Theory and Applications of Graphs

Volume 9 | Issue 2 Article 12

October 2022

Path-Stick Solitaire on Graphs

Jan-Hendrik de Wiljes
Freie Universität Berlin, jan.dewiljes@math.fu-berlin.de

Martin Kreh
University of Hildesheim, kreh@imai.uni-hildesheim.de

Recommended Citation

de Wiljes, Jan-Hendrik and Kreh, Martin (2022) "Path-Stick Solitaire on Graphs," *Theory and Applications of Graphs*: Vol. 9: Iss. 2, Article 12.
DOI: 10.20429/tag.2022.090212
Available at: https://digitalcommons.georgiasouthern.edu/tag/vol9/iss2/12

Abstract

In 2011, Beeler and Hoilman generalised the game of peg solitaire to arbitrary connected graphs. Since then, peg solitaire and related games have been considered on many graph classes. In this paper, we introduce a variant of the game peg solitaire, called path-stick solitaire, which is played with sticks in edges instead of pegs in vertices. We prove several analogues to peg solitaire results for that game, mainly regarding different graph classes. Furthermore, we characterise, with very few exceptions, path-stick-solvable joins of graphs and provide some possible future research questions.

1 Introduction

In [4], Beeler and Hoilman introduced the game of peg solitaire on graphs as a generalisation of the classical peg solitaire game:

Given a connected, undirected graph G with vertex set $V(G)$ and edge set $E(G)$, pegs can be put in the vertices of G. Given three vertices u, v, w with pegs in u and v and a hole in w such that $uv, vw \in E(G)$, we can jump with the peg from u over v into w, removing the peg in v (cf. Figure 1). This jump will be denoted as $u \cdot \overrightarrow{v} \cdot w$.

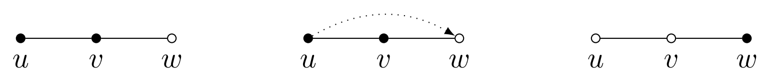

Figure 1: A jump in peg solitaire.

If, starting with a single hole in one of the graphs vertices, by some series of jumps all pegs but one can be removed, then the graph is called (peg) solvable. Characterising solvable graphs is one of the main goals in current research on peg solitaire. To tackle this problem, the game has been considered for several classes of graphs, including path graphs, complete graphs, star graphs, double stars, and caterpillars (for more results and variants see [3–9, 11, 12, 14, 15]).

In this paper, we define a variant of this game as follows: Given a graph G, we put *sticks* in the edges of G. Let $e, f, g \in E(G)$ be three edges such that e is incident to f and f is incident to g, but no vertex of G is incident to all three of these edges. If sticks are in e and f and no stick (i.e., a *hole*) is in g, then we can jump with the stick from e over f into g, removing the stick in f. This jump will be denoted as $e \cdot \overrightarrow{f} \cdot g$ (cf. Figure 2). Note that the edges e, f and g form (in this order) a path or cycle of length 3, hence the name path-stick solitaire.

A variant of this game, called stick solitaire, is introduced in [13] to investigate peg solitaire on line graphs. The *line graph* $L(G)$ of a graph G has vertex set $E(G)$ and two edges of G are adjacent in $L(G)$ if and only if they are incident in G (cf. Figure 3).

In stick solitaire, additional to the jump possibilities described above, the jump $e \cdot \overrightarrow{f} \cdot g$ is allowed whenever e, f, g are edges incident to the same vertex. In particular, any move that is allowed in path-stick solitaire is also allowed in stick solitaire. Conversely, any graph that can be solved in stick solitaire without using the additional type of jump is also solvable in path-stick solitaire. This is in particular the case, whenever there are no vertices of degree at least 3. Note, moreover, that playing stick solitaire on G is the same as playing peg solitaire

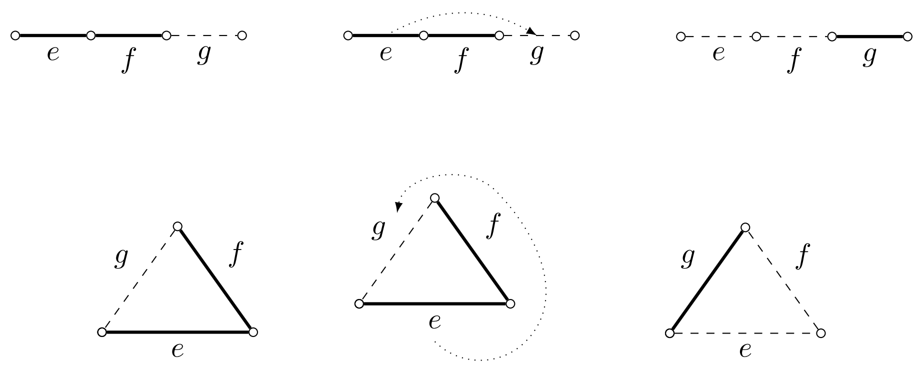

Figure 2: Possible jumps in path-stick solitaire.

