UCT for tactical assault planning

Radha-Krishna Balla and Alan Fern, 2009

Seminar paper

Đorđe Relić

dorde.relic@stud.unibas.ch

University of Basel

November 2015

Content overview

- Problem: tactical planning in RTS games
- AI planners for RTS games
- Monte Carlo and UCT
- UCT in tactical assault planning
- Experimental results

Real-time strategy games

- Warcraft II (1995 by Blizzard)

Real-time strategy games

- Problem : defeat the enemy
- Three sub-problems:
  - Unit control
  - Strategic planning
  - Tactical planning

AI planners for RTS games

- Choosing an action sequence that leads to victory
  - Scripting
  - Dynamic scripting
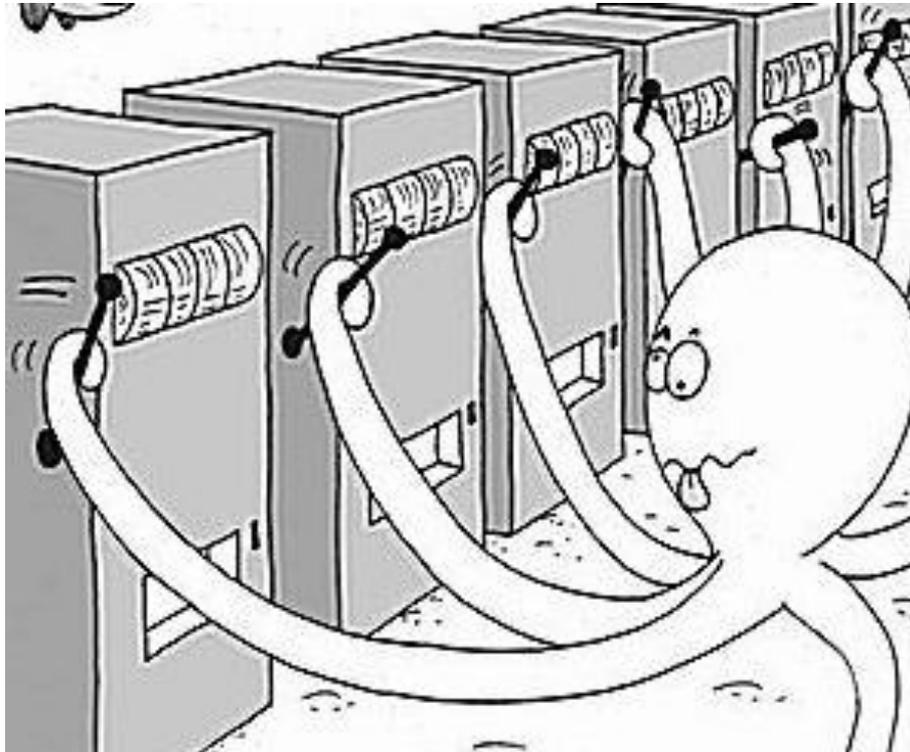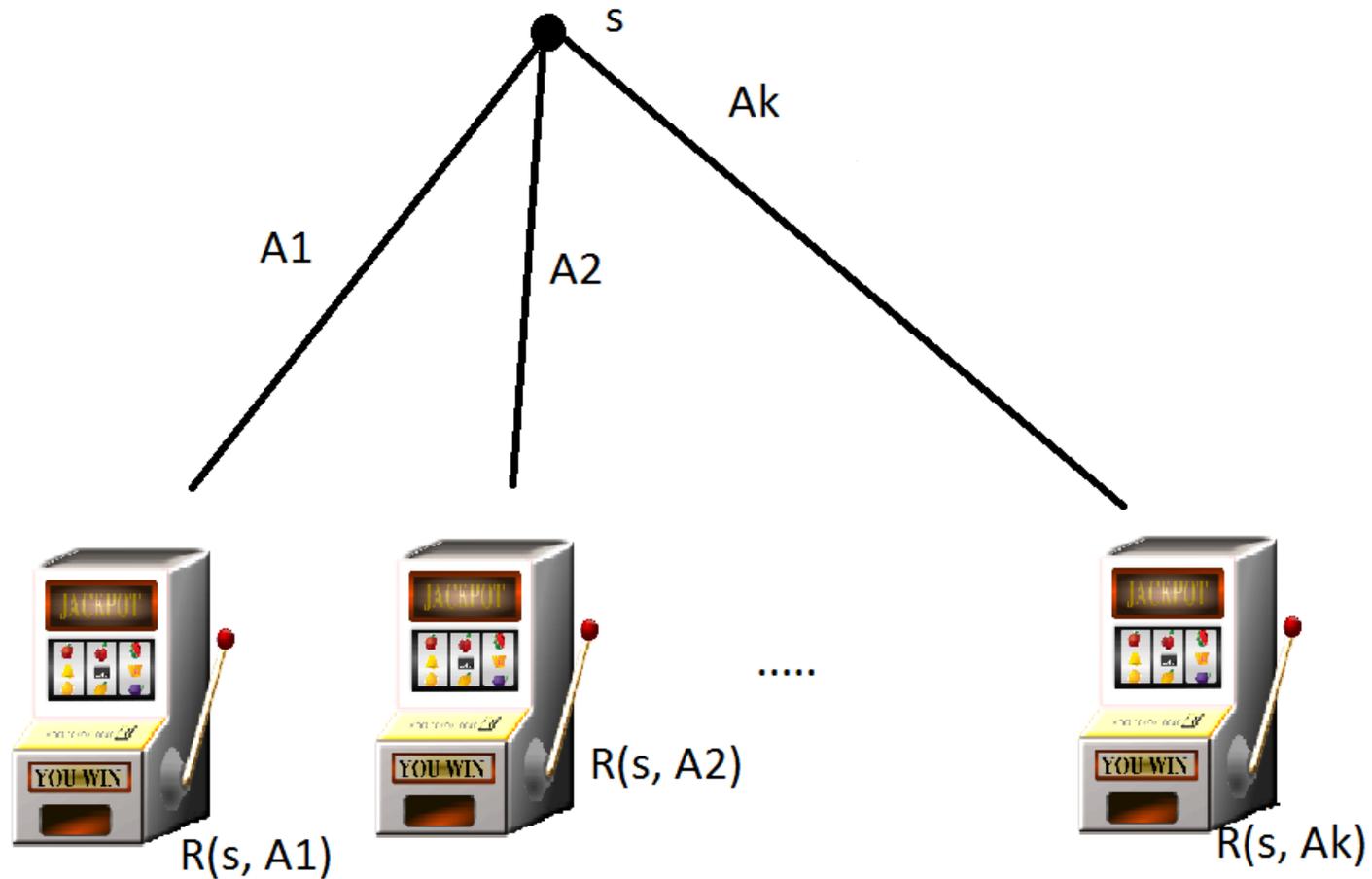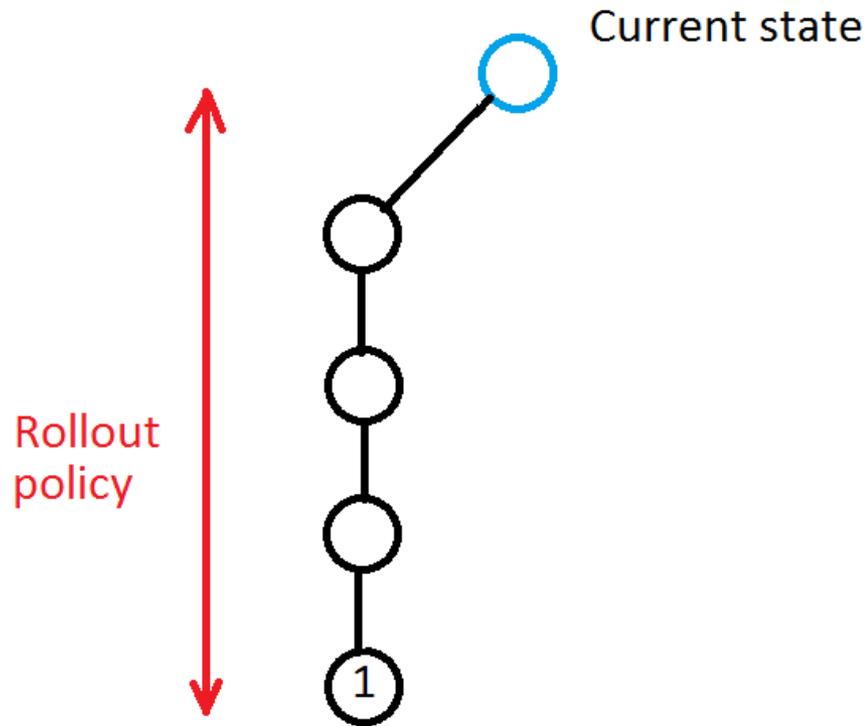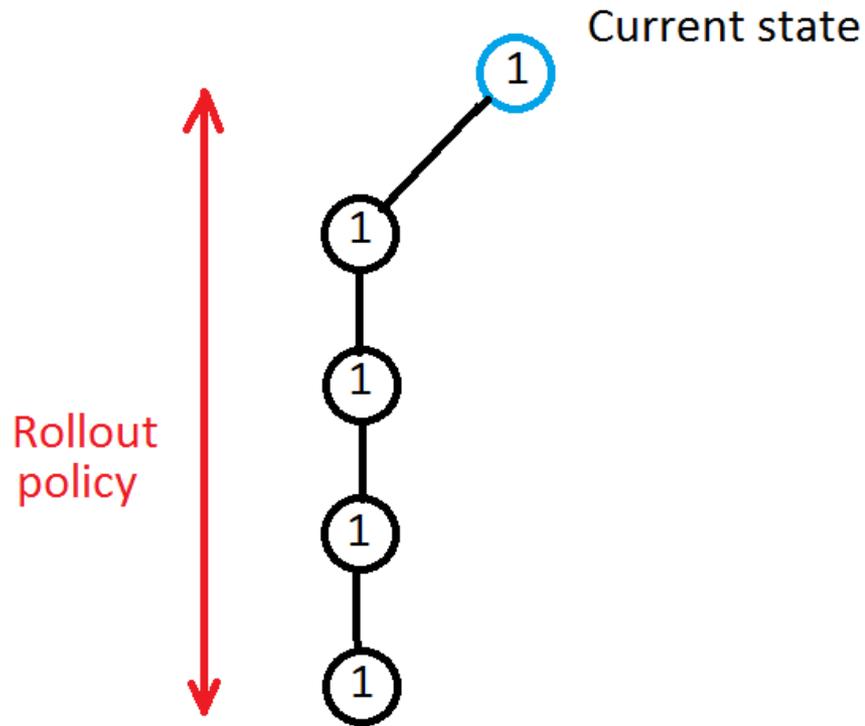 - Monte Carlo planning

