Real-Time Monte Carlo Tree Search in *Ms Pac-Man*

Original authors: Tom Pepels, Mark H. M. Winands, Marc Lanctot

December 7, 2017

Background

Ms Pac-Man

Ms Pac-Man is one of *Pac-Man* game series.

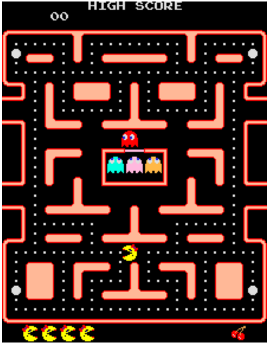


Figure: Snapshot of *Ms Pac-Man*

Background

Ms Pac-Man

- Players goal is to avoid being eaten by ghosts and gaining higher scores by collecting pills.
- In *Ms Pac-Man*, there are four different mazes, and the behavior of the ghost is unpredictable.
- *Ms Pac-Man* has become an interesting topic for AI research.

Monte Carlo Tree Search

Introduction

- Monte Carlo Tree Search (MCTS) is a best-first search method based on random sampling of the state space for a specified domain.
- MCTS has been successfully applied in many turn-based games, such as *Go* and *Hex*.
- In MCTS, a tree is built incrementally, and each node keeps statistics corresponding to the reward of that node, and times the nodes have been visited.
- MCTS copes well when limited time is available between moves, because it can stop anytime to select a move.

Monte Carlo Tree Search

How it works

MCTS consists of four steps, which are performed iteratively.

1. Starting from the root node, choose a child according to some policy, iterating until a leaf node that does not represent a terminal state is reached.
2. Add children to the selected node given available moves.
3. Run a simulated playout from the current state randomly, or according to a heuristic strategy, until a terminal is reached.
4. The result is propagated backward to update the information of the nodes.

Monte Carlo Tree Search

How it works

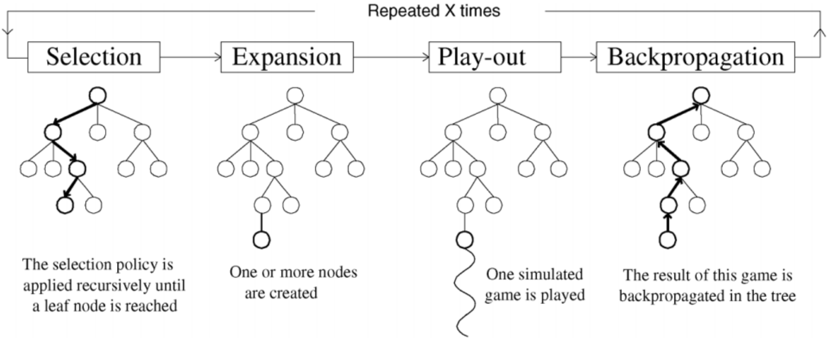

Repeated X times
Selection
Expansion
Play-out
Backpropagation
The selection policy is applied recursively until a leaf node is reached
One or more nodes are created
One simulated game is played
The result of this game is backpropagated in the tree

Figure: Four steps of Mote Carlo tree search

MCTS in *Ms Pac-Man*

Unlike most of the turn-base games, *Ms Pac-Man* is a real-time strategy game. Such games are usually considered more complex than turn based games, including

- The existence of randomness adds uncertainty to the game;
- The state space is usually very large;
- The game is open-ended.

MCTS in *Ms Pac-Man*

Unlike most of the turn-base games, *Ms Pac-Man* is a real-time strategy game. Such games are usually considered more complex than turn based games, including

- The existence of randomness adds uncertainty to the game;
- The state space is usually very large;
- The game is open-ended.

We would like to apply MCTS framework, but we need to add some enhancements to make it work better.

MCTS in *Ms Pac-Man*

Search Tree and Variable Depth

Each node stores reward values for different tactics.

