# Computer Generation of Birds of a Feather Puzzles

**Todd W. Neller, Daniel Ziegler**
Gettysburg College
{tneller,ziegda03}@gettysburg.edu

## Abstract

In this article, we describe a computer-aided design process for generating high-quality Birds of a Feather solitaire card puzzles. In each iteration, we generate puzzles via combinatorial optimization of an objective function. After solving and subjectively rating such puzzles, we compute objective puzzle features and regress our ratings onto such features to provide insight for objective function improvements. Through this iterative improvement process, we demonstrate the importance of the halfway solvability ratio in quality puzzle design. We relate our observations to recent work on tension in puzzle design, and suggest next steps for more efficient puzzle generation.

## Introduction

Birds of a Feather (Neller 2016) is an original perfect-information solitaire game played with a standard 52-card deck. After shuffling, the player deals the cards face-up into an $r$-by-$c$ grid of cards. In this paper we focus on the case where $r = c = 4$.

The object of Birds of a Feather is to move all grid cards into a single stack. Think of each grid cell as initially containing a 1-card stack. A stack may be moved on top of another stack in the same row or same column if the top cards of the two stacks have the same suit, the same rank, or adjacent ranks. Aces and kings are considered low and high, respectively, and are non-adjacent to each other.

For example, consider the initial game state shown in Figure 1(a). Having the same rank in the same column, the JS (Jack of Spades) stack can move onto the JC (Jack of Clubs) (Fig. 1(b)). Having adjacent rank in the same column, the TS (Ten of Spades) stack can move onto the 9H (9 of Hearts) (Fig. 1(c)). Having the same suit in the same row, the JS stack can move onto the 5S (Fig. 1(c)). We can solve this deal, moving all cards into a single stack, with this sequence of moves: JS→JC, TS→9H, JS→5S, KS→3S, KS→KC, JS→KS, JS→TS, 6H→7D, 6H→5C, 6H→8H, QH→AH, QH→TH, QH→3H, QH→JS, QH→6H.

Our paper begins with an outline of our design methodology and definitions of puzzle features we found to be of potential interest through our process. We then share our experience of the iterative design process and what we learned about the most important features to optimize for quality puzzle design. Next, we relate our insights to recent work in characterizing tension in puzzle design. Finally, we suggest future work that could make our puzzle generation algorithm more efficient.