Multi-armed bandit

Photo credits Microsoft Research

Monte Carlo planning

UCT

Current state

Rollout policy

1

UCT

UCT

Current state
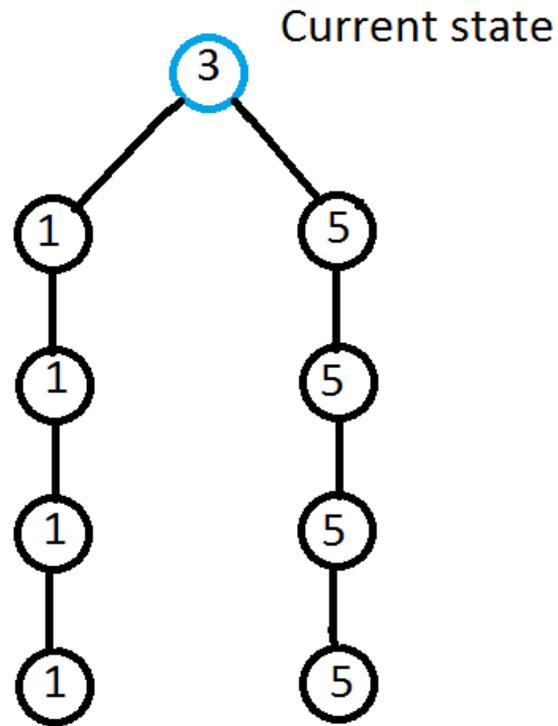
UCT
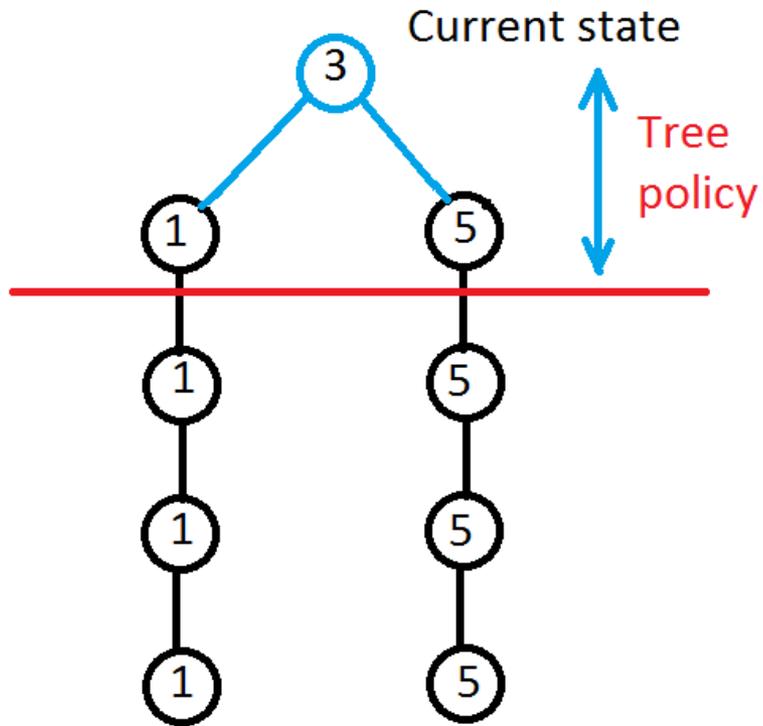
UCT