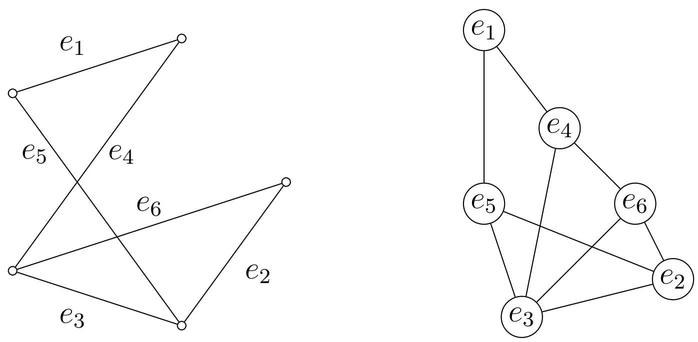

Figure 3: A graph G (left) and its line graph $L(G)$ (right).

on $L(G)$. In contrast to that, peg solitaire results cannot be applied to path-stick solitaire in general in the same way. For instance, the edges e_2, e_3 and e_5 of the graph G displayed in Figure 3 cannot be used for the same jump in path-stick solitaire, but the corresponding nodes in $L(G)$ form a triangle and can therefore certainly be part of the same jump in peg solitaire.

The following notation is equivalent to the respective notation for peg solitaire. In general, we begin with a *starting state* $S \subset E$ of edges that are empty, i.e., without sticks; all other edges contain a stick. A *terminal state* $T \subset E$ is a set of edges that contain sticks at the end of the game such that no more jumps are possible; all other edges are empty. A terminal state T is *associated* to a starting state S if T can be obtained from S by a series of jumps. We will always assume that the starting state S consists of a single edge.

As mentioned above, the goal of the original peg solitaire game is to remove all pegs but one. Analogously, we want to achieve a terminal state consisting of a single edge. Since this is not possible for all graphs, we introduce the following (analogue) notations and notions. A graph G is called

- *path-stick-solvable*, in short *ps-solvable*, if there is some $e \in E$ such that the starting state $S = \{e\}$ has an associated terminal state consisting of a single edge.

- *freely ps-solvable* if, for all $e \in E$, the starting state $S = \{e\}$ has an associated terminal state consisting of a single edge.

- *k-ps-solvable* if there is some $e \in V$ such that the starting state $S = \{e\}$ has an associated terminal state consisting of k edges.
- *strictly k-ps-solvable* if G is k-ps-solvable but not ℓ-ps-solvable for any $\ell < k$. In that case G has *path-stick solitaire number* $\mathrm{t}(G) = k$.

Since this paper serves as an introduction to path-stick solitaire, we begin by determining the path-stick solitaire number of (at least in the context of peg solitaire) well-known graph classes in Section 2. As the construction of ps-solvable graphs from known examples is an important step to classifying all ps-solvable graphs, we consider joins and Cartesian products of graphs in Section 3. We almost completely characterise ps-solvable joins, which is our main result.

