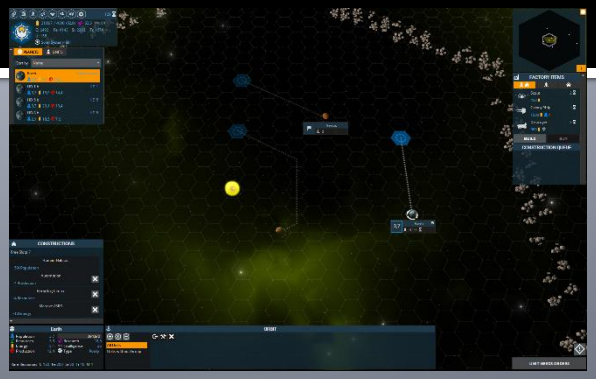
MFF UK
Prague
25.10.2018

GAME DEVELOPMENT
@CUNI.CZ

AI for Children of the Galaxy

MCTS to Rule Them All…

Pavel Šmejkal, Jakub Gemrot



AI in Games…

Why?

Would it be a game

without ...

Source: https://wall.alphacoders.com/big.php?i=324425

Would it be a game

without ...

Adapted from: https://wall.alphacoders.com/big.php?i=324425

Would it be a game

if you have to ...

Would it be a game

if you have to …

Games are still used as a source of interesting (and hard) problems.

Mastering Games

Means to compete with humankind

Source: iSTAN HONDA/AFP/GETTY IMAGES

Source: AP/Lee Jin-man

Source: ihttps://www.inverse.com/article/13630

Source: GETTY IMAGES/NICOLAS_

Go 19x19

Game tree ~ 10^{360}
Avg. game length ~ 150 plies

AlphaZero (is claimed) to require >4.6yr of TPU time to master it.
Learned in 8hr on a farm of 5064 TPUs.

Silver, David, et al. "Mastering chess and shogi by self-play with a general reinforcement learning algorithm." arXiv preprint arXiv:1712.01815 (2017).

What's on the game horizon now?

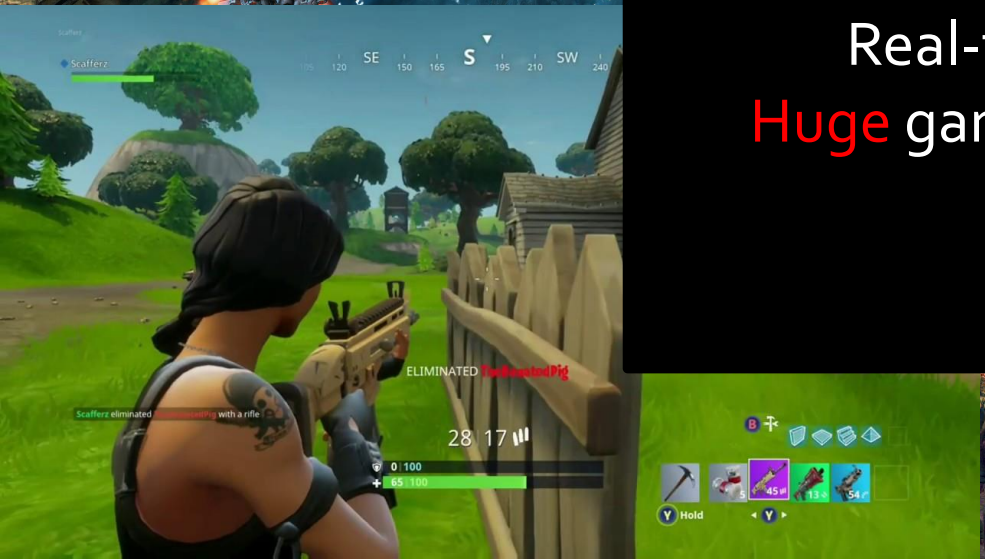
Modern Computer Games

Now for "real" problems…

Modern Computer Games

We have solved Go, now for "real" problems...

Stochastic
Partially observable
Simultaneous
Real-time
Huge game trees

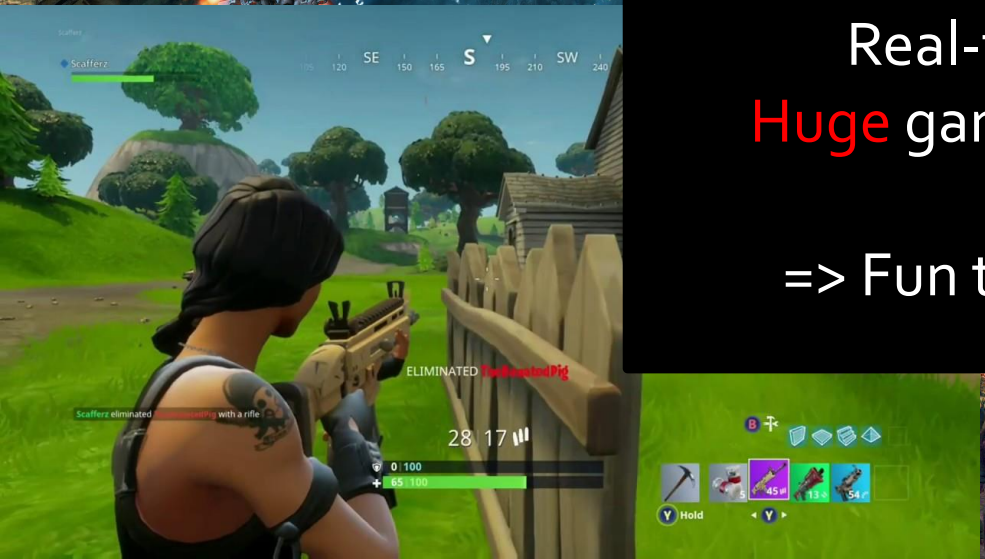


Modern Computer Games

We have solved Go, now for "real" problems...

Stochastic
Partially observable
Simultaneous
Real-time
Huge game trees

=> Fun to play!



Star Craft II

Targeted by Google, OpenAI, Facebook

Star Craft: Brood War

In scope of various unis for some time now

Star Craft: Brood War

Game size?

Game	State-space	Branching	Depth
Chess	10^{47}	35	80
Go	10^{171}	80	200
SCBW			

Star Craft: Brood War

Game size?

Game	State-space	Branching	Depth (pl.)
Chess	10^{57}	~35	~80
Go	10^{430}	~250	~211
SCBW	10^{1685}		

StarCraft map: 128x128
Maximum number of units: 400

Considering only unit positions:
$(128x128)^{400}=16384^{400}\approx10^{1685}$

Star Craft: Brood War

Game size?

Game	State-space	Branching	Depth
Chess	10^{57}	~35	~80
Go	10^{430}	~250	~211
SCBW	10^{1685}	10^{4} - 10^{200}	

Units: 4 – 200
Actions per unit: 10

Branching factor: 10^{50} - 10^{200}

Star Craft: Brood War

Game size?

Game	State-space	Branching	Depth
Chess	10^{57}	~35	~80
Go	10^{430}	~250	~211
SCBW	10^{1685}	10^{4} - 10^{200}	36000

Length of a game: 25 minutes

25 min x 60 sec x 24 iteration/sec = 36000

How to tackle such a space?

Star Craft: Brood War

Divide et impera

Layers of control

- Strategic – Army/Base level
  - Build, research, muster, expand, manage groups
- Tactical – Group level
  - Move, attack, siege, defend
- Reactive – Unit Level
  - Engage, withdraw, use ability

STRATEGIC LAYER (SCBW)

Star Craft: Brood War

Strategic Layer

- Not addressed much
- Partial observability is a big problem as the first encounter with the enemy is done usually after 2-4 minutes (depth 2 880 – 5 760)
- Even though we have a lot of replays, if you consider the number of maps, combination of races and different initial positions, the data set is not big enough in each bucket
- Human players have already converged to many viable opening strategies

Star Craft: Brood War

Strategic Layer

"Poor man's solution"

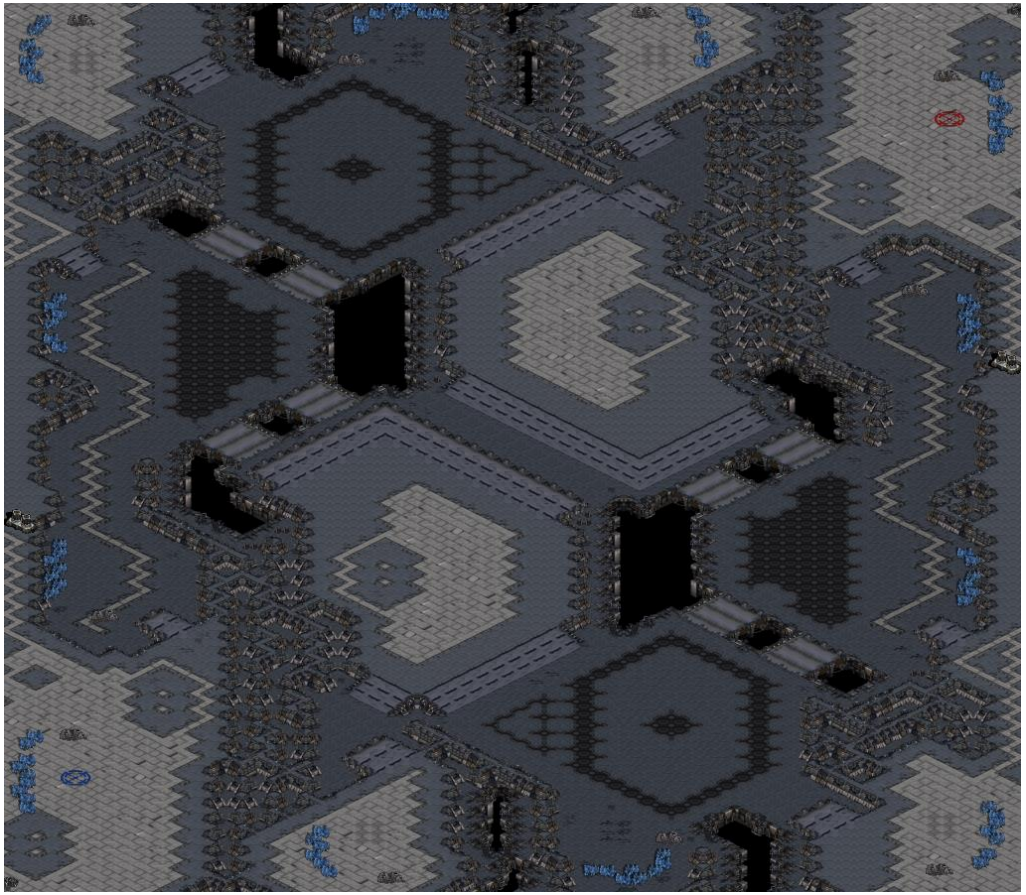
- Pick existing strategy and implement its build order via rule-based systems
- Zerg: 6-pool rush, Lurker rugh, Mutarush, …
- Terran: Bunker-push, Tank-push, …
- Protoss: Zealot rush, Photon cannon rush, …
- Suitability depends on the map and initial base positions
- Typically each bot implements one to a few strategies

TACTICAL LAYER (SCBW)

Star Craft: Brood War

Tactical layer

Abstraction
for map Benzene



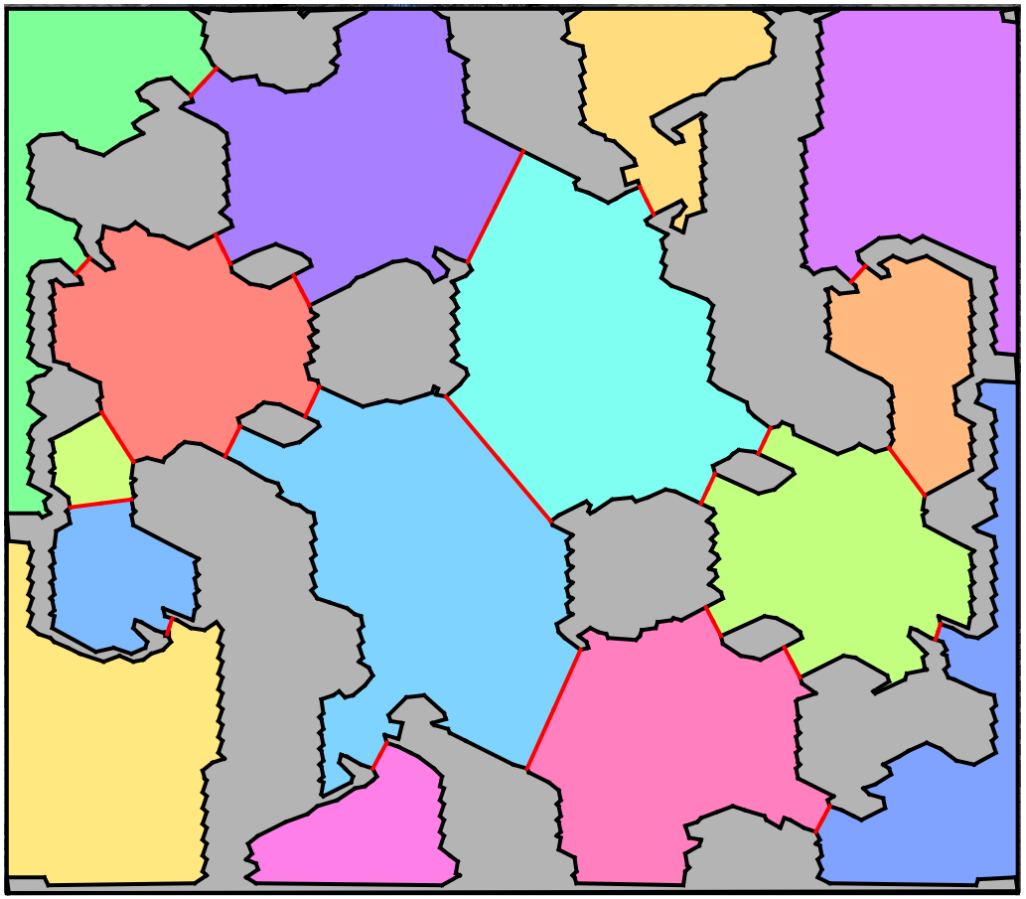
Uriarte, Alberto, and Santiago Ontañón. "Game-tree search over high-level game states in RTS games." *Tenth Artificial Intelligence and Interactive Digital Entertainment Conference*. 2014.

