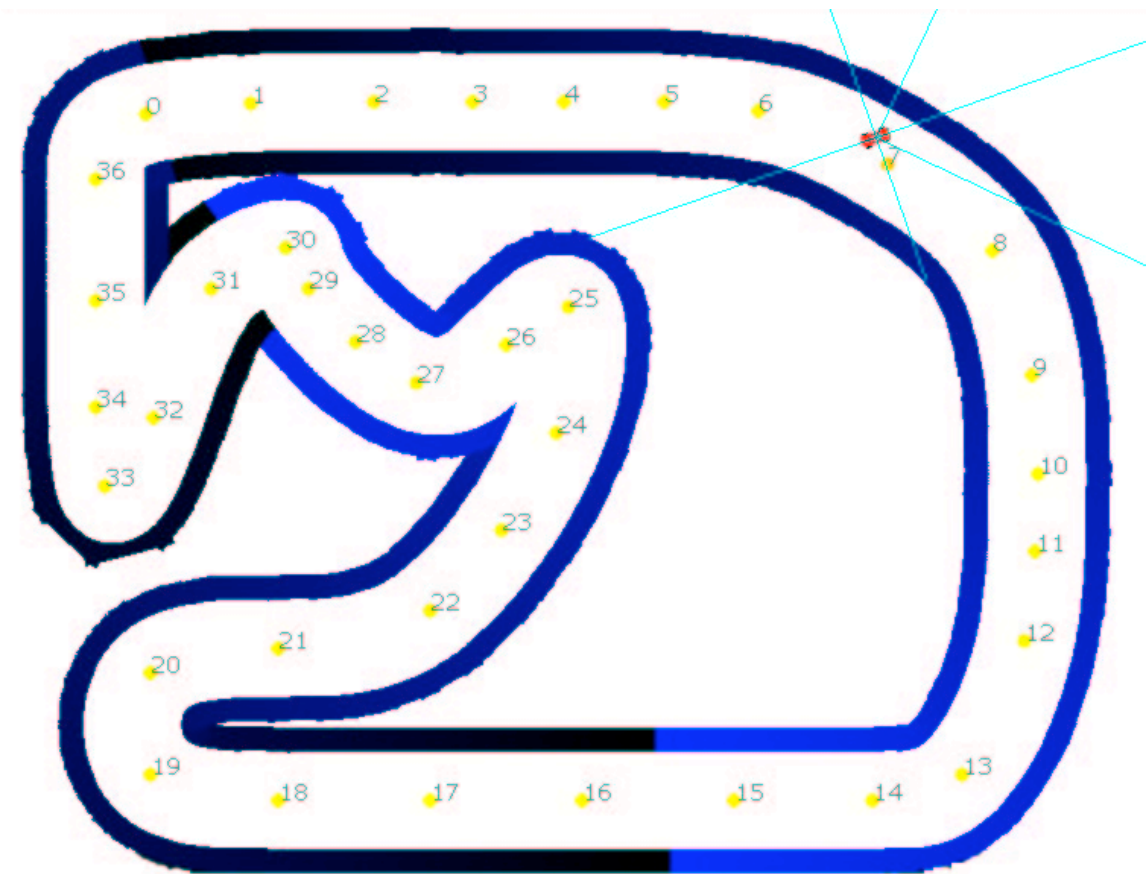
IEEE CIG 2007

Evolving racing tracks

1. Representation: b-splines
2. Mutation: randomly perturbing parameters
3. Fitness function: based on a player model