We use the notations P_n, C_n, and K_n for the *path graph*, the *cycle graph*, and the *complete graph* on n vertices. Furthermore, $K_{m,n}$ denotes the *complete bipartite graph* on $m + n$ vertices. Two additional (not as common as the ones before) graph classes are considered in this paper. For $R, L \geq 1$, let $DS(L, R)$ denote the *double star* with $L + R$ pendant vertices, i.e., the graph which is obtained by connecting the centres of $K_{1,R}$ and $K_{1,L}$ by an (additional) edge. A *windmill variant* $W(P, B)$ is a graph with a universal vertex u (that is adjacent to every other vertex), P pendant vertices, that are only adjacent to u, and B blades consisting of two vertices each, such that these two vertices are adjacent (and vertices in a blade are not adjacent to vertices in another blade). A closed trail containing all edges of a graph G is an *Eulerian cycle* of G. The *union* $G \cup H$ of two graphs G and H is understood as their disjoint union in this paper, i.e., it has vertex set $V(G) \cup V(H)$, where $V(G) \cap V(H) = \emptyset$, and edge set $E(G) \cup E(H)$, where $E(G) \cap E(H) = \emptyset$. As a special case, nG denotes the union of n disjoint copies of a graph G. Two other binary graph operations will also be considered in this paper. For graphs G and H, we denote the Cartesian product of G and H by $G \,\square\, H$ and use the (common) notation $(g, h) \in V(G \,\square\, H)$ for the vertex induced by $g \in V(G)$ and $h \in V(H)$. The join $G \vee H$ of two graphs G and H is $G \cup H$ together with additional edges connecting every pair of vertices g, h with $g \in V(G)$ and $h \in V(H)$. Furthermore, we use the common abbreviations $[i, j] = \{i, i + 1, \ldots, j\}$ and $[j] = [1, j]$ for integers $i < j$.

2 Graph classes

First of all, if no vertices of degree larger than 2 exist, then we may apply results from peg solitaire (cf. [4]).

Proposition 2.1. *Let n be a positive integer.*

1. *For $n \geq 4$, we have $\mathrm{t}(P_n) = 1$ if $n = 4$ or $2 \nmid n$ and $\mathrm{t}(P_n) = 2$ otherwise. In the second case, it is possible to achieve a terminal state $\{v_1v_2, v_3v_4\}$ with $v_2v_3 \in E(P_n)$ and certain $v_i \in V(P_n)$.*
2. *For $n \geq 3$, we have $\mathrm{t}(C_n) = 1$ if $n = 3$ or $2 \mid n$ and $\mathrm{t}(C_n) = 2$ otherwise. In the second case, for any $v_1, v_2, v_3, v_4 \in V(C_n)$ with $v_1v_2, v_2v_3, v_3v_4 \in E(C_n)$, it is possible to achieve the terminal state $\{v_1v_2, v_3v_4\}$.*
3. *If G contains an even Eulerian cycle, then we have $\mathrm{t}(G) = 1$.*

Although this proposition follows immediately from the previously mentioned connection to line graphs, for illustrative reasons, we explicitly explain how to solve G when it contains an even Eulerian cycle. Let $v_1, v_2, \ldots, v_{2n}$ be an even Eulerian cycle of G. Start with a hole in the edge v_2v_3. Start jumping $v_4v_5 \cdot \overrightarrow{v_3v_4} \cdot v_2v_3, v_1v_2 \cdot \overrightarrow{v_2v_3} \cdot v_3v_4$. Now we can proceed jumping back and forth, ending with a stick in the edge $v_{2n-1}v_{2n}$.

Proposition 2.2. *For every integer $n \geq 3$, we have* $\mathrm{t}(K_n) = 1$.

Proof. We distinguish several cases. If $n \equiv 1 \bmod 4$, then K_n contains an even Eulerian cycle and the statement follows from Proposition 2.1. If $n \equiv 3 \bmod 4$, then we can use Fleury's algorithm to obtain an Eulerian cycle starting with $v_1v_2v_3v_1$, where $v_i \in V(K_n)$. This can be solved using Proposition 2.1 such that the final sticks are in v_1v_2 and v_3v_1. Using the jump $v_2v_1 \cdot \overrightarrow{v_1v_3} \cdot v_3v_2$ solves this case. If n is even, then let $V(K_n) = \{v_1, v_2, \ldots, v_n\}$. The graph $K_n \setminus \{v_1v_2, v_3v_4, \ldots, v_{n-1}v_n\}$ contains an even Eulerian cycle, hence it is ps-solvable by Proposition 2.1. Without loss of generality let $\{v_nv_1\}$ be the terminal state. Now we can solve K_n in the same way using the additional jumps $v_nv_1 \cdot \overrightarrow{v_1v_2} \cdot v_2v_3, v_2v_3 \cdot \overrightarrow{v_3v_4} \cdot v_4v_5, \ldots, v_{n-2}v_{n-1} \cdot \overrightarrow{v_{n-1}v_n} \cdot v_nv_1$ in the end. □

Stars and double stars pose a huge problem for path-stick solitaire as can be seen in the following (easily provable) result.

Proposition 2.3. *Let n, L, and R be positive integers.*

- *We have* $\mathrm{t}(K_{1,n}) = n - 1$.
- *We have* $\mathrm{t}(DS(L, R)) = L + R - 1$.

