AUSTRALASIAN JOURNAL OF COMBINATORICS
Volume **63(3)** (2015), Pages 321–332

Peg solitaire on trees with diameter four

Robert A. Beeler   Clayton A. Walvoort

Department of Mathematics and Statistics
East Tennessee State University
Johnson City, TN 37614-1700
U.S.A.
beelerr@etsu.edu   cwalvoort@gmail.com

Abstract

In a 2011 paper by Beeler and Hoilman, the traditional game of peg solitaire is generalized to graphs in the combinatorial sense. One of the important open problems was to classify solvable trees. In this paper, we give necessary and sufficient conditions for the solvability for all trees with diameter four. We also give the maximum number of pegs that can be left on such a graph under the restriction that we jump whenever possible.

1 Introduction

Peg solitaire is a table game which traditionally begins with "pegs" in every space except for one which is left empty (in other words, a "hole"). If in some row or column two adjacent pegs are next to a hole (as in Figure 1), then the peg in x can jump over the peg in y into the hole in z. In [2], peg solitaire is generalized to graphs.

A graph, $G = (V, E)$, is a set of vertices, V, and a set of edges, E. If there are pegs in vertices x and y and a hole in z, then we allow x to jump over y into z, provided that $xy, yz \in E$. Such a jump will be denoted $x \cdot \overrightarrow{y} \cdot z$. For all undefined graph theory terminology, refer to Chartrand [8].

A graph G is *solvable* if there exists some vertex s so that, starting with a hole in s, there exists an associated terminal state consisting of a single peg. A graph G is *freely solvable* if for all vertices s so that, starting with a hole in s, there exists an

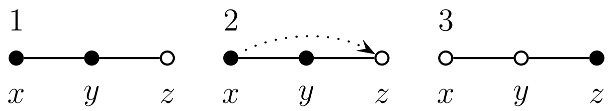

Figure 1: A Typical Jump in Peg Solitaire, $x \cdot \overrightarrow{y} \cdot z$

associated terminal state consisting of a single peg. A graph G is *k-solvable* if there exists some vertex s so that, starting with a hole in s, there exists an associated minimum terminal state consisting of k nonadjacent pegs [2]. For more information on traditional peg solitaire, refer to [1, 7]. For a variation of peg solitaire that allows an additional move, refer to [9].

In [2, 3], the solvability of several families of graphs was determined. One of the more important open problems in [2] was to classify the solvability of trees. In this paper, we consider trees of a fixed diameter, in other words, the maximum distance between any two vertices. We note that the only tree of diameter one is the path on two vertices, which is trivially freely solvable. The trees of diameter two are precisely the stars with n arms, denoted $K_{1,n}$. These graphs are $(n-1)$-solvable [2]. The trees of diameter three are precisely the double stars. A *double star* consists of two adjacent vertices x and y_1. The vertex x (y_1) is adjacent to c (a_1) pendant vertices, where $c \geq a_1 \geq 1$. The double star with parameters c and a_1 is denoted $K_{1,1}(c; a_1)$. The solvability of double stars is given below.

Proposition 1.1 *[3] The double star $K_{1,1}(c; a_1)$ is freely solvable if and only if $c = a_1$ and $a_1 \neq 1$; $K_{1,1}(c; a_1)$ is solvable if and only if $c \leq a_1 + 1$; $K_{1,1}(c; a_1)$ is $(c - a_1)$-solvable in all other cases.*

We are motivated by the above comments to determine the solvability of all trees of diameter four. Any tree of diameter four can be obtained by appending pendant vertices to the existing vertices of $K_{1,n}$. Label the center of the star as x and its arms as $y_1, \ldots, y_n$. Suppose that we append c pendant vertices to x, namely $x_1, \ldots, x_c$ and a_i pendant vertices to y_i, namely $y_{i,1}, \ldots, y_{i,a_i}$ for $i = 1, \ldots, n$. Note that for $i \neq j$ and for any ℓ and m, the vertices $y_{i,\ell}$, y_i, x, y_j, and $y_{j,m}$ induce a path of length four. Thus, this construction gives all trees of diameter four. The resulting graph will be denoted $K_{1,n}(c; a_1, \ldots, a_n)$. An example is shown in Figure 2.

For convenience of notation, we will denote the sets of vertices $X = \{x_1, \ldots, x_c\}$ and $Y_i = \{y_{i,1}, \ldots, y_{i,a_i}\}$ for $i = 1, \ldots, n$. The set of support vertices $\{y_1, \ldots, y_n\}$ will be denoted N. The combination of a support vertex and a single pendant will be called a *leg*. Without loss of generality, assume that $a_1 \geq \cdots \geq a_n \geq 1$. This ensures that each tree of diameter four has a unique parametrization under this notation.

To aid in our constructions, it is useful to define the *dual* of a configuration. The *dual* of a peg configuration T, denoted T', is the state resulting from reversing the roles of pegs and holes [5]. The relationship between the dual of a configuration and whether it is a valid terminal state is given below.

