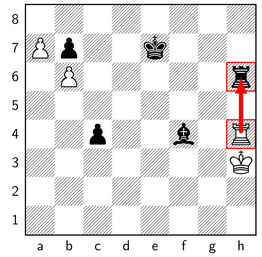
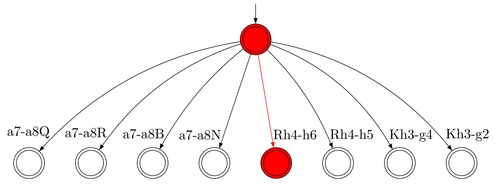
Split Moves for Monte-Carlo Tree Search

Jakub Kowalski[1], Maksymilian Mika[1], Wojciech Pawlik[1], Jakub Sutowicz[1], Marek Szykuła[1], Mark H. M. Winands[2]

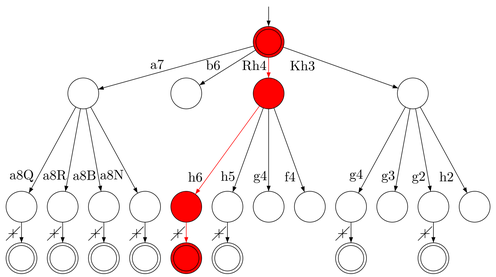
[1]University of Wrocław, Faculty of Mathematics and Computer Science [2]Maastricht University, Department of Data Science and Knowledge Engineering



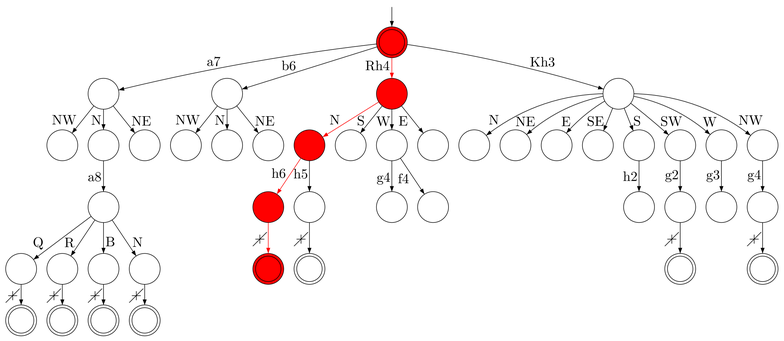
Orthodox move



Split move: *Mod* split strategy



Split move: *ModPlus* split strategy



Motivation

- Applicable to *any* game-playing algorithm: Monte-Carlo Tree Search, Min-Max, evolutionary search, neural networks, ...
- Main effects:
  - Improving efficiency.
  - Reducing branching factor.
  - Sharing information between moves.

Variants

- Different method in selection and simulation phases.
  - Orthodox (selection or simulation).
  - Semisplit (selection or simulation).
  - Roll-up (selection) – progressive switch from semisplit to orthodox.
- *raw* or *nodal* expansion (one node or with a full move).
- Heuristics MAST and RAVE:
  - Split (statistics on semimoves).
  - Join (statistics on full moves).
  - Context (statistics on prefixes of moves).
  - Mixed (both Split and Context, weighted).

MCTS variant		Standard		Split		Join		Context	
Tree	Simulation	MAST	RAVE	MAST	RAVE	MAST	RAVE	MAST	RAVE
orthodox	orthodox	✓	✓	✓	–	✓	✓	✓	✓
orthodox	semisplit	–	–	✓	–	–	✓	✓	✓
semisplit	orthodox	–	–	✓	✓	✓	–	✓	✓
semisplit	semisplit	✓	✓	✓	✓	–	–	✓	✓
roll-up	orthodox	–	–	✓	–	✓	–	✓	✓
roll-up	semisplit	–	–	✓	–	–	–	✓	✓

- Split strategies of different granularity:
  - Mod (split at each modification of a board square / game variable).
  - Plus (split at each decision).
  - Shift (split during selecting a board square).

Implementation

- Based on *Regular Boardgames* General Game Playing system.
- The compiler generates a reasoner with split moves according to the given *split strategy*.
- *Just-in-time* compilation which takes into account the game rules, the algorithm of the agent, configuration parameters.
- Many optimizations, dedicated data structures.
- Over 700 available configurations.

Future work

- Applying to other algorithms.
- How to select the best variant and split strategy for the given game?
- How strongly splitting the game can distort the agent's results, and how hard is the problem?

Differences in speed

Game	Orthodox MCTS		Semisplit MCTS	
	States/sec.	Avg. branching factor	Speed-up factor	Avg. branching factor
Amazons	236 269	457.40	11.42	6.91
Breakthrough	2 495 330	25.69	2.03	7.70
Breakthru	11 088	12 958.00	174.38	12.13
Chess	285 631	22.80	4.96	2.96
Chess no-check	710 881	33.22	3.89	3.92
English Draughts	4 411 795	5.22	1.09	2.50
Fox and hounds	13 940 118	4.12	0.95	2.65
Go	173 452	130.35	0.33	72.56
Knightthrough	2 159 981	37.32	2.51	8.65
Pentago	492 027	171.24	3.33	15.09
Skirmish	679 837	34.35	4.07	4.29
The Mill Game	2 298 726	14.85	1.91	4.25

Win Ratios

Tree	Simulation	Strategy	Amazons	Breakthrough	Breakthru	Chess	Chess (no check)	English Draughts	Fox And Hounds	Go	Knightthrough	Pentago	Skirmish	The Mill Game
Orthodox	Split	Mod	86	64	63	96	56	55	48	59	82	72	81	43
Orthodox	Split	ModPlus	73	63	61	97	75	52	45	52	79	72	83	38
Split-nodal	Split	ModPlus	89	49	100	96	62	44	49	83	87	88	83	40
Roll-up-nodal	Orthodox	Mod	44	44	97	42	19	51	51	46	55	55	48	49
Split-nodal, RAVE-split	Split, MAST-split	Mod	66	38	100	97	22	43	61	66	71	68	53	17
Roll-up-nodal	Split, MAST-split	Mod	67	79	100	93	60	68	32	65	82	74	79	21

Different timelimits and equivalent state budgets