As soon as cycles (or longer paths) exist, it is possible to remove more sticks. In particular, for $K_{m,n}$ and the windmill variant we can prove results similar to the corresponding statements in peg solitaire.

Proposition 2.4. *Let m and n be non-negative integers with $m + n \geq 3$. Then we have* $\mathrm{t}(K_{m,n}) = 1$ *if and only if* $(m, n) \in \{(1, 2), (2, 1)\}$ *or* $m, n \geq 2$.

Proof. If $m = 1$ or $n = 1$, then the statement follows from Proposition 2.3. Hence, we assume $m, n \geq 2$ from now on. If both parameters are even, then $K_{m,n}$ is ps-solvable because of Part 3 in Proposition 2.1. If $2 \nmid m$ or $2 \nmid n$, then we reduce the situation to the even case. Denote by $w_1, w_2, \ldots, w_n$ the vertices in the independent set of size n. Start with a hole in an edge uw_1, choose some non-neighbour $v \neq u$ of u, and jump $w_nv \cdot \overrightarrow{vw_1} \cdot w_1u$. For $i = n-1, n-2, \ldots, 2$, execute the jump $uw_i \cdot \overrightarrow{w_iv} \cdot vw_{i+1}$. After $w_nu \cdot \overrightarrow{uw_1} \cdot w_1v$, we have eliminated all sticks from edges incident to u and are left with sticks everywhere else except for vw_2. This procedure will be completed (at most twice) until we only have to solve a subgraph $K_{m',n'}$ with even m', n'. □

The following proposition is an analogue to the result in [1, Corollary 2.2] on the peg unsolvability of graphs with large maximum degree and large number of leaves. This might be especially useful when investigating trees.

Proposition 2.5. *Let G be a graph with $|V(G)| \geq 4$ and let $v \in V(G)$ be a vertex that is adjacent to at least $\frac{1}{2}|E(G)|$ leaves. Then G is not ps-solvable.*

Proof. Denote by $e_1, e_2, \ldots, e_p$ the p edges connecting v with leaves, let $f_1, f_2, \ldots, f_m$ be the other edges incident to v, and let $g_1, g_2, \ldots, g_n$ be the edges not incident to v. Then we have $p \geq m + n$. To remove the sticks from the edges e_i, the only possible move is $e_i \cdot \overrightarrow{f_j} \cdot g_k$ for some j, k. It is best possible to start with a hole in some f_j or g_k, since otherwise the first move will increase the number of sticks in the edges e_i. For each stick in an edge e_i, we need a stick in an edge f_j. To get a stick in f_j, there are two possibilities: We can either jump $g_k \cdot \overrightarrow{g_\ell} \cdot f_j$ or $g_k \cdot \overrightarrow{f_\ell} \cdot f_j$. Only the first of these two jumps increases the number of sticks in the edges f_j. Since the first move requires two sticks in edges not adjacent to v and each jump that removes a stick from an edge e_i increases the number of sticks in the edges g_k by 1, the maximum number of sticks that can be removed from the edges e_i is $m + 2n - n - 1 = m + n - 1 < p$. Hence G is not ps-solvable. □

Equivalence does not hold in Proposition 2.5. There are even infinitely many graphs being unsolvable in path-stick solitaire without having the mentioned property.

Proposition 2.6. *Let P and B be non-negative integers with $P + 2B \geq 3$. Then $\mathrm{t}(W(P, B)) = 1$ if and only if $P \leq 2B$.*

Proof. Let us first consider $P \leq 2B$. Denote by $e_1, e_2, \ldots, e_P$ the edges incident to some pendant vertex and by $f_1, f_2, \ldots, f_B$ the edges not incident to the central vertex u of $W(P, B)$. Finally, let a_i, b_i be the edges incident to f_i. We start with a hole in f_1 and distinguish the following cases.

1. If $P \in \{0, 1\}$, then we execute the jumps $a_i \cdot \overrightarrow{a_1} \cdot f_1, b_1 \cdot \overrightarrow{f_1} \cdot a_1, f_i \cdot \overrightarrow{b_i} \cdot b_1$ for $i \in [2, B]$. We solve the graph via $a_1 \cdot \overrightarrow{b_1} \cdot f_1$, if $P = 0$, or via $e_1 \cdot \overrightarrow{b_1} \cdot f_1, a_1 \cdot \overrightarrow{f_1} \cdot b_1$, if $P = 1$.

