A Doppelkopf Player Based on UCT

Silvan Sievers Malte Helmert
University of Basel, Switzerland

Doppelkopf: Game Information

- Trick based card game for four players
- 48 cards: double deck from nine to ace
- Two parties: re and kontra
- Solo and normal games
- Unique feature: parties not known in advance during normale games
- Goal: collect 121 (re) or 120 (kontra) card points

Doppelkopf: Game Rules

Announcements

- All reveal party of the announcing player
- All increase game value
- Some claim to win the game
- Some increase card points required for winning

Game Evaluation: Score Points

- $+1$ for winning
- $+1/+2$ for announcements
- $+1$ for every 30 card points achieved extra
- Extra score points for special tricks

The UCT Algorithm (Kocsis and Szepesvári 2006)

- Monte Carlo tree search algorithm based on sampling
- State of the art for many problems of acting under uncertainty

High Level Description

- Repeatedly perform rollouts starting in the current state
- Balance exploration and exploitation
- Incorporate rewards from rollouts into a game tree

Variations of the UCT Algorithm

Single-UCT

- One UCT computation
- Each rollout with a different card assignment

Ensemble-UCT

- Several UCT computations
- Fix a card assignment for each UCT computation

The Card Assignment Problem (CAP)

CAP

- Assign all remaining cards to all other players
- Respect all available information about other players

- Goal for unbiased players: compute solutions to the CAP uniformly at random
- Requirement: solve #CAP (#-complete)
  $\rightarrow$ infeasible

The Card Assignment Algorithm

While there are cards left to be assigned:
  If a card can be assigned to exactly one player:
    Assign that card to that player
  If a player requires as many cards as he can have:
    Assign those cards to that player
  If a player requires a ♣Q:
    Assign a ♣Q to that player
  Otherwise:
    Assign a random card to a random player

Experiments: Setup

- Two UCT players against two random players
- 1000 games with random card deals
- Repeat every game in every possible permutation of positions
- Total of 10000 rollouts for every decision
- Results: averge score points per game with 95% confidence interval

Experiments: Ensemble-UCT Configurations

- X/Y: number of single UCT computations/rollouts

ensemble-UCT (5/2000)	ensemble-UCT(10/1000)	random
1.67 ± 0.12	$\mathbf{1.83 \pm 0.11}$	(-1.75 ± 0.05)

ensemble-UCT (10/1000)	ensemble-UCT (20/500)	random
$\mathbf{2.10 \pm 0.11}$	1.70 ± 0.10	(-1.90 ± 0.05)

$\rightarrow$ trade-off between the number of different card assignments and the quality of the computation per card assignment

Experiments: Influence of Announcement Making

announcing ensemble-UCT	no announcing ensemble-UCT	random
$\mathbf{1.70 \pm 0.07}$	0.79 ± 0.05	(-1.25 ± 0.04)

announcing single-UCT	no announcing single-UCT	random
$\mathbf{0.48 \pm 0.06}$	0.19 ± 0.05	(-0.33 ± 0.04)

$\rightarrow$ making announcements crucial for performance

Experiments: Ensemble-UCT versus Single-UCT

ensemble-UCT	single-UCT	random
$\mathbf{4.52 \pm 0.11}$	-1.25 ± 0.08	(-1.63 ± 0.05)

$\rightarrow$ using few, but fixed card assignments better than using many

Experiments: Playing Against a Human

- 24 games human vs. ensemble-UCT

human	ensemble-UCT	random	random
43	-9	$(-15$	$-19)$
15	7	$(-35$	$13)$

- Analysis of ensemble-UCT playing style:
  - Too many solos (works well against random players)
  - Always makes announcements when playing solo, but rarely in normal games
  - The fewer options remaining, the stronger the game play (not a surprise)

Possible Improvements

- Separate hand evaluation algorithm
- Analyze and reduce bias of card assignment algorithm
- Domain specific knowledge for simulation phase of rollouts
- Drop assumption that opposing players behave like UCT players
- Reuse information from decisions at previous game states

Contributions

- Doppelkopf as a benchmark problem
- Baseline UCT players
- Card assignment algorithm
- Ensemble-UCT for more stable UCT performance