# Solving Peg Solitaire (BrainVita)

Zohaib Hassan Soomro, Syed Ahmed Shah

Software Engineering Department, Mehran University of Engineering and Technology Jamshoro Pakistan.

19SW42@students.muet.edu.pk

19SW44@students.muet.edu.pk

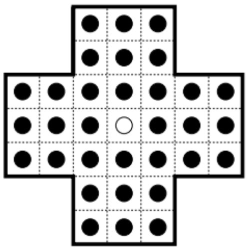
**Abstract:**

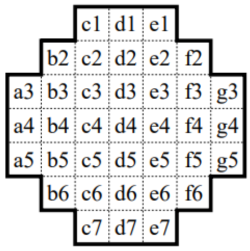
We study the classical game of peg solitaire (BrainVita) where diagonal jumps are allowed. We prove that on many boards, one can begin from a full board with one peg missing, and finish with

one peg anywhere on the board. We then consider the problem of finding solutions that minimize the number of moves (where a move is one or more jumps by the same peg) and find the shortest solution to the “central game”, which begins and ends at the center. In some cases, we can prove analytically that our solutions are the shortest possible, in other cases we apply A* or bidirectional search heuristics.

**Introduction:**

Peg solitaire (BrainVita) is a puzzle that has been popular for over 300 years; it is most played on the 33-hole or 37-hole boards. We refer to a board location as a hole, because on an actual board there is a hole or depression in which the peg (or marble) sits. The game begins with pegs in all the holes except one). The player jumps one peg over another into an empty hole, removing the jumped peg from the board. The goal is to select a sequence of jumps that finish with one peg. In the standard version of the game, only horizontal and vertical jumps are allowed (along rows and columns), and there can be at most four different jumps into any hole. In this paper, we’ll explore the version of the game where diagonal jumps in both directions are also allowed—there can then be up to eight jumps into a hole and this will be called 8-move solitaire. An intermediate version, in which diagonal jumps are allowed in only one direction, is called 6-move solitaire, and is equivalent to solitaire played on a triangular grid.



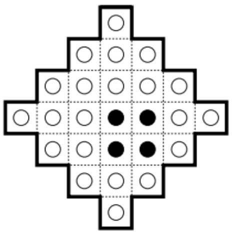



In this paper, we will consider 8-move solitaire on a square symmetric board: the “standard” 33-hole board. The central game is the problem which begins with a full board with one peg missing at the centre and finishes with one peg in the centre.

To identify the holes on the board, we use a notation where the rows are labelled, top to bottom, 1,2,3, . . . , and the columns are labelled left to right, a,b,c. . . , as in above figure.

A board position B will be denoted by a capital letter, and by B′ we mean the complement of this board position, where every peg is replaced by a hole, and vice versa.

A solitaire jump is denoted by the starting and ending coordinates for the jump, separated by a dash, i.e., d4-f4 for the rightward jump from the centre. If the same peg makes one or more jumps, we will call this a move. To denote this, we add the intermediate coordinates of the jumps, for example d4-f4-d6-d4 for the triple-jump move from the centre in Figure below



A common type of peg solitaire problem begins with one peg missing and finishes with one peg. such problems are referred to as **single vacancy to single survivor problems**, or SVSS problems.

The solution to any SVSS problem on the standard 33-hole board has exactly 31 jumps, because we begin with 32 pegs and finish with one, and one peg is lost per jump. However, we can also consider the number of moves in a solution, which has a natural interpretation as the number of pegs that must be touched during the solution (not counting those removed from the board). A solution's length will always be measured in moves.

**Related Work:**

Single vacancy to single survivor (SVSS) problems: -

- **Reversibility, categories, and position classes**

  if a sequence of jumps takes one from board position A to B, then these jumps executed in the same direction, but in reverse order will take one from B′ to A ′. This stems from the fact that the basic solitaire jump itself takes the complement of three consecutive board locations. As stated in Winning Ways, "Backwards solitaire is just forward solitaire with the notions "empty" and "full" interchanged." One consequence of this is that any solution to a complement problem, when played in reverse, is a different solution to the same complement problem.