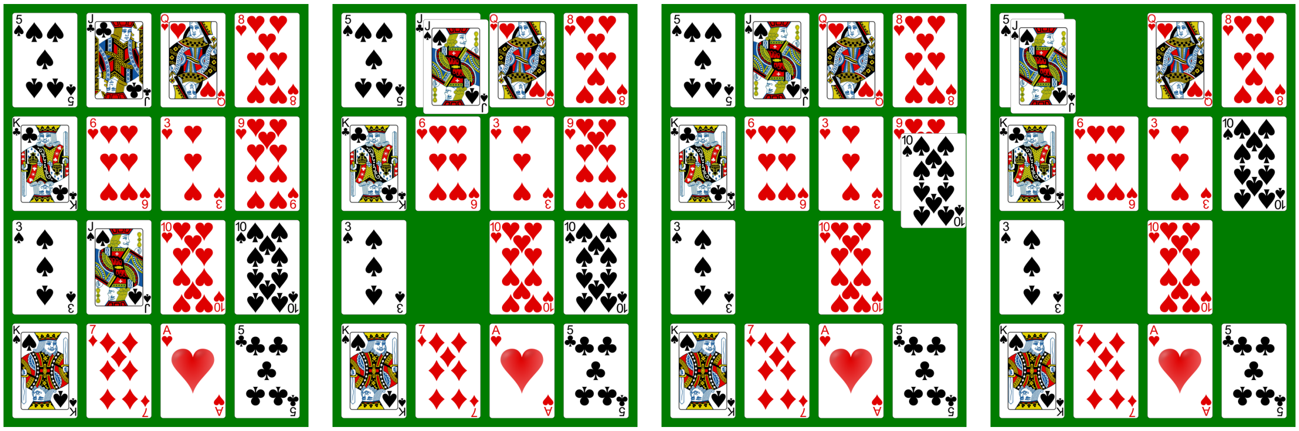
## Objective and Methodology

The objective of this work was to create an algorithm to generate high-quality Birds of a Feather puzzles for varying numbers of cards within a 4-by-4 card grid. We envision the implementation of a puzzle application where the player progresses through puzzles with greater numbers of cards (up to 16) as they meet success with puzzles having smaller numbers of cards. Thus, we are faced with two primary challenges: (1) defining a measure of puzzle quality, and (2) generating such puzzles efficiently.

We first describe our general approach before delving into specifics. Our puzzle generation methodology is similar to that of (Neller et al. 2011), in which puzzle generation is approached as a combinatorial optimization in the space of possible puzzles:

1. Define an objective function measure of puzzle "badness" that we seek to minimize.
2. Generate a set of puzzles by taking randomly generated initial puzzles and approximately optimizing them for a fixed number of iterations of stochastic local search (SLS) (Hoos and Stützle 2004).
3. Manually solve this set of generated puzzles and rate them according to our subjective enjoyment of their challenges. If we are satisfied with the puzzle quality, we terminate design.
4. Look for relationships between our subjective puzzle ratings and objective puzzle features in order to gain insight into what would improve our objective function. We then redefine our objective function, possibly introducing new features or re-weighting previous features according to such insight.
5. Return to step 2.

We now describe our step-by-step process in greater detail. The objective function we minimize is referred to as our *energy function* using the energy-minimization physics analogy of simulated annealing. In our very first iteration, we began with the simplest of puzzle energy ("badness") measures, defining a two-valued measure with a high (1000)



(a) Initial state

(b) After JS→JC

(c) After TS→9H

(d) After JS→5S

Figure 1: Example Birds of a Feather moves

or low (0) value for unsolvable or solvable puzzles, respectively. Unsolvability is the most undesirable feature of any puzzle, so seeking to create unbiased solvable puzzles is a reasonable starting point.

For the next step of our process, our choice of SLS algorithm was a probabilistic iterative improvement algorithm called Hill-Descent with Random Uphill Steps (Neller 2005). The stochastic local step mutates our current puzzle state in one of two ways: For 30% of local steps, we swap a random unused card in for a random card of our puzzle. The other 70% of local steps consist of swapping a random card in our grid with another grid location's contents (empty or not).

After this local step to a next state, the next state energy is evaluated. If it is found to be greater (i.e. worse) than the previous state, we accept it with probability 0.005, and reject it otherwise, reverting to our previous state. This is repeated for 2500 iterations for fewer than 10 cards, 800 iterations for 10 to 12 cards, and 50 iterations for more than 12 cards.

After repeated applications of such puzzle generation through optimization with SLS, we collected a new set of puzzles. We then used a graphical user interface for manual solution of these puzzles. We solved each puzzle, recorded our rating on a scale from 0 to 10, and logged our puzzle observations as plain text. This subjective data was then stored into a comma-separated value (CSV) data file along with objective computed puzzle feature data as described in the next section. In each of these phases we collected between 15 and 40 puzzle ratings.

We next computed a linear regression of our ratings on our normalized numeric features and noted regression coefficients. Since puzzle solution and rating is a laborious process and sample sizes are relatively small, we found sample sizes insufficient to demonstrate statistical significance for prediction of ratings using puzzle features. Nonetheless, the feature coefficient signs were consistent with the (un)desirability of features, and the coefficient magnitudes helped us to both determine the relative importance of normalized puzzle features and suggested improvements to our energy function. We would also observe that puzzle solving skill, puzzle preferences, and puzzle ratings may change over a person's experience of a puzzle, yet this data-driven approach nonetheless provided a meta-level iterative improvement process for puzzle design.

If the suggested improvements to our energy function did not correlate with our subjective puzzle preferences, we looked to add to or adjust our weightings of objective puzzle features in order to more accurately model our subjective preferences. Finally, we returned to puzzle generation, repeating this iterative process until we were satisfied with puzzle quality.