Proposition 1.2 *[2, 5] Suppose that S is a starting state of G with associated terminal state T. Let S' and T' be the duals of S and T, respectively. It follows that T' is a starting state of G with associated terminal state S'.*

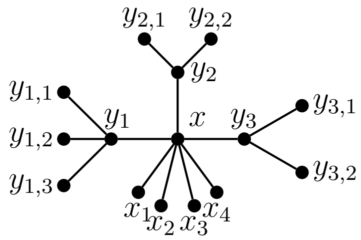

Figure 2: The graph $K_{1,3}(4; 3, 2, 2)$

2 Packages and purges

In an effort to streamline our main result, terminology from [7] will be introduced. A *package* is a collection of vertices which satisfy a specific configuration of pegs and holes such that a sequence of jumps will preserve the locations of certain pegs and holes and remove the remaining pegs. When a package is used to remove pegs, it is called a *purge*. The pegs and holes which are restored to their original locations are called the *catalyst*. We will define five useful packages along with the associated purges. Additional packages and purges on graphs were discussed in [6].

The *wishbone* package consists of $K_{1,2}(1; 1, 1)$ with the hole in x. The wishbone purge is $y_{2,1}\cdot\overrightarrow{y_2}\cdot x$, $x_1\cdot\overrightarrow{x}\cdot y_2$, $y_{1,1}\cdot\overrightarrow{y_1}\cdot x$, and $y_2\cdot\overrightarrow{x}\cdot x_1$. The wishbone purge removes two legs, while the catalyst is x (hole) and x_1 (peg).

The *trident* package consists of $K_{1,3}(1; 1, 1, 1)$ with the hole in x and pegs elsewhere. The trident purge is $y_{3,1}\cdot\overrightarrow{y_3}\cdot x$, $x_1\cdot\overrightarrow{x}\cdot y_3$, $y_{2,1}\cdot\overrightarrow{y_2}\cdot x$, $y_3\cdot\overrightarrow{x}\cdot y_2$, $y_{1,1}\cdot\overrightarrow{y_1}\cdot x$, and $y_2\cdot\overrightarrow{x}\cdot x_1$. This purge removes three legs, while the catalyst is x (hole) and x_1 (peg).

The *spider(N)* package consists of a $K_{1,3}(2; 1, 1, 1)$ with the hole in y_1 and pegs elsewhere. The spider(N) purge is $x_1\cdot\overrightarrow{x}\cdot y_1$, $y_{1,1}\cdot\overrightarrow{y_1}\cdot x$, $x_2\cdot\overrightarrow{x}\cdot y_1$, $y_{2,1}\cdot\overrightarrow{y_2}\cdot x$, $x\cdot\overrightarrow{y_1}\cdot y_{1,1}$, and $y_{3,1}\cdot\overrightarrow{y_3}\cdot x$. Note that the spider($N$) purge removes two pegs from X and two legs, while y_1 (hole), x (peg) and $y_{1,1}$ (peg) are the catalyst.

The *spider(x)* package has the hole in x. The associated purge is $y_{1,1}\cdot\overrightarrow{y_1}\cdot x$, $x_1\cdot\overrightarrow{x}\cdot y_1$, $y_{2,1}\cdot\overrightarrow{y_2}\cdot x$, $x\cdot\overrightarrow{y_1}\cdot y_{1,1}$, $y_{3,1}\cdot\overrightarrow{y_3}\cdot x$, and $x_2\cdot\overrightarrow{x}\cdot y_1$. The spider($x$) purge removes two pegs from X and two legs, while x (hole), y_1 (peg), and $y_{1,1}$ (peg) are the catalyst.

The *double star* package will consist of a $K_{1,1}(d; d)$ with a hole in x. The associated purge to remove d pendants from each side of the double star is accomplished with the moves $y_{1,i}\cdot\overrightarrow{y_1}\cdot x$ and $x_i\cdot\overrightarrow{x}\cdot y_1$ for $i = 1, \ldots, d$ [3]. This purge is denoted $\mathcal{DS}(Y_1, X, d)$. The catalyst is x (hole) and y_1 (peg).

Illustrations of the graphs used for the wishbone, trident, and spider purges are given in Figure 3. For each graph, the catalyst is placed in a box. The hollow vertex in the box represents the initial hole for the purges.

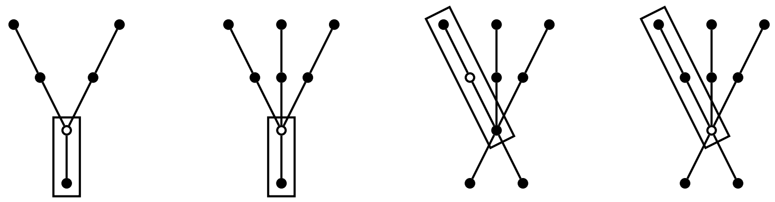

Figure 3: The graphs for the wishbone, trident, and spider(N), and spider(x) purges

3 The solvability of trees of diameter four

With the previous section in mind, we now proceed with our main result. Namely, we provide necessary and sufficient conditions for the solvability of all trees of diameter four. The strategy for solving $K_{1,n}(c; a_1, \dots, a_n)$ will be to begin by performing double star purges. Therefore, we introduce a new parameter, $k = c - s + n$, where $s = \sum_{i=1}^{n} a_i$. This gives the number of pegs remaining in X after $a_i - 1$ pegs have been removed from Y_i for $i = 1, \dots, n$ using double star purges. We begin with the case where $a_1 \geq 2$.

Theorem 3.1 *The conditions for solvability of $K_{1,n}(c; a_1, \dots, a_n)$ where $a_1 \geq 2$ are as follows:*

(i) The graph $K_{1,n}(c; a_1, \dots, a_n)$ is solvable if and only if $0 \leq k \leq n+1$.

(ii) The graph $K_{1,n}(c; a_1, \dots, a_n)$ is freely solvable if and only if $1 \leq k \leq n$.