**Proposed Solution**:

- **A* search technique**

A* search, uses an estimate h(B) of the number of moves from the current board position B to the desired goal (usually a one peg finish. To apply an A* search in our search by levels, at each level i, we accept only board positions satisfying the constraint

$$i + h(B) \leq m,$$

where m is the length of the longest solution we will accept. This is not a traditional A* search, in which the node selected for expansion is the one with the smallest value of i+h(B).

To find the shortest solution to a problem starting from board position S, we set m = h(S) and run a search by levels with constraint. If this finishes with no solution found, we set m = h(S) + 1 and repeat the search from the start, and continue this process, increasing m by one each time, until a solution is found. This technique is very similar to "iterative deepening A*", the main difference being that it is based on a breadth-first rather than a depth-first search. One advantage of our technique is that we can find all solutions, rather than just the first one.

If h(B) is admissible, meaning that it never overestimates the actual number of moves remaining, then an A* search is guaranteed to find the shortest solution, if one exists. For this reason, we will always select h(B) that are admissible. One requirement of admissibility is that if T is a target board position, h(T) = 0.

**Discussion:**

The reason a breadth-first rather than a depth-first search is used to find short solutions in peg solitaire is the amount of repetition encountered while searching. For example, if one can play from board position A to B in 8 moves, it is common to have millions of possible move sequences that can take one from A to B. To effectively apply a depth-first search, it is necessary to store board positions encountered previously, and this can fill memory for the largest boards. This memory limitation is also present in a breadth-first search, but we have the option to store only board positions at the current level, rather than all previously seen. A breadth-first search in peg solitaire can also be easily split into smaller pieces to be worked on separately, and we can also eliminate searches over boards that are equivalent by symmetry.

**Performance evaluation:**

- **Corner constrained board**

We call a board location a corner if there is no jump that can capture a peg at this board location. In 4-move solitaire every hole at the edge of the board is a corner. A corner peg is defined as any peg that is in the same category as some corner peg, or equivalently the peg can jump into some corner.

A peg which begins in a corner cannot be removed in its original location but must be first moved to the centre. Notice that no move can remove more than one corner peg, the central peg. Thus, the process of removing each peg that begins in a corner involves at least two moves: moving it to the centre and then jumping over it.

**1**. All corner pegs are in the same category (which implies that no corner move can remove a corner peg).

**2.** There is a limit to the number of corner pegs that can be removed by one move. Strictly speaking, of course, there is always some limit to the number of pegs removed by one move, but this property is reserved for boards for which this limit is small enough to constrain the length of solutions.

A board with the above two properties is called corner constrained, because the length of the shortest solution is limited primarily by the removal of corner pegs.

- **Back to 4-move solitaire**

The A* heuristic finds Bergholt's 18-move solution to the central game in under a minute, after expanding 4.6×10^5 nodes in a standard 33-hole board. This is a reduction by about a factor of 50 over an unconstrained search, for which the total number of reachable board positions is 2.3 × 10^7.

**Conclusion:**

Table below summarizes our result on short solutions to SVSS problem for a standard 33-hole board.

| Board | Holes/Corners | Type | shortest solution in moves to The central game | shortest solution in moves to Any SVSS problem |
|---|---|---|---|---|
| Standard 33-hole | 33/8 | Corner-constrained | 15 | 13 |

- **In a Nutshell**

We have discussed a heuristic technique that can speed up the search for the shortest solution which is a modified A* search using an estimate of the number of remaining moves.

- **Future work**

Given enough computational power, it is possible to find minimum length solutions on all of these boards by exhaustive search. However, in this paper we have tried to avoid a brute-force approach, always looking for clever search heuristics to find solutions quickly. Using an A* search, it can be faster to find minimum length solutions on the same board under 8-move solitaire than under 4-move solitaire. This is somewhat surprising since the number of board positions reachable is about 16 times larger.

**References:**


1. Brainvita (itgurussoftware.com)
2. https://www.researchgate.net/publication/2840090_Depth-first_search_solves_Peg_Solitaire
3. https://www.researchgate.net/search/researcher?q=solving%2Bpeg%2Bsolitaire
4. https://www.researchgate.net/publication/233485086_A_Fresh_Look_at_Peg_Solitaire
5. https://www.semanticscholar.org/