Design and Evaluation of an Extended Learning Classifier-based StarCraft Micro AI

Stefan Rudolph, Sebastian von Mammen, Johannes Jungbluth, Jörg Hähner
stefan.rudolph@informatik.uni-augsburg.de
sebastian.von.mammen@informatik.uni-augsburg.de
The Defender assembles his troops to defend the buildings. The Attacker storms towards the enemy's buildings to attack. The units of the Explorer swarm in different directions from the base. The Strategist decided on attacking enemy units.

In particular, we implemented and co-evolved one defensive AI, one aggressive one, one that focuses on exploration and one where XCS takes global strategic decisions. Screen shots of their respective activities are seen in Figure 4.

Right at the beginning of the match, all units move to the upper of two buildings on the map and stay there for its defense until the end. Using the full functionality of an XCS, the Hydralisks' as well as the Queen's behaviors are learned. For the Hydralisks, the condition part of the classifier rules considers the distance to the next visible enemy. Six actions are ordered: (1) Approach and attack the next enemy unit can trigger escape or ensnare airborne units or to hurl parasites, (2) airborne enemy, (3) move to a predefined point, (4) support a friendly unit, (5) protect the hatchery, or (6) burrow. For the Queen the proximity to the closest ground or airborne enemy, (3) move to a predefined point, (4) support a friendly unit, (5) protect the hatchery, or (6) burrow. For the Queen the proximity to the closest ground or airborne enemy, (3) move to a predefined point, (4) support a friendly unit, (5) protect the hatchery, or (6) burrow.
The Defender enemy's buildings to attack. The units of the Explorer swarm in different directions from the base. The Strategist decided on attacking enemy units. The Attacker storms towards the base. The units have built up a great distance to their buildings.

In particular, we implemented and co-evolved one defensive AI, one aggressive one, one that focuses on exploration and one where XCS takes global strategic decisions. Screen shots of representative behaviors of the four implemented AIs are seen in Figure 4. The XCS' reinforcement component positively rewards any attacks, whereas the other actions are only remunerated, if the player is attacked itself or if the next enemy unit can trigger escape or ensnare a friendly unit. For the Queen the proximity to the closest ground or airborne enemy, move to a predefined point, support distance to the next visible enemy. Six actions are considered: (1) Approach and attack, (2) move to the closest ground or airborne enemy, (3) move to a predefined point, (4) support distance to the next visible enemy. For the Hydralisks, the condition part of the classifier rules considers the full functionality of an XCS, the Hydralisks' as well as the Queen's behaviors are considered. Two buildings on the map are built and stay there for its defense until the end. Using the explorer, the defender assembles his troops to defend the buildings. The units of the Explorer swarm in different directions from the base. The Strategist decided on attacking enemy units. The Attacker storms towards the base. The units have built up a great distance to their buildings.
Learning Approach

Diagram:
- **Environment**
  - **Detector**
  - **Population [N]**
  - **Match set [M]**
  - **Action set [A]**
  - **Prediction Array**
  - **GA + cover**
  - **Effector**
  - **Reward** (delay = 1)

Flow:
- Action selection
- Match set [M] to Action set [A]
- Prediction Array to GA + cover

Connections:
- From Environment to Detector
- From Detector to Population [N]
- From Population [N] to Match set [M]
- From Match set [M] to Prediction Array
- From Prediction Array to Action set [A]
- From Action set [A] to Effector
- From Effector to Reward

Feedback:
- Delay = 1
Experiment Setup

300 micro matches for training the AIs
150 micro matches to evaluate the trained AIs
Right at the beginning of the match, all units move to the upper of the base. The Strategist decided on attacking enemy units. The units of the Explorer swarm in different directions from the base. The Attacker storms towards the enemy’s buildings to attack. The defender assembles his troops to defend the buildings. The Attacker storms towards the enemy’s buildings to attack. The units of the Explorer swarm in different directions from the base. The defender assembles his troops to defend the buildings. The Attacker storms towards the enemy’s buildings to attack. The defender assembles his troops to defend the buildings. The Attacker storms towards the enemy’s buildings to attack.

In particular, we implemented and co-evolved one defensive AI, one aggressive one, one that focuses on exploration and one where XCS takes global strategic decisions. Screen shots of their respective activities are seen in Figure 4. The defender (30 wins) attacker (105 wins) strategist (7 wins) explorer (269 wins).
### Co-Evolutionary Learning/Evaluation

<table>
<thead>
<tr>
<th>Defender</th>
<th>enemy</th>
<th>matches</th>
<th>frames</th>
<th>hp left</th>
<th>winner</th>
</tr>
</thead>
<tbody>
<tr>
<td>Explorer</td>
<td>100</td>
<td>+</td>
<td>+</td>
<td>0</td>
<td>+</td>
</tr>
<tr>
<td>Attacker</td>
<td>100</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Strategist</td>
<td>100</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Strategist</td>
<td>50</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Explorer</td>
<td>50</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Attacker</td>
<td>50</td>
<td>-</td>
<td>0</td>
<td>+</td>
<td>+</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Strategist</th>
<th>enemy</th>
<th>matches</th>
<th>frames</th>
<th>hp left</th>
<th>winner</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attacker</td>
<td>100</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Explorer</td>
<td>100</td>
<td>+</td>
<td>+</td>
<td>0</td>
<td>+</td>
</tr>
<tr>
<td>Defender</td>
<td>100</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Explorer</td>
<td>50</td>
<td>-</td>
<td>0</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Attacker</td>
<td>50</td>
<td>+</td>
<td>0</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Defender</td>
<td>50</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0</td>
</tr>
</tbody>
</table>

- "+" indicates a positive outcome (defender won or strategist made a successful move).
- "-" indicates a negative outcome (attacker won or explorer made a successful move).
- "0" indicates no change in health points.

The tables above illustrate the outcomes of various matches between different roles (Explorer, Attacker, Strategist, Defender). The matches are divided into two sets: one with 100 matches and another with 50 matches. Each cell represents the result of a match, with frames indicating the number of frames played, hp left showing the health points remaining, and winner denoting the winning role.
Baseline Evaluation against Non-Learning "AI"