Fitness function

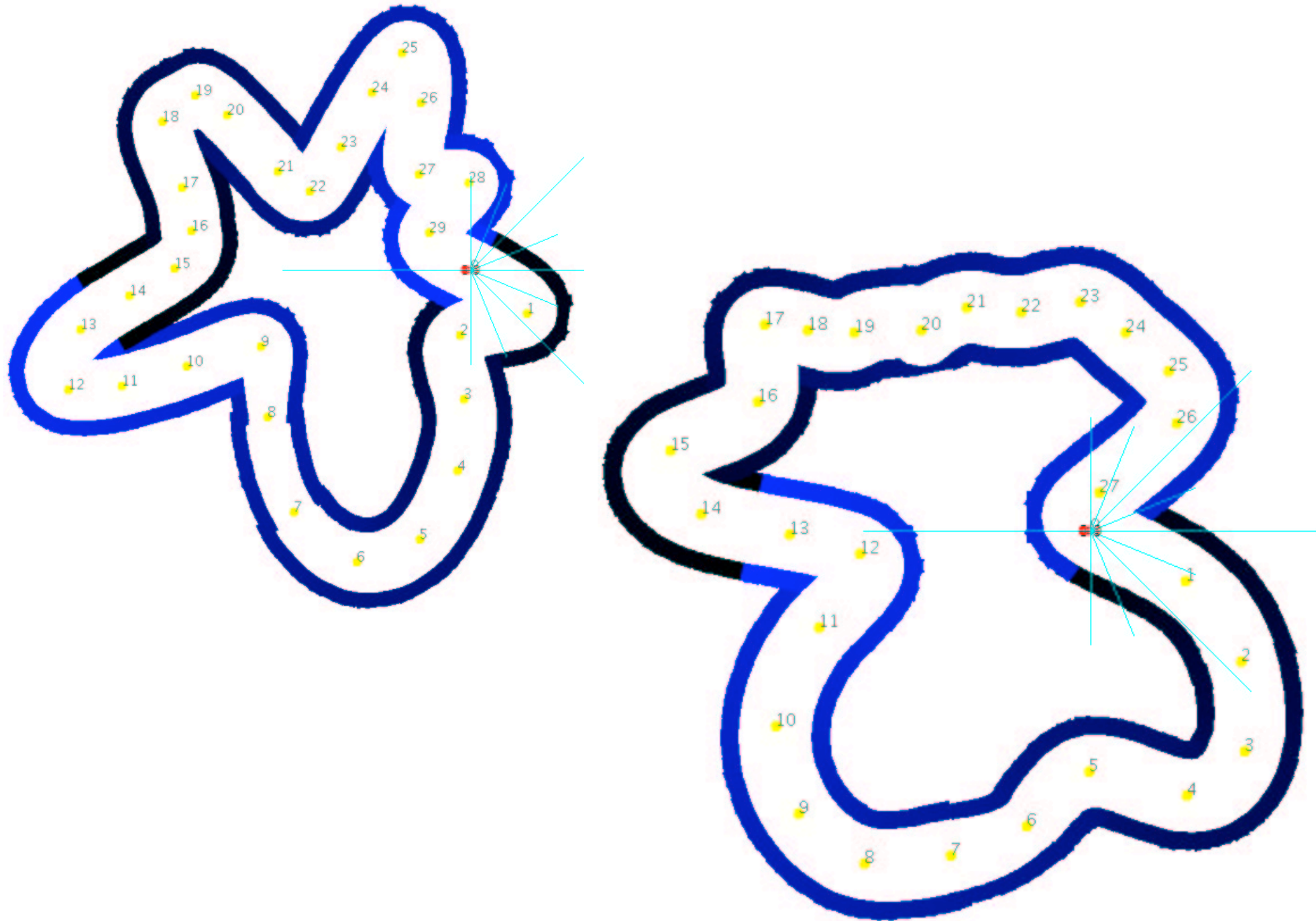
- Players drive on a test track
- Neural networks learn to replicate their driving styles
- New tracks are tested with these neural networks



