Adversarial Search with Procedural Knowledge Heuristic

Viliam Lisý, Branislav Bošanský, Michal Jakob and Michal Pechoucek

Sun Wei
Weixun Ge
2012 Fall
Concepts

- **Adversarial Search**
  - search for planning in a competitive multi-agent environment, in which the agents’ goals are in conflict.
  - single agent planning
  - interactions consideration

- **Procedural Knowledge**
  - not facts, but how to achieve the goals
Agenda

- Game tree
- Game tree search
  - minimax game tree search
  - max^n game tree search
- Goal-based game tree search
- Summary
Game Tree

- Game
  - S0: the initial state.
  - Player(s): has the move in state s.
  - Action(s): legal moves in state s.
  - Result(a, s): transition model.
  - Terminal-Test(s): terminal test.
  - Utility(s, p): final numeric value for a game.
- zero–sum game: constant sum of payoffs
Game Tree

- A tree where the nodes are game states and the edges are moves.
Game Tree Search

- minimax algorithm
  - MINIMAX(s)
Game Tree Search

- minimax algorithm
  - MINIMAX(s)
Game Tree Search

- minimax algorithm
  - MINIMAX(s)
- max^n algorithm
  - a multi-player version of minimax
  - fix order
Simple Scenario

- **Players:** government, humanitarian organization, and separatists
- **Units:** police (cop) and gangster (gng), 2 food trucks, 1 explosive truck
Simple Scenario

- **Players:** government, humanitarian organization, and separatists
- **Units:** police (cop) and gangster (gng), 2 food trucks, 1 explosive truck
- \((4^5)^6\): about \(10^{18}\)
Utilizing Procedural Background Knowledge in Game-tree Search

- **Algorithms**
  - Produce atomic actions leading to the fulfillment of the associated goal

- **Conditions**
  - Define world states in which pursuing the goals is meaningful

- **Evaluation Function**
  Assign to each player and world state a numeric value representing desirability of the game state for the player

Sun Wei, Weixun Ge Fall 2012 Northeastern University
Goal-based Game-tree Search

\[(\mathcal{P}, \mathcal{U}, \mathcal{A}, \mathcal{W}, \mathcal{T})\] the domain of algorithm

\[\mathcal{P}\] a set of players

\[\mathcal{U} = \bigcup_{p \in \mathcal{P}} \mathcal{U}_p\] a set of units/resources

\[\mathcal{A} = \times_{u \in \mathcal{U}} \mathcal{A}_u\] a set of combinations of actions
the units can perform

\[\mathcal{W}\] a set of possible world states

\[\mathcal{T} : \mathcal{W} \times \mathcal{A} \rightarrow \mathcal{W}\] transition changing world via actions of all units
Goals

- A goal is defined as a pair $(I_g, A_g)$.
- $I_g(W, U)$ is the initiation condition of the goal.
- $A_g$ is an algorithm that outputs the next action that leads to fulfilling the goal.
- A goal can be abandoned if $A_g$ is finished.
GBSearch(W, d, G) Main Procedure

Input: W ∈ W: current world state, d: search depth,
      G[U]: map from units to goals they pursue
Output: an array of values of the world state (one value for each player)

1. curW = W
2. while all units have goals in G do
3.     Actions = ∅
4.     foreach goal g in G do
5.         Actions = Actions ∪ NextAction(A_g)
6.         if A_g is finished then
7.             remove g from G
8.         end
9.     end
10.    curW = T(curW, Actions)
11.    d = d - 1
12.    if d=0 then
13.        return Evaluate(curW)
14.    end
15. end
16. u = GetFirstUnitWithoutGoal(G)
17. foreach goal g with satisfied I_g(curW, u) do
18.     G[u] = g
19.     V[g] = GBSearch(curW, d, Copy(G))
20. end
21. g = arg max_g V[g][Owner(u)]
22. return V[g]

Sun Wei, Weixun Ge Fall 2012 Northeastern University
Adversarial Search with Procedural Knowledge Heuristic

Search Reduction

AB algorithm $\max^n$): 1–6 steps

GB algorithm: 1–19 steps

Sun Wei, Weixun Ge Fall 2012 Northeastern University
Complex Scenario

A graph with 2574 vertices and two sets of units

2 police, 2 gangster,
1 engineer, 1 stone truck,
3 commodity truck

1 police, 1 gangster,
1 engineer, 1 stone truck,
3 commodity truck

Sun Wei, Weixun Ge Fall 2012 Northeastern University
Conclusions

- Extends multi-player game-tree search with a heuristic based on procedural knowledge.

- The new algorithm can be applied to larger scenarios.

- Further optimizations are needed for simulating a real-world scenario.

- A good algorithm representing goals in the background knowledge is needed.