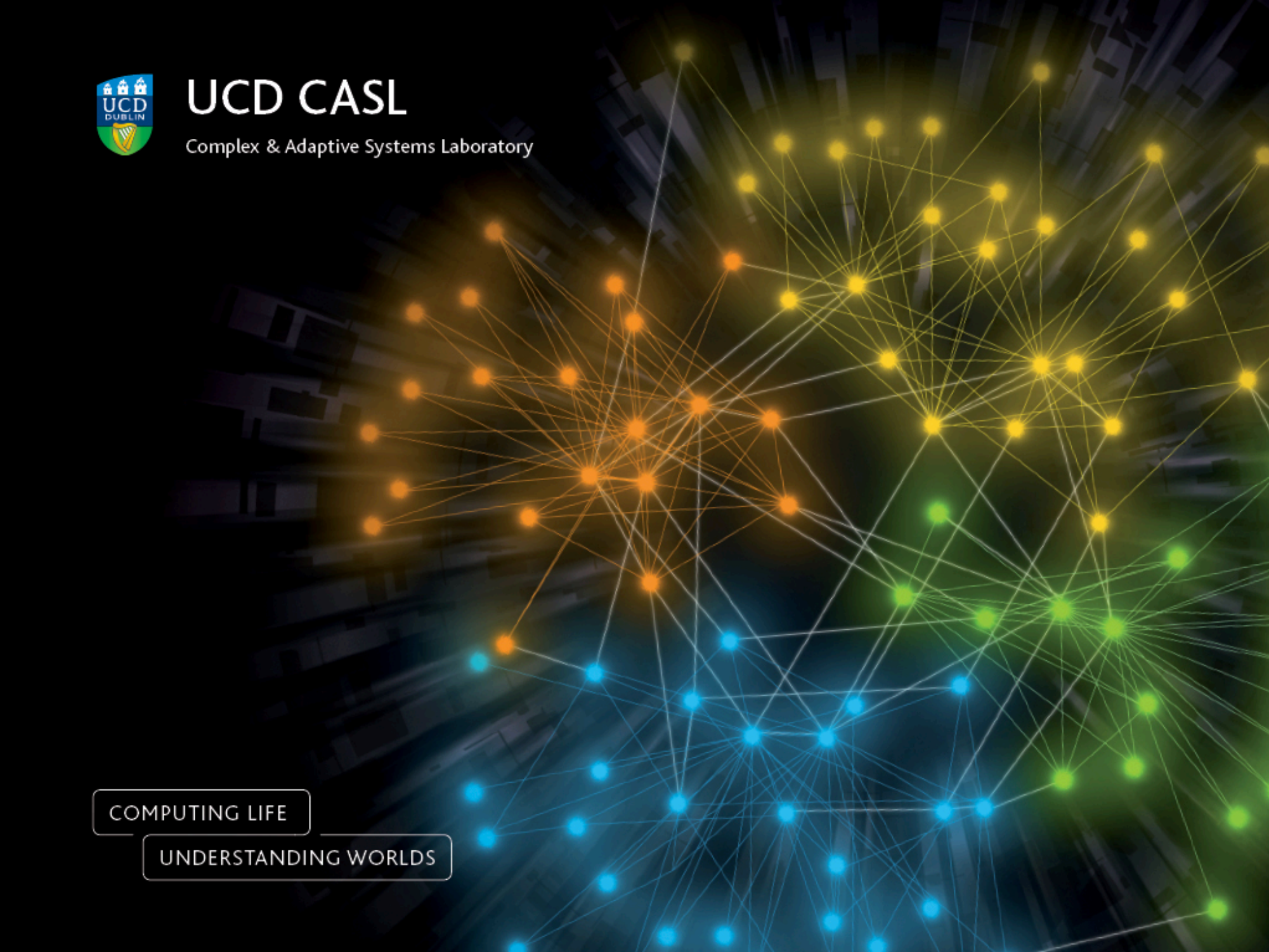

UCD CASL
Complex & Adaptive Systems Laboratory
COMPUTING LIFE
UNDERSTANDING WORLDS

D. Perez, M. Nicolau, M. O'Neill, A. Brabazon

Reactiveness and Navigation in Computer Games: Different Needs, Different Approaches

03.09.11

Outline

Behaviour Trees for Mario AI

Previous Work

- Evolved Behaviour Trees for Mario AI using Grammatical Evolution;
- Killer reactive behaviour;
- Navigation relied on high-level sub-routines;
- Achieved 4^{th} place at CIG-2010 competition.

