Opponent modelling for case-based adaptive game AI

Alexander Römelt

University of Heidelberg
Seminar: Artificial Intelligence for Games

July 18, 2019
Agenda

- Introduction
- Case-based adaptive game AI
- Incorporating opponent modelling
- Experiments
- Generic approach
- Conclusion

Opponent modelling for case-based adaptive game AI
"In general, an opponent model is an abstracted description of a player or a player's behaviour in a game" Herik, Donkers, and P. H. M. Spronck n.d.

Build a model of the opponent player and utilize it for actual play

Goal: Adapt to opponent and exploit his weaknesses!

Example: rock-paper-scissors
Possible other applications of opponent modelling:

- Military
- Robotics industry
- Understanding and representation of human models
General role of opponent modelling in classic games:

- Apply search techniques to find possible actions of the opponent and construct a model
- Guide the search process towards improved results
Introduction

Short history:

• 1970s:
  • contempt factor in chess programs
  • chance of performing a non-rational action
  • rudimentary knowledge in the search process

• 1993: opponent-model search (research groups from Haifa and Maastricht)

• 1994: search technique to speculate on the fallibility of the opponent

• 2000s: probabilistic opponent models

• 2009: Opponent modelling for case-based adaptive game AI

• 2013: Generic opponent modelling approach for RTS games
Video Games

Two possible roles:

As a companion:
- Behave according to the human player’s expectations
- Avoid being annoying

As an opponent:
- Adapt to the human players playing style
- Match the human players skills (play neither too weak nor too strong)

Goal: raise the entertainment factor
Challenges:

1. Realistic and complex game environments
2. Little time for observation
3. Often only partial observability of the environment

Opponent modelling has to be performed in parallel to graphics rendering, rudimentary AI, ...
Explicit opponent models:
- Specification of opponents attributes separated from decision-making process

Implicit opponent models:
- Game AI is finetuned to a specific type of opponent (without actually referring the attributes)

Approaches:
- Modelling opponent’s actions
- Modelling opponent’s preferences
Preference-based approach:
- Opponent modelling as a classification problem
- Classification as one of multiple models based on data collected during the game
- AI behaves based on the classification

Successful implementations in RTS games (Civilization IV), shooters, ...
Adaptive game AI:

- Game AI capable of adapting to changing circumstances
- Typically implemented with machine-learning techniques
- Learning effective behaviour while the game is in progress

Problems with ’online learning’:

- Often too many learning trials necessary for practical use
- Characters die or the game finishes before effective behaviour is learned
- Establishing effective behaviour of game AI in a stable and reliable manner is difficult
Case-based adaptive game AI:
- Game AI automatically gathers domain knowledge
- Results are immediately exploited

Particularly effective if observations from online games (MMOs) are available
Case-based adaptive game AI

Opponent modelling for case-based adaptive game AI
Case Base:
- Extracted from character and environment observations
- Structured in standard format with timestamp
- Taken from a multitude of games

Case Base is used to extract:
1. Opponent models
2. An evaluation function
Incorporating opponent modelling

Game environment: Spring
Incorporating opponent modelling

Adaptation Mechanism

Game AI

Evaluation Function

Opponent Model

Case Base

Observations

Game Character

Game Environment

commands

interactions

observations

cases

abstracted observations

cases

updates

case evaluation

opponent classification
Adaptation mechanism:

- Indexes collected games and clusters observations (offline)
- Initialized with previously successful game strategy
- Strategy selection using similarity matching (online)

**Game Strategy**

"Configuration of parameters that determine strategic behaviour"
- (overall 27 parameters in the used game AI)
Incorporating opponent modelling

Adaptation Mechanism

Evaluation Function

Case Base

Observations

Game AI

Game Character

Game Environment

Opponent Model

Case Base

abstracted observations

observations

observations

observes

observations

observations

commands

interaction

opponent classification

case evaluation

cases

cases

cases

updates
Evaluation Function

\[ V(p) = W_p V_1 + (1 - W_p) V_2 \]

Parameter:
- \( p \) = phase of the game (opening, end game, ...)