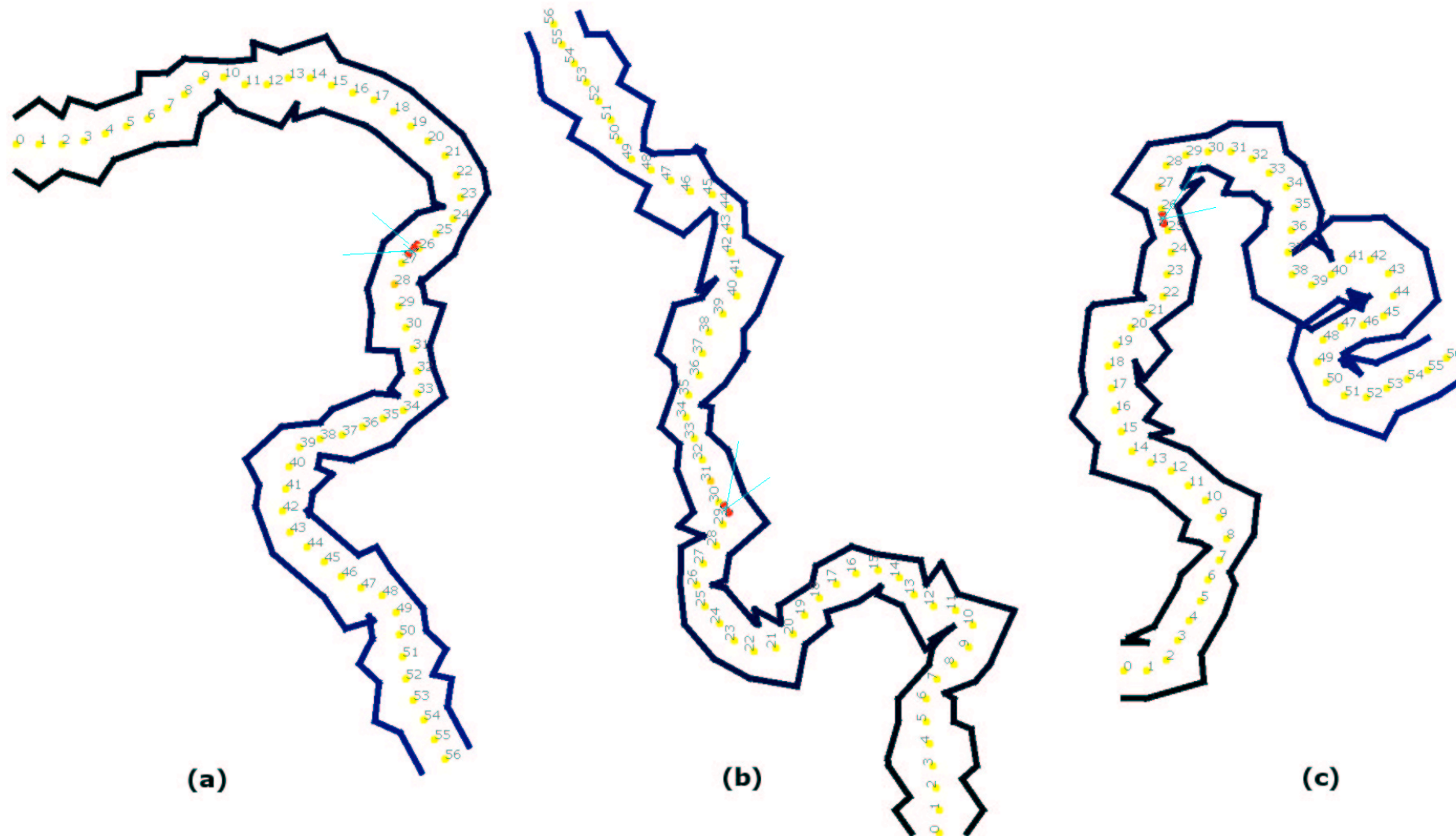
Fitness function

- Loosely adapted from Thomas Malone's theory of player entertainment
- Based on how the trained neural network drives on the new track
 - Progress (not too much or too little)
 - Progress (variation between trials)
 - Speed (variation within a lap)

Evolved tracks



Evolved tracks



Evolving Content in the Galactic Arms Race Video Game

Erin J. Hastings, Ratan K. Guha, and Kenneth O. Stanley

IEEE CIG 2009

Main idea

- Provide an infinite amount of content to players in a massively multiplayer game
 - increasing replay value
 - decreasing development cost
- Find appropriate content for each player
- Let players themselves affect the content generation

Galactic Arms Race

- Third person space shooter
- Massively multiplayer (about 1000 people)
- Both NPC and human enemies
- Weapons spread out in space
- Several weapons can be kept by the player
- Using a weapon = increasing its fitness

Galactic Arms Race



Weapon selection

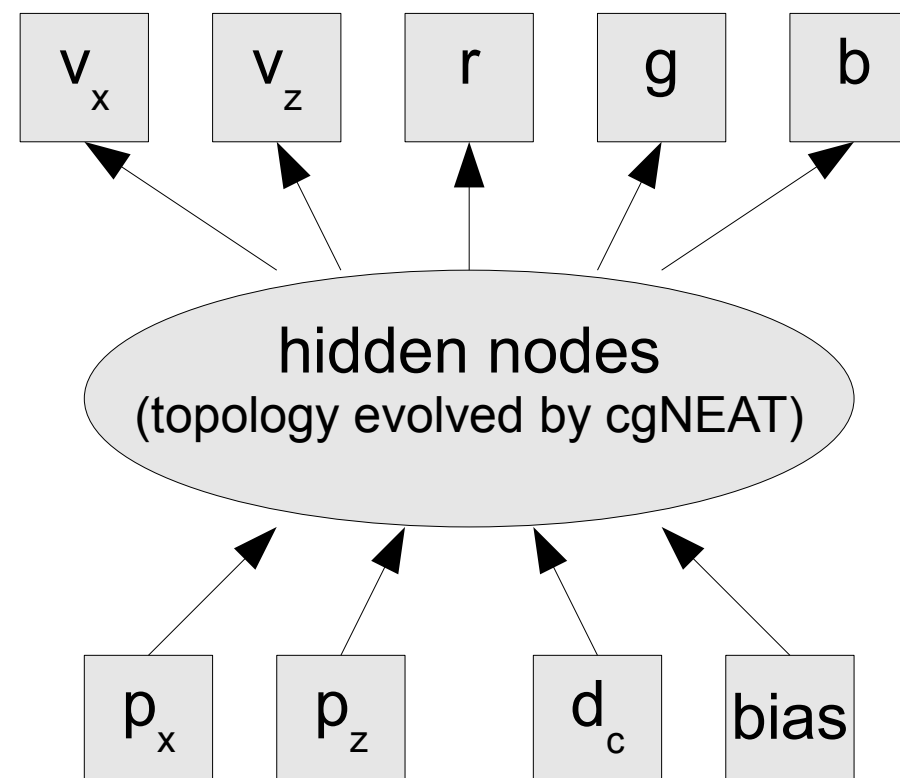
- One global population!
- New “individuals” can be picked up and/or discarded by players
- More used weapons get higher fitness, are used for reproduction
- Complicated EA: niching, innovation preservation etc.