(iii) The graph $K_{1,n}(c; a_1, \dots, a_n)$ is $(1-k)$-solvable if $k \leq -1$. The graph $K_{1,n}(c; a_1, \dots, a_n)$ is $(k-n)$-solvable if $k \geq n+2$.

Proof. We begin by showing that $k \geq 0$ is necessary for solvability. If $k < 0$, then $c - s + n < 0$. It is necessary to remove all pegs in Y_i for $i = 1, \dots, n$. To remove a peg from a fixed (but arbitrary) Y_i, there must first be a peg in y_i. The only moves that accomplish this are $x_p \cdot \overrightarrow{x} \cdot y_i$ and $y_j \cdot \overrightarrow{x} \cdot y_i$ for $1 \leq p \leq c$ and $j \neq i$. Therefore, the double star purges $\mathcal{DS}(Y_i, X, d)$ and $\mathcal{DS}(Y_i, N, d)$ are necessary to remove pegs in Y_i. Notice this is analogous to $K_{1,1}(s; n+c)$, but $s \geq n + c + 1$. As shown in [3], in order to optimally solve $K_{1,1}(s; n+c)$, the initial hole must be in y_i. However, all of the elements of N are in the other side of the double star. Thus this is not solvable. Since the double star purges are necessary and the initial hole is in N, at least $s - n - c$ pegs will remain in $\cup_{i=1}^{n} Y_i$ after the double star purges [2]. Also, there was a peg in x at the beginning of the game and there will be a peg in x when the last move of the final double star purge is made. Therefore, this peg must be added as well, meaning there will be at least $s - n - c + 1 = 1 - k$, pegs remaining when $k < 0$. Thus, when $k < 0$ the graph is at best $(1-k)$-solvable. This establishes the lower bound on the necessary condition in *(i)* and the first part of *(iii)*.

Now, we show that $k \leq n+1$ is necessary for solvability. If $k \geq n+2$, then $s \leq c-2$. Similar to above, $\mathcal{DS}(X, Y_i, d)$ is the only way to remove pegs from X. Thus, we leave at least $c - s = k - n$ pegs when $k \geq n+2$. Hence the graph is at best $(k-n)$-solvable. This establishes the upper bound on the necessary condition in *(i)* and the second part of *(iii)*.

To show sufficiency for *(i)*, we give an algorithm to solve $K_{1,n}(c; a_1, \ldots, a_n)$ for $0 \leq k \leq n+1$. Begin with the hole in x and perform $\mathcal{DS}(Y_{n-i+1}, X, a_{n-i+1} - 1)$ for $i = 1, \ldots, n$. Without loss of generality, the last peg in Y_i is in $y_{i,1}$. We now have k pegs remaining in X.

If $k = 0$ and $n = 2$, then begin by performing one less double star purge such that the "extra pegs" are in $y_{1,2}$ and x_c. Now move $y_{2,1} \cdot \overrightarrow{y_2} \cdot x$, $x_c \cdot \overrightarrow{x} \cdot y_2$, $y_{1,2} \cdot \overrightarrow{y_1} \cdot x$, $y_2 \cdot \overrightarrow{x} \cdot y_1$, and $y_{1,1} \cdot \overrightarrow{y_1} \cdot x$ to solve with the final peg in x. If $k = 0$ and $n \geq 3$, then eliminate the remaining legs by using a combination of wishbone and trident purges with x and y_1 as the catalyst. Finally, move $y_{1,1} \cdot \overrightarrow{y_1} \cdot x$ to solve. It will later be shown that the case where $k = 0$ is not freely solvable.

If $k = 1$ and $n = 2$ or $n = 3$, then the double star purges will remove pegs from all pendants except for x_1, $y_{1,1}$, $y_{2,1}$, and (in the case where $n = 3$) $y_{3,1}$. Ignoring all pendants with the exception of $y_{1,2}$ and those listed, the resulting subgraph is $K_{1,2}(1; 2, 1)$ or $K_{1,3}(1; 2, 1, 1)$, respectively. In either case, the configuration has holes in x and $y_{1,2}$ and pegs elsewhere. If $n \geq 4$, then we can use wishbone and trident purges to remove additional pendants and their supports. Ignoring the purged pendants and supports, the resulting graph is isomorphic to $K_{1,2}(1; 2, 1)$ with holes in x and $y_{1,2}$. It can be checked using [4] that both configurations can be solved with the final peg located in any vertex. If our target vertex is in one of the vertices from $K_{1,2}(1; 2, 1)$ or $K_{1,3}(1; 2, 1, 1)$, then we can also solve the graph with a hole in that vertex. Otherwise, we simply relabel the target vertex to match one of the vertices of $K_{1,2}(1; 2, 1)$ or $K_{1,3}(1; 2, 1, 1)$. For example, if we wanted our final peg in $y_{i,j}$ for $i \neq 1, 2$, then we instead label $y_{i,j}$ as $y_{2,1}$ and proceed with the above method. Ergo, $K_{1,n}(c; a_1, \ldots, a_n)$ is freely solvable when $k = 1$ by Proposition 1.2.