Star Craft: Brood War

Tactical layer

Abstraction
for map Benzene

Perkins' algorithm
to decompose
a map into
regions
and **chokepoints**.



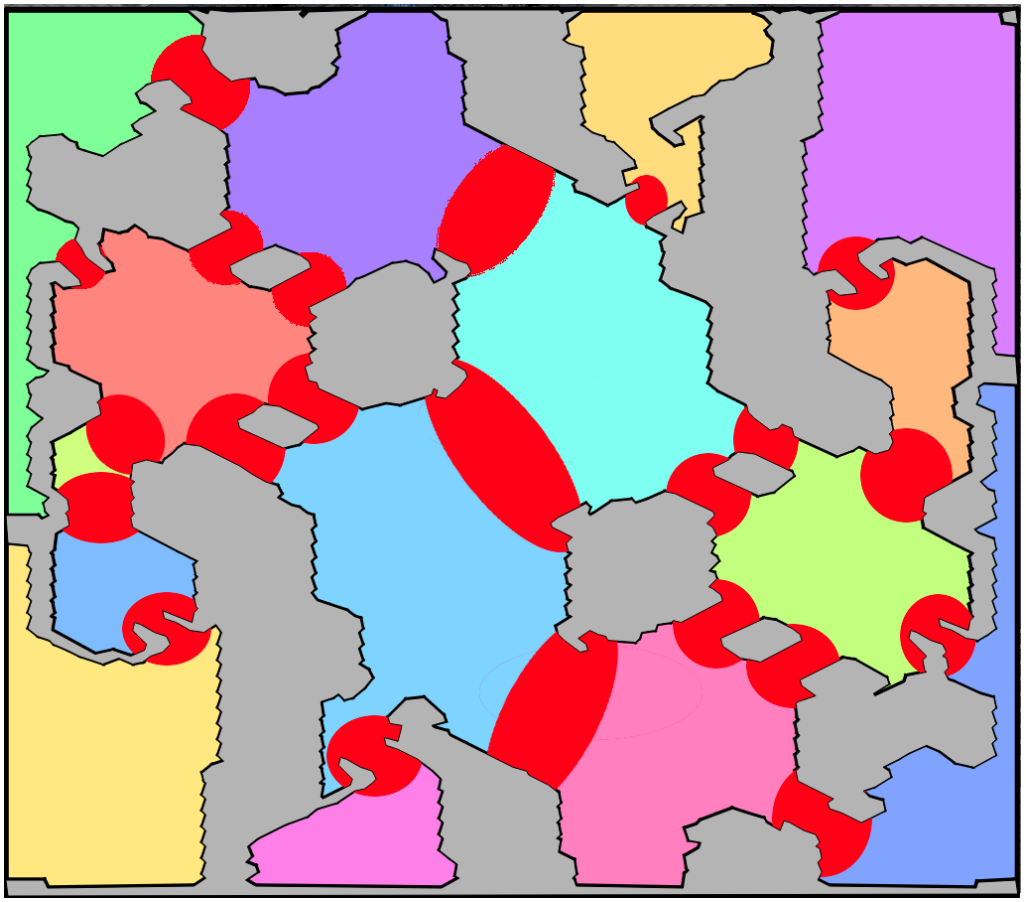
Uriarte, Alberto, and Santiago Ontañón. "Game-tree search over high-level game states in RTS games." *Tenth Artificial Intelligence and Interactive Digital Entertainment Conference*. 2014.

Star Craft: Brood War

Tactical layer

Abstraction
for map Benzene

Chokepoints (20)
are deviding
regions (15).



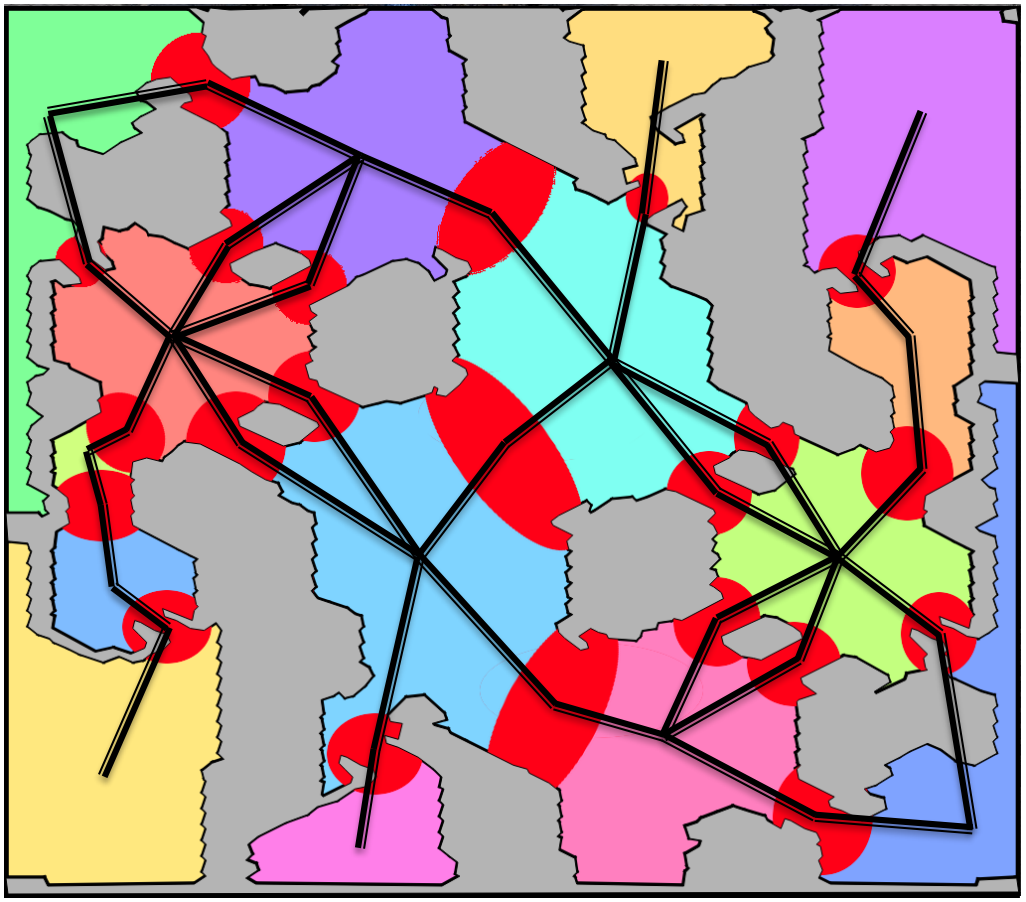
Uriarte, Alberto, and Santiago Ontañón. "Game-tree search over high-level game states in RTS games." *Tenth Artificial Intelligence and Interactive Digital Entertainment Conference*. 2014.

Star Craft: Brood War

Tactical layer

Abstraction
for map Benzene

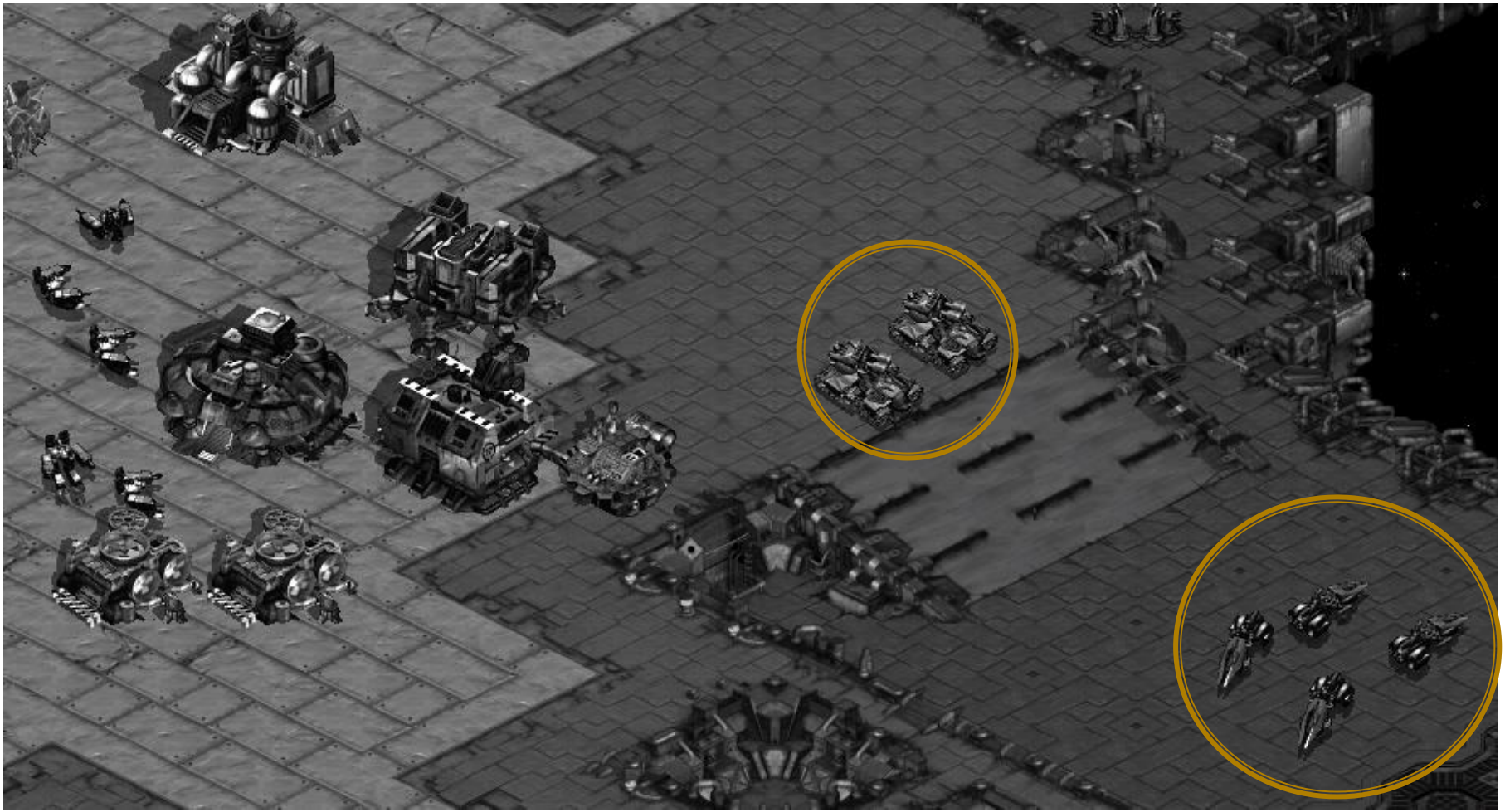
Distance matrix
precomputed
between regions.
(Mind the air units.)



Uriarte, Alberto, and Santiago Ontañón. "Game-tree search over high-level game states in RTS games." *Tenth Artificial Intelligence and Interactive Digital Entertainment Conference*. 2014.

