11-9-2017

The Birds of a Feather Research Challenge

Todd W. Neller
Gettysburg College

Follow this and additional works at: https://cupola.gettysburg.edu/csfac

Part of the Computer Sciences Commons, Game Design Commons, and the Science and Mathematics Education Commons

Share feedback about the accessibility of this item.


This is the author's version of the work. This publication appears in Gettysburg College's institutional repository by permission of the copyright owner for personal use, not for redistribution. Cupola permanent link: https://cupola.gettysburg.edu/csfac/45

This open access presentation is brought to you by The Cupola: Scholarship at Gettysburg College. It has been accepted for inclusion by an authorized administrator of The Cupola. For more information, please contact cupola@gettysburg.edu.
The Birds of a Feather Research Challenge

Abstract
Neller presented a set of research challenges for undergraduates that allow an excellent formative experience of research, writing, peer review, and potential presentation and publication through a top-tier conference. The focus problem is the analysis of a newly-designed solitaire card game, Birds of a Feather, so potentials for discovery abound. Open access talk slides, research code, solvability data sets, research tutorial videos, and more are also available at http://cs.gettysburg.edu/~tneller/puzzles/boaf.

Keywords
undergraduate research, games, Birds of a Feather, artificial intelligence

Disciplines
Computer Sciences | Game Design | Science and Mathematics Education

Comments
This talk was originally presented at the Gettysburg College Computer Science Colloquium on November 9th, 2017. A follow-up research workshop was held for Gettysburg students at our Computer Science Colloquium on February 1st, 2018. It was also presented at the Eighth Symposium on Educational Advances in Artificial Intelligence 2018 on February 3rd, 2018 and Dickinson College on February 27th, 2018.

This presentation is available at The Cupola: Scholarship at Gettysburg College: https://cupola.gettysburg.edu/csfac/45
The Birds of a Feather Research Challenge

Todd W. Neller
Gettysburg College
November 9th, 2017
Outline

• Backstories:
  – Rook Jumping Mazes
  – Parameterized Poker Squares
  – FreeCell

• Birds of a Feather
  – Rules
  – 4x4 Single Stack Play
  – Experiments
  – Brainstorming
Rook Jumping Maze Design

- **Rook Jumping Mazes** - logic mazes with simple rules based on Chess rook moves
- Few maze designers in history had the skill to create these.
- We worked together to create a metric to rate the quality of mazes and performed combinatorial optimization to **generate high quality mazes**.
Example Maze

- **Specification**: grid size, start state (square), goal state, jump numbers for each non-goal state.

- **Jump number**: Move *exactly* that many squares up, down, left, right. *(Not diagonally.)*

- **Objectives**:
  - Find a path from start to goal.
  - Find the shortest of these paths.
Publication

Rook Jumping Maze Design Considerations

Todd W. Neller¹, Adrian Fisher², Munyaradzi T. Choga¹,
Samir M. Lalvani¹, and Kyle D. McCarty¹

¹ Gettysburg College, Dept. of Computer Science, Gettysburg, Pennsylvania, 17325, USA,
tneller@gettysburg.edu,
WWW home page: http://cs.gettysburg.edu/~tneller
² Adrian Fisher Design Ltd., Portman Lodge, Durweston, Dorset, DT11 0QA England,
adrian@adrianfisherdesign.com,
WWW home page: http://www.adrianfisherdesign.com

Abstract. We define the Rook Jumping Maze, provide historical perspective,
and describe a generation method for such mazes. When applying stochastic local
search algorithms to maze design, most creative effort concerns the definition of
an objective function that rates maze quality. We define and discuss several maze
features to consider in such a function definition. Finally, we share our preferred
design choices, make design process observations, and note the applicability of
these techniques to variations of the Rook Jumping Maze.

in van den Herik, H. Jaap, Iida, Hiroyuki, and Plaat, Aske, eds., LNCS 6515: Computers and Games,
7th International Conference, CG 2010, Kanazawa, Japan, September 24-26, 2010, Revised
Parameterized Poker Squares

• Materials:
  – shuffled standard (French) 52-card deck,
  – paper with 5-by-5 grid, and
  – pencil

• Each turn, a player draws a card and writes the card rank and suit in an empty grid position.