## Puzzle Features of Interest

In order to generate puzzles through optimization, one must carefully design the objective function for optimization. The development of our function explored multiple features of puzzle states. In this section, we define and discuss these features.

Two cards are said to be *flockable* if they have the same suit, the same rank, or adjacent ranks. A *flockability graph* is a graph where each card is a node and there is an edge between two nodes if and only if the two corresponding cards are flockable. The *average flockability* of a game state is the total number of flockability graph edges divided by the number of cards in the grid. Flockability is different from legal move relations in that we ignore same-row/columns constraints in order to gain an estimate of future "mobility" in play.

*Suit frequency*, i.e. how often a suit occurs in the puzzle, allows for helpful measures of state suit diversity. A puzzle that has a high frequency of a particular suit and low frequency of all remaining suits is uninteresting because a sweep of same-suit plays often allows for a simple, uninteresting solution. We use suit frequency to calculate the *dominant suit ratio*, which is the suit with the highest frequency divided by the total number of cards on the grid. We use this ratio as one measure of card diversity.

One can also use characteristics of a subgraph of the flockability graph based on same/adjacent rank, called the *rank-flockability graph*. Cards in the same *rank cluster* share a connected component of the rank-flockability graph. For example, the cards 2D, 3C, and 4H would all be in the same rank cluster. The number of rank clusters is a significant feature that contributes to the flockability because fewer rank clusters means that cards have more connections based on rank.

The *solvability ratio* of a puzzle state is a feature of great importance. More specifically, we refer to the *depth $d$ solvability ratio* as the ratio of solvable to total states after $d$ moves. When $d = \lfloor \frac{n}{2} \rfloor$, where $n$ is the number of initial cards, we call this the *halfway solvability ratio*. If the ratio of solvable to total states is high/low at a certain depth, it means that there is a high/low probability that a random legal move will lead to a solvable state. In order to calculate this ratio, we perform a full search that prunes repeated states.

## Experimental Results

Throughout our experimentation, we changed our feature set and objective function over time so as to generate more interesting puzzles. In this section, we will discuss the evolution of our objective function, which we refer to as our *energy function*.

### Iteration 1: Suit Diversity, Rank Diversity

We started our process by generating and solving 5 different 4-card-puzzles and 5 different 5-card-puzzles. These puzzles were generated with solvability as the only criterion. After solving each puzzle, we recorded our thoughts on the quality of the puzzle. We came across two 4-card-puzzles that yielded insights. One of the 4-card-puzzles had the cards 4D, 4S, 4H, and 4C. The other 4-card-puzzle had the cards 5H, 9H, TH, and KH. From these two puzzles, we observed that puzzles with a low suit diversity or a low rank diversity are uninteresting because their solution tends to be trivial.

We thus introduced the features of suit diversity and rank diversity as our first preferred features in our objective function. The energy was calculated with the *suit diversity*, represented by the number of different suits, and the *rank diversity*, represented by the number of different ranks. For all of our energy functions, we continue to assign a fixed energy value of 1000 for unsolvable states. However, for our next energy function, solvable state energies are computed by modifying a base energy value of 0 as follows:

- If the number of suits $>$ 1 and the number of suits $<$ the total number of cards, subtract 10 from the energy value.
- If the number of ranks $>$ 1, subtract 10 from the energy value.

Using our new energy function, we generated puzzles and repeated the process of puzzle solving and subjective data collection. We solved 9 different 5-card-puzzles and 6 different 6-card puzzles. We collected our data on these puzzles along with our own personal ratings from 0-10, with 0 being a low rating and 10 being the highest rating. Each rating was saved along with the features preferred in our energy function. We computed a linear regression of our ratings on our normalized numeric features. The regression coefficients, shown in Table 1, suggested that we should be maximizing the number of different ranks and minimizing the number of different suits. This regression feedback did not correlate with our subjective ratings, because we concluded from our initial puzzle generation that a minimal number of suits is a bad thing.

| Features | Coefficients | Std. Error |
|---|---:|---:|
| Intercept | 2.7273 | 5.6351 |
| Suit Diversity | -0.1515 | 1.1937 |
| Rank Diversity | 0.9091 | 0.9571 |

Table 1: Regression coefficients for iteration 1.