If $2 \leq k \leq n-1$ and k is odd, then perform the double star purges as above. Next, perform the spider(x) purge $\frac{k-1}{2}$ times using x, y_1, and $y_{1,1}$ as the catalyst. The graph has been reduced to the case of $k = 1$ with holes in x and $y_{1,2}$. As shown above, the final peg can now be located anywhere. Hence, the graph is freely solvable by Proposition 1.2. If k is even, then begin the game with the initial hole in y_1, and make the move $x_c \cdot \overrightarrow{x} \cdot y_1$. Ignoring x_c reduces this to the case when k is odd, with the initial hole in x. Thus $K_{1,n}(c; a_1, \ldots, a_n)$ is freely solvable when $2 \leq k \leq n-1$.

If $k = n$, then instead begin with the initial hole in y_1. Move $x_c \cdot \overrightarrow{x} \cdot y_1$ and ignore x_c. This reduces to the case of $k = n-1$ with the initial hole in x, which can have the final peg in any location. Thus $K_{1,n}(c; a_1, \ldots, a_n)$ is freely solvable when $k = n$. Note that this means that we have established the sufficient conditions for *(ii)* as well.

If $k = n+1$, then instead begin with the initial hole in y_1 and move $x_c \cdot \overrightarrow{x} \cdot y_1$. Ignoring the hole in x_c, this reduces this to the case when $k = n$ and the initial hole is in x, which we know to be solvable. Next, we will show that the case where $k = n + 1$ is not freely solvable.

Throughout the proof, it has been shown that the conditions given for $K_{1,n}(c; a_1, \ldots, a_n)$ to be freely solvable are sufficient. We now show that the conditions given in *(ii)* are necessary. For $K_{1,n}(c; a_1, \ldots, a_n)$ and $k = 0$, consider $K_{1,2}(1; 2, 1)$, which is not solvable if the initial hole is in X or N. If $k = n + 1$, consider $K_{1,2}(4; 2, 1)$, which is not solvable if the initial hole is in x or in any Y_i. These can be verified using an exhaustive computer search [4]. We will now show that any $K_{1,n}(c; a_1, \ldots, a_n)$ where $a_1 \geq 2$ and $k = 0$ or $k = n + 1$ reduces to $K_{1,2}(1; 2, 1)$ or $K_{1,2}(4; 2, 1)$, respectively.

Note that when a double star purge is performed, we can ignore the empty pendants. This results in a reduced graph that has the same value of k as the original graph. So, we can append one "extra" vertex to X and one "extra" vertex to one Y_i and k will not change. Also, $\mathcal{DS}(Y_i, X, 1)$ will remove the recently added pegs. Ignoring these now empty vertices will result in the original $K_{1,n}(c; a_1, \ldots, a_n)$. The double star purges are necessary, as shown earlier. Thus as many as desired of these "extra" vertices can be added in pairs and the new tree will reduce to the original. Similarly, we can append the set of vertices $\{x_{c+1}, y_{n+1}, y_{n+1,1}, y_{n+1,2}\}$ without changing the value of k. Further, $y_{n+1,1} \cdot \overrightarrow{y_{n+1}} \cdot x$, $x_c \cdot \overrightarrow{x} \cdot y_{n+1}$, $y_{n+1,2} \cdot \overrightarrow{y_{n+1}} \cdot x$, and $x_{c+1} \cdot \overrightarrow{x} \cdot x_c$ will remove the newly added vertices. This sequence of moves is analogous to a double star purge, which we have argued is necessary. By using combinations of these two "addition" methods, any diameter four tree with $k = 0$ or $k = n + 1$ can be constructed from $K_{1,2}(1; 2, 1)$ and $K_{1,2}(4; 2, 1)$, respectively. Therefore, all such trees must reduce (via purges) to either $K_{1,2}(1; 2, 1)$ or $K_{1,2}(4; 2, 1)$, both of which are not freely solvable. Hence, the conditions given in *(ii)* are necessary for a graph to be freely solvable.

If the above algorithm is used on a diameter four tree with $k \geq n + 2$, then the remaining $k - n$ pegs will be in X. In particular, if $k = n + 2$, then this results in two pegs that are distance 2 apart. This gives the second part of the sufficient condition in *(iii)*.

We now show that the first part of *(iii)* is sufficient. In this case, $k \leq -1$ and a different technique is required. Again, we begin with the hole in x and use the double star purges as above until there are no pegs remaining in X. Now there are $s - c$ pegs left in the Y_i and n pegs remaining in the support vertices. We remove the remaining support vertices using a combination of wishbone and trident purges with x and y_1 as the catalyst. This removes an additional $2n - 2$ pegs. After the final move of $y_{1,1} \cdot \overrightarrow{y_1} \cdot x$, there are $s - c + n - 1 - (2n - 2) = 1 - k$ pegs remaining. Further, we have $-k$ pegs in Y_i and one peg in x. In particular, if $k = -1$, then we have two pegs that are distance 2 apart. ■

We now deal with the case when all of the $a_i = 1$. We note that in this case, $k = c \geq 0$. For this reason, we give our conditions in terms of c.

Theorem 3.2 *The conditions for solvability of $K_{1,n}(c; 1, \ldots, 1)$ are as follows:*

(i) The graph $K_{1,2t}(c; 1, \ldots, 1)$ is solvable if and only if $0 \leq c \leq 2t$ and $(t, c) \neq (1, 0)$. The graph $K_{1,2t+1}(c; 1, \ldots, 1)$ is solvable if and only if $0 \leq c \leq 2t + 2$.

(ii) The graph $K_{1,n}(c;1,\ldots,1)$ *is freely solvable if and only if* $1 \le c \le n-1$.