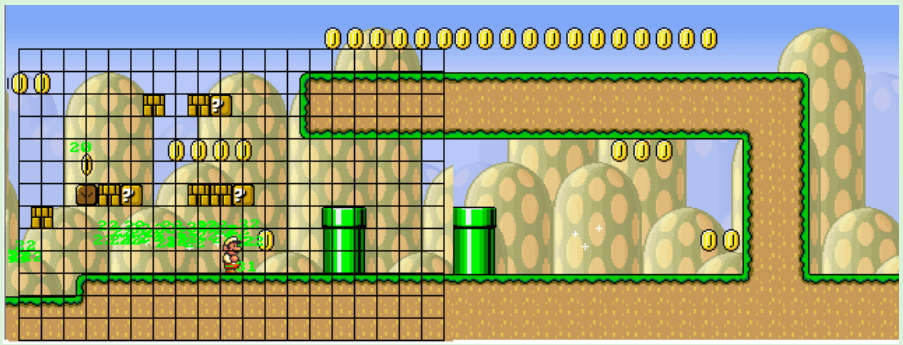
Mario AI

The Mario AI Benchmark

- Open source version of game, developed by Togelius et al.

Environment and Control

- 21x21 matrix around Mario with different levels of information:
  - Geometry, enemies, position, status, mode, stats;



Mario AI

Navigation

- Need to navigate through structural hazards;
- Dynamic path: blocks can be broken;
- Instant calculation of path through level:
  - Use dynamically updated A* navigation.

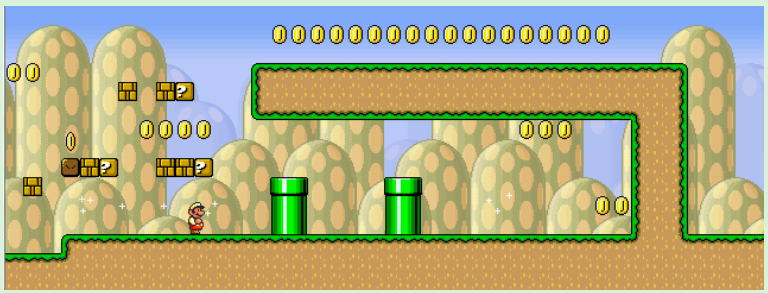
Reactiveness

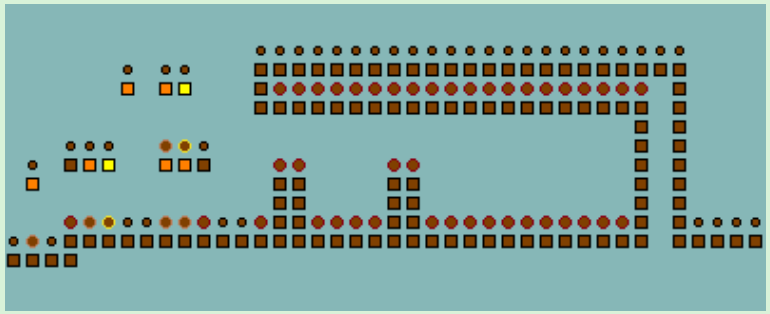
- Instant, reactive behaviour required;
- Too dynamic for online learning;
- Well encoded as condition-action associations:
  - Evolve Behaviour Trees offline using Grammatical Evolution.

Navigation: A*

Graph creation

- Map not available at start, must be built during navigation;
- Use discrete coordinate system to store tiled version of level;
- Identify nodes:
  - Places where Mario can stand;
  - Store extra information (such as what is over each node).