Looking back at the different puzzles that we had solved, we judged that desired card diversity was not modeled well. We decided that a better feature for expressing desired suit diversity was the dominant suit ratio. As seen in the following 6-card puzzle, a puzzle could have multiple suits represented, but could still have a high dominant suit ratio.

| | | | |
|---|---|---|---|
| 4D | -- | -- | -- |
| -- | 5D | 2C | 2D |
| 3D | -- | AD | -- |
| -- | -- | -- | -- |

This puzzle is too simple, as one can same-rank flock over the one lone suit, followed by a sweep of remaining same-suit plays. The dominant suit ratio of this puzzle would be 5/6, which we found to be high.

Another feature that we introduced to better represent rank diversity was the number of rank clusters. We came to this conclusion that because the 6-card-puzzle above has multiple different ranks represented but only has one rank cluster. This creates excess card mobility.

### Iteration 2: Dominant Suit Ratio, Number of Rank Clusters

With the introduction of the dominant suit ratio and the number of rank clusters, we changed our objective function to prefer puzzles with a low dominant suit ratio and more than one rank cluster.

Before we began to generate puzzles and collect more subjective play data, we introduced 2 new features that would be calculated for each puzzle. The depth 1 solvability ratio and the average flockability were features that we believed would contribute to the quality of a puzzle. The average flockability was introduced because it provided a measure of card mobility. The depth 1 solvability ratio was introduced in an effort to see how crucial the first move was to the quality of a puzzle. Although our energy function only considered the dominant suit ratio and the number of rank clusters, exporting all current features of interest would eventually yield insight into the development of higher quality puzzles.

Our observations that quality puzzles have a low dominant suit ratio and at least one rank cluster led us to the formation

| Features | Coefficients | Std. Error |
|---|---:|---:|
| Intercept | 5.1556 | 1.9873 |
| Depth 1 Solvability | -8.5229 | 1.7883 |
| Dominant Suit Ratio | 3.0158 | 3.7160 |
| Number of Rank Clusters | 0.4377 | 0.2607 |
| Average Flockability | 0.2167 | 0.4530 |

Table 2: Regression Coefficients from Iteration 2.

of a different energy function for solvable states, modifying a base energy value of 0 as follows:

- If the dominant suit ratio $< 0.5$, subtract 10 from the energy value.
- If the number of rank clusters $> 1$, subtract 10 from the energy value.

Using the energy function above, we generated puzzles and repeated the process of puzzle solving and subjective data collection. We solved 4 different 5-card-puzzles, 11 different 6-card puzzles, 5 different 7-card puzzles, 18 different 8-card puzzles, and 2 different 9-card puzzles. The computed linear regression coefficients for our normalized numeric features are shown in Table 2.

From the regression in Table 2, we see that the dominant suit ratio and number of rank clusters both had positive coefficients. Although we prefer a higher number of rank clusters, we do not prefer a high dominant suit ratio. Since the high coefficient for the dominant suit ratio does not coincide with our subjective preferences, we disregarded the high coefficient and kept the preference of a dominant suit ratio less than 0.5. Since it is a subjective preference to have more than 1 rank cluster, we decided to keep the condition in our energy function, preferring puzzles with more than one rank cluster.

As for the depth 1 solvability ratio, the large negative coefficient means that the lowest ratios yield the highest ratings, which correlates with our subjective puzzle preferences. This coefficient led us to adjust our energy function in an effort to minimize the depth 1 solvability ratio.

Since the average flockability coefficient had a small positive magnitude, we looked to our highest-rated individual puzzle feature data in order to judge a desirable average flockability. We observed that higher-rated puzzles have an average flockability between 2 and 4, so we adjusted our energy function to reflect this observation.

