NSG Challenge: Parameterized Poker Squares

Todd W. Neller
Gettysburg College
tneller@gettysburg.edu

Introduction

The inaugural EAAI NSG Challenge[1] will be to create AI to play a parameterized form of the game Poker Squares. We here describe the game of Poker Squares, our parameterization of the game, and the Java interface competitors will use to facilitate evaluation.

Poker Squares

Poker Squares[2] (a.k.a. Poker Solitaire, Poker Square, Poker Patience) is a folk sequential placement optimization game[3] appearing in print as early as 1949, but likely having much earlier origins. Using a shuffled 52-card French deck, the rules of (Morehead and Mott-Smith 1949, p. 106) read as follows:

> Turn up twenty-five cards from the stock, one by one, and place each to best advantage in a tableau of five rows of five cards each. The object is to make as high a total score as possible, in the ten Poker hands formed by the five rows and five columns. Two methods of scoring are prevalent, as follows:

HAND	ENGLISH	AMERICAN
Royal flush	30	100
Straight flush	30	75
Four of a kind	16	50
Full house	10	25
Flush	5	20
Straight	12	15
Three of a kind	6	10
Two pairs	3	5
One pair	1	2

The American system is based on the relative likelihood of the hands in regular Poker. The English system is based on the relative difficulty of forming the hands in Poker Solitaire.

[1]Whereas DARPA has its "grand challenges", ours are not so grand.

[2]http://www.boardgamegeek.com/boardgame/41215/poker-squares, http://cs.gettysburg.edu/~tneller/games/pokersquares

[3]http://www.boardgamegeek.com/geeklist/152237/sequential-placement-optimization-games

You may consider that you have "won the game" if you total 200 (American) or 70 (English).

Note that the single remaining Poker hand classification of "high card", which does not fit any of the above classifications, scores no points.

Parameterized Poker Squares

As David Parlett observed, "British scoring is based on the relative difficulty of forming the various combinations in this particular game, American on their relative ranking in the game of Poker." (Parlett 2008, pp. 552–553) We observe that different point systems give rise to different placement strategies.

For example, in playing with British or American scoring, one often has a row and column where one dumps unwanted cards so as to form higher scoring combinations in the other rows and columns. However, a very negative score (i.e. penalty) for the "high card" category would discourage leaving any such row or column without a high probability of alternative scoring.

In our parameterization of Poker Squares, we parameterize the score of each of the 10 hand categories as being an integer in the range $[-128, 127]$. Given a vector of 10 integers corresponding to the hand classification points as ordered in the table above, the player then plays Poker Squares according to the given point system.

The goal is to design Poker Squares AI with high expected score performance across the distribution of possible score parameters.

The EAAI NSG Challenge

Since the purpose of this contest is to promote undergraduate student-faculty research collaboration, each contest team must include a full-time undergraduate and a college faculty member. A faculty member may mentor more than one team of undergraduates. Important dates:

- (one month before the AAAI-16 paper submission deadline) - contest submissions due

- (EAAI-16 paper submission deadline) - papers describing the research and development of contest entries due

At EAAI-16, contest results will be announced, and papers accepted for publication will be presented. Such papers will appear with EAAI papers in the AAAI conference

proceeding and thus must meet AAAI publication formatting and quality specifications. Thus, undergraduates will be guided in the research, publication, and presentation practices of the AAAI research community.

Contest Structure

Contest participants will come to a consensus on a small selection of point systems to be used in evaluating entries. Possibilities include:

- American - (given above)
- English (a.k.a. British) - (given above)
- Ameritish - a randomized hybrid of American and British point systems; includes American and English systems
- Random - points for each hand category are chosen randomly in the range $[-128, 127]$
- Hypercorners - points for each hand category are chosen with equal probability from $\{-1, 1\}$
- Single Hand - only one hand category scores 1 point; all other categories score no points

Hand categories are decided according to the rules of Poker, with higher ranking hand categories taking precedence. For example, a three of a kind also contains one pair, but for scoring purposes will be counted as a three of a kind, even if one pair scores more points than three of a kind. Note that the high card hand category may be awarded points in non-Ameritish systems.

A selection of such systems will be used to evaluate contest players.

Java Contest Code

Contest support code for testing and development will be supplied in Java. Entrants will provide code implementing the `PokerSquaresPlayer` interface. Full Javadoc documentation and Java contest code is available online[4]. If there is a desire to create and support a `PokerSquaresPlayer` that interfaces to another programming language and facilitates student development in another language, please contact Todd Neller[5].