Star Craft: Brood War

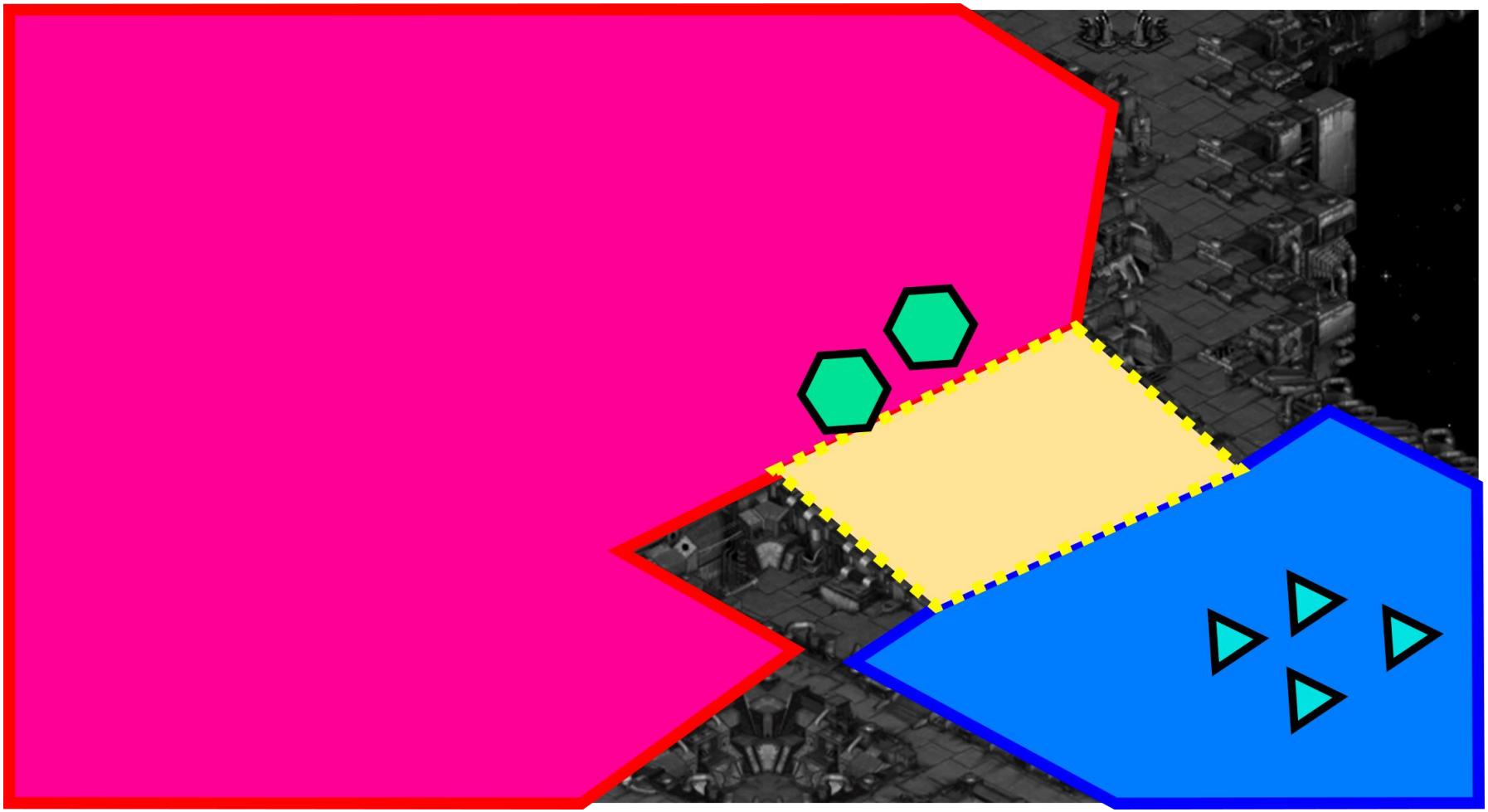
Tactical layer



Uriarte, Alberto, and Santiago Ontañón. "Game-tree search over high-level game states in RTS games." *Tenth Artificial Intelligence and Interactive Digital Entertainment Conference*. 2014.

Star Craft: Brood War

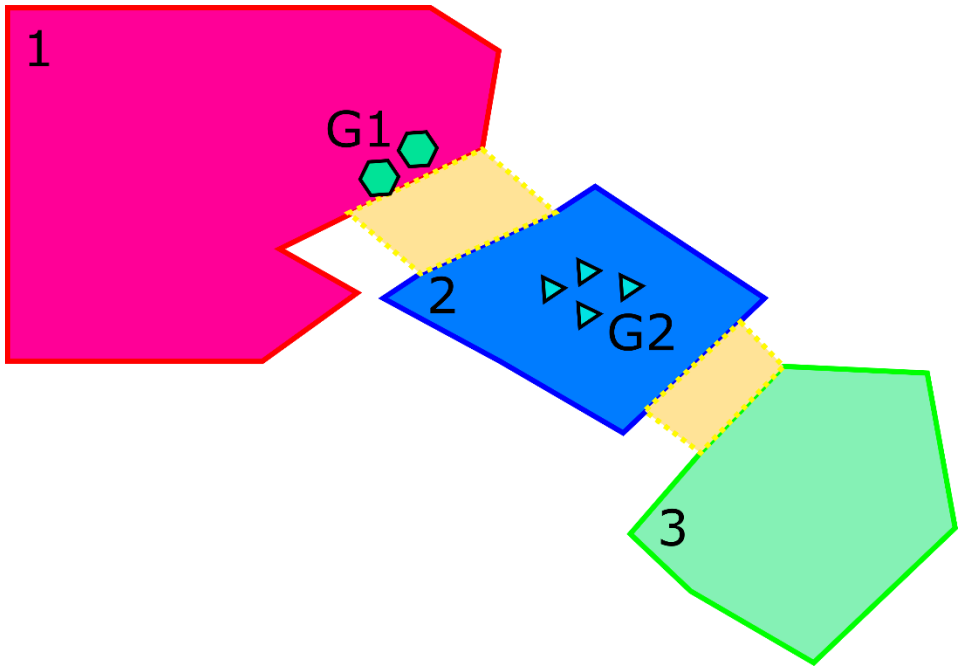
Tactical layer



Uriarte, Alberto, and Santiago Ontañón. "Game-tree search over high-level game states in RTS games." *Tenth Artificial Intelligence and Interactive Digital Entertainment Conference*. 2014.

Star Craft: Brood War

Tactical layer



G1:
move 2, idle
G2:
move 1, move 3, idle

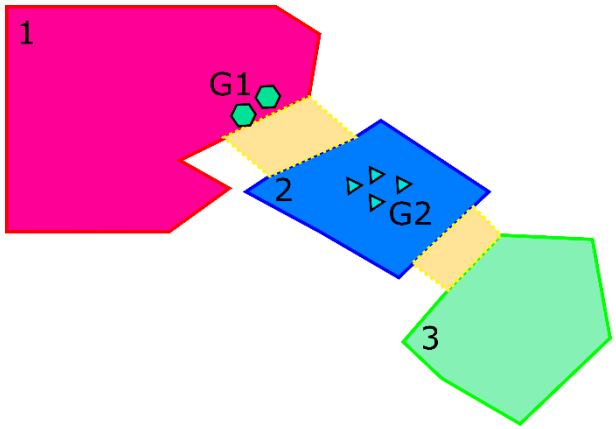
=> Branching factor 6

Uriarte, Alberto, and Santiago Ontañón. "Game-tree search over high-level game states in RTS games." *Tenth Artificial Intelligence and Interactive Digital Entertainment Conference*. 2014.

Star Craft: Brood War

Tactical layer

SCBW player is managing about 8 groups.



Avg.# of region links ~ 4

⇒ 4 move + 1 idle action

⇒ $5^8 = 4.10^5$ branching factor in a late game phase

Much smaller during early/mid game phases.

Uriarte, Alberto, and Santiago Ontañón. "Game-tree search over high-level game states in RTS games." *Tenth Artificial Intelligence and Interactive Digital Entertainment Conference*. 2014.

Star Craft: Brood War
Tactical layer

Search is doable

- ABCD
- MCTSCD

(discussed later)

Uriarte, Alberto, and Santiago Ontañón. "Game-tree search over high-level game states in RTS games." *Tenth Artificial Intelligence and Interactive Digital Entertainment Conference*. 2014.

REACTIVE LAYER (SCBW)

Star Craft: Brood War

Reactive layer

Star Craft: Brood War

Reactive layer

Star Craft: Brood War

Reactive layer

Red player: 22 units (4 station.)

Possible actions (roughly):

$9^{18} \times 8^{4} \sim 10^{20}$

Blue payer: 47 unit

Possible actions (roug.):

$9^{47} \sim 10^{87}$

Local "search" to prune the action space.

Star Craft: Brood War

Reactive layer

Action space -> Script space

Instead of actions, we use scripts.

`Script: S -> A`

For a given state s, it gives an action to perform. Usually O(N).

Closest	attack closest unit
Kiting	attack closest unit than escape
AV	attack highest `dpf(u)/hp(u)`
NOK-Closest	attack closest unit if not to receiving lethal dmg
NOK-AV	NOK but attack via AV
Kiting-AV	hit and run, choose target via AV
Kiting-NOK-AV	kiting but choose NOK-AV

Random	Weakest	Closest	AV	Kiter	Kite-AV	NOK-AV
1.00	0.98	0.98	0.98	0.97	0.97	0.95

Churchill, David, Abdallah Saffidine, and Michael Buro. "Fast Heuristic Search for RTS Game Combat Scenarios." *Eighth Artificial Intelligence and Interactive Digital Entertainment Conference*. 2012.

Star Craft: Brood War

Reactive layer

Red player: 22 units (4 station.)

Possible actions (**low-level**):

$$9^{18} \times 8^{4} \sim 10^{20}$$

Blue payer: 47 unit

Possible actions (**low-l.**):

$$9^{47} \sim 10^{87}$$

Local "search" to prune the action space.

Star Craft: Brood War

Reactive layer

Red player: 22 units (4 station.)

Possible actions (**2 scripts**):

$$2^{18} = 262\,144$$

Blue payer: 47 unit

Possible actions (**2 scr.**):

$$2^{47} \sim 10^{53}$$

Star Craft: Brood War

Reactive layer

Red player: 22 units (4 station.)
Possible actions (**2 scripts**):

$$2^{18} = 262\,144$$

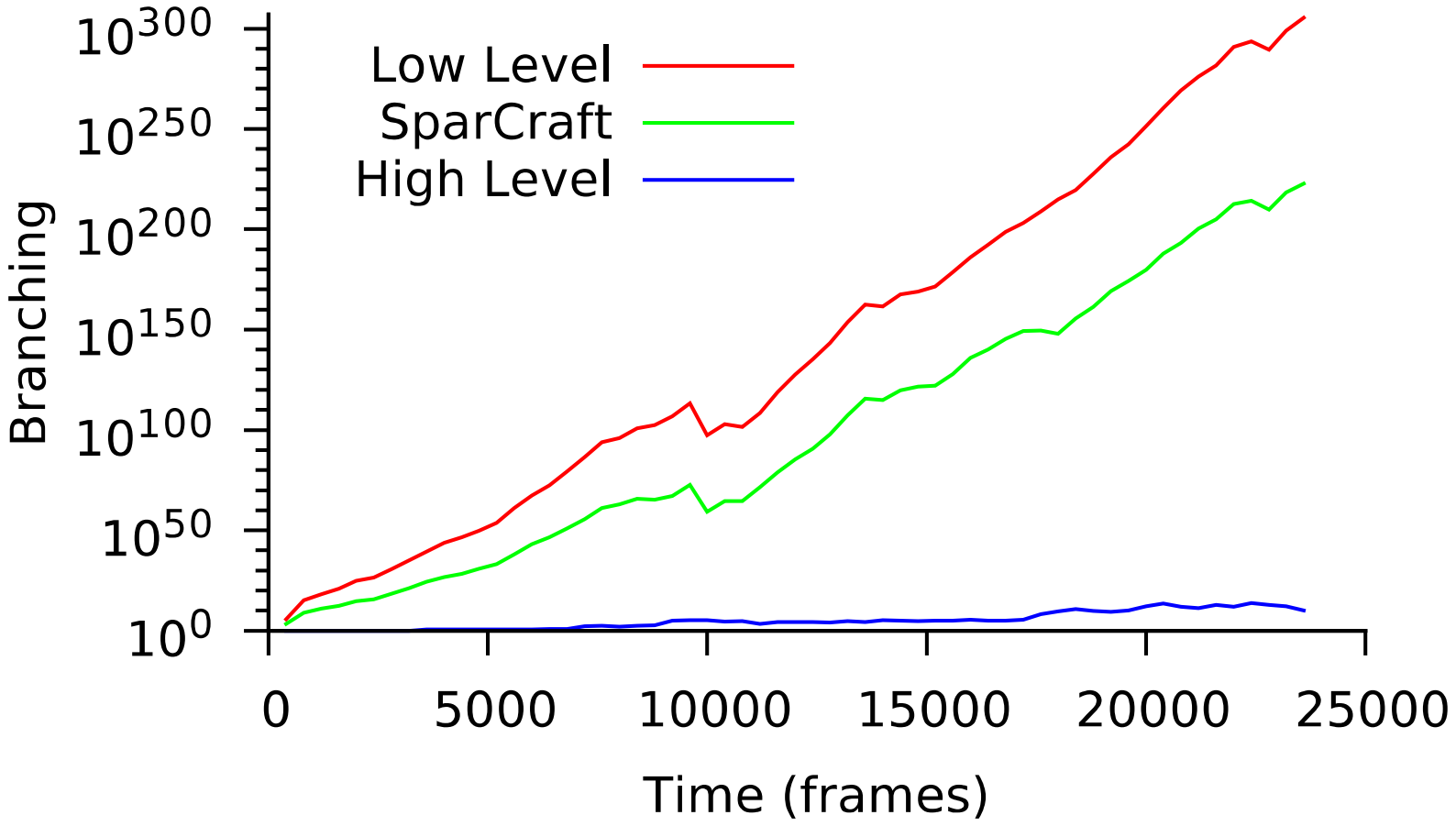
Blue payer: 47 unit
Possible actions (**1 scr.**):

1

Evaluating a script costs non-trivial time, typically O(N)!