(iii) The graph $K_{1,2t}(c;1,\ldots,1)$ *is* $(c-2t+1)$*-solvable if* $c \ge 2t+1$*. The graph* $K_{1,2t+1}(c;1,\ldots,1)$ *is* $(c-2t-1)$*-solvable if* $c \ge 2t+3$.

Proof. Using the argument from Theorem 3.1, $K_{1,n}(c;1,\ldots,1)$ is at best $(c-n)$-solvable when $c \ge n+2$. If $n=2$ and $c=0$, then the graph is the path on five vertices, which is not solvable [2]. This establishes part of the necessary conditions in *(i)*. Additional necessary conditions will be discussed later.

If $c=0$ and $n \ge 3$, then begin with the initial hole in x. Remove $n-1$ legs using a combination of wishbone and trident purges, as in Theorem 3.1. Finally, use $y_{1,1}\cdot\overrightarrow{y_1}\cdot x$ to solve the graph.

If $1 \le c \le n-1$ and c is odd, then start with the initial hole in x. Perform $\frac{c-1}{2}$ spider(x) purges, then use wishbone and trident purges to reduce the graph to $K_{1,2}(1;1,1)$ or $K_{1,3}(1;1,1,1)$ with the hole in x. It can be checked using [4] that $K_{1,2}(1;1,1)$ and $K_{1,3}(1;1,1,1)$ with the hole in x can have the final peg located anywhere except x. Thus, $K_{1,n}(c;1,\ldots,1)$ is freely solvable when $1 \le c \le n-1$ and c is odd by Proposition 1.2.

If $1 \le c \le n-1$ and c is even, then start with the initial hole in y_1 and make the move $x_c\cdot\overrightarrow{x}\cdot y_1$. Ignoring x_c reduces this to the case when c is odd with the initial hole in x. We have shown this case can have the final peg anywhere except x. If the initial hole is in x, then use spider(x) purges to reduce to the case where $c=0$. Therefore, $K_{1,n}(c;1,\ldots,1)$ is freely solvable when $1 \le c \le n-1$ and c is even. This paragraph and the preceding one establishes the sufficient condition in *(ii)* and part of the sufficient conditions for *(i)*.

If $c=n$, then let the initial hole be in y_1 and use the spider(N) purge $\lceil \frac{c}{2} \rceil - 1$ times. Ignoring the vertices cleared by the spider(N) purges reduces this to $K_{1,1}(1;1)$ or $K_{1,2}(2;1,1)$, depending on whether n is odd or even, respectively. Note that both $K_{1,1}(1;1)$ and $K_{1,2}(2;1,1)$ have the hole in y_1. For $K_{1,1}(1;1)$, move $x_1\cdot\overrightarrow{x}\cdot y_1$ and $y_{1,1}\cdot\overrightarrow{y_1}\cdot x$ to solve. For $K_{1,2}(2;1,1)$, move $y_2\cdot\overrightarrow{x}\cdot y_1$, $y_{1,1}\cdot\overrightarrow{y_1}\cdot x$, $x_2\cdot\overrightarrow{x}\cdot y_2$, $y_{2,1}\cdot\overrightarrow{y_2}\cdot x$, and $x_1\cdot\overrightarrow{x}\cdot y_1$ to solve.

If $c=n+1$ and n is odd, then start with the initial hole in y_1 and perform the spider(N) purge $\frac{n-1}{2}$ times. This reduces to $K_{1,1}(2;1)$ with the hole in y_1, which is a solvable double star with the final peg in y_1. However, when n is even, spider(N) purges will reduce the graph to $K_{1,2}(3;1,1)$, which is not solvable. We will show that $K_{1,2t}(2t+1;1,\ldots,1)$ will reduce to $K_{1,2}(3;1,1)$ because spider(N) purges are necessary. Begin with the initial hole in y_1. If we make the initial jump $y_2\cdot\overrightarrow{x}\cdot y_1$, then $y_{1,1}\cdot\overrightarrow{y_1}\cdot x$ is forced. This is essentially $K_{1,2t-1}(2t+1;1,\ldots,1)$ with a hole in y_1, which is unsolvable. Thus, up to automorphism, the first two moves are $x_1\cdot\overrightarrow{x}\cdot y_1$ followed by $y_{1,1}\cdot\overrightarrow{y_1}\cdot x$. As before, the move $y_2\cdot\overrightarrow{x}\cdot y_1$ will lead to an unsolvable graph. Therefore, $x_2\cdot\overrightarrow{x}\cdot y_1$ and $y_{2,1}\cdot\overrightarrow{y_2}\cdot x$ are forced. Again, $y_1\cdot\overrightarrow{x}\cdot y_2$ will lead to an unsolvable graph. Thus $x\cdot\overrightarrow{y_1}\cdot y_{1,1}$ and $y_{3,1}\cdot\overrightarrow{y_3}\cdot x$ are forced. This concludes the exact moves of the spider(N) purge. Since the spider purges are necessary, if the initial hole is not in y_1, then the graph will reduce to a case that is not solvable by a similar argument. This completes the proof of *(i)*.

It has been shown that the conditions provided for $K_{1,2t}(c;1,\ldots,1)$ and $K_{1,2t+1}(c;1,\ldots,1)$ to be freely solvable are sufficient. To complete the proof of *(ii)* we show that these conditions are necessary. For $K_{1,n}(c;1,\ldots,1)$ with $c=0$, assume the initial hole is in y_1. Up to automorphism, the moves $y_2\cdot\overrightarrow{x}\cdot y_1$ and $y_{1,1}\cdot\overrightarrow{y_1}\cdot x$ are forced, which clears the first leg. Therefore, the legs must be removed one at a time until the path on five vertices remains, which is not solvable [2].