The cumulative sum of rewards and mean reward are

$$S^p_{tactic} = \sum_{n=1}^{N} R^P_{tactic,n}, \overline{S}^p_{tactic} = \frac{1}{N} \tag{1}$$

The maximum mean reward is defined as

$$M^p_{tatic} = \begin{cases} \overline{S}^p_{tactic} & \text{if p is a leaf} \\ -\infty & \text{if p is not in the tree} \\ \max_{i \in C(p)} M^i_{tactic} & \text{otherwise} \end{cases} \tag{2}$$

MCTS in *Ms Pac-Man*

Search Tree and Variable Depth

The search path is variably determined by a distance limit T_{path}. A leaf is only expanded is the length of the path to the root node does not exceed T_{path}.

- this might enable the agent to find safer path when in danger;
- the scoring potentia over all possible paths in the tree is normalized due to the uniform length of each path.

MCTS in *Ms Pac-Man*

Tactics

At any time, one of the following tactics is active:

- If the survival rate is below a threshold, the survival tactic is used;
- Otherwise, if a power pill is eaten and edible ghosts are in the range of Pac-Man, the ghost score tactic is selected;
- Otherwise, the default pill score tactic is applied.

MCTS in *Ms Pac-Man*

Search tree reuse

There are two ways to reuse the tree:

Rule-based reuse Unless some special situations occur, the search tree is preserved;

Continuous decay The values stored in nodes are not discarded, but multiplied by a decay factor λ. Simulation results suggest that decaying these values $(0 < \lambda < 1)$ can be better compared to no decay $(\lambda = 0)$ and no reuse $(\lambda = 1)$.

MCTS in *Ms Pac-Man*

Selection and Expansion

The policy that determines which child to select is the one that maximized the following equation

$$X_i = v_i + C\sqrt{\frac{\ln n_p}{n_i}}$$

When one or more of the children's visit counts are below a threshold, a random uniform selection is made.

MCTS in *Ms Pac-Man*

Simulation

It is neither necessary nor computationally possible to run numerous simulations until the game terminates within strict time limit.

So, during playout, moves are made until one of the following conditions applies:

- A preset number of time units T_{time} have passed;
- Pac-Man is considered dead;
- The next maze is reached.

MCTS in *Ms Pac-Man*

Simulation

The goals for Pac-Man are

- Keep survived;
- Eat more pills;
- Eat more ghosts.

The goals for ghosts are

- Ensure that Pac-Man loses a life by trapping her;
- Avoid being eaten by Pac-Man;
- Limit the numbers of pills Pac-Man can eat.

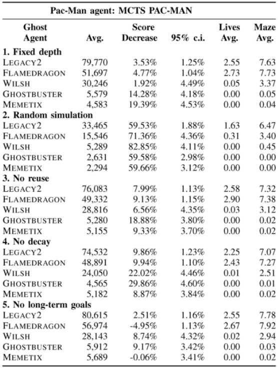
MCTS in *Ms Pac-Man*

Backpropagation and move selection

- If the maximal survival rate is below the threshold, survival tatic should be applied;
- Otherwise, scores are determined based on the current tactic;
- If the current tactic provides no feasible reward, it is replaced according to the ordering: ghost, pill, survival.