Star Craft: Brood War

Tactical vs. Reactive Layer

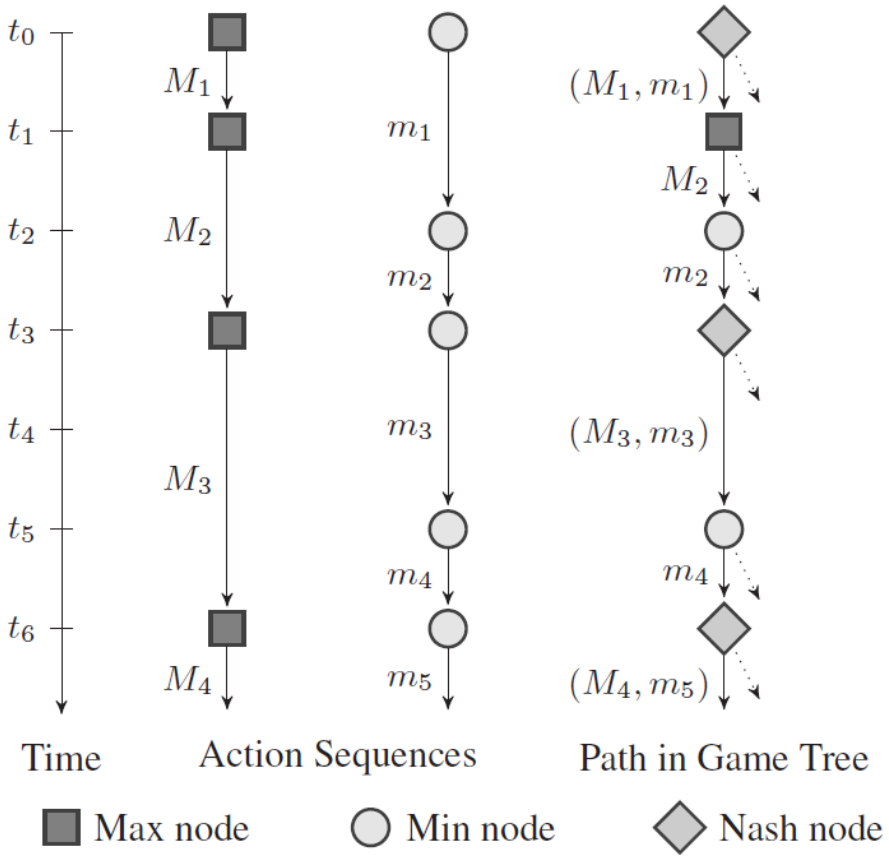


Uriarte, Alberto, and Santiago Ontañón. "Game-tree search over high-level game states in RTS games." *Tenth Artificial Intelligence and Interactive Digital Entertainment Conference*. 2014.

ALPHA-BETA CONSIDERING DURATION (ABCD)

Star Craft: Brood War

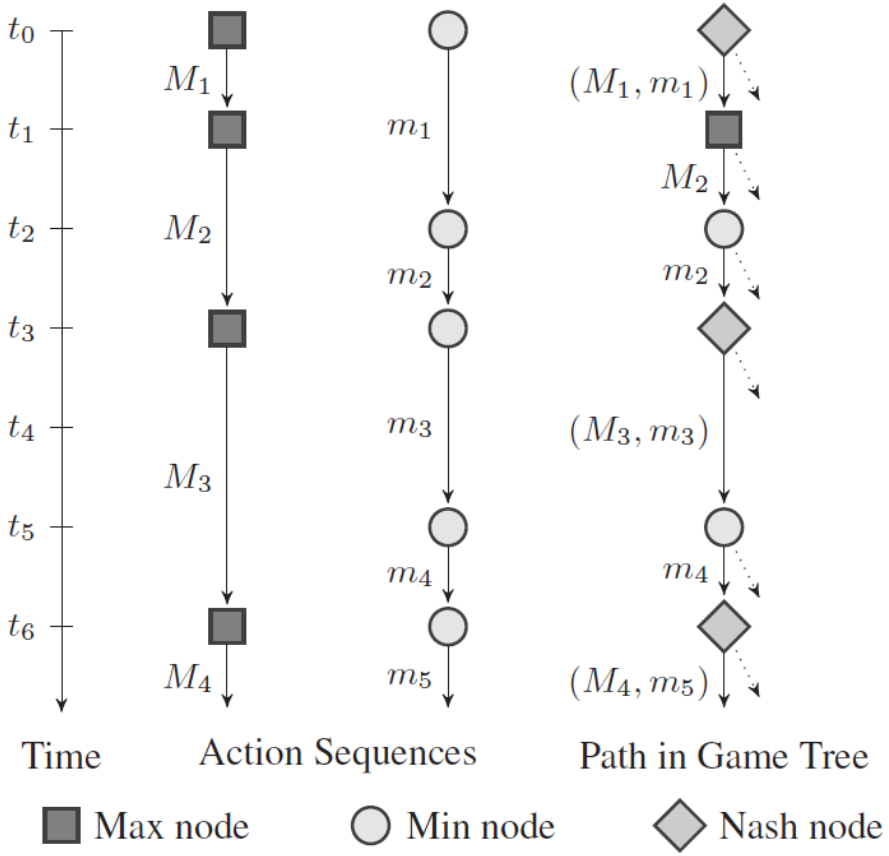
Tactical layer - ABCD

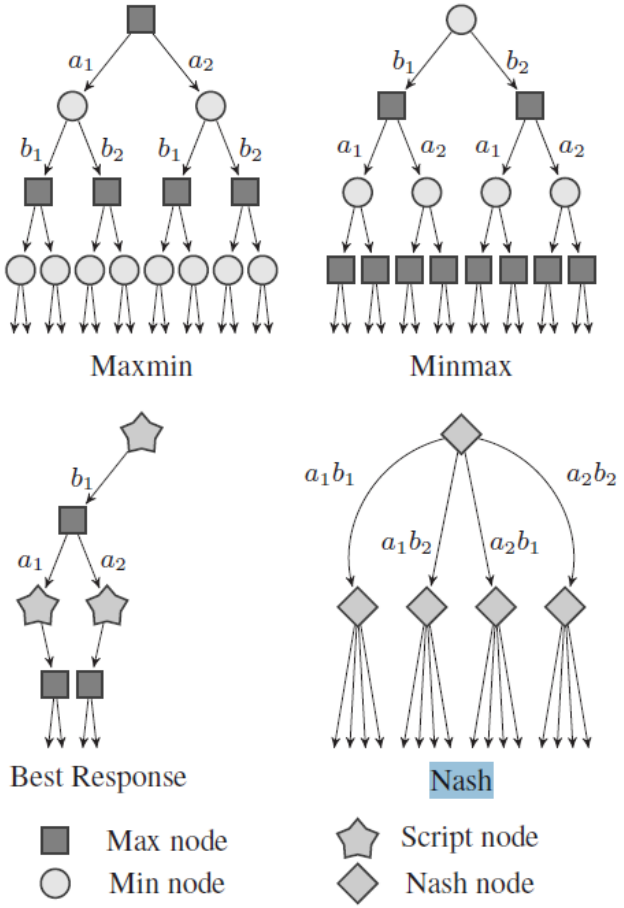


Churchill, David, Abdallah Saffidine, and Michael Buro. "Fast Heuristic Search for RTS Game Combat Scenarios." *Eighth Artificial Intelligence and Interactive Digital Entertainment Conference*. 2012.

Star Craft: Brood War

Tactical layer - ABCD





Churchill, David, Abdallah Saffidine, and Michael Buro. "Fast Heuristic Search for RTS Game Combat Scenarios." *Eighth Artificial Intelligence and Interactive Digital Entertainment Conference*. 2012.

Star Craft: Brood War

Tactical layer - ABCD

Algorithm 1 Alpha-Beta (Considering Durations)

```
1:  procedure ABCD(s, d, m_0, α, β)
2:      if computationTime.elapsed then return timeout
3:      else if terminal(s, d) then return eval(s)
4:      toMove ← s.playerToMove(policy)
5:      while m ← s.nextMove(toMove) do
6:          if s.bothCanMove and m_0 = ∅ and d ≠ 1 then
7:              val ←ABCD(s, d − 1, m, α, β)
8:          else
9:              s' ← copy(s)
10:             if m_0 ≠ ∅ then s'.doMove(m_0)
11:             s'.doMove(m)
12:             v ←ABCD(s', d − 1, ∅, α, β)
13:         if toMove = MAX and (v > α) then α ← v
14:         if toMove = MIN and (v < β) then β ← v
15:         if α ≥ β then break
16:     return toMove = MAX ? α : β
```

Churchill, David, Abdallah Saffidine, and Michael Buro. "Fast Heuristic Search for RTS Game Combat Scenarios." *Eighth Artificial Intelligence and Interactive Digital Entertainment Conference*. 2012.

Star Craft: Brood War

Tactical layer - ABCD

$$e(s) = \sum_{u \in U_1} \mathrm{hp}(u) - \sum_{u \in U_2} \mathrm{hp}(u)$$

$$\mathrm{dpf}(u) = \frac{\mathrm{damage}(w(u))}{\mathrm{cooldown}(w(u))}$$

$$\mathrm{LTD}(s) = \sum_{u \in U_1} \mathrm{hp}(u) \cdot \mathrm{dpf}(u) - \sum_{u \in U_2} \mathrm{hp}(u) \cdot \mathrm{dpf}(u)$$

$$\mathrm{LTD2}(s) = \sum_{u \in U_1} \sqrt{\mathrm{hp}(u)} \cdot \mathrm{dpf}(u) - \sum_{u \in U_2} \sqrt{\mathrm{hp}(u)} \cdot \mathrm{dpf}(u)$$

`NOK-AV(s) = NOK-AV DFS`

Churchill, David, Abdallah Saffidine, and Michael Buro. "Fast Heuristic Search for RTS Game Combat Scenarios." *Eighth Artificial Intelligence and Interactive Digital Entertainment Conference*. 2012.

Star Craft: Brood War

Tactical layer - ABCD

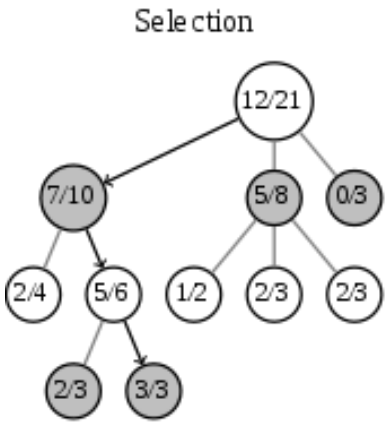
Opponent	ABCD Search Setting				
	Alt LTD	Alt LTD2	Alt NOK-AV	Alt' Playout	RAB' Playout
Random	0.99	0.98	1.00	1.00	1.00
Kite	0.70	0.79	0.93	0.93	0.92
Kite-AV	0.69	0.81	0.92	0.96	0.92
Closest	0.59	0.85	0.92	0.92	0.93
Weakest	0.41	0.76	0.91	0.91	0.89
AV	0.42	0.76	0.90	0.90	0.91
NOK-AV	0.32	0.64	0.87	0.87	0.82
Average	0.59	0.80	0.92	0.92	0.91

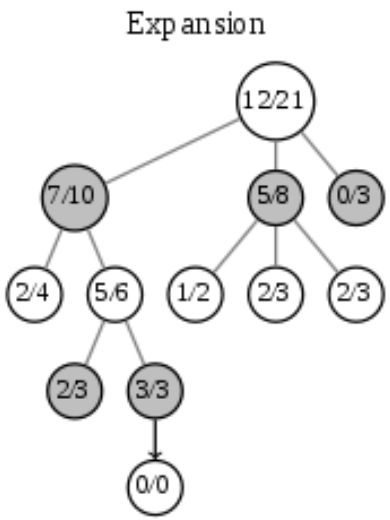
Churchill, David, Abdallah Saffidine, and Michael Buro. "Fast Heuristic Search for RTS Game Combat Scenarios." *Eighth Artificial Intelligence and Interactive Digital Entertainment Conference*. 2012.

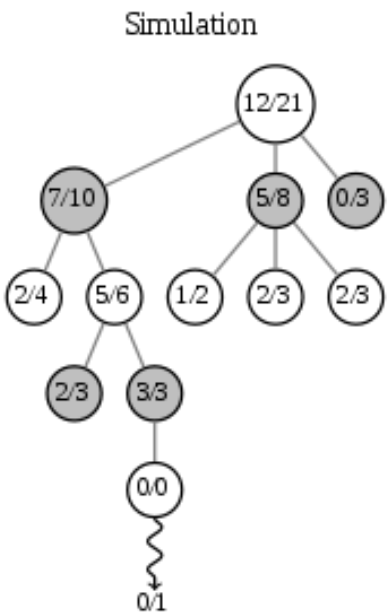
MONTE-CARLO TREE SEARCH CONSIDERING DURATION (MTCSCD)

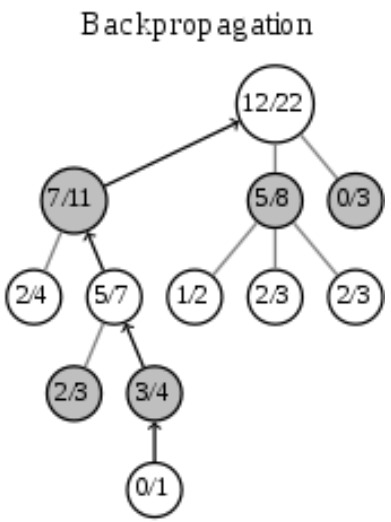
Monte-Carlo tree search
Overview

- Heuristic search algorithm, similar to minimax but expands the tree in „asymmetric" fashion
- 4 steps (Nodes are annotated `[#wins]/[#visits]`)









$$v_i + C \times \sqrt{\frac{\ln(N)}{n_i}}$$

value estimate; tunable parameter; total number of trials; num trials for arm i

Source: Wikipedia

Monte-Carlo tree search

Considering Durations (used for Tactical layer)

Algorithm 1 MCTS Considering Durations

```
1:  function MctsSearch(s_0)
2:      n_0 ← CreateNode(s_0,∅)
3:      while withing computational budget do
4:          n_l ← TreePolicy(n_0)
5:          △ ← DefaultPolicy(n_l)
6:          Backup(n_l, △)
7:      return (BestChild(n_0)).action
8:
9:  function CreateNode(s,n_0)
10:     n.parent ← n_0
11:     n.lastSimult ← n_0.lastSimult
12:     n.player ← PlayerToMove(s,n.lastSimult)
13:     if BothCanMove(s) then
14:         n.lastSimult ← n.player
15:     return n
16:
17: function DefaultPolicy(n)
18:     lastSimult ← n.lastSimult
19:     s ← n.s
20:     while withing computational budget do
21:         p ← PlayerToMove(s,lastSimult)
22:         if BothCanMove(s) then
23:             lastSimult ← p
24:         simulate game s with a policy and player p
25:     return s.reward
```

Uriarte, Alberto, and Santiago Ontañón. "Game-tree search over high-level game states in RTS games." *Tenth Artificial Intelligence and Interactive Digital Entertainment Conference*. 2014.