A puzzle that received a high rating with this energy function is the following:

| | | | |
|---|---|---|---|
| -- | -- | KH | -- |
| 2S | -- | JD | QC |
| -- | -- | TS | JS |
| -- | -- | 3C | -- |

This puzzle has 2 rank clusters and has a dominant suit ratio of 3/7. This puzzle had the lowest depth 1 solvability ratio out of all of the puzzles generated by this energy function. The regression coefficient for the depth 1 solvability ratio did not come as a surprise because a puzzle that is initially more difficult tends to yield a higher subjective rating. The low depth 1 solvability ratio makes the puzzle very difficult with many first moves that lead to an unsolvable state.

## Iteration 3: Depth 1 Solvability Ratio, Average Flockability

We observed that high quality puzzles have many possible missteps within the first few moves. This observation led us to the introduction of two new features that would be calculated for each puzzle. These features are the depth 2 and depth 3 solvability ratios.

After adjusting, adding, and reweighting features, our new energy function begins with a base energy value of the depth 1 solvability ratio times the puzzle size, and modifies this value as follow:

- If the number of rank clusters $> 1$ and the dominant suit ratio $< 0.5$, subtract 5 from the energy value.
- If $2 \leq$ the average flockability $\leq 4$, subtract 5 from the energy value.

Using the energy function above, we generated puzzles and repeated the process of puzzle solving and subjective data collection. We solved 5 different 5-card-puzzles, 10 different 6-card puzzles, 9 different 7-card puzzles, and 10 different 8 card puzzles.

Through this iteration, we found that there was a large increase in puzzle difficulty. The following puzzle is an example 8-card puzzle generated by the iteration 2 energy function:

| | | | |
|---|---|---|---|
| -- | -- | -- | 5S |
| JD | 7C | TC | 3D |
| -- | TS | -- | 3S |
| -- | -- | -- | TH |

Compare the previous puzzle to this next example of an 8-card puzzle generated using the iteration 3 energy function:

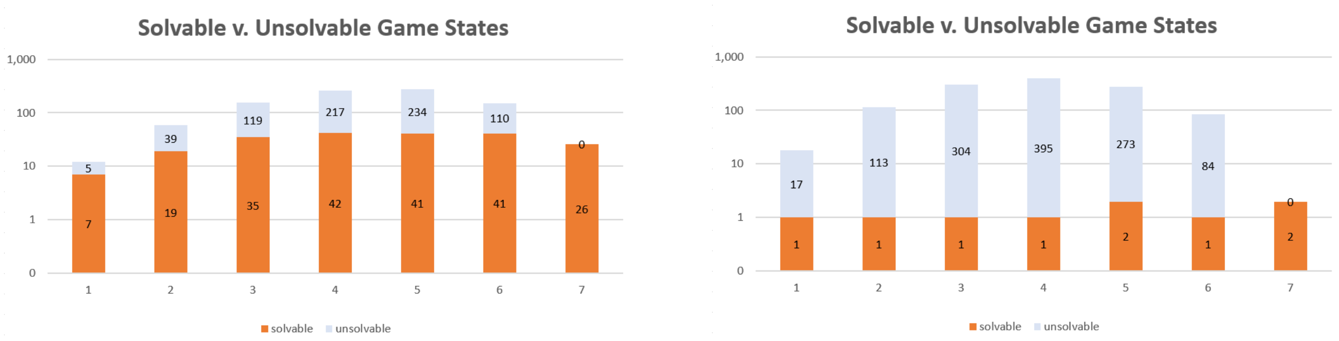
| | | | |
|---|---|---|---|
| -- | -- | 5D | -- |
| 4D | -- | 5H | -- |
| -- | KD | 5C | -- |
| 9S | 9C | 4C | -- |

There is a significant difference in difficulty between these two puzzles. Figure 2 shows us that this difficulty is related to the low ratio of solvable states. Each puzzle's bar chart shows the number of solvable versus unsolvable states at each depth on a logarithmic scale.

Computing a linear regression of our ratings on our normalized numeric features, we observe the regression coefficients shown in Table 3.