• After 25 turns, the grid is full and the player scores each grid row and column as a 5-card poker hand according to a point system.
# American Point System

<table>
<thead>
<tr>
<th>Poker Hand</th>
<th>Points</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Royal Flush</td>
<td>100</td>
<td>A 10-J-Q-K-A sequence all of the same suit</td>
<td>10♣, J♣, Q♣, K♣, A♣</td>
</tr>
<tr>
<td>Straight Flush</td>
<td>75</td>
<td>Five cards in sequence all of the same suit</td>
<td>A♦, 2♦, 3♦, 4♦, 5♦</td>
</tr>
<tr>
<td>Four of a Kind</td>
<td>50</td>
<td>Four cards of the same rank</td>
<td>9♣, 9♦, 9♥, 9♠, 6♥</td>
</tr>
<tr>
<td>Full House</td>
<td>25</td>
<td>Three cards of one rank with two cards of another rank</td>
<td>7♠, 7♣, 7♥, 8♠, 8♣</td>
</tr>
<tr>
<td>Flush</td>
<td>20</td>
<td>Five cards all of the same suit</td>
<td>A♥, 2♥, 3♥, 5♥, 8♥</td>
</tr>
<tr>
<td>Straight</td>
<td>15</td>
<td>Five cards in sequence; Aces may be high or low but not both</td>
<td>8♣, 9♠, 10♥, J♠, Q♣</td>
</tr>
<tr>
<td>Three of a Kind</td>
<td>10</td>
<td>Three cards of the same rank</td>
<td>2♠, 2♥, 2♦, 5♣, 7♠</td>
</tr>
<tr>
<td>Two Pair</td>
<td>5</td>
<td>Two cards of one rank with two cards of another rank</td>
<td>3♥, 3♦, 4♣, 4♠, A♣</td>
</tr>
<tr>
<td>One Pair</td>
<td>2</td>
<td>Two cards of one rank</td>
<td>5♦, 5♥, 9♣, Q♠, A♥</td>
</tr>
<tr>
<td>High Card</td>
<td>0</td>
<td>None of the above</td>
<td>2♦, 3♣, 5♠, 8♥, Q♣</td>
</tr>
</tbody>
</table>
Scoring Examples

<table>
<thead>
<tr>
<th>Hand Type</th>
<th>Points</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Royal Flush</td>
<td>100</td>
<td>0</td>
</tr>
<tr>
<td>Straight Flush</td>
<td>75</td>
<td>0</td>
</tr>
<tr>
<td>Four of a Kind</td>
<td>50</td>
<td>0</td>
</tr>
<tr>
<td>Full House</td>
<td>26</td>
<td>0</td>
</tr>
<tr>
<td>Flush</td>
<td>20</td>
<td>0</td>
</tr>
<tr>
<td>Straight</td>
<td>15</td>
<td>0</td>
</tr>
<tr>
<td>Three of a Kind</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>Two Pair</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>One Pair</td>
<td>2</td>
<td>0</td>
</tr>
</tbody>
</table>

WON

Total: 229
## Parameterized Poker Squares Results

### Players' Mean Scores by Point System

<table>
<thead>
<tr>
<th>Players</th>
<th>American</th>
<th>Amerish</th>
<th>British</th>
<th>Hypercorner</th>
<th>Random</th>
<th>High Card</th>
<th>One Pair</th>
<th>Two Pair</th>
<th>3 of a Kind</th>
<th>Straight</th>
<th>Flush</th>
<th>Full House</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMO_V2</td>
<td>125.27</td>
<td>105.54</td>
<td>54.50</td>
<td>1.10</td>
<td>437.77</td>
<td>9.37</td>
<td>9.12</td>
<td>4.46</td>
<td>3.20</td>
<td>2.97</td>
<td>3.43</td>
<td>1.82</td>
</tr>
<tr>
<td>DevneilPlayer</td>
<td>14.36</td>
<td>15.27</td>
<td>7.51</td>
<td>-9.52</td>
<td>-86.92</td>
<td>5.22</td>
<td>4.10</td>
<td>0.45</td>
<td>0.21</td>
<td>0.04</td>
<td>0.05</td>
<td>0.03</td>
</tr>
<tr>
<td>Gettysburg</td>
<td>123.94</td>
<td>110.28</td>
<td>53.38</td>
<td>1.24</td>
<td>429.89</td>
<td>9.37</td>
<td>9.17</td>
<td>4.47</td>
<td>3.02</td>
<td>2.71</td>
<td>3.46</td>
<td>1.93</td>
</tr>
<tr>
<td>SRulerPlayer</td>
<td>51.83</td>
<td>55.39</td>
<td>30.29</td>
<td>-5.10</td>
<td>242.85</td>
<td>9.34</td>
<td>8.84</td>
<td>4.04</td>
<td>2.10</td>
<td>1.58</td>
<td>1.98</td>
<td>0.61</td>
</tr>
<tr>
<td>JoTriz</td>
<td>116.75</td>
<td>109.03</td>
<td>53.59</td>
<td>-0.78</td>
<td>351.07</td>
<td>9.31</td>
<td>9.15</td>
<td>4.59</td>
<td>3.03</td>
<td>2.59</td>
<td>3.36</td>
<td>1.67</td>
</tr>
<tr>
<td>xRandomRolloutPruningPlayer</td>
<td>116.12</td>
<td>111.26</td>
<td>53.92</td>
<td>-2.20</td>
<td>411.78</td>
<td>9.35</td>
<td>9.16</td>
<td>4.52</td>
<td>2.89</td>
<td>2.94</td>
<td>3.41</td>
<td>1.82</td>
</tr>
<tr>
<td>MonteCarloTreePlayer</td>
<td>15.47</td>
<td>15.31</td>
<td>7.61</td>
<td>-9.30</td>
<td>-86.83</td>
<td>4.80</td>
<td>4.53</td>
<td>0.45</td>
<td>0.20</td>
<td>0.05</td>
<td>0.02</td>
<td>0.00</td>
</tr>
<tr>
<td>RandomPlayer</td>
<td>14.25</td>
<td>15.67</td>
<td>7.71</td>
<td>-9.66</td>
<td>-106.80</td>
<td>5.20</td>
<td>4.31</td>
<td>0.42</td>
<td>0.23</td>
<td>0.01</td>
<td>0.01</td>
<td>0.01</td>
</tr>
</tbody>
</table>