Conclusion

Result



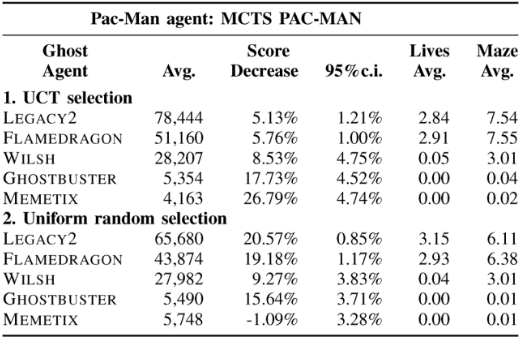
Pac-Man agent: MCTS PAC-MAN

Ghost		Score		Lives	Maze
Agent	Avg.	Decrease	95% c.i.	Avg.	Avg.
1. Fixed depth					
Legacy2	79,770	3.53%	1.25%	2.55	7.63
Flamedragon	51,697	4.77%	1.04%	2.73	7.73
Wilsh	30,246	1.92%	4.49%	0.05	3.37
Ghostbuster	5,579	14.28%	4.18%	0.00	0.05
Memetix	4,583	19.39%	4.53%	0.00	0.04
2. Random simulation					
Legacy2	33,465	59.53%	1.88%	1.63	6.47
Flamedragon	15,546	71.36%	4.36%	0.31	3.40
Wilsh	5,289	82.85%	4.11%	0.00	0.45
Ghostbuster	2,631	59.58%	2.98%	0.00	0.00
Memetix	2,294	59.66%	3.12%	0.00	0.00
3. No reuse					
Legacy2	76,083	7.99%	1.13%	2.58	7.32
Flamedragon	49,332	9.13%	1.15%	2.90	7.38
Wilsh	28,816	6.56%	4.35%	0.03	3.12
Ghostbuster	5,280	18.88%	3.80%	0.00	0.02
Memetix	5,155	9.33%	3.70%	0.00	0.02
4. No decay					
Legacy2	74,532	9.86%	1.23%	2.25	7.07
Flamedragon	48,891	9.94%	1.10%	2.43	7.27
Wilsh	24,050	22.02%	4.46%	0.01	2.51
Ghostbuster	4,565	29.86%	4.60%	0.00	0.01
Memetix	5,182	8.87%	3.84%	0.00	0.02
5. No long-term goals					
Legacy2	80,615	2.51%	1.16%	2.55	7.78
Flamedragon	56,974	-4.95%	1.13%	2.67	7.92
Wilsh	28,143	8.74%	4.32%	0.02	2.94
Ghostbuster	5,912	9.17%	3.42%	0.00	0.03
Memetix	5,689	-0.06%	3.41%	0.00	0.02

Figure: Single enhancement disabled

Conclusion

Result

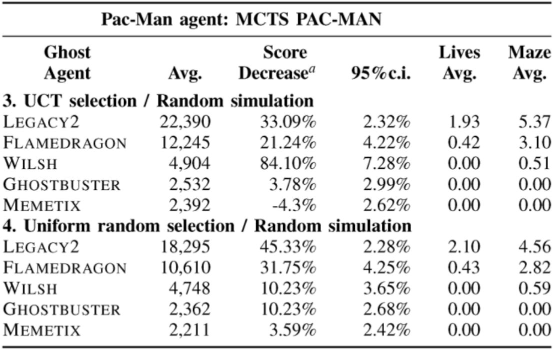


Pac-Man agent: MCTS PAC-MAN					
Ghost Agent	**Avg.**	**Score Decrease**	**95%c.i.**	**Lives Avg.**	**Maze Avg.**
1. UCT selection					
Legacy2	78,444	5.13%	1.21%	2.84	7.54
Flamedragon	51,160	5.76%	1.00%	2.91	7.55
Wilsh	28,207	8.53%	4.75%	0.05	3.01
Ghostbuster	5,354	17.73%	4.52%	0.00	0.04
Memetix	4,163	26.79%	4.74%	0.00	0.02
2. Uniform random selection					
Legacy2	65,680	20.57%	0.85%	3.15	6.11
Flamedragon	43,874	19.18%	1.17%	2.93	6.38
Wilsh	27,982	9.27%	3.83%	0.04	3.01
Ghostbuster	5,490	15.64%	3.71%	0.00	0.01
Memetix	5,748	-1.09%	3.28%	0.00	0.01

Figure: Depth-1 search, simulation strategy

Conclusion

Result



Pac-Man agent: MCTS PAC-MAN					
Ghost Agent	Avg.	Score Decrease[a]	95%c.i.	Lives Avg.	Maze Avg.
3. UCT selection / Random simulation					
LEGACY2	22,390	33.09%	2.32%	1.93	5.37
FLAMEDRAGON	12,245	21.24%	4.22%	0.42	3.10
WILSH	4,904	84.10%	7.28%	0.00	0.51
GHOSTBUSTER	2,532	3.78%	2.99%	0.00	0.00
MEMETIX	2,392	-4.3%	2.62%	0.00	0.00
4. Uniform random selection / Random simulation					
LEGACY2	18,295	45.33%	2.28%	2.10	4.56
FLAMEDRAGON	10,610	31.75%	4.25%	0.43	2.82
WILSH	4,748	10.23%	3.65%	0.00	0.59
GHOSTBUSTER	2,362	10.23%	2.68%	0.00	0.00
MEMETIX	2,211	3.59%	2.42%	0.00	0.00

Figure: Depth-1 search, random simulation

Q & A