According to the regression, the depth 2 and depth 3 solvability ratios have a strong negative coefficient.

Looking specifically at the depth 3 solvability ratio, the majority of the 6- through 8-card puzzles from this generation were halfway solved at depth 3. We thus hypothesized that the highest quality Birds of a Feather puzzles have a low (but not zero) solvability ratio at a depth approximately halfway to the solution. This observation led us to the introduction of the *halfway solvability ratio*.



(a) Iteration 2 puzzle

(b) Iteration 3 puzzle

Figure 2: Logarithm of solvable vs. unsolvable game states by depth

| Features | Coefficients | Std. Error |
|---|---:|---:|
| Intercept | 10.75160 | 3.11937 |
| Depth 1 Solvability | 2.18117 | 4.65598 |
| Depth 2 Solvability | -8.48164 | 10.09565 |
| Depth 3 Solvability | -12.87851 | 6.15367 |
| Dominant Suit Ratio | 1.87682 | 3.12199 |
| Rank Clusters | -0.07801 | 0.41600 |
| Average Flockability | -1.20047 | 0.71890 |

Table 3: Regression coefficients from iteration 3.

| Features | Coefficients | Std. Error |
|---|---:|---:|
| Intercept | 11.2486 | 2.5070 |
| Depth 1 Solvability | 3.0164 | 2.8270 |
| Halfway Solvability | -40.7491 | 27.9119 |
| Dominant Suit Ratio | -3.0697 | 2.3928 |
| Rank Clusters | 0.2660 | 0.4185 |
| Average Flockability | -1.0141 | 0.6472 |

Table 4: Regression coefficients from iteration 4.

### Iteration 4: Solvability Ratio at the Halfway Point

Our data showed that the number of rank clusters and the dominant suit ratio were not as significant as the halfway solvability ratio. We thus created our new energy function to largely consider this ratio, but still retain prior useful features. Our next energy function has a base energy value of 1000 times the halfway solvability ratio times the puzzle size and then modifies this value as follow:

- If the number of rank clusters $> 1$ and the dominant suit ratio $< 0.5$, subtract 10 from the energy value.
- If $2 \leq$ the average flockability $\leq 4$, subtract 10 from the energy value.

We repeated our process of puzzle generation with this new energy function, gave subjective puzzle ratings, and collected feature data. After solving 3 different 6-card puzzles, 7 different 7-card puzzles, 16 different 8-card puzzles, and 5 different 9-card puzzles, we computed a linear regression of our ratings on our normalized numeric features. The regression coefficients are shown in Table 4.

According to the regression, the halfway solvability ratio has a very strong negative coefficient. Given the importance of this normalized feature, we decided to try only preferring a minimum non-zero halfway solvability ratio in our energy function. We found that making this change to our energy function did not appear to degrade the good perceived quality of the puzzles generated. Thus, we concluded our iterative design process.

Looking back at our design process, we note that there were a number of times when a close examination of specific puzzle instances and ratings caused us to disregard regression coefficient signs or the seeming relative importance of a feature. Certainly, our puzzle sample sizes were relatively small and such statistical information may seem of little value. Nonetheless, we credit these statistical checks with providing us the “scent” that put us on the trail of the important halfway solvability ratio that was our key finding.

## Puzzle Tension

In this section, we summarize Cameron Browne’s recent exploration of the concept of puzzle tension (Browne 2017), characterize our work according to Browne’s classification system, and show how our observations are in support of his efforts to formalize a notion of tension in puzzle design.

Popular game designer Wolfgang Kramer noted *tension* as a key design factor of good games, introducing a notion of *tension curves*, but not going so far as to define a measure that could be graphed to produce such curves (Kramer 2000). Browne informally described tension as follows:

> Tension is related to the amount of impact that players’ choices have on the outcome of the game.

Browne further proposes a simple metric for estimating the tension $T$ in puzzle state $S$:

$$T(S) = 1 - \frac{M_{S_w}}{M_S}$$

where $M_S$ is the number of moves available in $S$ and $M_{S_w}$ is the number of winning moves available in $S$.