| Max                   | 125.27   | 111.26  | 54.50   | 1.24        | 437.77 | 9.37      | 9.17     | 4.59     | 3.20        | 2.97     | 3.46  | 1.93       |
| Min                   | 14.25    | 15.27   | 7.51    | -9.66       | -106.80| 4.80      | 4.10     | 0.42     | 0.20        | 0.01     | 0.01  | 0.00       |

### Normalized Scores

<table>
<thead>
<tr>
<th>Players</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMO_V2</td>
<td>11.821</td>
</tr>
<tr>
<td>DevneilPlayer</td>
<td>0.190</td>
</tr>
<tr>
<td>Gettysburg</td>
<td>11.763</td>
</tr>
<tr>
<td>SRulerPlayer</td>
<td>7.149</td>
</tr>
<tr>
<td>JoTriz</td>
<td>11.170</td>
</tr>
<tr>
<td>xRandomRolloutPruningPlayer</td>
<td>11.334</td>
</tr>
<tr>
<td>MonteCarloTreePlayer</td>
<td>0.192</td>
</tr>
<tr>
<td>RandomPlayer</td>
<td>0.153</td>
</tr>
</tbody>
</table>
Monte Carlo Approaches
to Parameterized Poker Squares

Todd W. Neller\textsuperscript{(2)}, Zuozi Yang\textsuperscript{1}, Colin M. Messinger\textsuperscript{1}, Calin Anton\textsuperscript{2},
Karo Castro-Wunsch\textsuperscript{3}, William Maga\textsuperscript{2}, Steven Bogaerts\textsuperscript{3},
Robert Arrington\textsuperscript{3}, and Clay Langley\textsuperscript{3}

BeeMo, a Monte Carlo Simulation Agent for Playing Parameterized Poker Squares

Karo Castro-Wunsch, William Maga, Calin Anton
MacEwan University, Edmonton, Alberta, Canada
karantonio@gmail.com, magaw@mymacmewan.ca, antonc@macmewan.ca

Abstract
Parameterized Poker Squares is a game playing agent. We organized the agent in three dimensions: partial hand abstraction, algorithms, and partial hand utility. To an abstract hand, we implemented five algorithms, among which we selected BeeMo, our final product. BeeMo utilizes Monte-Carlo search. The search is based on hand patterns utilized through an iterative improvement of Monte-Carlo simulations and optimization

Using Domain Knowledge to Improve Monte-Carlo Tree Search Performance in Parameterized Poker Squares

Robert Arrington, Clay Langley, and Steven Bogaerts
Department of Computer Science
DePauw University
Greencastle, IN, USA
{robertarrington.2015, claylangley.2017, stevenbogaerts}@depauw.edu

Learning and Using Hand Abstraction Values for Parameterized Poker Abstraction Squares

Todd W. Neller and Colin M. Messinger and Zuozi Yang
Gettysburg College
tneller@gettysburg.edu

