An Independent General Game Player

Christoph Otte

Faculty of Computer Science and Mathematics
University Leipzig

christoph.otte@hotmail.de

In General Game Playing, an artificial intelligence research area, independence of software is an essential property. This project developed a high independent general game player, which used a knowledge base to store its collected experience. To play across games without the direct help of humans, it compared two approaches; the Naïve Storing of essential informations and artificial intelligence algorithms called Learning Classifier Systems.

1 Purpose

Artificial Intelligence (AI) is a research area in natural sciences, which tries to mimic various properties of human intelligence. One of these properties is independence. The latter describes the ability of software to make decisions depending on the current situation without any help of a human. It belongs to a set of properties used for the AI research area called General Game Playing (GGP) [GLP05]. GGP is an application of AI to simulate human game play (especially the analysis and act behaviour). The key purpose of GGP is to create software, which is able to learn game playing without any direct help of humans. In GGP there are several strategies, which have their own pros and cons in various games. To reach a higher degree of independence, this project deals with the development of a high independent general game player called Long Time Player (LTP). The LTP expands the use of artificial algorithms to learn game playing by the decision procedure to prefer the most effective algorithms for different game categories. The GGP belongs to basic research, but its key aspects can be applied for various areas of practical use i.e. for medical decision systems or trade marketing programs.
2 Methods

To play different game categories, the LTP needs a memory function. This ability was implemented in form of a knowledge base, which was used to store the obtained LTP experience. This experience was collected by playing a large corpus of games. Therefore the LTP contains a special training mode to analyse games by a categorisation and store the resulting information in the knowledge base. Subsequent to this training the LTP can play new (unknown) games by a look up for a similar pattern in the knowledge base to prefer an adequate game playing strategy.

LTP’s implementation is based on the following algorithms: choosing the playing actions randomly, Depth Search [Th01] - a searching algorithm for graphs, Monte Carlo Methods [MR95] - simulation based algorithms and Upper Confidence Bounds for Trees [FY08] - an extension of the Monte Carlo Methods.

Within the LTP implementation two different approaches were applied in parallel and finally compared to each other. The first approach is the Naive Storing of the following information via simulation: the category of the analysed game, the used strategy and the average win points reached in all performed simulations of this strategy applied for this game category. To play a new (unknown) game, the LTP uses the strategy with the highest average win points value. The second approach use Learning Classifier Systems (LCS) [BLS02] to perform a respectively more effective learning process. LCS is a rule-based algorithm of AI, which uses Reinforcement Learning (RL) to create problem-concerning rule sets. The integration of LCS within LTP was performed by storing of rules in the according knowledge base and extending the abilities of the LTP’s training mode by RL.

3 Results

Both approaches and the four implemented GGP algorithms were tested in separate sessions on the same corpus of test games (which is not overlapping the previously mentioned corpus of training games). Free player positions in games were occupied with algorithms, which choose their playing actions randomly. The resulting rankings represent the according learning curves shown in Figure 1. Naive Storing and LCS showed different learning curves: while the first adjust its results on the second rank, the results of the second fluctuate between the first and the second rank. A rank describes here the average points of an algorithm in all games of a test run.
4 Discussion

The difference can be explained by the LCS rules. Their key purpose is to acquire as much procedural knowledge as possible. They can generalize some structures in games, to detect these in new games. Nevertheless, within the classification process, it is not ensured that newly created rules work as well as the newly deleted. That's why the learning curve of the second approach is fluctuating. Probably the fluctuations shrink with a tendency to the first rank, when LCS trains even more training games.

For the further development of LTP three strategies can be applied. The first strategy is the use of a more balanced corpus of training games. This can have the effect that the increase of learning process raises faster. The second strategy is an extension of the LTP’s repertoire. More algorithms implicate a finer allocation structure and have thus perhaps a better overall result. The third strategy is to compare a further approach with Naive Storing and LCS. For example a neuronal network representing a sub symbolic learning algorithm can be used to extend this project.

References

[FY08] Hilmar Finnsson, Yngvi Björnsson: Simulation Based Approach to General Game Playing, School of Computer Science, Reykjavik University, Iceland, 2008