If $n=2t$ and $c=n$, then we have shown that spider(N) or spider(x) purges are necessary. Hence, up to automorphism, the initial hole must be in y_1, x, or X to be solvable. If $n=2t+1$ and $c=n+1$, then spider(N) purges are necessary. Therefore, up to automorphism on the vertices, the initial hole must be in y_1 or X.

Consider $K_{1,2t+1}(c;1,\ldots,1)$, where $c\geq 2t+3$. If the hole is in y_1, then t spider(N) purges reduce the graph to $K_{1,1}(c-2t;1)$ with the hole in y_1. After the moves $x_{c-2t}\cdot\overrightarrow{x}\cdot y_1$, $y_{1,1}\cdot\overrightarrow{y_1}\cdot x$, and $x_{c-2t-1}\cdot\overrightarrow{x}\cdot y_1$, there are $c-2t-2$ pegs in X and 1 peg in y_1. In particular, if $c=2t+3$, then the remaining two pegs are distance 2 apart. This shows that second part of *(iii)* is sufficient.

Similarly, for $K_{1,2t}(c;1,\ldots,1)$, where $c\geq 2t+1$, the $t-1$ spider(N) purges will reduce the graph to $K_{1,2}(c-2t+2;1,1)$ with a hole in y_1. After making the jumps $x_{c-2t+2}\cdot\overrightarrow{x}\cdot y_1$, $y_{2,1}\cdot\overrightarrow{y_2}\cdot x$, $x_{c-2t+1}\cdot\overrightarrow{x}\cdot y_2$, $y_{1,1}\cdot\overrightarrow{y_1}\cdot x$, and $y_2\cdot\overrightarrow{x}\cdot x_{c-2t+1}$, there are $c-2t+1$ pegs in X. In particular, if $c=2t+1$, then the final two pegs are distance 2 apart. This shows that first part of *(iii)* is sufficient. ■

These results imply a more general theorem. This theorem was confirmed for all trees with twelve vertices or less using an exhaustive computer search [4].

Theorem 3.3 *Let T be a tree with maximum degree $\Delta(T)$, $n(T)$ vertices, and c pendants adjacent to the vertex of maximum degree. If $\Delta(T)\geq n(T)-c+1$, then T is not solvable.*

Proof. As shown in [2, 3], the above bound holds for trees of diameter three or less. Hence, we may assume that T is a tree with a diameter of at least four. Choose x to be a vertex of maximum degree. Denote the set of pendants adjacent to x as X such that $|X|=c$. In order for T to be solvable, every peg must be removed from X. Recall that for trees of diameter four, a peg in X can only be removed using a peg in one of the Y_i, which is distance three away from X. For a tree of larger diameter, the Y_i position may be filled by another peg that is greater than distance three away from X. Doing so removes additional pegs that are also greater than distance three away from X and eventually reduces to a diameter four tree. Therefore, the upper bound for solvability of diameter four trees, $k\leq n+1$ can be used to determine when T is not solvable. Note that this is equivalent to $c\leq s+1$.

In a tree of diameter four, there are $s+n$ vertices remaining when x and X are excluded. In the general case, there are $n(T)-c-1$ vertices remaining. Since T is being treated as a diameter four tree, $s+n = n(T) - c - 1$. The upper bound for diameter four trees can be now be manipulated to become $s+1 = n(T) - c - n$. Since $n = \Delta(T) - c$, $s+1 = n(T) - \Delta(T)$. Hence, if $c > s+1$, then $c > n(T) - \Delta(T)$. Therefore, $\Delta(T) \geq n(T) - c + 1$ implies that T is not solvable. ∎

Notice that there are solvable trees with $\Delta(T) = n(T) - c$. An infinite class of examples are double stars of the form $K_{1,1}(c+1; c)$. Therefore, the above bound is sharp.

4 Fool's solitaire

Fool's solitaire is a variation of peg solitaire where the goal is to have the maximum number of pegs possible remaining at the end of the game under the caveat that the player jumps whenever possible. The *fool's solitaire number* of a graph G, denoted $Fs(G)$, is the cardinality of the largest terminal state T that is associated with a starting state consisting of a single hole. A terminal state T is a *fool's solitaire solution* if the cardinality of T is equal to $Fs(G)$ [5]. If G is a connected graph, then a sharp upper bound for the fool's solitaire number is $Fs(G) \leq \alpha(G)$, where $\alpha(G)$ denotes the independence number of G. The fool's solitaire number for stars is $Fs(K_{1,n}) = n$ [5] and double stars is $Fs(K_{1,1}(c; a_1)) = c + a_1$ [3]. Additional results about fool's solitaire were given in [10].

In [5], it is conjectured that if G is a connected graph, then $Fs(G) \geq \alpha(G) - 1$. However, it can be checked using [4] that $K_{1,3}(0; 2, 2, 2)$ violates this conjecture, because $Fs(K_{1,3}(0; 2, 2, 2)) = 5 = \alpha(K_{1,3}(0; 2, 2, 2)) - 2$. This example is far from unique. In fact, the diameter four trees provide an infinite class of counterexamples to the above conjecture. For this reason, we are motivated to find $Fs(K_{1,n}(c; a_1, \ldots, a_n))$.