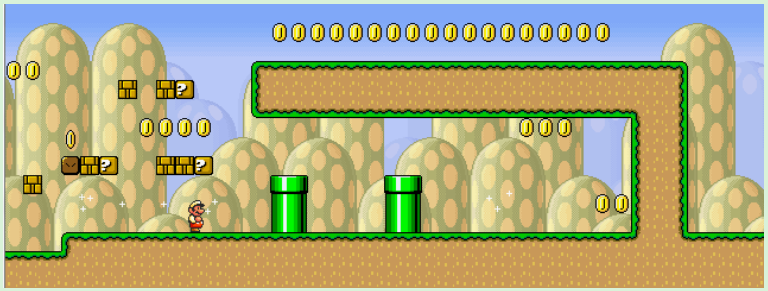


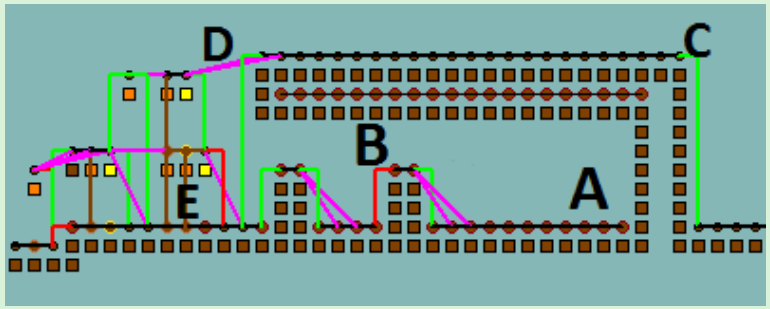


Navigation: A*

Graph creation

- Non-zenithal perspective: horizontal edges $\neq$ vertical edges;
- Different types of links:
  A: Walk links; **D**: Faith jump links;
  B: Jump links; **E**: Break jump links.
  C: Fall links;
- Cost: Manhattan distance + modifier for jumps.





Reactiveness: Behaviour Trees

Overview

- Introduced as a means to specify system requirements;
- Lately used to encode game AI (*Halo*, *Spore*, *Façade*, *Defcon*).

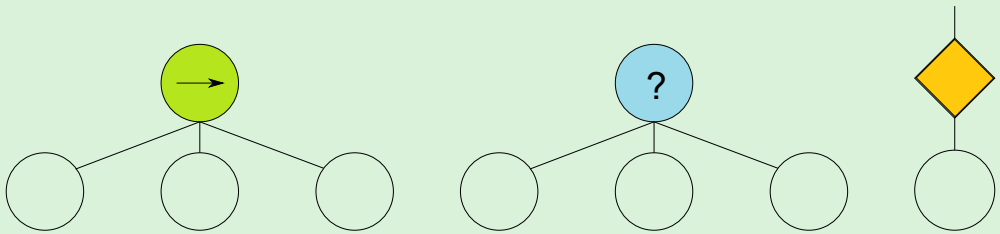
AI Application

- Hierarchically encode behaviours by decreasing order of complexity:
  - ⋆*soldierBehaviour*
    - ⋆*attack*
      - ⋆*aimingAlgorithm*
        - ⋆*tracking*
          - ...
            - ⋆*playSprite*

Behaviour Trees

Components

- *Control Nodes*: **Sequence**, **Selector**, **Filter**;



- *Leaf Nodes*: **Conditions**, **Actions**.

Behaviour Trees for Mario

Integration

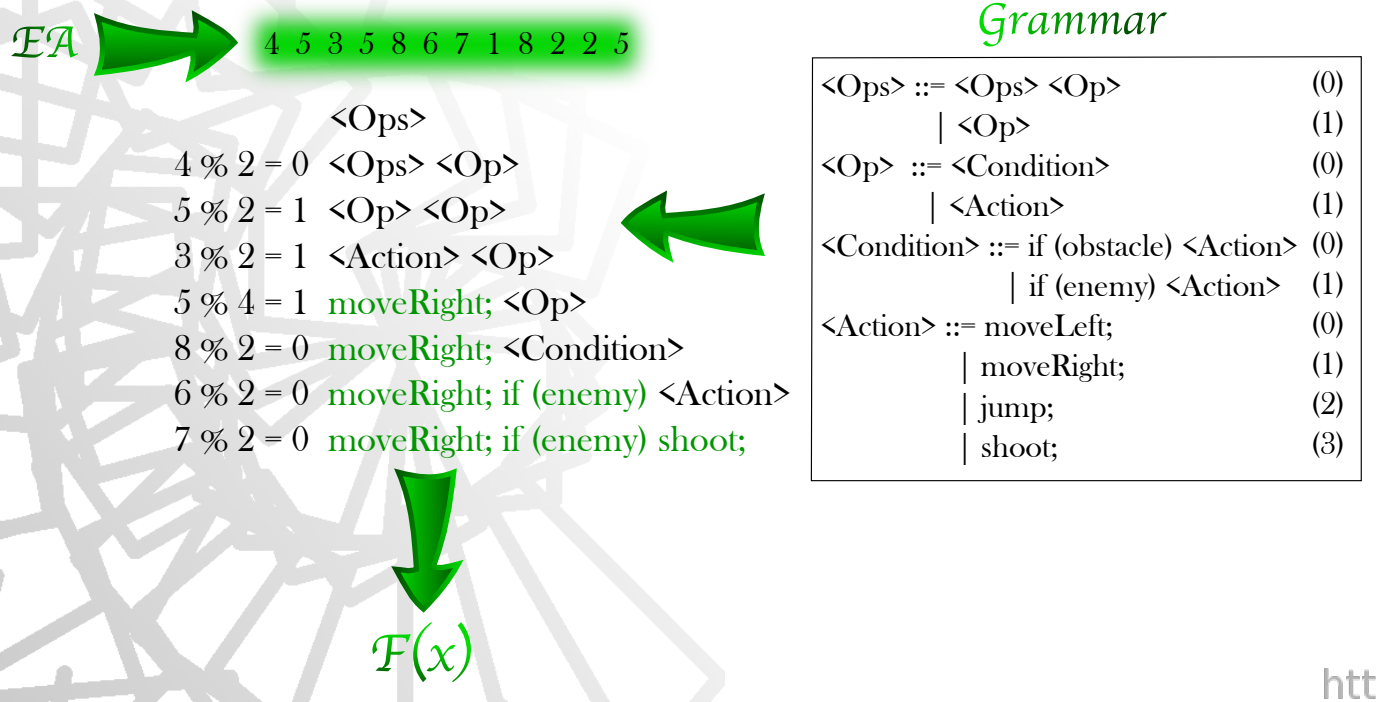
- Synchronise Mario and BT execution:
  - BT parsing resumes from previously reached point.
- Provide nodes for evolution:

Filters	Conditions	Actions	Sub-trees
Loops	EnemyAhead	*LRUDJF*	JumpRightLong
NON	UnderQuestion	RunRight	VerticalJump
...	...	GetPathToRightMostPosition	...
...	...	GetPathToClosestQuestion	...

Grammatical Evolution

Key Characteristics

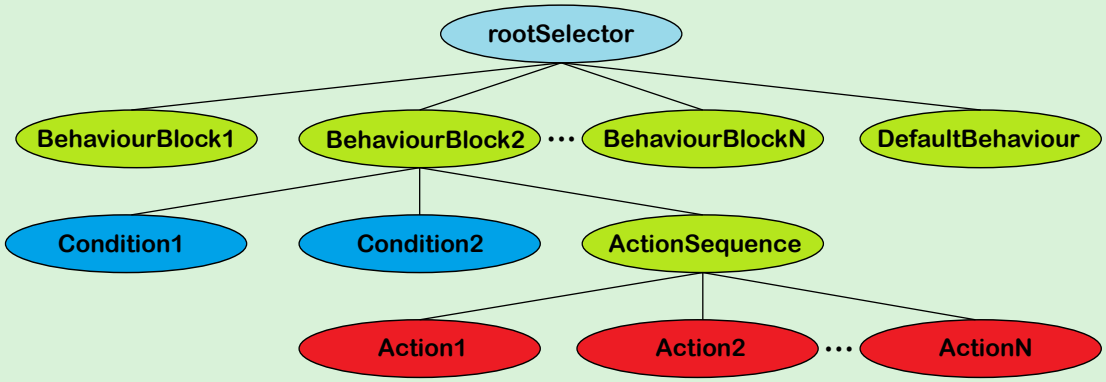
- Numerical strings evolved by any search algorithm:
- Syntax of solutions specified by grammar.



Grammatical Evolution

Evolving Behaviour Trees

- XML syntax specified through grammar;
- All conditions (18), actions (15), sub-trees (14) and filters (4);
- Limit syntax combinations through grammar (and-or trees).



Experiments

Experimental Setup

GE	Population Size	500
	Evaluations	125000
	Marked 2-point Crossover Ratio	50%
	Marked Swap Crossover Ratio	50%
	Average Mutation Events per Individual	1
Mario	Level Difficulties	0 1 2 3 4
	Level Types	0 1

Generalisation Score

- Measured on 360 unseen levels:
  - 20 different map sets;
  - 9 level difficulties;
  - 2 level types.