Weapon representation

- Particle systems
- Evolved neural network (NEAT) maps relative particle position to velocity and color



(a)



Weapon evolution



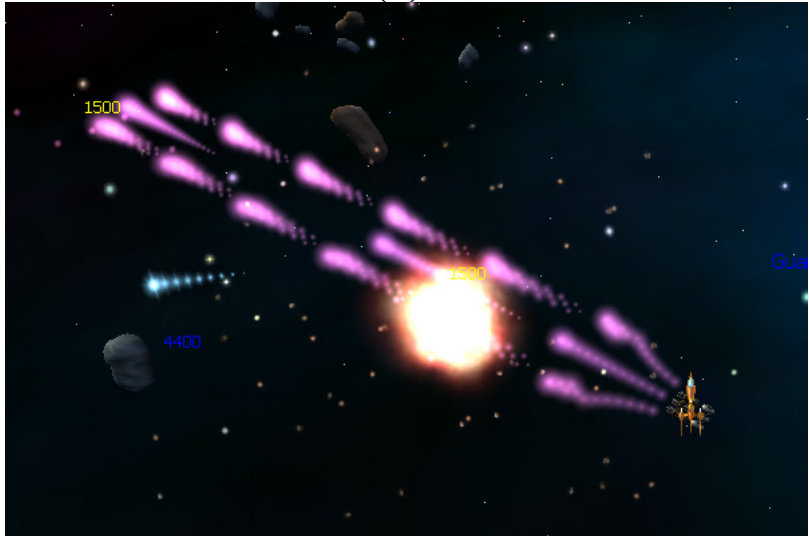
(a)



(b)



(c)



(d)



(e)



(f)



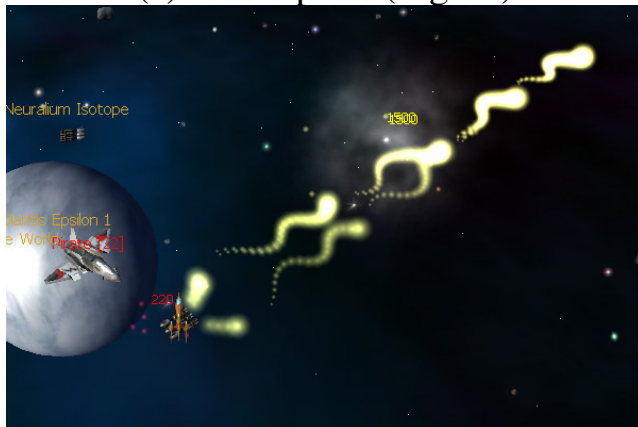
(a) Multispeed (7 gens)



(b) Ultrawide (13 gens)



(c) Three Prong (3 gens)



(d) Corkscrew (3 gens)



(e) Yellow Ladder (35 gens)



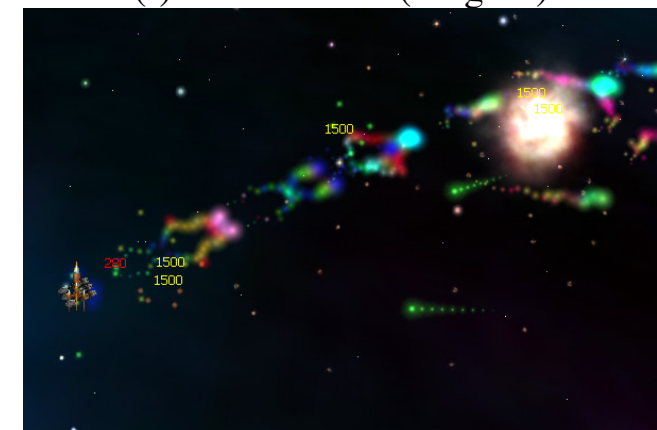
(f) Blue Ladder (42 gens)



(g) Double Bolt (12 gens)



(h) Trident (2 gens)



(i) Subatomic Heat (9 gens)



(j) Wallmaker (14 gens)



(k) Double Wallmaker (15 gens)



(l) Tunnelmaker (2 gens)