Evaluative terms:
- \( V_1 = \) material strength
- \( V_2 = \) commander safety

\( W \in [-1...1] = \text{free parameter} \)
Incorporating opponent modelling

Adaptation Mechanism

Game AI

Evaluation Function

Opponent Model

Case Base

Observations

Game Character

Game Environment

case evaluation

cases

abstracted observations

observations

updates

opponent classification

commands

interaction

observations
Incorporating opponent modelling

10 features of high-level strategy:

<table>
<thead>
<tr>
<th>Nr.</th>
<th>Feature</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td># observed k-bot units</td>
<td>Global strategic preference</td>
</tr>
<tr>
<td>2.</td>
<td># observed tank units</td>
<td></td>
</tr>
<tr>
<td>3.</td>
<td># observed air units</td>
<td></td>
</tr>
<tr>
<td>4.</td>
<td># tech. adv. constructions</td>
<td>Technological development</td>
</tr>
<tr>
<td>5.</td>
<td># metal extractors</td>
<td>Economy strength</td>
</tr>
<tr>
<td>6.</td>
<td># solar panels</td>
<td></td>
</tr>
<tr>
<td>7.</td>
<td># wind turbines</td>
<td></td>
</tr>
<tr>
<td>8.</td>
<td>![clock] first attack on metal extractors</td>
<td>Aggressiveness</td>
</tr>
<tr>
<td>9.</td>
<td>![clock] first attack on solar panels</td>
<td></td>
</tr>
<tr>
<td>10.</td>
<td>![clock] first attack on wind turbines</td>
<td></td>
</tr>
</tbody>
</table>
Opponent models are automatically established based on case base.

Models are utilised in a game state ...

- ... early enough too have a strong impact on the outcome
- ... not too early for observing strategic choices

Usually models are established after 150 game states

(\approx 10 \textit{minutes of realtime play})
Incorporating opponent modelling

Generation of opponent models:

- Cluster feature data using k-means algorithm
- Measure differences in opponent behaviour using Euclidean distance

Utilising opponent models:

- Offline processing: label each game in case base with information about the opponent
- Classify opponent based on identified clusters (nearest-neighbour)
Incorporating opponent modelling

Offline game AI initialisation:
1. Choose the most observed opponent as the most likely to be pitted against
2. Initialise game AI with the game strategy that has been observed to be the most effective against this particular opponent

Online strategy selection:
• Select strategy in phases of transition
• Choose opponent models if available
Incorporating opponent modelling

If no opponent models available:

1. Select N games from the Case Base that are similar to the current game
2. Select M games from the preselected N games that satisfy a goal criterion
3. Perform the strategy of the game most similar to the current game state

**Fitness value**

Metric for desired behaviour:
100 = significant victory / 0 = tied situation
Incorporating opponent modelling

Continuous validation of the chosen strategy:

- Measure difference in fitness value of current game and selected game
- Compensate by estimating ultimate fitness value with current strategy applied
If opponent models are available:

- Additional moment, when the opponent can be classified accurately
- Adapt game strategy when initially predicted opponent does not match observed opponent model

Opponent classification process is incorporated in strategy selection
Experiments

1. How well does the case based AI adapt to the original AI with medium strength?

2. How well does the case based AI adapt to a previously unobserved opponent playing a randomly generated strategy?
Experiments

Setup:

- Open source game AI: extended with case base vs. original version
- Three different maps
- All trials repeated 150 times

Baseline: All experiments performed with disabled case base and randomly selected strategy

Experiments are performed in basic mode and in a mode with incorporated opponent modelling
Experiments