We begin with some observations about the maximum independent set A for $G = K_{1,n}(c; a_1, \ldots, a_n)$. First note that A will contain each of the Y_i where $a_i \geq 2$. If $c = 0$, then $x \in A$. If $c = 1$, then either x or x_1 will be in A. In this case, we choose that $x_1 \in A$ for the purpose of the fool's solitaire problem. If $c \geq 2$, then $X \subset A$. If $a_i = 1$ and $c = 0$, then $y_{i,1} \in A$, but $y_i \notin A$. However, if $a_i = 1$ and $c \geq 1$, then we have a choice whether to include $y_{i,1}$ or y_i into A. In any case, $\alpha(G) = s + c + 1$ when $c = 0$ and $\alpha(G) = s + c$ when $c \geq 1$. These cases will be instrumental in proving the following theorem.

Theorem 4.1 *Consider the diameter four tree $G = K_{1,n}(c; a_1, \ldots, a_n)$, where $a_i \geq 2$ for $1 \leq i \leq n - \ell$, $a_i = 1$ for $n - \ell + 1 \leq i \leq n$, and $n \geq 2$.*

(i) If $c = 0$ and $\ell = 0$, then $Fs(G) = s + c - \left\lfloor \frac{n}{3} \right\rfloor$.

(ii) If $c \geq 1$ and $\ell = 0$, then $Fs(G) = s + c - \left\lfloor \frac{n+1}{3} \right\rfloor$.

(iii) If $\ell \geq 1$, then $Fs(G) = s + c - \left\lfloor \frac{n-2m+1}{3} \right\rfloor$, where $m = \min\{\ell, \left\lfloor \frac{n}{2} \right\rfloor\}$.

Proof. *(i)* First, consider the case where $c = 0$ and $\ell = 0$. As noted above, the maximum independent set is $T = Y_1 \cup \cdots \cup Y_n \cup \{x\}$. The dual of this configuration is $T' = \{y_1, \ldots, y_n\}$. This has $n \geq 2$ pegs, none of which are adjacent. Hence, we can not obtain the upper bound of $\alpha(G) = s + c + 1$ pegs. Thus some pegs must be removed from the maximum independent set to obtain the fool's solitaire solution. Equivalently, some pegs must be added to the dual of the maximum independent set in order to obtain a solvable configuration. We will determine the minimum number of pegs that need to be added to the dual.

Up to automorphism, there are two places where we can add a peg to the dual, namely to x or to one of the Y_i. If we add a peg to x, then we can remove one peg from N with the move $x \cdot \overrightarrow{y_1} \cdot y_{1,1}$. Hence this will not solve the dual. Adding an additional peg to Y_2 will remove an additional three pegs from N using the moves $y_{2,1} \cdot \overrightarrow{y_2} \cdot x$, $y_3 \cdot \overrightarrow{x} \cdot y_1$, $y_{1,1} \cdot \overrightarrow{y_1} \cdot x$, and $x \cdot \overrightarrow{y_4} \cdot y_{4,1}$. However, if a peg is added to Y_1 rather than x, then we can remove two pegs using the moves $y_{1,1} \cdot \overrightarrow{y_1} \cdot x$ and $x \cdot \overrightarrow{y_2} \cdot y_{2,1}$. Adding an additional peg to Y_3 will remove an additional three pegs using a similar sequence as above. Thus, it is more efficient to add a peg to one of the Y_i than it is to add a peg to x. Similarly, adding two or more pegs to a single Y_i is not advantageous, as this would deny us to ability to jump its corresponding support vertex. Further, considering an alternate independent set that includes y_i would result in a dual that includes the vertices in Y_i. Hence, this is not advantageous for the same reasons as above.

Thus, if $n = 3t + r$, where $t, r \in \mathbb{Z}$ and $0 \leq r \leq 2$, then we must add at least $t + 1$ pegs to the dual. Equivalently, the fool's solitaire number is at most $s + c - t$, where $t = \left\lfloor \frac{n}{3} \right\rfloor$. To show equality, it is sufficient to provide the dual of the fool's solitaire solution and the sequence of moves that will reduce this to a single peg. We claim that $T' = \{y_1, \ldots, y_n, y_{1,1}, y_{3i,1} : i \leq t\}$. Begin with the moves $y_{1,1} \cdot \overrightarrow{y_1} \cdot x$ and $x \cdot \overrightarrow{y_2} \cdot y_{2,1}$. For $i = 1, \ldots, t-1$, make the sequence of jumps $y_{3i,1} \cdot \overrightarrow{y_{3i}} \cdot x$, $y_{3i+1} \cdot \overrightarrow{x} \cdot y_{3i-1}$, $y_{3i-1,1} \cdot \overrightarrow{y_{3i-1}} \cdot x$, and $x \cdot \overrightarrow{y_{3i+2}} \cdot y_{3i+2,1}$. If $r = 0$, then we replace $x \cdot \overrightarrow{y_{3t-1}} \cdot y_{3t-1,1}$ with $y_{3t-1} \cdot \overrightarrow{x} \cdot y_{3t-2}$ and make the additional jumps $y_{3t,1} \cdot \overrightarrow{y_{3t}} \cdot x$ and $y_{3t-2} \cdot \overrightarrow{x} \cdot y_{3t}$. If $r = 1$, then we make the additional jumps $y_{3t,1} \cdot \overrightarrow{y_{3t}} \cdot x$, $y_{3t+1} \cdot \overrightarrow{x} \cdot y_{3t-1}$, and $y_{3t-1,1} \cdot \overrightarrow{y_{3t-1}} \cdot x$. If $r = 2$, then we make the additional jumps $y_{3t,1} \cdot \overrightarrow{y_{3t}} \cdot x$, $y_{3t+1} \cdot \overrightarrow{x} \cdot y_{3t-1}$, $y_{3t-1,1} \cdot \overrightarrow{y_{3t-1}} \cdot x$, and $x \cdot \overrightarrow{y_{3t+2}} \cdot y_{3t+2,1}$. Thus $Fs(G) = s + c - t$, where $t = \left\lfloor \frac{n}{3} \right\rfloor$ and $c = 0$.

