General Game Playing

By:
Mohit Agarwal
Ujjwal K. Singh

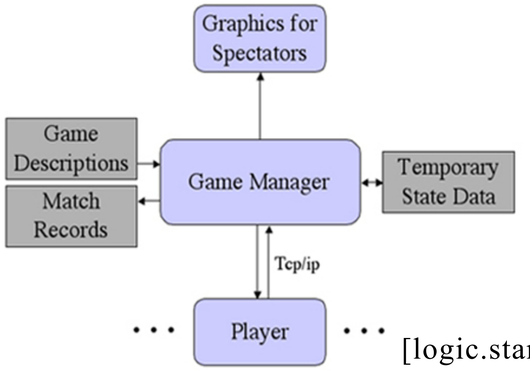
Overview

Definition

Intelligent agents capable of playing more than one game successfully at any expert level without any human intervention and any prior knowledge of game

Game Descriptive Language(GDL)

A formal logic programming language used to communicate rules of the game and conceptualization of game in terms of entities, actions and players.



[logic.stanford.edu]

Previous Work and Our Work

Remarkable research progress in field after organization of GGP competition in AAAI conference from 2005.

Clune Player Winner of 2005	CadiaPlayer Winner of 2007, 08 and 12	Ary Winner of 2009 and 10
Used Heuristic Evaluation functions that represent exact value of specified games and using that as an approximation of original game	Cadia Player employs UCT search for playing any game.It traverse through tree by decending down using random sampling and UCT selection policy.	Ary employed basic implementation of Monte-Carlo Tree Search.

Our Work:

We have used the source code for basic player in java from www.ggp-potsdam.de/browser/basicplayer/Basic.tar.gz?format=raw
We have integrated Monte Carlo Search Tree with Upper Confidence Bound.

For implementation of MCTS we have used the source http://mcts.ai/code which has the explanation of the alogorithm and its implementation.

Advantage
It optimizes the best move in initial phase of game.

Future Work
If for a simulation game reaches non-terminating path then it results in timeout. This problem can be solved by early cutoff by limiting the no. of moves in simulation phase to reach terminating state.

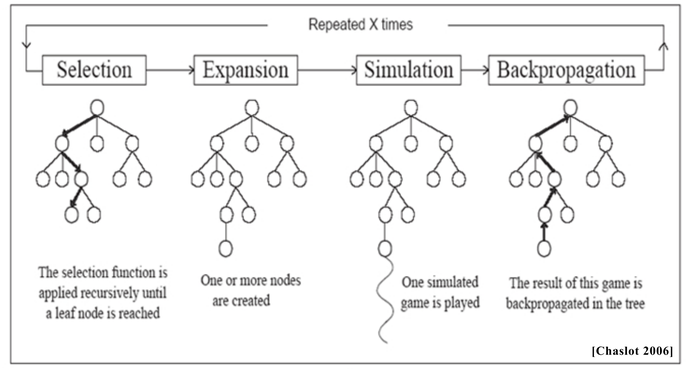
At initial phase there should be the need to explore more and exploit less and at end phase explore less and exploit more that can be achieved by adjusting contant in UCB formula but this extension degrades the player.

We are thinking to exploit tree in reverse order but we are not sure of the outcome

Algorithm and Results

Each node contain two important pieces of information: an estimated value based on simulation results and the no. of times it has been visited.

Selection of nodes is done on the basis of UCB formula



$$k = argmax_{i \in I}\left(v_i + C \times \sqrt{\frac{\ln n_p}{n_i}}\right)$$

C maintains balance between exploitation and exploration part

Main Reference:
Hilmar Finnsson (2012)Generalized Monte-Carlo Tree Search Extensions for General Game Playing
http://cadia.ru.is/wiki/_media/public:cadiaplayer:hif_aaai12a.pdf

Results **StartClock: 10s and PlayClock: 5s**

Game(Single Player)	Random	Legal	Minimax	Basic Player	Our Player
Buttons	Runtime: 32ms Score: Lose	Runtime: 39ms Score: Lose	Runtime: 195ms Score: Lose	32466ms Score: Win	32465ms Score: Win
Maze	10 steps Lose	10 steps Lose	10 steps Lose	7 steps Win	7 steps Win
Snake	6ms Score(100): 6	19ms 15	265ms 15	63779ms 42	106726ms 100
Peg	215ms 20	215ms 0	51645ms 0	71529ms 48	106674ms 100
memory_small	81ms 25	72ms 0	344ms 0	24640ms 75	24632ms 100
blocks_world	3ms 0	2ms 0	200ms 0	20726ms 100	20723ms 100

Game	Vs Random	Vs Basic Player
Tictactoe	Results : 100 0 runtime (in ms):103609 23	results: 100 0 runtime (in ms): 36372 36376
		results: 50 50 runtime (in ms): 44243 44242
Checkers	results: 100 40 runtime (in ms):709219 771	results: 80 100 runtime (in ms): 393917 392234
		100/90 337430ms 337234ms
Mini Chess	Results: 100 0 runtime (in ms):85731 23	results: 100 0 runtime (in ms): 20755 20748
		results: 100 0 runtime (in ms): 20755 20751
Conn4	Results : 100 0 runtime (in ms): 95051 317	results: 100 0 runtime (in ms): 98990 99002
		results: 0 100 runtime (in ms): 118648 118692

Conclusion

As, our player is improved version of MCST i.e. integrated with UCB, we are expected to have more intelligent gaming agent than BasicPlayer(which employed simple Monte Carlo).