(a) SmallDivide
(b) TheRing
(c) MetaHeckv2
### Experiments

1. Experiment:

<table>
<thead>
<tr>
<th>Adaptation mode</th>
<th>Trials</th>
<th>Goal achv.</th>
<th>Goal achv. (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMALLDIVIDEB</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Disabled</td>
<td>150</td>
<td>59</td>
<td>39</td>
</tr>
<tr>
<td>Basic</td>
<td>150</td>
<td>115</td>
<td>77</td>
</tr>
<tr>
<td>OM</td>
<td>150</td>
<td>135</td>
<td>90</td>
</tr>
<tr>
<td>THERING</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Disabled</td>
<td>150</td>
<td>90</td>
<td>60</td>
</tr>
<tr>
<td>Basic</td>
<td>150</td>
<td>122</td>
<td>81</td>
</tr>
<tr>
<td>OM</td>
<td>150</td>
<td>127</td>
<td>85</td>
</tr>
<tr>
<td>METALHECKv2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Disabled</td>
<td>150</td>
<td>70</td>
<td>47</td>
</tr>
<tr>
<td>Basic</td>
<td>150</td>
<td>124</td>
<td>83</td>
</tr>
<tr>
<td>OM</td>
<td>150</td>
<td>130</td>
<td>87</td>
</tr>
</tbody>
</table>
Experiments

Map: SmallDivide

- Opponent Modelling
- Basic
- Disabled

Median fitness value over game trials vs. Percentage of game time.
2. Experiment:

<table>
<thead>
<tr>
<th>Adaptation mode</th>
<th>Trials</th>
<th>Goal achv.</th>
<th>Goal achv. (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMALLDIVIDE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Disabled</td>
<td>150</td>
<td>71</td>
<td>47</td>
</tr>
<tr>
<td>Basic</td>
<td>150</td>
<td>96</td>
<td>64</td>
</tr>
<tr>
<td>OM</td>
<td>150</td>
<td>136</td>
<td>91</td>
</tr>
<tr>
<td>THERING</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Disabled</td>
<td>150</td>
<td>76</td>
<td>51</td>
</tr>
<tr>
<td>Basic</td>
<td>150</td>
<td>93</td>
<td>62</td>
</tr>
<tr>
<td>OM</td>
<td>150</td>
<td>93</td>
<td>62</td>
</tr>
<tr>
<td>METALHECKV2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Disabled</td>
<td>150</td>
<td>54</td>
<td>36</td>
</tr>
<tr>
<td>Basic</td>
<td>150</td>
<td>60</td>
<td>40</td>
</tr>
<tr>
<td>OM</td>
<td>150</td>
<td>79</td>
<td>53</td>
</tr>
</tbody>
</table>
## Experiments

Per opponent:

<table>
<thead>
<tr>
<th>Opponent</th>
<th>Trials</th>
<th>Adaptation mode</th>
<th>Disabled</th>
<th>Basic</th>
<th>OM</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>4</td>
<td>9</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>6</td>
<td>8</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>3</td>
<td>10</td>
<td>5</td>
<td>5</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>4</td>
<td>10</td>
<td>4</td>
<td>3</td>
<td>5</td>
<td>9</td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td>7</td>
<td>9</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>6</td>
<td>10</td>
<td>7</td>
<td>6</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>7</td>
<td>10</td>
<td>6</td>
<td>7</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>8</td>
<td>10</td>
<td>7</td>
<td>7</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>9</td>
<td>10</td>
<td>3</td>
<td>6</td>
<td>10</td>
<td>9</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>5</td>
<td>7</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>11</td>
<td>10</td>
<td>6</td>
<td>8</td>
<td>5</td>
<td>9</td>
</tr>
<tr>
<td>12</td>
<td>10</td>
<td>6</td>
<td>8</td>
<td>7</td>
<td>9</td>
</tr>
<tr>
<td>13</td>
<td>10</td>
<td>5</td>
<td>7</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>14</td>
<td>10</td>
<td>5</td>
<td>8</td>
<td>7</td>
<td>9</td>
</tr>
<tr>
<td>15</td>
<td>10</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>9</td>
</tr>
<tr>
<td>Goal achv. avg. (%)</td>
<td>55%</td>
<td>69%</td>
<td>81%</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Experiments