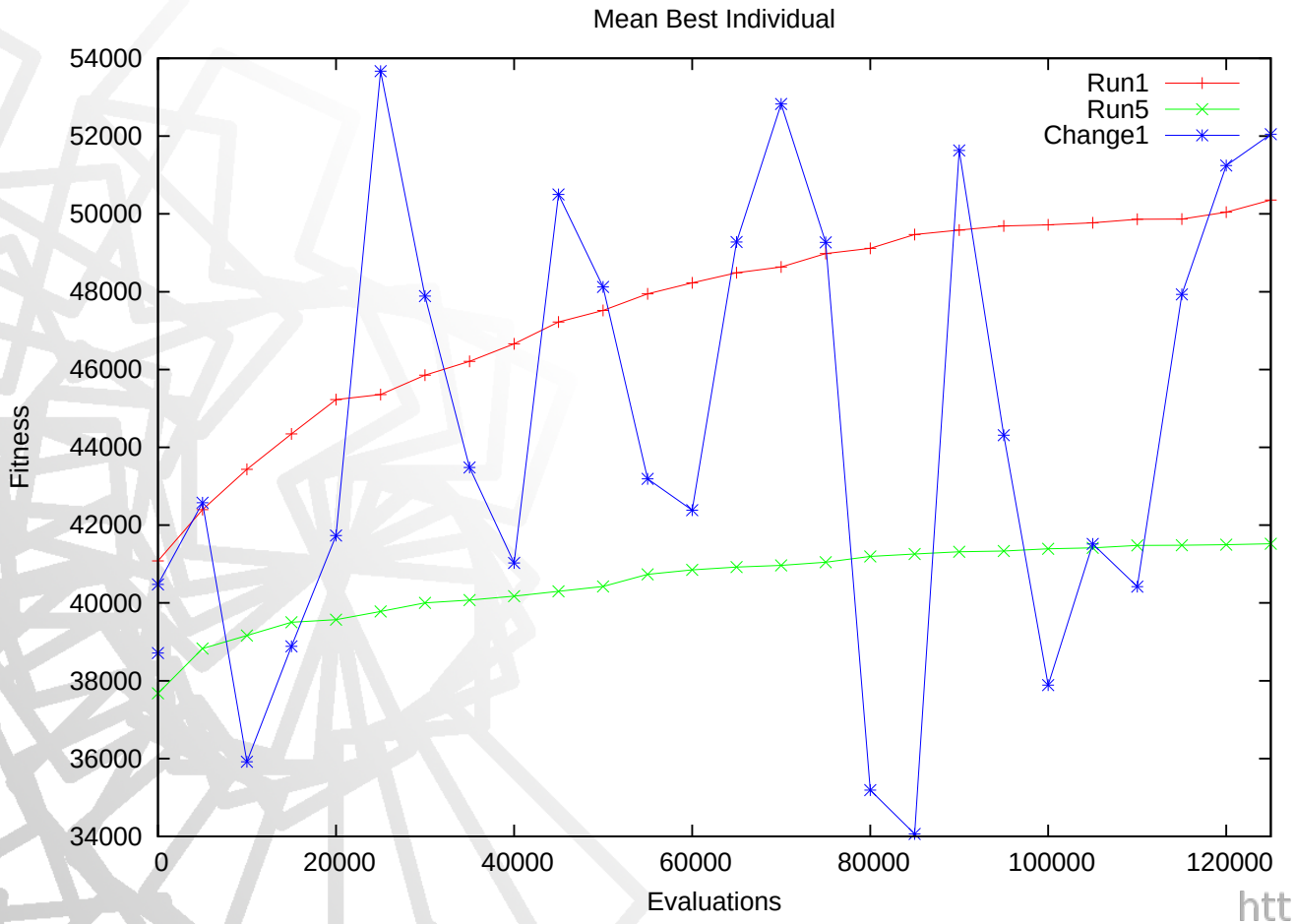
Experiments

Generalisation Issues

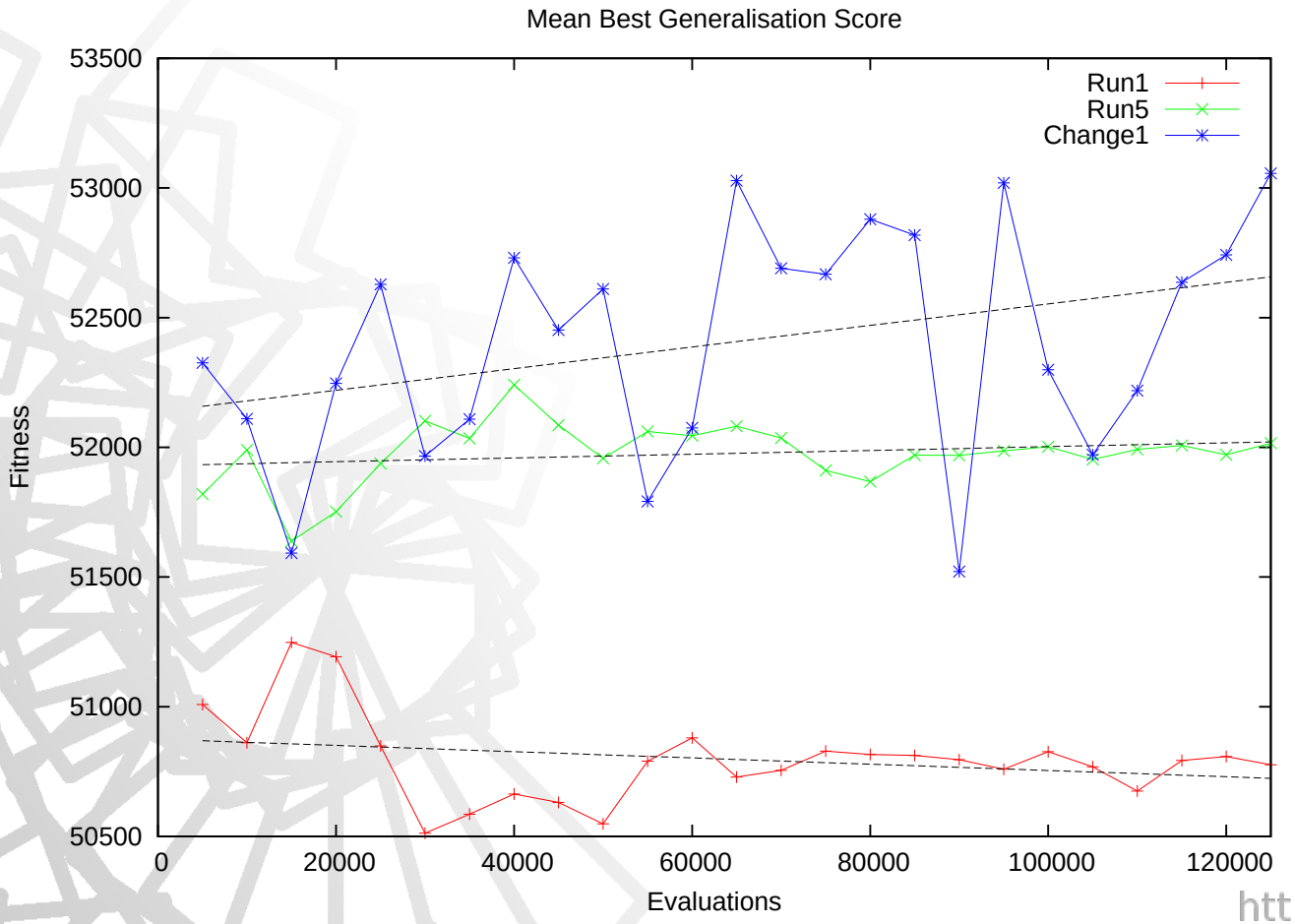
- Dynamic problem: controllers tested in unseen maps;
- Three approaches:

	Run1	**Run5**	**Change1**
Map sets per evaluation	1	5	1
Change sets between evaluations	No	No	Yes

Results


Mean Best Individual
Fitness
Evaluations
Run1
Run5
Change1
54000
52000
50000
48000
46000
44000
42000
40000
38000
36000
34000
0
20000
40000
60000
80000
100000
120000

Results


Mean Best Generalisation Score
Run1
Run5
Change1
Fitness
Evaluations
53500
53000
52500
52000
51500
51000
50500
0
20000
40000
60000
80000
100000
120000

Conclusions

Navigation & Reactiveness

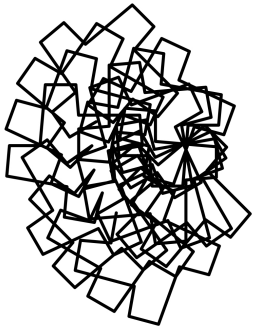
- Dynamic A* provides updated path following routines;
- Behaviour Tree routines enable fast-reacting behaviour;
- Grammatical Evolution combines both through evolution:
  - Grammar facilitates syntax specification;
  - Genetic operators exchange meaningful *building-blocks*.

Future Work

- Improve generalisation performance:
  - Training and generalisation tests;
  - Sliding windows.

Acknowledgements

- SFI support grant number 08/IN.1/I868;
- A big **Thank You!** to all NCRA members.



Fondúireacht Eolaíochta Éireann
Science Foundation Ireland