UCT
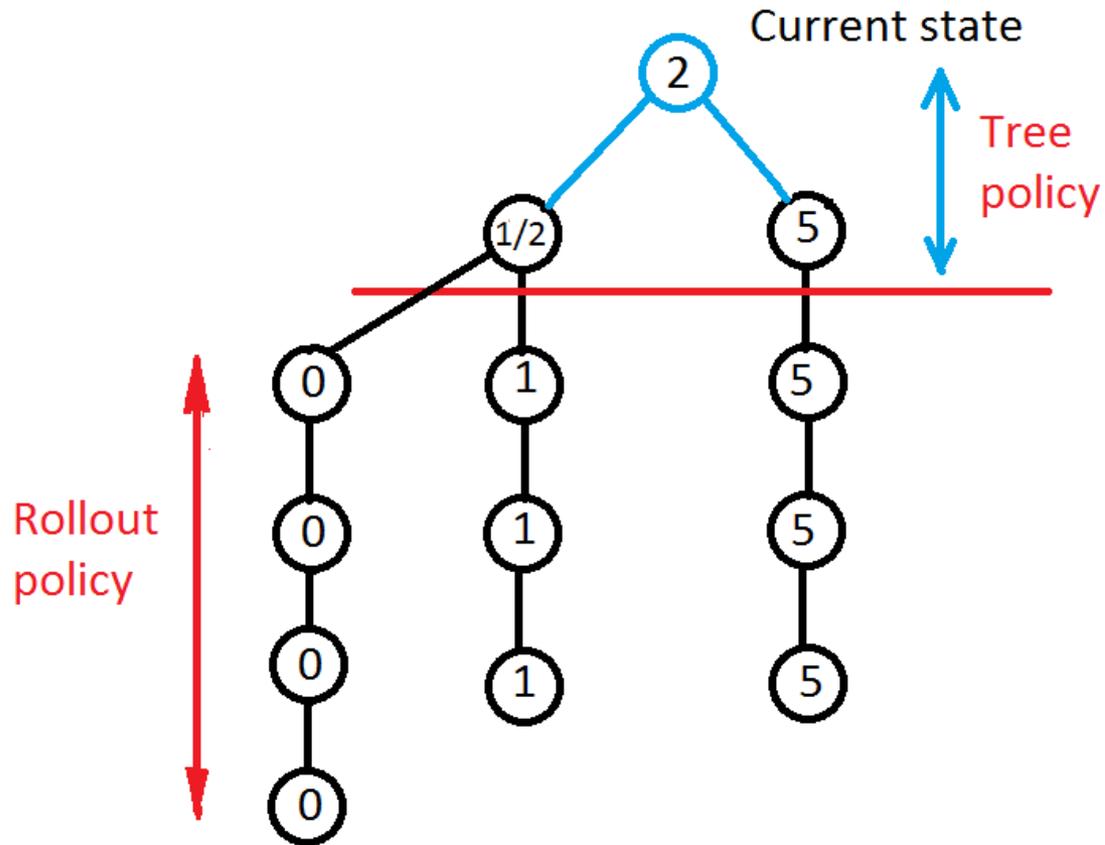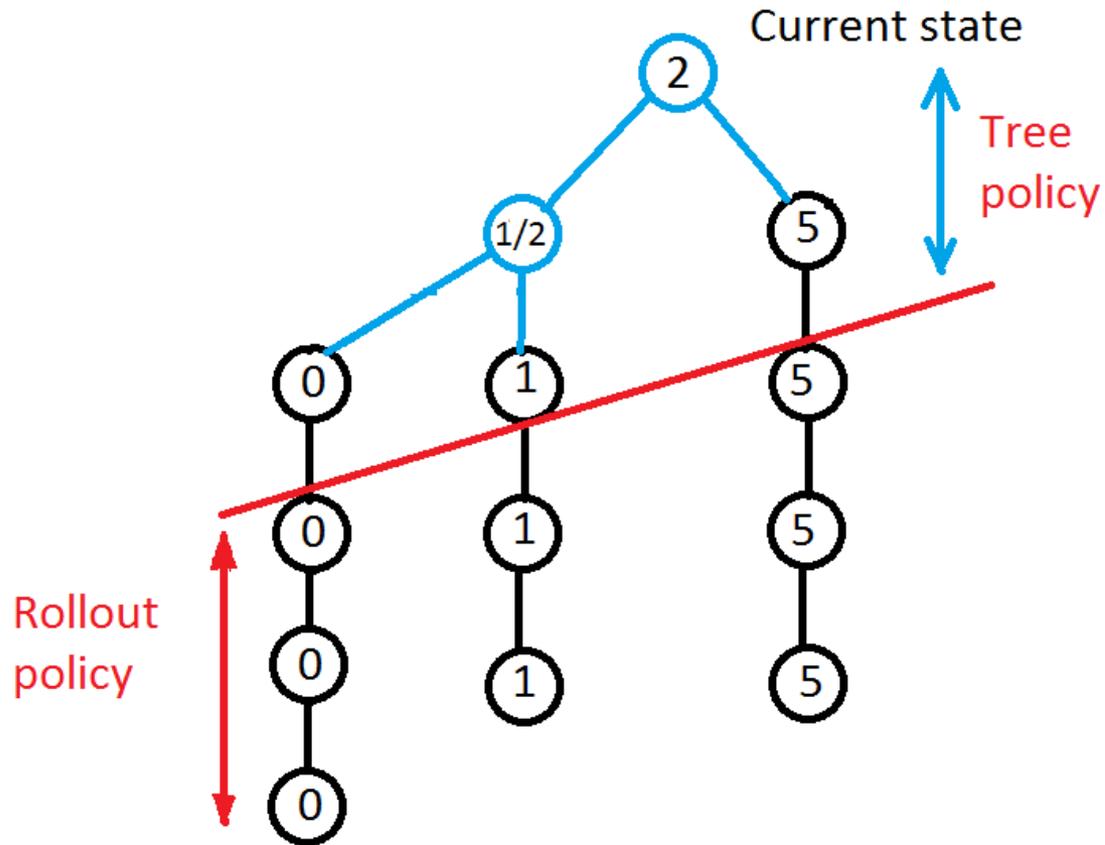