Automatic Heuristic Construction in a Complete General Game Player

Gregory Kuhlmann    Kurt Dresner    Peter Stone

Learning Agents Research Group
Department of Computer Sciences
The University of Texas at Austin

AAAI 2006
Computer Game Playing

- One of AI’s biggest success stories
  - checkers, chess, scrabble, othello, connect-4
• One of AI’s biggest success stories
  • checkers, chess, scrabble, othello, connect-4

• **Search** is universal in game playing

• Bound search for large state spaces
  • Board evaluation function (**heuristic**)

• Game analysis
  • Traditionally performed by **human designers**
  • Specific to a **single game**
General Game Playing

- Single system plays **many games** in a class
- Analysis performed by **system itself**
- Player inputs game rules for unknown game
  - Game description allows simulation
  - Expand game tree

- If not exhaustively searchable, what to do?
  - Look for hints in game description
General Game Playing

- Single system plays many games in a class
- Analysis performed by system itself
- Player inputs game rules for unknown game
  - Game description allows simulation
  - Expand game tree

![Game Tree Diagram]

- If not exhaustively searchable, what to do?
  - Look for hints in game description
**Game Players** run as servers

**Game Manager** sends rules to players
- Game Description in **GDL**
- Start clock
  - Time to analyze description (1–40 minutes)
- Play clock:
  - Time to make moves (10–120 seconds)

---

![Diagram](attachment:diagram.png)

**GM**

- **Rules**
- **Move**
- **Move**
- **...**

**GP**

- **Start**
- **Play**
- **Play**
Class of Games

Deterministic, Perfect Information Games

![Chess](image1)

![Backgammon](image2)

![Checkers](image3)

![Card Game](image4)

**YES**

**NO**
Game Description Language

- First order logic (KIF)
- **State**: database of provable facts
- **Constructs**
  - **init**: initial state
  - **legal**: legal moves
  - **next**: state transitions
  - **terminal**: termination conditions
  - **goal**: value of terminal states
(role white)  (role black)
(init (cell a 1 b))  (init (cell a 2 b))
(init (cell a 1 b))  (init (cell a 2 bk))
(init (cell a 1 wr))  (init (cell a 2 b))
(init (cell a 1 b))  (init (cell a 2 b))
(init (control white))  (init (step 1))
(<= (legal white (move wk ?u ?v ?x ?y))
    (true (control white))
    (true (cell ?u ?v wk))
    (kingmove ?u ?v ?x ?y)
    (true (cell ?x ?y b)))
(<= (next (step ?y))
    (true (step ?x))
    (succ ?x ?y))
(succ 1 2)  (succ 2 3)  (succ 3 4)  (succ 4 5)
(<= (goal white 100)
    checkmate)
(<= terminal
    (true (step 10)))

- Simulate with **theorem prover** (Prolog)
- How can we do better than just legal play?
(role white)  (role black)
(init (cell a 1 b))  (init (cell a 2 b))
(init (cell a 1 b))  (init (cell a 2 bk))
(init (cell a 1 wr))  (init (cell a 2 b))
(init (cell a 1 b))  (init (cell a 2 b))
(init (control white))  (init (step 1))
(<= (legal white (move wk ?u ?v ?x ?y))
    (true (control white))
    (true (cell ?u ?v wk))
    (kingmove ?u ?v ?x ?y)
    (true (cell ?x ?y b)))
(<= (next (step ?y))
    (true (step ?x))
    (succ ?x ?y))
(succ 1 2)  (succ 2 3)  (succ 3 4)  (succ 4 5)
(<= (goal white 100)
    checkmate)
(<= terminal
    (true (step 10)))

- Simulate with **theorem prover** (Prolog)
- How can we do better than just legal play?
Identify structures from common game elements

Successor Relations

(\text{succ} 1 \ 2) \quad (\text{angel paper table})
(\text{succ} 2 \ 3) \quad (\text{angel table bottom})
(\text{succ} 3 \ 4) \quad (\text{angel bottom mellow})
(\text{succ} 4 \ 5) \quad (\text{angel mellow yard})

Tokens will be scrambled. Based on structure alone.

Bridge between logical and numerical representations
Identifying Structures (cont.)

Find rules matching templates

Step Counters

\[
\begin{align*}
& (\leq (\text{next} \ (\text{step} \ ?x)) \quad (\leq (\text{next} \ (\text{foo} \ ?u))) \\
& (\text{true} \ (\text{step} \ ?y)) \quad (\text{true} \ (\text{foo} \ ?v)) \\
& (\text{succ} \ ?y \ ?x)) \quad (\text{bar} \ ?v \ ?u))
\end{align*}
\]

Again no lexical clues used.