Monte-Carlo tree search

Considering Durations (used for Tactical layer)

Algorithm 1 MCTS Considering Durations

```
1:  function MctsSearch(s_0)
2:      n_0 ← CreateNode(s_0,∅)
3:      while withing computational budget do
4:          n_l ← TreePolicy(n_0)
5:          △ ← DefaultPolicy(n_l)
6:          Backup(n_l, △)
7:      return (BestChild(n_0)).action
```

```
9:  function CreateNode(s,n_0)
10:     n.parent ← n_0
11:     n.lastSimult ← n_0.lastSimult
12:     n.player ← PlayerToMove(s,n.lastSimult)
13:     if BothCanMove(s) then
14:         n.lastSimult ← n.player
15:     return n
16:
17: function DefaultPolicy(n)
18:     lastSimult ← n.lastSimult
19:     s ← n.s
20:     while withing computational budget do
21:         p ← PlayerToMove(s,lastSimult)
22:         if BothCanMove(s) then
23:             lastSimult ← p
24:         simulate game s with a policy and player p
25:     return s.reward
```

ε-greedy tree policy with **ε** = 0.2

Default policy = random move selection

Simultaneous node = Alt policy

Limited the depth of the tree policy to 10

MCTSCD for 2,000 playouts with a length of 7,200 game frames.

Group actions: Idle, Move adjacent, attack

Uriarte, Alberto, and Santiago Ontañón. "Game-tree search over high-level game states in RTS games." *Tenth Artificial Intelligence and Interactive Digital Entertainment Conference*. 2014.

Monte-Carlo tree search

Considering Durations (used for Tactical layer)

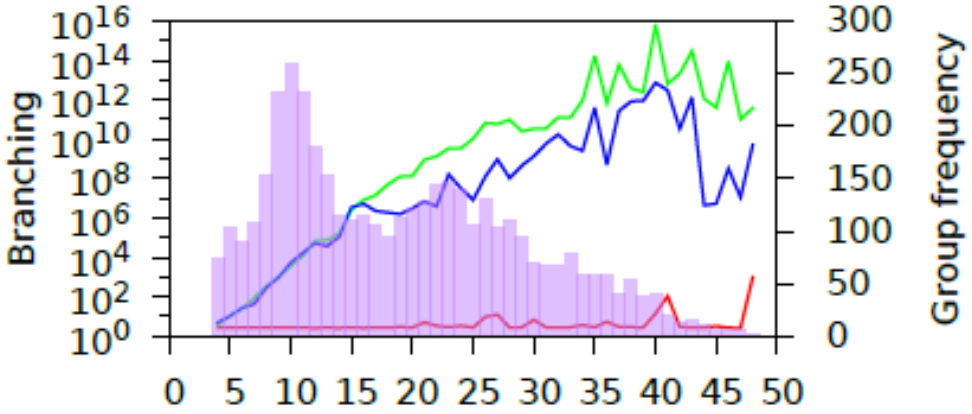
Algorithm	*Map*	*Avg. Eval*	*% > 0*	Avg. % overwrt.
Scripted	Benzene	35853.33	100.00	-
Scripted	Destination	32796.51	100.00	-
ABCD	Benzene	16020.45	81.82	83.05
ABCD	Destination	18226.32	87.72	88.05
MCTSCD	Benzene	16170.51	89.74	83.74
MCTSCD	Destination	20753.85	84.62	92.25

Uriarte, Alberto, and Santiago Ontañón. "Game-tree search over high-level game states in RTS games." *Tenth Artificial Intelligence and Interactive Digital Entertainment Conference*. 2014.

Monte-Carlo tree search

Considering Durations (used for Tactical layer)

Algorithm	*Map*	*Avg. Eval*	*% > 0*	Avg. % overwrt.
Scripted	Benzene	35853.33	100.00	-
Scripted	Destination	32796.51	100.00	-
ABCD	Benzene	16020.45	81.82	83.05
ABCD	Destination	18226.32	87.72	88.05
MCTSCD	Benzene	16170.51	89.74	83.74
MCTSCD	Destination	20753.85	84.62	92.25



Uriarte, Alberto, and Santiago Ontañón. "Game-tree search over high-level game states in RTS games." *Tenth Artificial Intelligence and Interactive Digital Entertainment Conference*. 2014.

PORTFOLIO GREEDY SEARCH (PGS)

Portfolio Greedy Search
Used for Reactive layer

Idea: let them fight against each other iterating best assignment of scripts to units while using playout to determine the outcome.

Algorithm 2 Portfolio Greedy Search

```
1:  Portfolio P                                   ▷ Script Portfolio
2:  Integer I                                     ▷ Improvement Iterations
3:  Integer R                                     ▷ Self/Enemy Improvement Responses
4:  Script D                                      ▷ Default Script
5:
6:  procedure PortfolioGreedySearch(State s, Player p)
7:      Script enemy[s.numUnits(opponent(p))].fill(D)
8:      Script self[] ← GetSeedPlayer(s, p, enemy)
9:      enemy ← GetSeedPlayer(s, opponent(p), self)
10:     self = Improve(s, p, self, enemy)
11:     for r = 1 to R do
12:         enemy = Improve(s, opponent(p), enemy, self)
13:         self = Improve(s, p, self, enemy)
        return generateMoves(self)
```

```
28: procedure Improve(State s, Player p, Script self[],
29:                                                 Script e[])
30:     for i = 1 to I do
31:         for u = 1 to self.length do
32:             if timeElapsed > timeLimit then return
33:             bestValue ← −∞
34:             Script bestScript ← ∅
35:             for Script c in P do
36:                 self[u] ← c
37:                 value ← Playout(s, p, self, e)
38:                 if value > bestValue then
39:                     bestValue ← value
40:                     bestScript ← c
41:             self[u] ← bestScript
42:     return self
```

Churchill, David, and Michael Buro. "Portfolio greedy search and simulation for large-scale combat in StarCraft." *Computational Intelligence in Games (CIG), 2013 IEEE Conference on*. IEEE, 2013.

Portfolio Greedy Search

Experiment setup

Alpha-Beta search:

- Time Limit: 40 ms
- Max Children: 20
- Evaluation: NOK-AV vs. NOK-AV Playout
- Transposition Table Size: 100000 (13.2 MB)

UCT search:

- Time Limit: 40 ms
- Max Children: 20
- Evaluation: NOK-AV vs. NOK-AV Playout
- Final Move Selection: Most Visited
- Exploration Constant: 1.6
- Child Generation: One-at-leaf
- Tree Size: No Limit (6 MB largest seen in 40 ms)

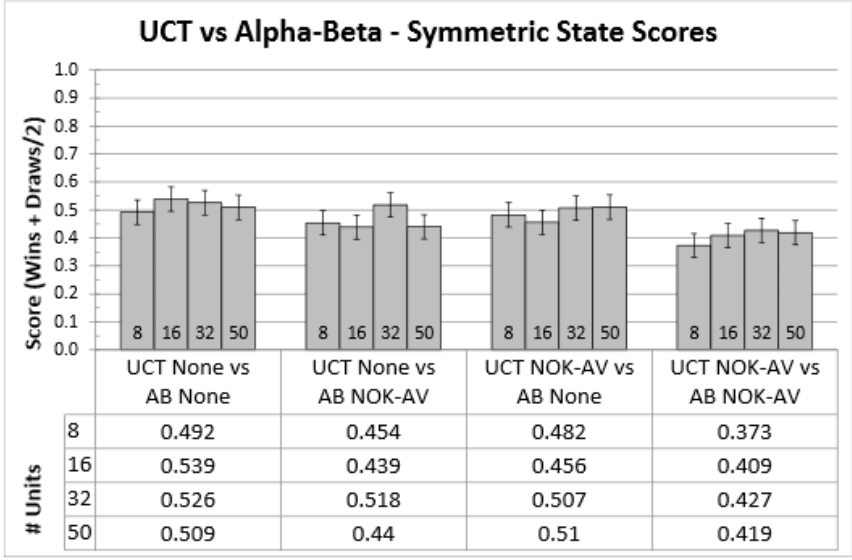
Portfolio Greedy search:

- Time Limit: 40 ms
- Improvement Iterations I: 1
- Response Iterations R: 0
- Initial Enemy Script: NOK-AV
- Evaluation: Improved Playout
- Portfolio Used: (NOK-AV, Kiter)

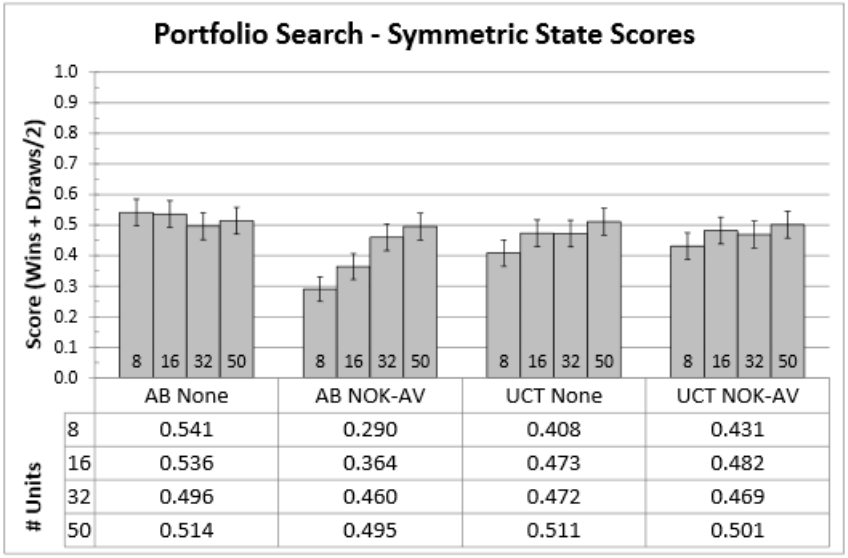
Churchill, David, and Michael Buro. "Portfolio greedy search and simulation for large-scale combat in StarCraft." *Computational Intelligence in Games (CIG), 2013 IEEE Conference on*. IEEE, 2013.

Portfolio Greedy Search

Experiment result

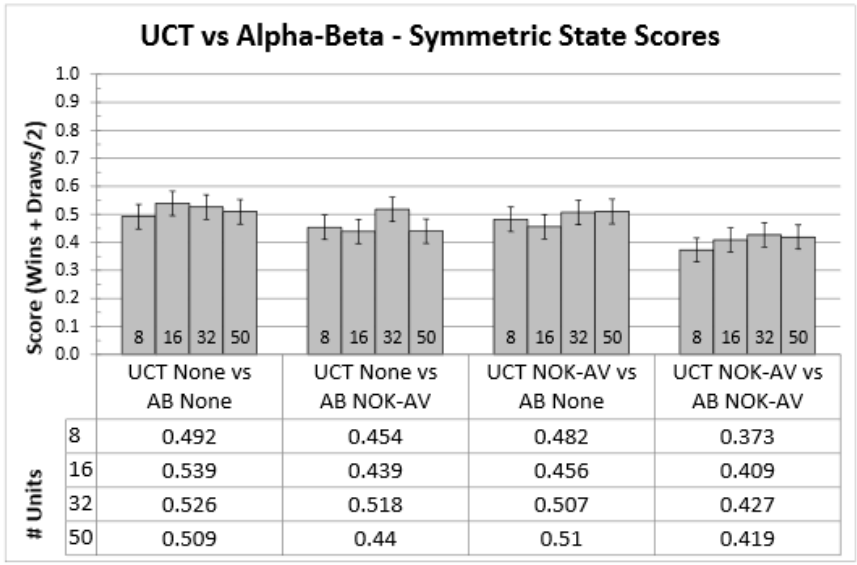


# Units	UCT None vs AB None	UCT None vs AB NOK-AV	UCT NOK-AV vs AB None	UCT NOK-AV vs AB NOK-AV
8	0.492	0.454	0.482	0.373
16	0.539	0.439	0.456	0.409
32	0.526	0.518	0.507	0.427
50	0.509	0.44	0.51	0.419