Abstract
BeeMo's learning and the hand abstraction

\begin{tabular}{|l|c|c|c|}
\hline
Hand Type & Am. & Brit. & El. \\
\hline
royal flush & 100 & 50 & 20 \\
straight flush & 75 & 30 & 10 \\
4 of a kind & 50 & 16 & 4 \\
straight & 25 & 10 & 2 \\
full house & 20 & 5 & 1 \\
3 of a kind & 15 & 12 & 3 \\
flush & 10 & 6 & 1 \\
2 pair & 5 & 3 & 0 \\
1 pair & 3 & 2 & 0 \\
high card & 0 & 0 & 0 \\
\hline
\end{tabular}
FreeCell Solitaire Card Game
FreeCell Characteristics

• Randomly generated, but no chance after face-up deal (perfect information) → Combinatorial game
• Self-generating puzzle that is solvable with high probability
• Invited many interesting research questions posed and solved by skilled enthusiasts
• Would have been great for undergraduate research, but largely harvested.
• To gain new low-hanging fruit, plant a new tree!
Birds of a Feather Characteristics

• Randomly generated, but no chance after face-up deal (perfect information) → Combinatorial game

• Self-generating puzzle that is solvable (for certain deal dimensions) with high probability

• Invites many interesting research questions (to be introduced later)

• But first, we introduce the game...
Birds of a Feather

- “Birds of a feather flock together.”
- Designed August 9, 2016
- Materials: a standard, shuffled 52-card deck
- Setup: Deal cards singly, face-up into a grid (e.g. 4-by-4).
- Object: Form a single stack of all cards.
- Play: A player may move one stack of cards onto another stack of cards in the same row or column if the cards on top of the stacks have either
  - (1) the same suits, or
  - (2) the same or adjacent ranks. Aces are low and not adjacent to kings, so rank adjacency is according the ordering A, 2, 3, ..., J, Q, K.
Birds of a Feather: Adjacent Rank
Birds of a Feather: Same Suit
Birds of a Feather: Same Rank
Birds of a Feather: Single Stack Goal
Birds of a Feather Java Project

• Import Eclipse Java project...
  – File → Import... → General → Existing Project into Workspace
  – Select “Select Archive File” radio button and click “Browse” button.
  – Select /Courses/Colloquia/tneller171109/BirdsOfAFeather.zip
  – Click “Finish”

• Open BirdsOfAFeather.java (GUI) in project BirdsOfAFeather

• Run
Birds of a Feather Example Deal
Birds of a Feather Example Solution

TS-9H  AH-TH  AH-3H
AH-QH  6H-7D  JS-JC
KS-3S  KS-KC  5S-KS
6H-5C  5S-TS  AH-8H
6H-AH  5S-6H  5S-JS
Experiment 1: Are all 4x4 deals solvable?

- Open Experiment1.java and run.
  - Program attempts to solve deal 0, 1, 2, ...
  - Are all solvable? If not, what is the first that isn’t solvable?
- Press the red square by the Eclipse console window to terminate the experiment.
- Run BirdsOfAFFeather and type ‘s’ to create seed 10 puzzle.
- What makes this unsolvable?
- Type ‘t’ to toggle connections between flockable card pairs.
- Let’s call a single unflockable card an odd bird.
Experiment 2: Do all unsolvable deals have an odd bird?

• For efficiency, we’ll divide the solving ranges and work in parallel. Wait for your input parameters.
• Open Experiment2.java and run. Odd bird deals will be identified and skipped.
• Every time you find an unsolvable deal (odd bird or otherwise), add it to your list of unsolvable deal numbers and note if it is an odd bird deal.
• After covering your range, examine any unsolvable deals using BirdsOfAFFeather.java. If there are no unsolvable deals or they’re easily understood, examine deal 1163 or 1264.
Example Research Questions

• m-by-n single-stack deal solvability/scoring:
  – What is the probability that a deal will have a single-stack solution?
  – What is the maximal score distribution of deals?
  – Which features of a puzzle can guide a computer efficiently to a successful solution?
  – Which algorithms work best to solve such puzzles?
  – What are characteristics of grids without single-stack solutions?

• Puzzle solving heuristics:
  – Which features of a puzzle can guide a player to a successful solution?

• Puzzle design:
  – How would you create a metric for a good BirdsOfAFeather puzzle?
  – Which techniques work best for designing puzzles for such a metric?
  – How can one generate a succession of gradually more difficult puzzles to guide and teach a player to solve such puzzles well?
Conclusion

• Birds of a Feather offers a new landscape for research exploration.
• Together, we can understand more about puzzle solving, puzzle design, etc. and gain research, writing, presentation, and publication experience in the process.
• Please contact Todd Neller (tneller@gettysburg.edu) if you’re interested!