2. If $P \geq 2$, then we jump

$$e_{2i-1} \cdot \overrightarrow{a_1} \cdot f_1, f_{i+1} \cdot \overrightarrow{b_{i+1}} \cdot a_1, e_{2i} \cdot \overrightarrow{a_{i+1}} \cdot f_{i+1}, f_1 \cdot \overrightarrow{a_1} \cdot a_{i+1}, f_{i+1} \cdot \overrightarrow{a_{i+1}} \cdot a_1 \tag{1}$$

   for $i \in \left[\left\lfloor \frac{P}{2} \right\rfloor - 1\right]$. In each step, we removed sticks from two e_j and all edges in one blade. Hence, this procedure yields a situation with only sticks (and a hole in f_1) in a subgraph isomorphic to $W(P', B')$ with $P' \in \{2, 3\}$ and $B' = B - \left\lfloor \frac{P}{2} \right\rfloor + 1$.

   If $P \leq 2B - 1$, which implies $B' \geq 2$, then we execute the same series of jumps as in (1) for $i = \left\lfloor \frac{P}{2} \right\rfloor$ and solve the graph as in the first case.

   If $P = 2B$, then the jumps $e_{P-1} \cdot \overrightarrow{b_1} \cdot f_1, f_1 \cdot \overrightarrow{a_1} \cdot b_1, e_P \cdot \overrightarrow{b_1} \cdot f_1$ solve the graph.

We now turn our attention to the necessity and, hence, consider $P > 2B$ (note that Proposition 2.5 implies the statement only for $P \geq 3B$). A stick in some e_i can only be removed by a jump $e_i \cdot \overrightarrow{a_j} \cdot f_j$ or $e_i \cdot \overrightarrow{b_j} \cdot f_j$, which removes a stick from some a_j or b_j as well. Since there is no jump which increases the number of sticks in the edges of blades incident to u, for $W(P, B)$ to be ps-solvable we need $P \leq 2B + 1$ and hence $P = 2B + 1$. Furthermore, we need to start with a hole in some e_i. But since the only two possible jumps in that case are of the form $f_j \cdot \overrightarrow{a_j} \cdot e_i$ or $f_j \cdot \overrightarrow{b_j} \cdot e_i$, the number of sticks in edges incident to leaves is, after the first jump, larger than the number of sticks in the edges of blades incident to u. This configuration is again not ps-solvable, due to the argument above. □

3 Binary graph operations

Some graph operations, such as Cartesian products, joins, and line graphs, have been considered for peg solitaire and its variants (cf. [2, 4, 14, 15]), in particular to construct new solvable graphs. Since we are mainly interested in characterising ps-solvable graphs, it also appears natural to investigate graph operations. The ones mentioned before seem, compared to peg solitaire, particularly problematic for path-stick solitaire as additional sticks appear. Nevertheless, a neat trick gives the following partial characterisation for ps-solvable joins.

Theorem 3.1. *Let G and H be graphs. The join $G \vee H$ is ps-solvable if $(|V(G)|, |V(H)|) \notin \{(x, y) \in \mathbb{N}^2 : (x = 1$ and $y \geq 3)$ or $(x \geq 3$ and $y = 1)\}$.*

Proof. The cases $(|V(G)|, |V(H)|) \in \{(1, 1), (1, 2), (2, 1)\}$ are trivial.

Begin with a hole in some edge $e = gh$, where $g \in V(G), h \in V(H)$. Now empty all edges in G by the following procedure. Let $f = ab$ be an edge in G, i.e., $a, b \in V(G)$, containing a stick. Assume without loss of generality that $a \neq g$. Jump $f \cdot \overrightarrow{ah} \cdot e$. Redefining f as e and iterating this step empties every edge of G. We can proceed analogously with the sticks in H. Lastly, we use Proposition 2.4 to ps-solve the complete bipartite graph induced by the edges between G and H. □

The excluded cases pose more problems. Proposition 2.5 implies the following result.

Corollary 3.2. *If G is a graph with at least $\frac{1}{2}(|E(G)| + |V(G)|)$ isolated vertices, then the join $G \vee K_1$ is not ps-solvable.*

This is not an equivalence as $W(5, 2)$, which is the join of $2K_2 \cup 5K_1$ and K_1, shows. Proposition 2.6 even yields infinitely many such examples. Nevertheless, windmill variants give the worst case examples in the sense, that $G \vee K_1$ is solvable if G has at most $\frac{1}{2}|V(G)|$ isolated vertices, as we will see in the next two theorems.