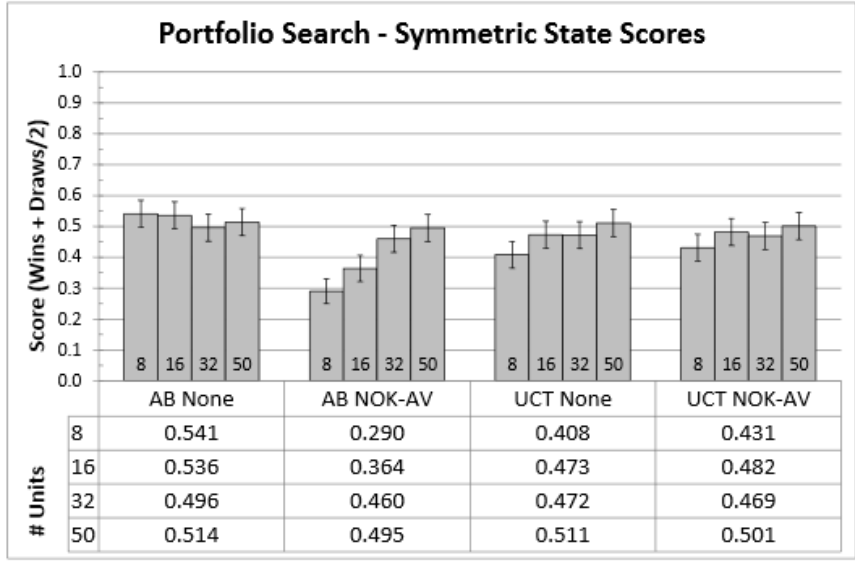


# Units	AB None	AB NOK-AV	UCT None	UCT NOK-AV
8	0.541	0.290	0.408	0.431
16	0.536	0.364	0.473	0.482
32	0.496	0.460	0.472	0.469
50	0.514	0.495	0.511	0.501

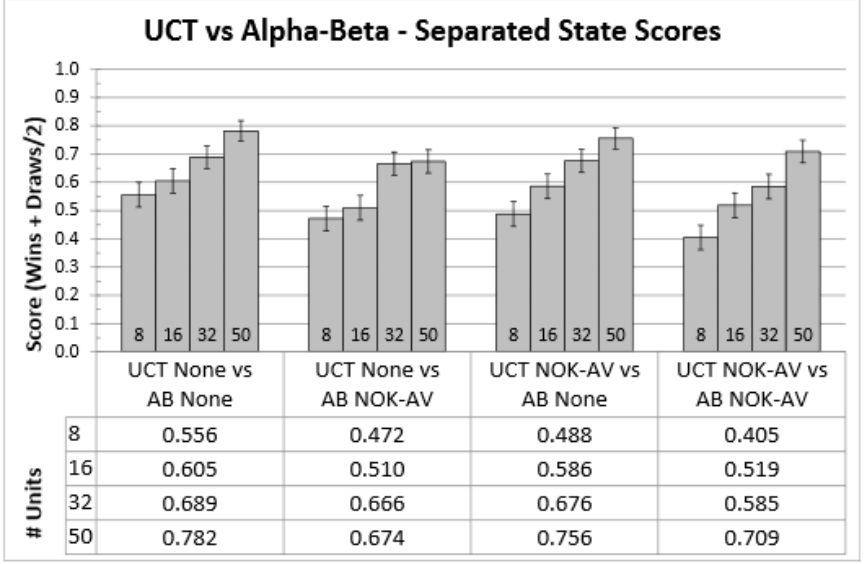
Churchill, David, and Michael Buro. "Portfolio greedy search and simulation for large-scale combat in StarCraft." *Computational Intelligence in Games (CIG), 2013 IEEE Conference on*. IEEE, 2013.



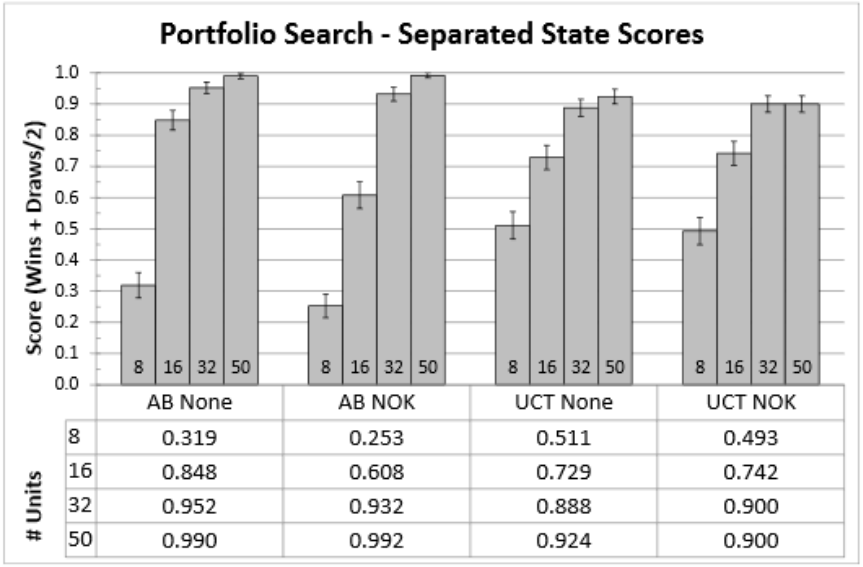
# Units	UCT None vs AB None	UCT None vs AB NOK-AV	UCT NOK-AV vs AB None	UCT NOK-AV vs AB NOK-AV
8	0.492	0.454	0.482	0.373
16	0.539	0.439	0.456	0.409
32	0.526	0.518	0.507	0.427
50	0.509	0.44	0.51	0.419



# Units	AB None	AB NOK-AV	UCT None	UCT NOK-AV
8	0.541	0.290	0.408	0.431
16	0.536	0.364	0.473	0.482
32	0.496	0.460	0.472	0.469
50	0.514	0.495	0.511	0.501



# Units	UCT None vs AB None	UCT None vs AB NOK-AV	UCT NOK-AV vs AB None	UCT NOK-AV vs AB NOK-AV
8	0.556	0.472	0.488	0.405
16	0.605	0.510	0.586	0.519
32	0.689	0.666	0.676	0.585
50	0.782	0.674	0.756	0.709

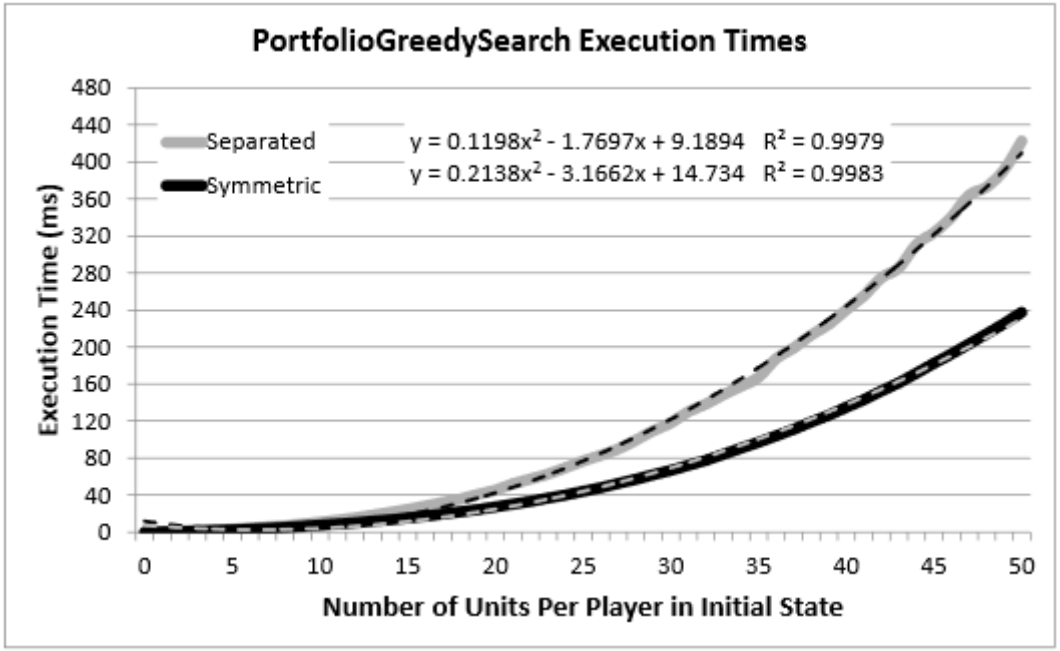


# Units	AB None	AB NOK	UCT None	UCT NOK
8	0.319	0.253	0.511	0.493
16	0.848	0.608	0.729	0.742
32	0.952	0.932	0.888	0.900
50	0.990	0.992	0.924	0.900

Churchill, David, and Michael Buro. "Portfolio greedy search and simulation for large-scale combat in StarCraft." *Computational Intelligence in Games (CIG), 2013 IEEE Conference on*. IEEE, 2013.

Portfolio Greedy Search

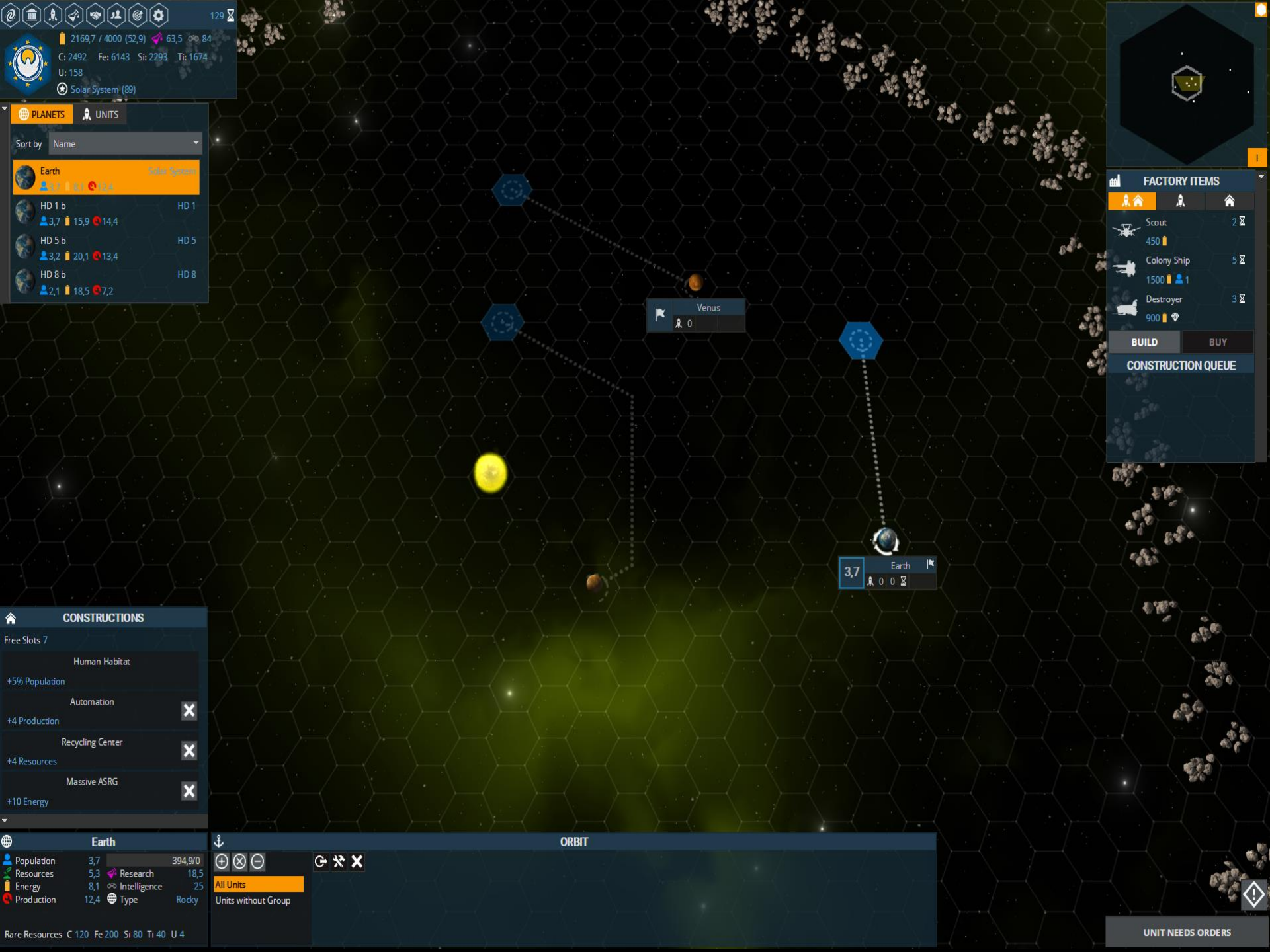
Experiment setup



Churchill, David, and Michael Buro. "Portfolio greedy search and simulation for large-scale combat in StarCraft." *Computational Intelligence in Games (CIG), 2013 IEEE Conference on*. IEEE, 2013.