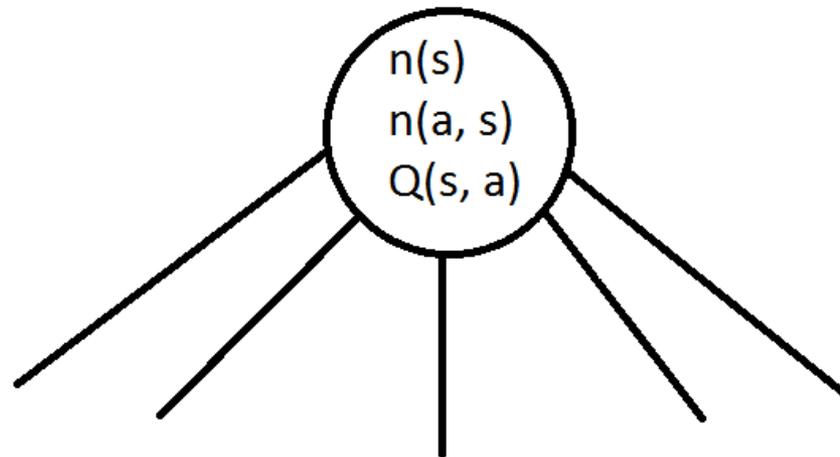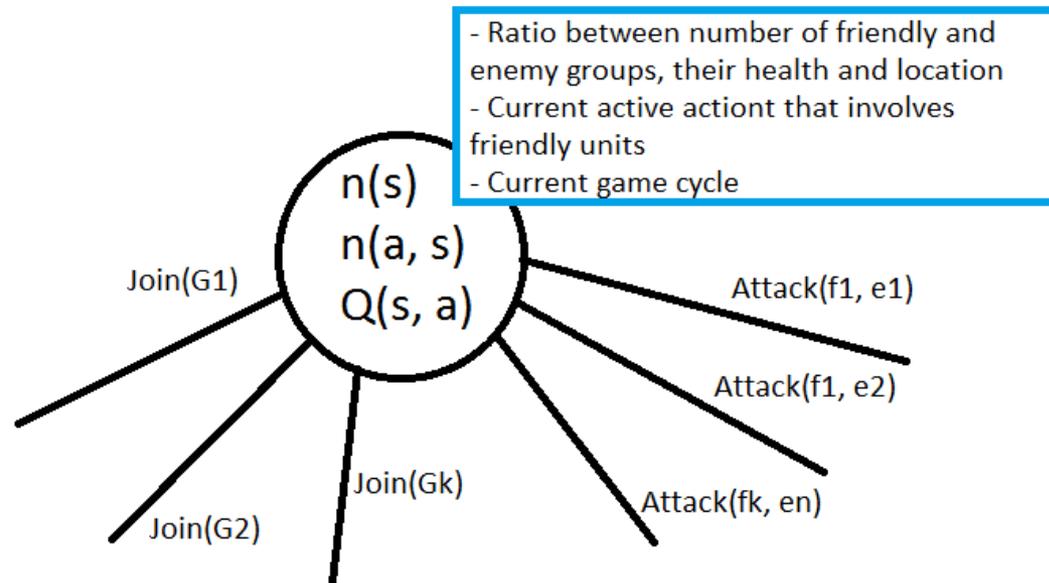
UCT