(ii) By a similar argument, if $c \geq 1$ and $\ell = 0$, then $T' = \{y_1, \ldots, y_n, x, y_{3i-1,1} : i \leq t\}$. If $n \equiv 2 \pmod 3$, then we also include $y_{n,1}$. In any case, we begin by making the jump $x \cdot \overrightarrow{y_1} \cdot y_{1,1}$. The current configuration of pegs is the same as in the previous case after the first two moves had been made. It follows that $Fs(K_{1,n}(c; a_1, \ldots, a_n)) = s + c - \left\lfloor \frac{n+1}{3} \right\rfloor$, where $c \geq 1$.

(iii) We now consider the case where $\ell \geq 1$. If $c = 0$, then the maximum independent set is $Y_1 \cup \cdots \cup Y_n \cup \{x\}$. As before, the dual of this configuration has no adjacent pegs. Hence, it is necessary to add at least one peg to the dual. We claim that adding x to the dual is the best choice. The reason is that we want to replace y_i with $y_{i,1}$ in the dual for all $i \geq m = \min\{\ell, \lfloor \frac{n}{2} \rfloor\}$. If x is in the fool's solitaire solution, then this is not possible because the fool's solitaire solution must be an

independent set. Further, this will allow us to "exchange" pegs in Y_i with pegs in N, where $i \geq n-m$. Since x will be in the dual of our fool's solitaire solution, the method described here will also work when $c \geq 1$. Thus in both cases we claim that $T' = \{y_1, \ldots, y_{n-m}, y_{n-m+1,1}, \ldots, y_{n,1}, x, y_{m+3i-1,1} : i = 1, \ldots, \lfloor \frac{n-2m}{3} \rfloor\}$. If $n-2m > 0$ and $n-2m \equiv 2 \pmod 3$, then we also include $y_{n-m,1}$ in the dual. We remove pegs from the dual using the moves $y_i \cdot \overrightarrow{x} \cdot y_{n-m+i}$ and $y_{n-m+i,1} \cdot \overrightarrow{y_{n-m+i}} \cdot x$ for $i = 1, \ldots, m$. Note that this is the same as the initial configuration in the case where $c \geq 1$ and $\ell = 0$. Thus, $Fs(G) = s + c - \lfloor \frac{n-2m+1}{3} \rfloor$. ■

We note that for trees of diameter four, the difference between $\alpha(G)$ and $Fs(G)$ can be arbitrarily large. However, $Fs(G) > 5\alpha(G)/6$ for all such trees. This is obtained by considering the ratio $Fs(G)/\alpha(G)$, where $G = K_{1,n}(0; 2, \ldots, 2)$ and n is sufficiently large. To what extent this bound holds for all connected graphs is unknown.

5 Open Problems

We end our discussion by giving several open problems.
(i) What are the conditions for the solvability of trees with diameter five?
(ii) Can Theorem 3.3 be generalized to describe all graphs?
(iii) Which graphs have $Fs(G) < \alpha(G) - 1$?
(iv) Is there a nontrivial lower bound for $Fs(G)$?

Acknowledgements

The authors wish to thank the anonymous referees for their many helpful and constructive comments.

References

[1] J. D. Beasley, *The ins and outs of peg solitaire*, vol. 2 of *Recreations in Mathematics*, Oxford University Press, Eynsham, 1985.

[2] R. A. Beeler and D. P. Hoilman, Peg solitaire on graphs, *Discrete Math.* 311 (20) (2011), 2198–2202.

[3] R. A. Beeler and D. P. Hoilman, Peg solitaire on the windmill and the double star, *Australas. J. Combin.* 52 (2012), 127–134.

[4] R. A. Beeler and H. Norwood, Peg solitaire on graphs solver applet, http://faculty.etsu.edu/BEELERR/solitaire/.

[5] R. A. Beeler and T. K. Rodriguez, Fool's solitaire on graphs, *Involve* 5 (4) (2012), 473–480.

[6] R. A. Beeler and C. A. Walvoort, Packages and purges for peg solitaire on graphs, *Congr. Numer.* 218 (2013), 33–42.

[7] E. R. Berlekamp, J. H. Conway, and R. K. Guy, *Winning ways for your mathematical plays*, Vol. 2, A.K. Peters Ltd., Natick, MA, second ed., 2003.

[8] G. Chartrand, L. Lesniak and P. Zhang, *Graphs & digraphs*, CRC Press, Boca Raton, FL, fifth ed., 2011.

[9] J. Engbers and C. Stocker, Reversible peg solitaire on graphs, *Discrete Math.* 338 (11) (2015), 2014–2019.

[10] S. Loeb and J. Wise, Fool's solitaire on joins and Cartesian products of graphs, *Discrete Math.* 338 (3) (2015), 66–71.

(Received 15 Mar 2014; revised 14 Dec 2014, 25 Sep 2015)