- Bounds tree depth
- Remove for longer internal games
- Remove from Transposition Table
Many games have a board of some type

State

(cell 1 1 bk) (cell 1 2 b)
(cell 1 3 wk) (cell 1 4 b)
(cell 2 1 b) (cell 2 2 b)
(cell 2 3 bk) (cell 2 4 b)
(cell 3 1 wr) (cell 3 2 b)
(cell 3 3 b) (cell 3 4 b)
(cell 4 1 b) (cell 4 2 b)
(cell 4 3 b) (cell 4 4 b)

Boards and Pieces

cell:0,1→2 ; [b, wk, wr, bk]
cell:0,2→1 ; [1, 2, 3, 4]
cell:1,2→0 ; [1, 2, 3, 4]

- Start with all ternary functions
- Divide slots into inputs and outputs
- Refine through internal simulation
Board Game Structures

Many games have a board of some type

State

(cell 1 1 bk) (cell 1 2 b)
(cell 1 3 wk) (cell 1 4 b)
(cell 2 1 b) (cell 2 2 b)
(cell 2 3 bk) (cell 2 4 b)
(cell 3 1 wr) (cell 3 2 b)
(cell 3 3 b) (cell 3 4 b)
(cell 4 1 b) (cell 4 2 b)
(cell 4 3 b) (cell 4 4 b)

Boards and Pieces

cell:0,1→2 ; [b, wk, wr, bk]

- Start with all ternary functions
- Divide slots into inputs and outputs
- Refine through internal simulation
Many games have a board of some type

### State

| Cell 1 1 bk | Cell 1 2 b |
| Cell 1 3 wk | Cell 1 4 b |
| Cell 2 1 b | Cell 2 2 b |
| Cell 2 3 bk | Cell 2 4 b |
| Cell 3 1 wr | Cell 3 2 b |
| Cell 3 3 b | Cell 3 4 b |
| Cell 4 1 b | Cell 4 2 b |
| Cell 4 3 b | Cell 4 4 b |

### Boards and Pieces

\[
\text{cell:0,1→2 ; [wk, wr, bk]}
\]

- **Start with all ternary functions**
- **Divide slots into inputs and outputs**
- **Refine through internal simulation**
### Identified Structure vs. Generated Features

<table>
<thead>
<tr>
<th>Identified Structure</th>
<th>Generated Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ordered Board w/ Pieces</td>
<td>Each piece’s X coordinate</td>
</tr>
<tr>
<td></td>
<td>Each piece’s Y coordinate</td>
</tr>
<tr>
<td></td>
<td>Manhattan distance between each pair of pieces</td>
</tr>
<tr>
<td></td>
<td>Sum of pair-wise Manhattan distances</td>
</tr>
<tr>
<td>Board w/o Pieces</td>
<td>Number of markers of each type</td>
</tr>
<tr>
<td>Quantity</td>
<td>Amount</td>
</tr>
</tbody>
</table>

- Board inputs **ordered** by successor relation(s)?
- Board has at least one **piece**?
- Non-board features also identified
Maximize single feature:

\[ H(s) = 1 + R^- + (R^+ - R^- - 2) \times V(s) \]

Or minimize single feature:

\[ H(s) = 1 + R^- + (R^+ - R^- - 2) \times [1 - V(s)] \]

- Example: Maximize white rook’s y-coordinate
- Actual win always better than heuristic value
- Actual loss always worse
During Start Clock:
- Candidate heuristics constructed from GD
- “Best” heuristic is chosen
  - Old approach: parallel search
  - New approach: internal tournament

During Play Clock:
- Iterative-deepening Minimax search
  - Minimax search w/ $\alpha\beta$ pruning
  - Transposition table and history heuristic
  - Extensions for > 2 players, simultaneous games
Experiments

- **Goal:** Identify impact of game analysis
- Three different games
  - created by competition organizers
- Heuristic chosen manually
  - simulates good method to choose heuristic
  - no experimentation after initial selection
- **Opponent:** constant heuristic (exhaustive search)
Othello variant

- more corner squares
- **opposite goal:** finish with *fewer* markers
Nothello

Othello variant

- more corner squares
- **opposite goal**: finish with *fewer* markers
Nothello

Othello variant

- more corner squares
- opposite goal: finish with fewer markers
Nothello

Othello variant

- more corner squares
- **opposite goal:** finish with fewer markers
Nothello

Othello variant

- more corner squares
- **opposite goal:** finish with *fewer* markers
**Heuristic:** minimize number of own markers