Rollout policy: select random unexlored action
Tree policy: select action a that maxmizes Q(s, a)

UCT

- UCB rule: $Q^+(s,\ a) = Q(s,a) + c \times \sqrt{\dfrac{\log n(s)}{n(s\,,a)}}$

- $n(s,\ a) \leftarrow n(s,\ a) + 1\ ; n(s) \leftarrow n(s) + 1$

- $Q(s,\ a) \leftarrow Q(s,\ a) + \dfrac{1}{n(s,a)}[R - Q(s,\ a)]$

UCT for tactical assault planning

- Ratio between number of friendly and enemy groups, their health and location
- Current active actiont that involves friendly units
- Current game cycle

n(s)
n(a, s)
Q(s, a)

Join(G1)

Join(G2)

Join(Gk)

Attack(f1, e1)

Attack(f1, e2)

Attack(fk, en)

Rollout policy: select random unexlored action
Tree policy: select action a that maximizes Q(s, a)
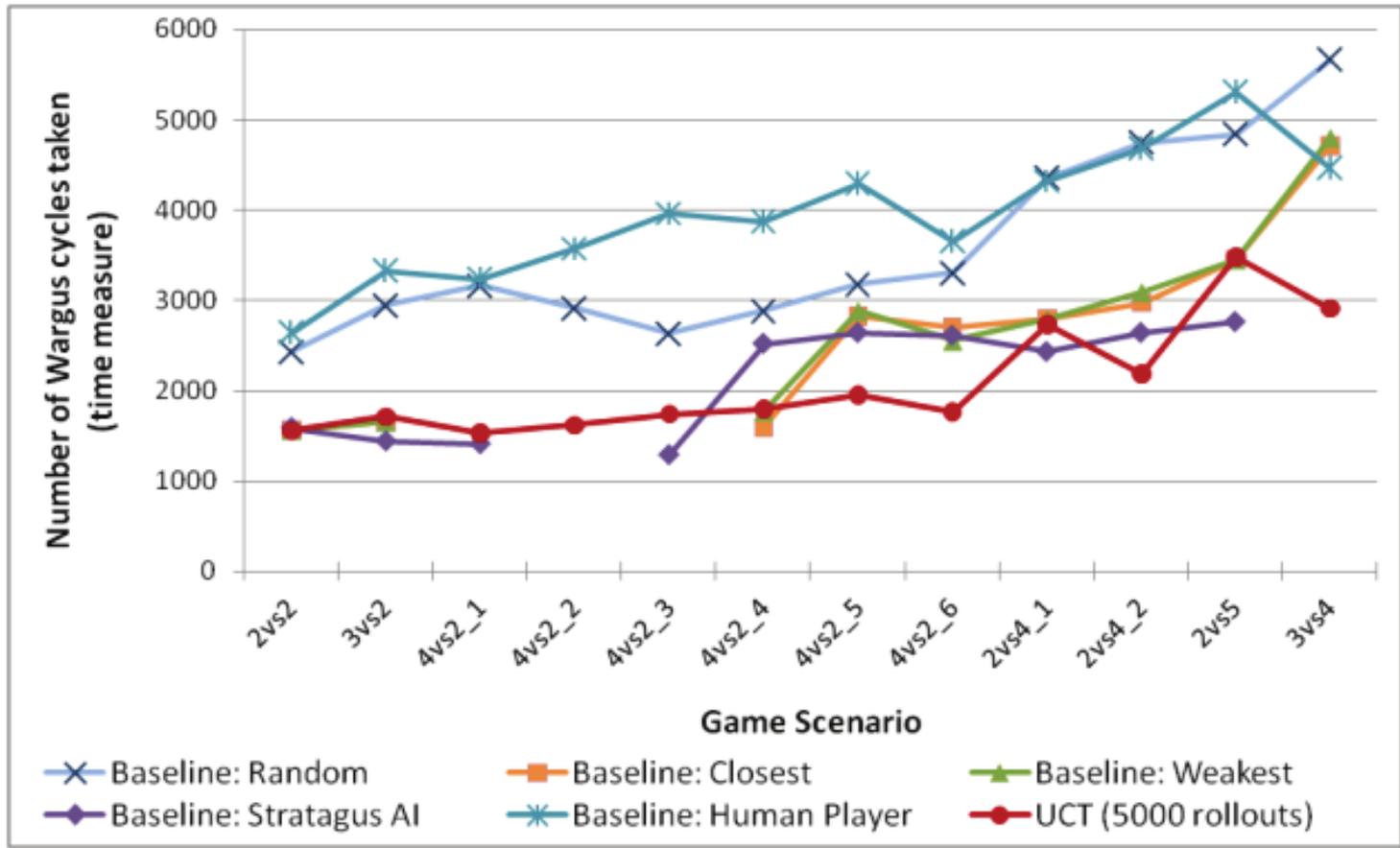
$$c = Q(s, a)$$
$$O(n^2 + n \times m)$$