For each point system tested in contest evaluation, each `PokerSquaresPlayer` will be given the point system and 5 minutes to perform preprocessing before beginning game play. For each game, each player will be given 30 seconds of total time for play decision-making. A player taking more than 30 seconds of total time for play decision-making or making an illegal play will score 10 times the minimum hand point score for the game. Players that crash, become non-responsive, or attempt a form of communication with another player will be removed from competition. However, this should be easily avoided as the contest evaluation code is being supplied for testing.

For each point system tested, each player's scores will be summed to a total score and then this total will be normalized to a floating point number ranging from 0 (lowest score

of all players) to 1 (highest score of all players). Players will be ultimately be ranked according to the sum of their normalized scores across all point system tests.

Please note that supplied code conveniently performs scoring of full/partial boards/hands, leaving participant free to focus on strategy. Demonstration players are also provided for baseline comparison and demonstration of simple time management and greedy Monte Carlo play.

At this point, we plan to perform all testing on a Dell Precision M4800 running Windows 7 (64-bit) with and Intel Core i7-4940MX CPU @ 3.1GHz, 32 GB RAM, and running Java version 1.8.0_20 or later.

Tips

Past experience with Poker Squares using only the American point system provided a number of lessons that may prove helpful for this challenge:

- The highest-performance player to date makes use of abstract partial/complete hand descriptions (e.g. "2 of one rank, 1 of another", "1 of one rank, 1 of another, with flush and straight potential"). Estimates of the expected value of *abstract* hands are formed using Monte Carlo methods[6], playing tens of thousands of games with an epsilon-greedy policy. Board positions are then statically evaluated using the sum of the partial/complete abstract hand expected score estimates.

- The second-highest-performance player to date makes use of a hand-crafted rule-based system. Other good performers use Monte Carlo simulation, neural networks. In other words, there are many potential approaches to this problem ranging from simple to complex. Start with the KISS principle.[7]

- Note that the structure of the game tree has alternating chance and choice nodes. This suggests that expectimax is the means of evaluating entire game/subgame trees. Because of the size of the problem, however, expectimax can only be applied to endgame subtrees or be depth-limited in some way (using static evaluation) so as to not exhaust limited time.

- One approach that has not been applied which has been fruitful for similar problems is Monte Carlo Tree Search. Cameron Brown et al has written an excellent survey (Browne et al. 2012). Richard Lorentz's work may also provide relevant MCTS insight (Lorentz 2012).

Given all of these observations, it is important to note that the challenge organizer has little idea what techniques may prove best to apply. However, along with the contest code, a player has been developed that shows it is possible to create a player that adapts well to different scoring systems and appears to play very well with respect to human play.

Thus, Parameterized Poker Squares presents a good research playground for undergraduate research.

---

[4] http://cs.gettysburg.edu/~tneller/games/pokersquares/eaai

[5] http://cs.gettysburg.edu/~tneller

[6] http://webdocs.cs.ualberta.ca/~sutton/book/ebook/node50.html

[7] http://en.wikipedia.org/wiki/KISS_principle

References

[Browne et al. 2012] Browne, C.; Powley, E.; Whitehouse, D.; Lucas, S.; Cowling, P. I.; Rohlfshagen, P.; Tavener, S.; Perez, D.; Samothrakis, S.; and Colton, S. 2012. A survey of Monte Carlo tree search methods. *IEEE Trans. on Computational Intell. and AI in Games* 4(1). also available at http://www.cameronius.com/cv/mcts-survey-master.pdf.

[Lorentz 2012] Lorentz, R. J. 2012. An MCTS program to play EinStein Würfelt Nicht! In van den Herik, H. J., and Plaat, A., eds., *Advances in Computer Games, Lecture Notes in Computer Science 7168, 13th International Conference, ACG 2011, Tilburg, The Netherlands, November 20-22, 2011, Revised Selected Papers*. Springer.

[Morehead and Mott-Smith 1949] Morehead, A. H., and Mott-Smith, G. 1949. *The Complete Book of Solitaire & Patience Games*. Grosset & Dunlap, 1st edition.

[Parlett 2008] Parlett, D. 2008. *The Penguin Book of Card Games*. Penguin Books, updated edition.