CHILDREN OF THE GALAXY
(Finally ☺)


129
2169,7 / 4000 (52,9)
63,5
84
C: 2492
Fe: 6143
Si: 2293
Ti: 1674
U: 158
PLANETS
UNITS
Sort by
Name
Earth
Solar System
3,7
8,1
12,4
HD 1 b
HD 1
3,7
15,9
14,4
HD 5 b
HD 8 b
HD 8
2,1
18,5
7,2
HD 11
0
0
7
HD 2
0
0
6
EMPTY SPACE
1
0
HD 22
0
0
6
HD 24
0
0
8
HD 23
0
0
7
HD 7
0
0
6
HD 1
3,7
0
1
7
Destroyer 13
Hull
10,5
Avail Energy
10
Engine Energy
4
Weapon Energy
3,5
Warp Energy
4
Sensor Energy
1,5
Warp Speed
3
Shield Energy
1,5
Rank 1
360/1000 +10/
UNIT NEEDS ORDERS

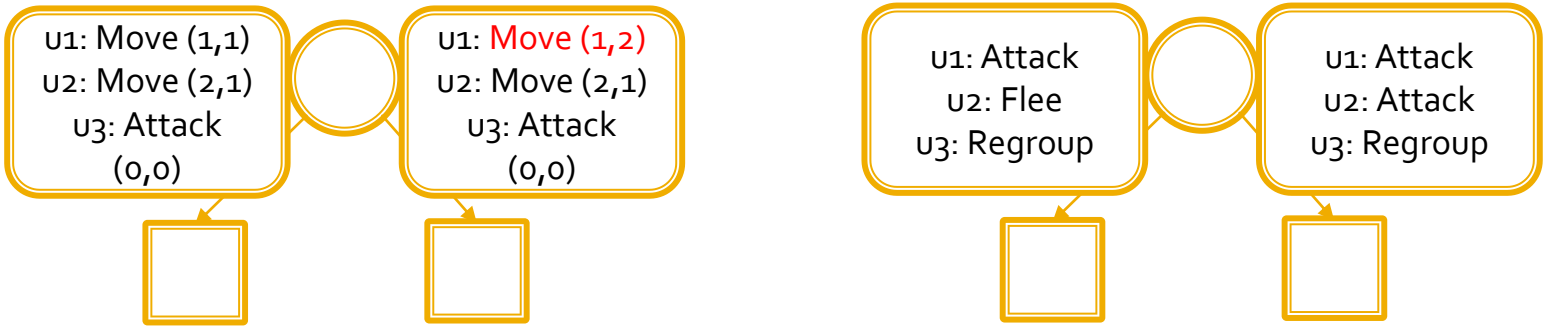

129
2169,7 / 4000 (52,9)
63,5
84
C: 2492
Fe: 6143
Si: 2293
Ti: 1674
U: 158
Solar System (89)
PLANETS
UNITS
Sort by
Name
Earth
Solar System
HD 1 b
HD 1
3,7
15,9
14,4
HD 5 b
HD 5
3,2
20,1
13,4
HD 8 b
HD 8
2,1
18,5
7,2
Venus
0
FACTORY ITEMS
Scout
2
450
Colony Ship
5
1500
1
Destroyer
3
900
BUILD
BUY
CONSTRUCTION QUEUE
3,7
Earth
0
0
CONSTRUCTIONS
Free Slots 7
Human Habitat
+5% Population
Automation
+4 Production
Recycling Center
+4 Resources
Massive ASRG
+10 Energy
Earth
Population
3,7
394,9/0
Resources
5,3
Research
18,5
Energy
8,1
Intelligence
25
Production
12,4
Type
Rocky
Rare Resources C 120 Fe 200 Si 80 Ti 40 U 4
ORBIT
All Units
Units without Group
UNIT NEEDS ORDERS


1000 (10) 4 18
C 60 Fe 100 Si 40 Ti 20 U 2
Physics 24
Skirmish (66)
ANETS
UNITS
Name
Earth Skirmish
1 10,1 1
VIDEO EXAMPLE

MCTS for CotG Combat

Script space

- Branching factor for movement of 7 units is about $2.7 * 10^{12}$
- Don't search in action space, search in script space



Branching factor for 7 units and 3 scripts is $3^7 = 2187$

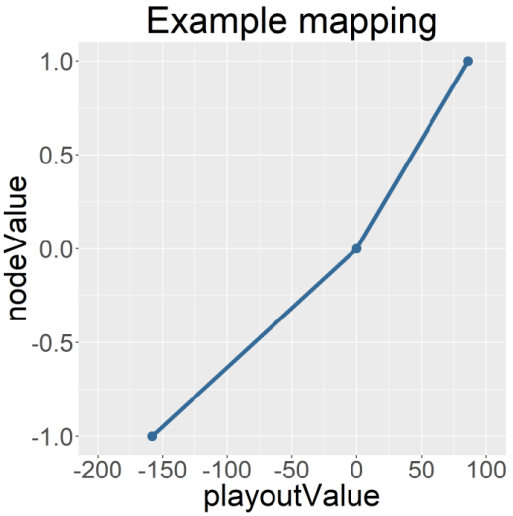
MCTS considering HP
(our small contribution)

MCTS for CotG Combat

Script space

- MCTS returns $i \in \{0, 1\}$ – lose/win
  - One bit of information
  - Statistically sufficient given many playouts
- Combat is just a subproblem

- MCTS_HP: Analyze the state and return $x \in [-1; 1]$ instead
  - Map HP remaining to interval $[-1; 1]$
  - Works for fewer playouts
  - Guides the search better



MCTS for CotG Combat

Experiment setup

- Round robin tournaments
- Various unit counts – from 3vs3 to 64vs64
- Scripts: Kiter, NOK-AV

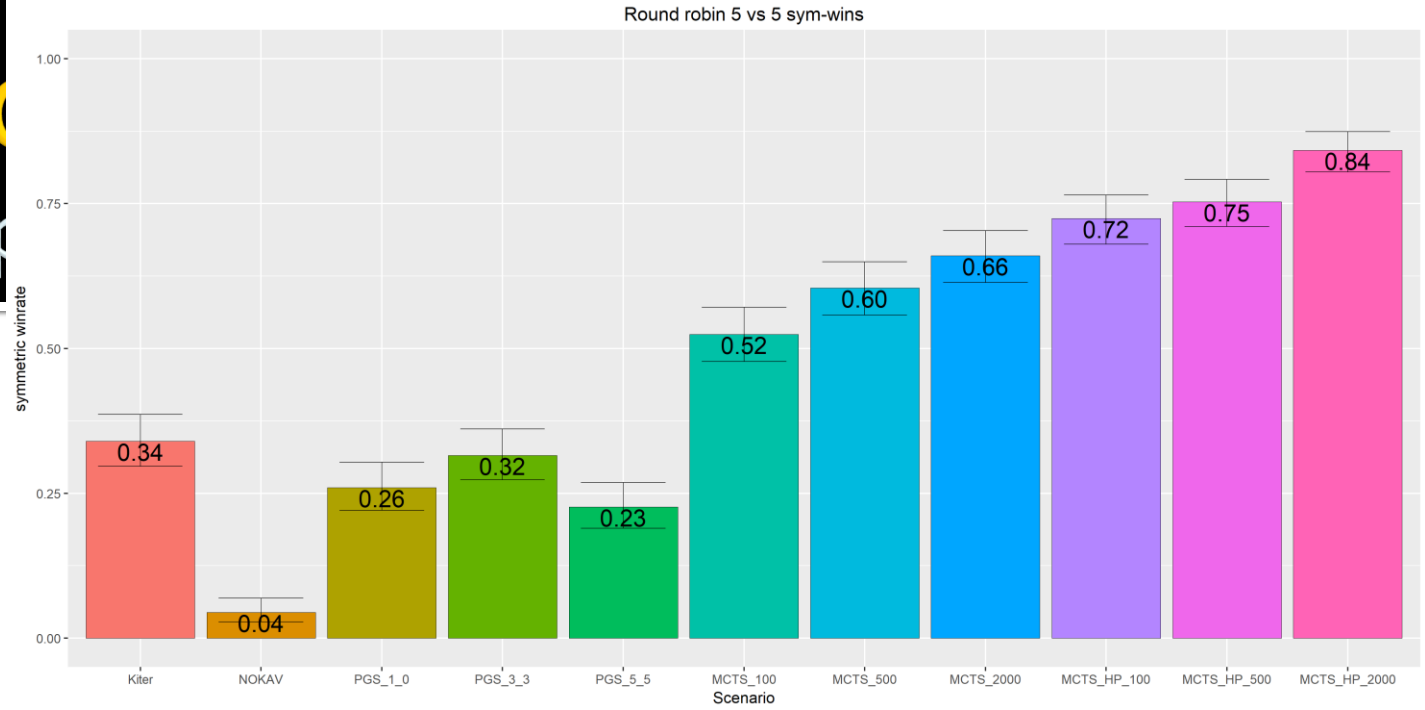
- Search methods
  - Portfolio greedy search (Churchill, Buro 2013)
    - Time limit 500ms, various I and R
  - MCTS in script space – similar to (Justesen et al. 2014)
    - Time limit: 100ms, 500ms, 2000ms
  - MCTS considering HP (our algorithm)
    - Time limit: 100ms, 500ms, 200ms

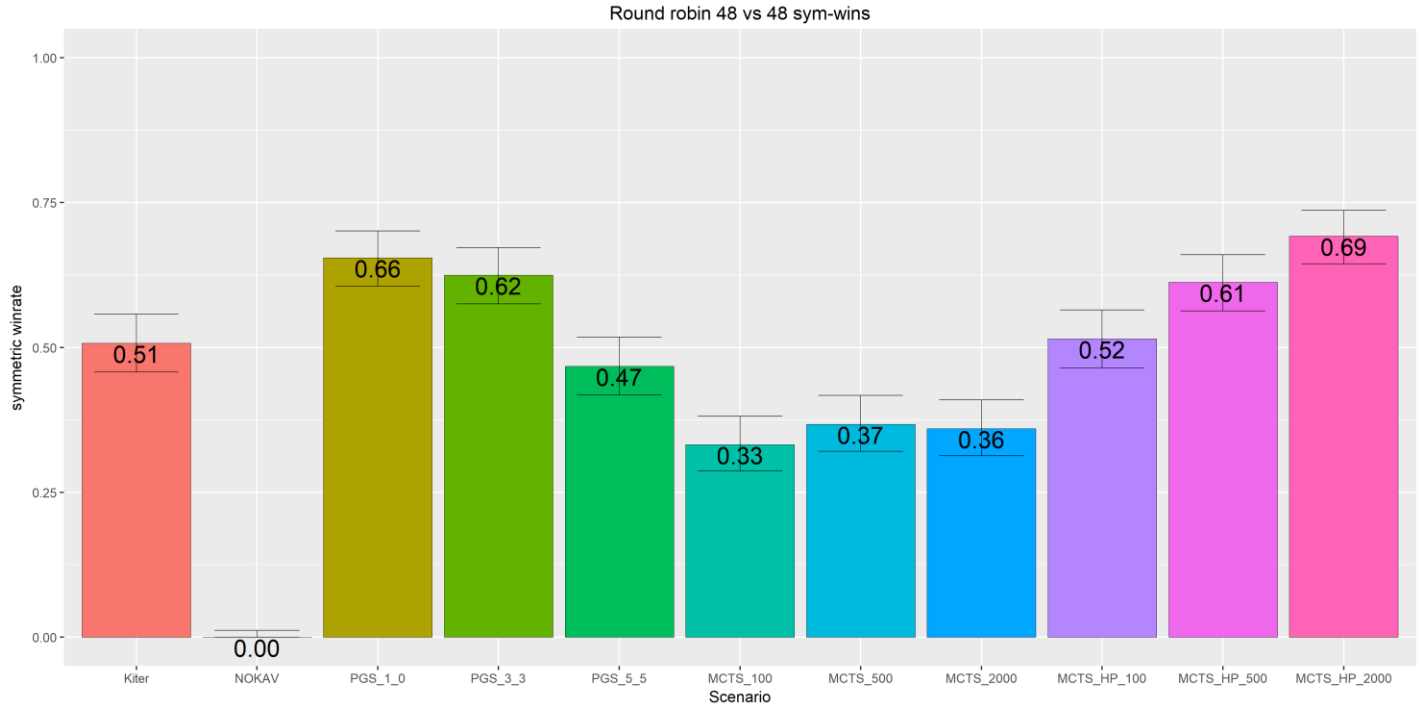
MCTS for CotG Combat

Experiment setup

Symmetric battle id	Battle 1: Player A starts	Battle 2: Player B starts	Sym result
1	A wins, HP: 10	B wins, HP: 20	B has more HP
2	A wins, HP: 12	B wins, HP: 15	B has more HP
3	A wins, HP: 5	A wins, HP: 2	A wins both
4	A wins, HP: 4	A wins, HP: 1	A wins both
5	A wins. HP: 8	B wins, HP: 11	B has more HP
Total	A: 7 wins, HP: 42 B: 3 wins, HP: 46		A: 2 sym-wins B: 3 sym-wins