Experiment results

- Wargus and Stratagus
- 12 scenarios
- Two objective functions
- Comparison with baseline algorithms:
  - Random
  - Attack closest
  - Attack weakest
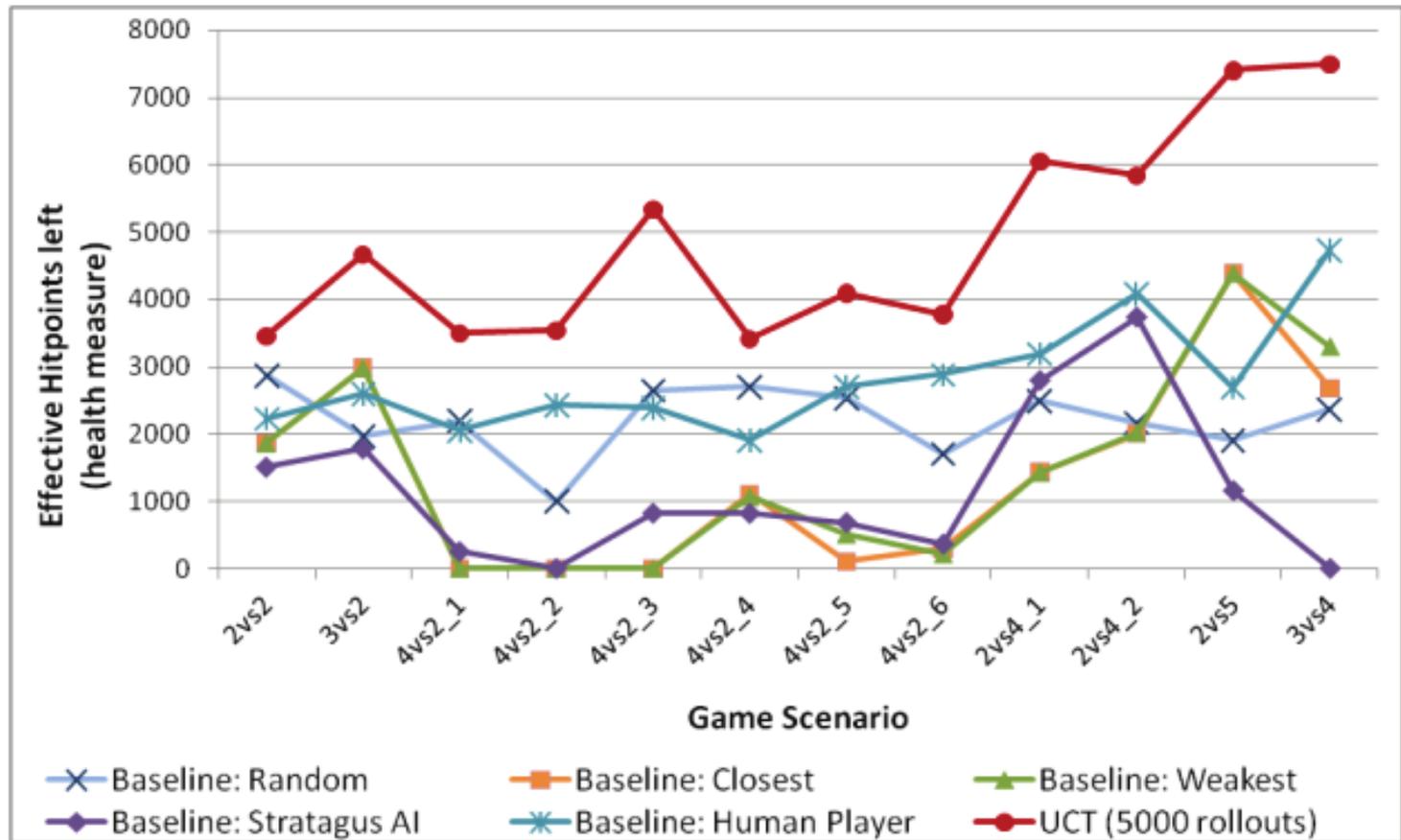 - Stratagus AI
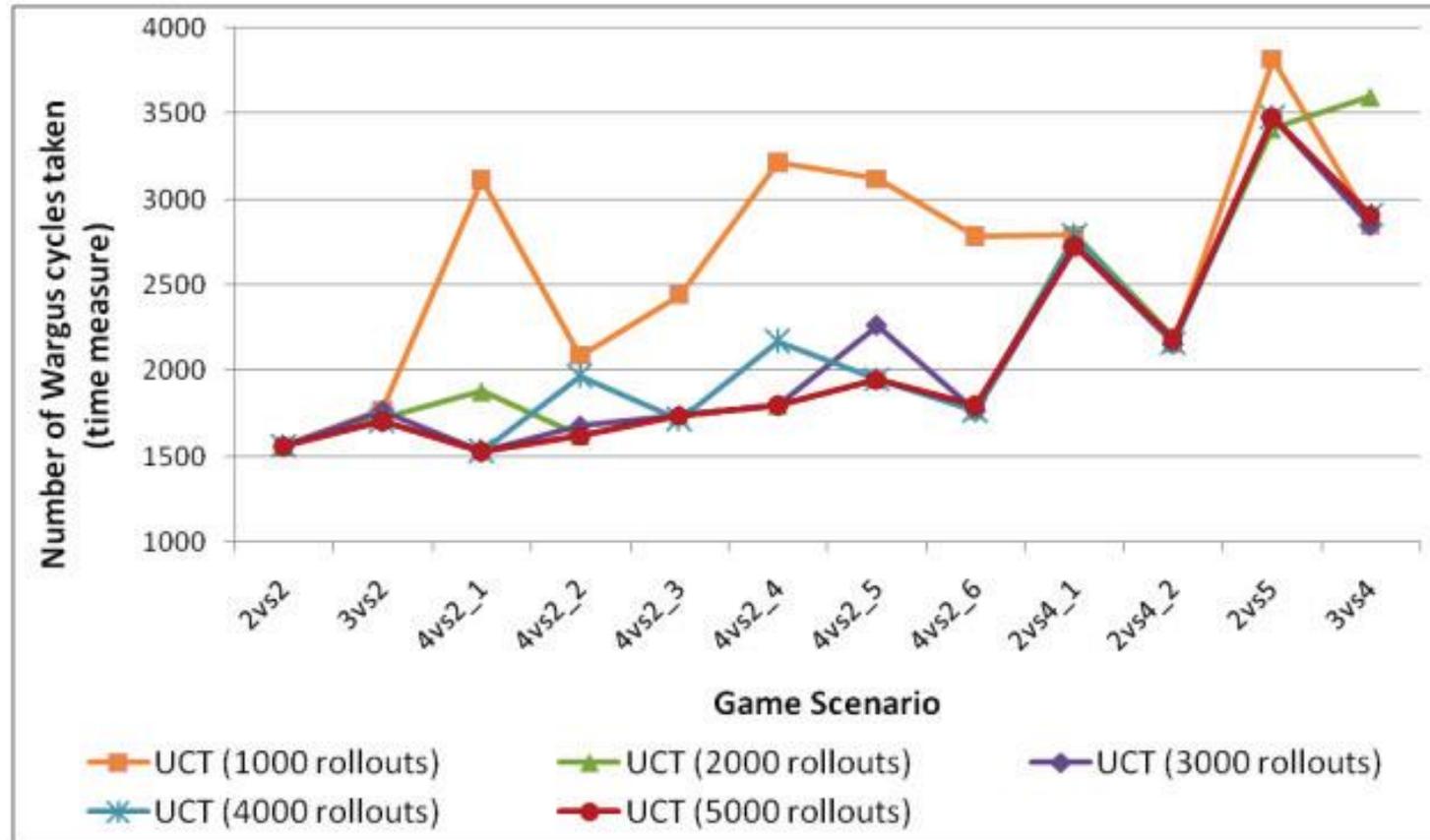 - Human player

Experiment results

Experimental results

Experimental results

Experimental results

Conclusion

- Brief intro to RTS games

- UCT algorithm

- UCT applied to RTS tactical assault

- Experimental results

Questions ?

Thank you for your attention