In the context of Birds of a Feather puzzles, we especially penalize lack of tension at the midgame, i.e. we seek to maximize midgame tension. This recognizes two common features of good puzzles we have observed: (1) while there may be a number of winning initial moves, there need to be many possible opening blunders to achieve a low ratio of midgame winning states, and (2) there will be no tension in the last winning play, as only two winning moves will remain with the last two stacks flocking together in either direction.

We would observe that many excellent puzzles share an "hourglass" structure of midgame tension. In difficult FreeCell puzzles (Keller 2016), there are often many opening move sequences leading to highly-constrained, high-tension midgame play which is followed by low-tension endgame playout. In difficult Sudoku puzzles, there are often easy opening play deductions leading to difficult midgame play deductions which are followed by easy endgame play deductions.

One even sees this hourglass or bottleneck state-space feature in mechanical take-apart puzzles such as the two-piece Devil's Claw[1]. Watching a video of the solution (tgreenless 2008), we see that the crucial part of the solution is very constrained. Holding one of the two pieces fixed, we can describe the relative position and orientation of the other piece with 6 parameters (e.g. a Euclidean fixed point on the piece with Euler angles describing the piece orientation with respect to that point), so that the state space may be represented as a 6-dimensional maze where maze "walls" are formed by the sets of infeasible states in which both puzzle pieces intersect. In this space, movement is somewhat constrained at first, until one finds the highly constrained key movement path apart.

Whether discrete or continuous, many good puzzles exhibit at least some hourglass or bottleneck point of high tension in its state space. In our experience, this is especially true of interesting Birds of a Feather puzzles.

In Browne's puzzle classification, Birds of a Feather puzzles are *dynamic*, i.e. state changes can be further modified, as in contrast to logic puzzles like Sudoku where a value filled in may not be changed. As a dynamic puzzle, our solvable Birds of a Feather puzzles are classed as *interesting*, as opposed to *uninteresting* trivial (one-move solvable) or unsolvable puzzles. They are *monotonic* challenges where "all state transitions progress towards a solution (or dead-end if a mistake is made)". Browne defines a *key state* as a state that must be visited on the solution path, and defines *structured* and *strict* monotonic puzzles as those for which some and all, respectively, of the states on a solution path are key states. Since our final move on a solution path is necessarily non-unique, our generated Birds of a Feather puzzles are thus dynamic, interesting, monotonic, structured puzzles according to Browne's taxonomy.

## Future Work

As mentioned earlier, the objective of this work was to create an algorithm to generate high-quality Birds of a Feather puzzles for varying numbers of cards within a 4-by-4 card grid. In our subjective judgment, we believe that we have succeeded in this via a stochastic local search that minimizes the halfway solvability ratio. While the simplicity of this insight is appealing, our algorithm is currently slow to generate full 16-card puzzles on the 4-by-4 grid.

Thus, our immediate future challenge is to increase the efficiency of our generator for large puzzles. We see two promising potential paths forward that do not require a complete search in order to provide an approximation of the halfway solvability ratio. In both cases, we can first apply the solver of (Neller et al. 2019) to perform a solvability check with a low average cost of 5.3 ms per random 16-card deal. While we would expect our difficult puzzles to have a cost higher than this, we believe that an initial check for a single solution can help us efficiently determine whether or not the halfway solvability ratio is nonzero.

One possible approximation method would be to randomly sample midgame states and perform efficient solvability searches on each, caching results to avoid repeated states across all samples. If we find no solvable midgame states in our sample, our initial solvability check would guarantee the existence of one. Taking this approach, one might seek to predict the total number of midgame states from the number of repeated states in our midgame state sample.

Another approximation method would be to perform a type of beam search, employing state flockability for our beam heuristic as recommended in (Neller et al. 2019). For this second approach, we might find all or a large proportion of solvable midgame states, but we would again need to make use of information from our beam search to predict the total number of midgame states. For this approach, one might predict the total number of midgame states from the product of pruning ratios for each level of the beam search.