Findings:

• Opponent modelling techniques increase the effectiveness of case-based adaptive game AI
• Approach works best in highly strategic environments

Possible improvements:

• Model opponent behaviour more detailedly with additional features
• Incorporate knowledge about feature weights
Main objectives:

1. **Generalization**: Can be generically applied, without needing knowledge about specific game features

2. **Robust adaptability**: Cope with opponents that switch strategy by continuous tracking of classification

3. **Efficiency**: Avoid inefficient online learning
Offline phase:

- Case base
- Feature selection module
  - Cases
  - Set of features
- Feature weighting module
  - Features & weights
- Induction module
  - Players' models
- Model base
Offline feature processing:

- Feature selection: search for feature subset and corresponding state space
- Induction algorithm: Cluster observations and evaluate accuracy
- Find best fitting state using Best-first search
Generic approach

Online phase:

- Model base
  - Players' models

- Classification module
  - State
  - Performance

- Evaluation module
  - State

- AI engine
  - Action
  - Loaded plan

- Game environment
  - State
Constant adaptation:

- Game AI informs evaluation module with the current state periodically
- Evaluation module estimates the AI player’s current performance against its opponent
- If performance is greater than or equal to a threshold: continue with current model
- Else: Reclassify the opponent model (k-nearest neighbour algorithm)

→ Increased robustness against opponents changing their strategy
Representation of the model base:

<table>
<thead>
<tr>
<th>Feature name</th>
<th>Feature index</th>
<th>Feature weight</th>
<th>G1</th>
<th>G2</th>
<th>G3</th>
<th>G4</th>
<th>G5</th>
<th>G6</th>
<th>G7</th>
<th>G8</th>
<th>G9</th>
<th>G10</th>
</tr>
</thead>
<tbody>
<tr>
<td>AIR_DEFENCE</td>
<td>F2</td>
<td>0.4</td>
<td>2</td>
<td>8</td>
<td>10</td>
<td>3</td>
<td>1</td>
<td>23</td>
<td>17</td>
<td>5</td>
<td>0</td>
<td>8</td>
</tr>
<tr>
<td>FAST_UNITS_RATE</td>
<td>F3</td>
<td>0.2</td>
<td>4</td>
<td>22</td>
<td>16</td>
<td>18</td>
<td>10</td>
<td>22</td>
<td>9</td>
<td>20</td>
<td>21</td>
<td>15</td>
</tr>
<tr>
<td>MAX_DEFENCES</td>
<td>F6</td>
<td>0.6</td>
<td>22</td>
<td>88</td>
<td>31</td>
<td>44</td>
<td>19</td>
<td>3</td>
<td>8</td>
<td>9</td>
<td>15</td>
<td>23</td>
</tr>
<tr>
<td>AIR_DEFENCE</td>
<td>F7</td>
<td>0.1</td>
<td>11</td>
<td>9</td>
<td>15</td>
<td>9</td>
<td>4</td>
<td>7</td>
<td>12</td>
<td>11</td>
<td>17</td>
<td>7</td>
</tr>
<tr>
<td>UNIT_SPEED_UNITGROUPS</td>
<td>F8</td>
<td>0.7</td>
<td>12</td>
<td>3</td>
<td>11</td>
<td>5</td>
<td>2</td>
<td>6</td>
<td>19</td>
<td>7</td>
<td>2</td>
<td>5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Opponent Model</th>
<th>Type 1</th>
<th>Type 2</th>
<th>Type 3</th>
<th>Type 1</th>
<th>Type 2</th>
<th>Type 3</th>
<th>Type 1</th>
<th>Type 1</th>
<th>Type 1</th>
<th>Type 1</th>
</tr>
</thead>
</table>
• The case-based approach avoids the shortcomings of resource-intensive online learning approaches

• Opponent modelling increases effectiveness of game AI significantly

• Further improvements have been made applying generic feature selection and learning feature weights