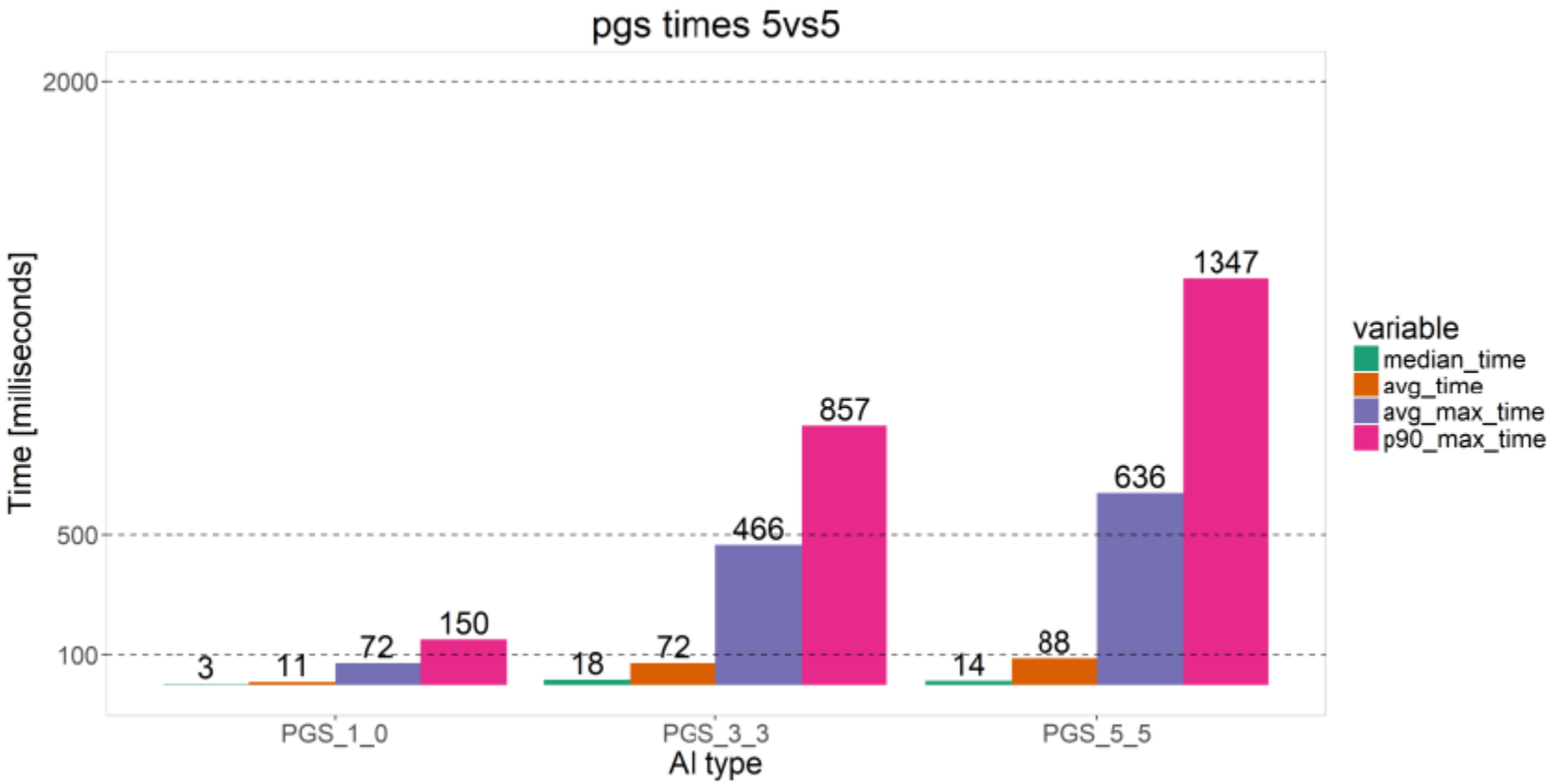
MC
Exp


Round robin 5 vs 5 sym-wins
symmetric winrate
1.00
0.75
0.50
0.25
0.00
0.34
0.04
0.26
0.32
0.23
0.52
0.60
0.66
0.72
0.75
0.84
Kiter
NOKAV
PGS_1_0
PGS_3_3
PGS_5_5
MCTS_100
MCTS_500
MCTS_2000
MCTS_HP_100
MCTS_HP_500
MCTS_HP_2000
Scenario


Round robin 48 vs 48 sym-wins
symmetric winrate
1.00
0.75
0.50
0.25
0.00
0.51
0.00
0.66
0.62
0.47
0.33
0.37
0.36
0.52
0.61
0.69
Kiter
NOKAV
PGS_1_0
PGS_3_3
PGS_5_5
MCTS_100
MCTS_500
MCTS_2000
MCTS_HP_100
MCTS_HP_500
MCTS_HP_2000
Scenario

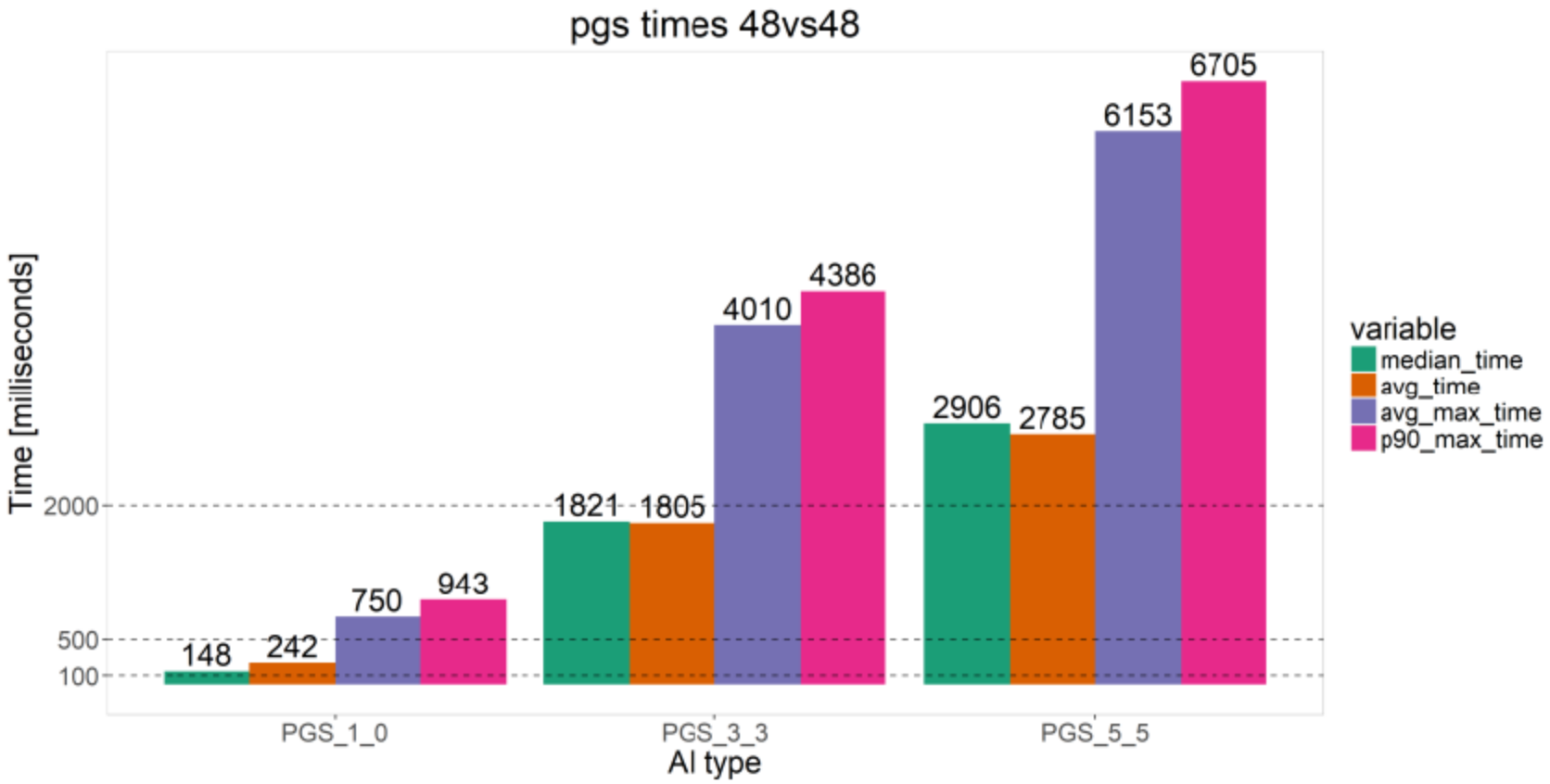
MCTS_HP for CotG Combat

Experiment setup



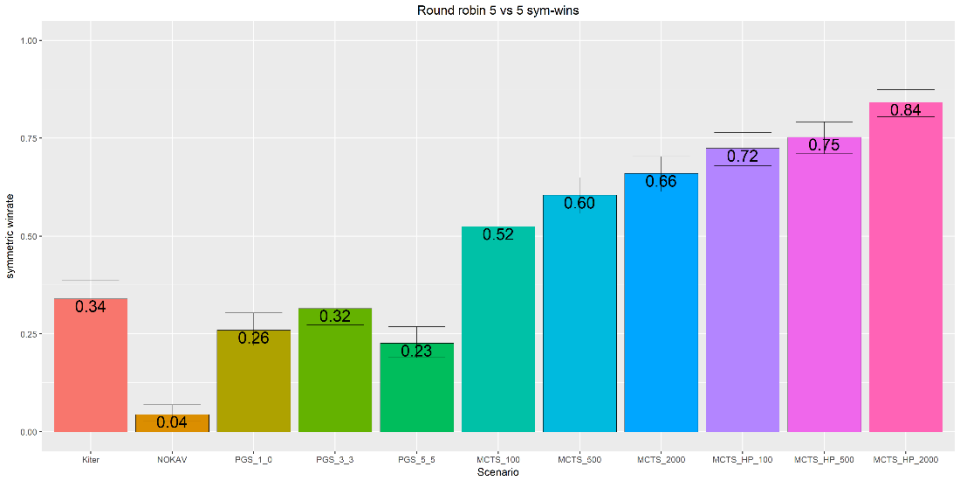
MCTS_HP for CotG Combat

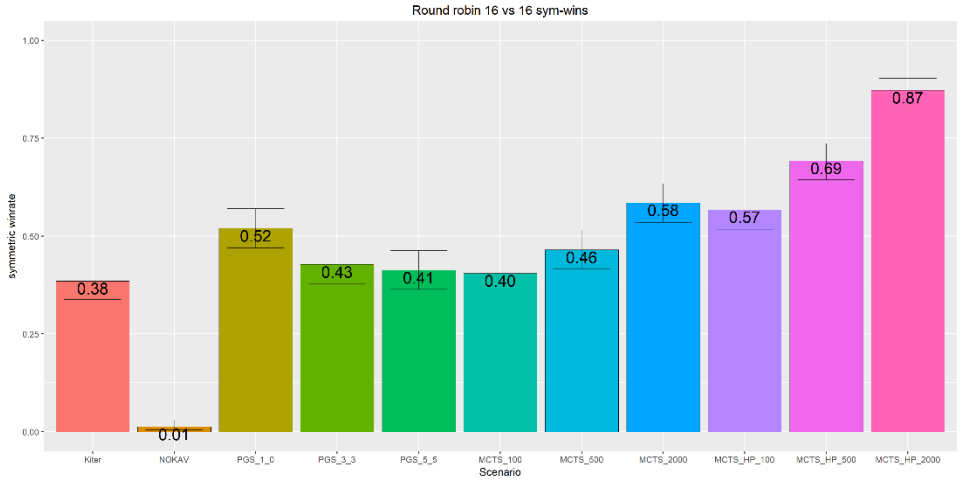
Experiment setup

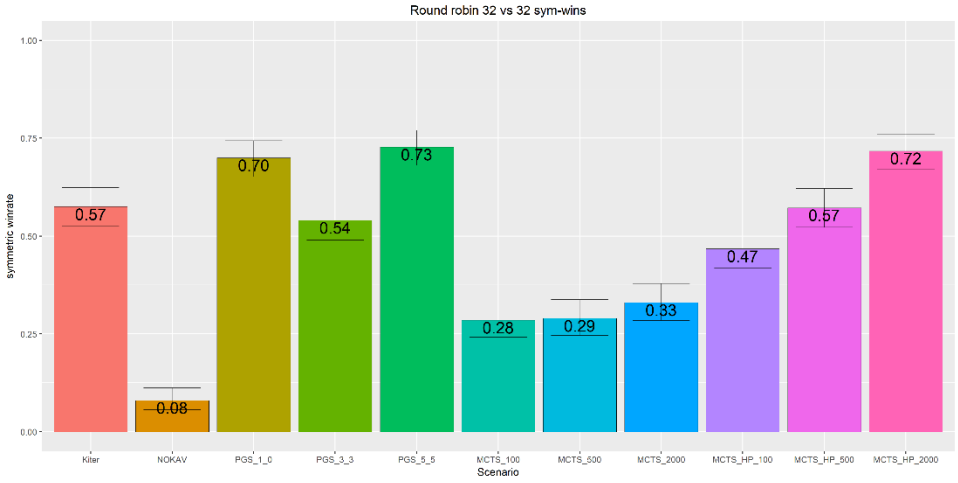


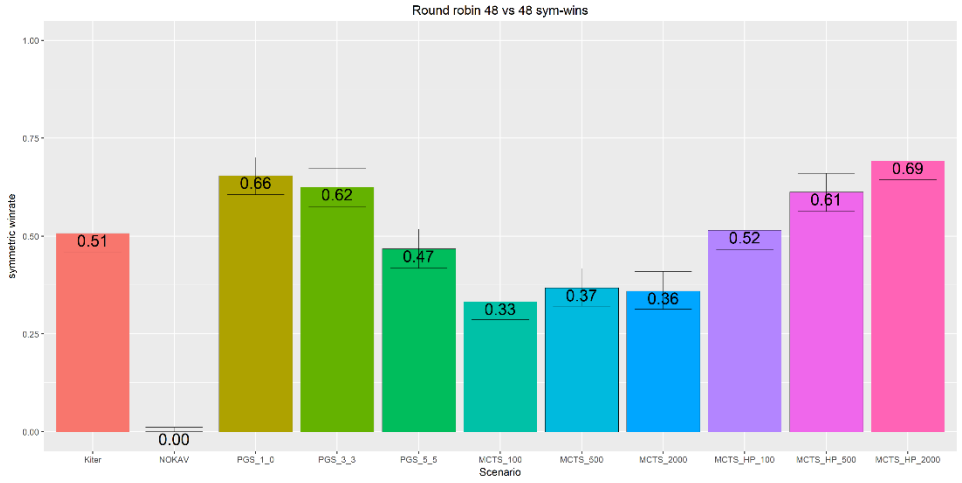
MCTS_HP for CotG Combat

Experiment setup









That’s all today ☺