<table>
<thead>
<tr>
<th>NumMarkers</th>
<th>H(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>74.25</td>
</tr>
<tr>
<td>8</td>
<td>79.20</td>
</tr>
<tr>
<td>8</td>
<td>79.20</td>
</tr>
<tr>
<td>9</td>
<td>76.73</td>
</tr>
</tbody>
</table>
Heuristic: minimize number of own markers

NumMarkers: 10  H(s): 74.25
NumMarkers: 8   H(s): 79.20
NumMarkers: 8   H(s): 79.20
NumMarkers: 9   H(s): 76.73
Heuristic: minimize number of own markers

NumMarkers: 10  H(s): 74.25
NumMarkers: 8  H(s): 79.20
NumMarkers: 8  H(s): 79.20
NumMarkers: 9  H(s): 76.73
Heuristic: minimize number of own markers

NumMarkers: 10  H(s): 74.25
NumMarkers: 8  H(s): 79.20
NumMarkers: 8  H(s): 79.20
NumMarkers: 9  H(s): 76.73
Hallway

Chess board with two pawns

- **Actions**: move pawn or place wall
- **Goal**: reach other side first
Hallway

Chess board with two pawns

- **Actions:** move pawn or place wall
- **Goal:** reach other side first
Hallway

Chess board with two pawns

- **Actions:** move pawn or place wall
- **Goal:** reach other side first
Hallway

Chess board with two pawns

- **Actions:** move pawn or place wall
- **Goal:** reach other side first
Chess board with two pawns

- **Actions**: move pawn or place wall
- **Goal**: reach other side first
Hallway

Chess board with two pawns

- **Actions**: move pawn or place wall
- **Goal**: reach other side first
Hallway

Chess board with two pawns

- **Actions:** move pawn or place wall
- **Goal:** reach other side first
Heuristic: maximize own pawn’s y-coordinate
Commodities trading game w/ three simultaneous players

Alice  Barney  Charlie

Commodities

Structures

Heuristic: maximize own money
Commodities trading game w/ three simultaneous players

Alice          Barney          Charlie

Commodities

Structures

Heuristic: maximize own money
## Experimental results

<table>
<thead>
<tr>
<th>Game</th>
<th>Matches</th>
<th>Expected Wins</th>
<th>Empirical Wins</th>
<th>$p$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nothello</td>
<td>15</td>
<td>7.5</td>
<td>15</td>
<td>$10^{-5}$</td>
</tr>
<tr>
<td>Hallway</td>
<td>15</td>
<td>3</td>
<td>15</td>
<td>$10^{-11}$</td>
</tr>
<tr>
<td>Farmers</td>
<td>25</td>
<td>8.3</td>
<td>11</td>
<td>0.234</td>
</tr>
</tbody>
</table>

## Competition Results
- **2005**: competitive but technical difficulties
- **2006**: very competitive (3rd place)
  - after 72 matches, gap with first: $\sim$ 3 games
Results

- Experimental results

<table>
<thead>
<tr>
<th>Game</th>
<th>Matches</th>
<th>Expected Wins</th>
<th>Empirical Wins</th>
<th>$p$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nothello</td>
<td>15</td>
<td>7.5</td>
<td>15</td>
<td>$10^{-5}$</td>
</tr>
<tr>
<td>Hallway</td>
<td>15</td>
<td>3</td>
<td>15</td>
<td>$10^{-11}$</td>
</tr>
<tr>
<td>Farmers</td>
<td>25</td>
<td>8.3</td>
<td>11</td>
<td>0.234</td>
</tr>
</tbody>
</table>

- Competition Results
  - 2005: competitive but technical difficulties
  - 2006: very competitive (3rd place)
    - after 72 matches, gap with first: $\sim$ 3 games
Conclusion and Future Work

- **General Game Playing**
  - Automate game analysis
- **Automatic Heuristic Construction**
  - Structures $\rightarrow$ Features $\rightarrow$ Heuristics
- Method incorporated into **complete agent**
- **Future Work**
  - Learn more complex evaluation functions
  - Understand game similarity
  - Transfer knowledge between games
Conclusion and Future Work

- **General Game Playing**
  - Automate game analysis

- **Automatic Heuristic Construction**
  - Structures → Features → Heuristics

- Method incorporated into complete agent

- **Future Work**
  - Learn more complex evaluation functions
  - Understand game similarity
  - Transfer knowledge between games