Other potential future work might include tuning of the ratio of different types of local steps and parameters of our stochastic local search. However, greater efficiency in energy function computation would be a necessary prerequisite to such work.

## Conclusions

In this work, we have explored the automated generation of high-quality Birds of a Feather puzzles for varying numbers of cards within a 4-by-4 card grid. Our basic generation algorithm was a stochastic local search called Hill Descent with Random Uphill Steps. While the algorithm is simple, much of the work in such puzzle design is the formation of the energy function that is minimized when seeking good puzzle designs.

Each iteration of our design process began with the definition of an energy function seeking to measure puzzle badness based on our observations of prior puzzles and (in subsequent iterations) insights from regressing our puzzle ratings on objective puzzle features. We then generated puzzles using this energy function, solved them, rated them, and took notes on what was observed in the design of such puzzles. The observations sometimes suggested new features that might be helpful in the improvement of our energy function. We then performed a regression of all puzzle ratings

[1] a.k.a. Hanayama's "Cast Devil", Binary Arts' "Twin Tangle", Professor Puzzle's "The Menace", "Double-W", "Double-M", and "M&W Puzzle",

onto the features of such puzzles in order to suggest changes to our energy function for the next iteration.

To our surprise, a single feature appears to be our strongest indicator of puzzle quality. The solvability ratio of depth $d$ is the ratio of solvable to unsolvable states after $d$ moves. When $d = \lfloor \frac{n}{2} \rfloor$, where $n$ is the number of initial cards, we call this the *halfway solvability ratio*. Minimizing this ratio above zero may be used to generate challenging high-quality Birds of a Feather puzzles.

This observation affirms Cameron Browne's recent work on puzzle tension (Browne 2017) that defines tension in terms of the fraction of successor states that lead to failure in solving a puzzle. We have characterized solvable Birds of a Feather puzzles in his taxonomy as dynamic, interesting, monotonic, and structured.

We have also suggested steps for future work, most important of which is increasing the efficiency of computing the halfway solvability ratio or some approximation thereof for larger numbers of cards.

## Acknowledgments

This work was supported, in part, by the Cross-Disciplinary Science Institute at Gettysburg College (X-SIG) and the Symposium on Educational Advances in Artificial Intelligence.

## References

Browne, C. 2017. Tension in puzzles. *Game & Puzzle Design Journal* 3(2):71–78.

Hoos, H., and Stützle, T. 2004. *Stochastic Local Search: Foundations & Applications*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Keller, M. 2016. FreeCell FAQ. http://solitairelaboratory.com/fcfaq.html. Accessed: 2016-09-30.

Kramer, W. 2000. What makes a game good? http://www.thegamesjournal.com/articles/WhatMakesaGame.shtml. Accessed: 2018-08-13.

Neller, T.; Fisher, A.; Choga, M.; Lalvani, S.; and McCarty, K. 2011. Rook jumping maze design considerations. In van den Herik, H.; Iida, H.; and Plaat, A., eds., *LNCS 6515: Computers and Games 2010*. Springer. 188–198.

Neller, T. W.; Berson, C.; Kharel, J.; and Smolik, R. 2019. Efficient solving of birds of a feather puzzles. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI-19), Honolulu, Hawaii, January 27 - February 1, 2019.* Menlo Park, CA, USA: AAAI Press.

Neller, T. W. 2005. Teaching stochastic local search. In *Proceedings of the 18th International FLAIRS Conference (FLAIRS-2005), Clearwater Beach, Florida, May 15-17, 2005.* Menlo Park, CA, USA: AAAI Press.

Neller, T. W. 2016. AI education: Birds of a feather. *AI Matters* 2(4):7–8. Accessed 2018-08-07 via https://sigai.acm.org/static/aimatters/2-4/AIMatters-2-4-03-Neller.pdf.

tgreenless. 2008. Metal wire puzzle double loop (video). https://www.youtube.com/watch?v=_hGo7Pg5HTs. Accessed: 2018-08-13.