Theorem 3.3. *Let G be a connected graph. Then $G \vee K_1$ is ps-solvable. In particular, if G is non-trivial, then it is possible to ps-solve $G \vee K_1$ with starting hole in an (arbitrary) edge lying at the end of a longest path of G and a final stick in some (not arbitrary) edge of G.*

Proof. Note that $G \vee K_1$ with $G \cong K_1$ is trivially ps-solvable, hence we exclude this case. We prove the theorem via induction over the length of a longest path in G. Note that the statement is certainly true for $K_2 \vee K_1$ (base case), hence assume G containing a path of length $m \geq 2$ from now on. The main idea is to remove all sticks from edges incident to an end-vertex of a longest path in G. The graph obtained by deleting this vertex from G is connected. To this end, let v_1 be an end-vertex of a longest path $v_1 v_2 \ldots v_m$ in G. Among the end-vertices of longest paths in $G' = G \setminus \{v_1\}$, let w be one of largest degree and let x be its successor on some longest path P of G'. Let further a be the single vertex of K_1. We start with a hole in $v_1 v_2$ and distinguish several cases and subcases depending on the degrees of the involved vertices.

<u>Case 1:</u> $d_G(v_1) \geq 3$. Let $N_G(v_1) \setminus \{v_2\} = \{b_1, b_2, \ldots, b_\ell\}$. If ℓ is even, then we jump $v_1 a \cdot \overrightarrow{a v_2} \cdot v_2 v_1, b_1 v_1 \cdot \overrightarrow{v_1 v_2} \cdot v_2 a$. If ℓ is odd, then we jump $b_1 a \cdot \overrightarrow{a v_2} \cdot v_2 v_1, b_2 v_1 \cdot \overrightarrow{v_1 a} \cdot a b_1, b_1 v_1 \cdot \overrightarrow{v_1 v_2} \cdot v_2 a$. In both cases, we eliminate all sticks from edges incident to v_1 except for the one in $v_1 b_\ell$

using the reduction procedure in Figure 4. After jumping $xw \cdot \overrightarrow{wa} \cdot av_1, b_\ell v_1 \cdot \overrightarrow{v_1 a} \cdot aw$, we proceed in $G' \vee K_1$ as before (use w in place of v_1) until we can use our induction hypothesis. Note that whenever we use the phrase "proceed ... as before" during the proof, we mean starting at the beginning of the proof with renamed objects until the induction hypothesis can be applied.

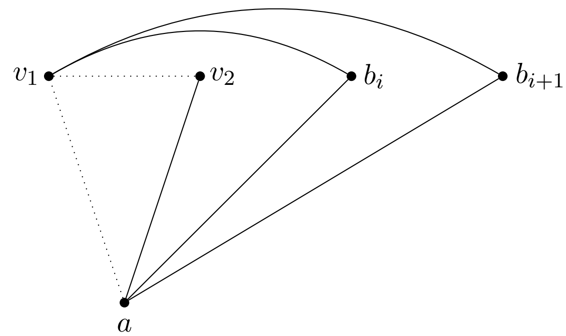

Figure 4: The elimination process for Theorem 3.3, Case 1. Repeatedly (for different i) perform the jumps $v_1 b_{i+1} \cdot \overrightarrow{b_{i+1} a} \cdot av_1, b_i v_1 \cdot \overrightarrow{v_1 a} \cdot ab_{i+1}$.

Case 2: $d_G(v_1) = 1$. Whenever we can choose v_2 as w, we will do that. In that case, after jumping $av_3 \cdot \overrightarrow{v_3 v_2} \cdot v_2 v_1, v_2 v_1 \cdot \overrightarrow{v_1 a} \cdot av_3$, we are done by using the induction hypothesis on G', since a longest path in G' starting with v_2 has length $m-1$. Hence assume now that $w = v_2$ is impossible. If $x = v_2$, then we jump $wa \cdot \overrightarrow{av_1} \cdot v_1 v_2, v_1 v_2 \cdot \overrightarrow{v_2 w} \cdot wa$ and proceed in G' as before. Now consider $x \neq v_2$. We make the following distinction.

Case 2.1: $d_{G'}(w) \geq 2$. Let $y \neq x$ be another neighbour of w in G'. If $d_{G'}(w)$ is odd, then we jump $ya \cdot \overrightarrow{av_1} \cdot v_1 v_2, aw \cdot \overrightarrow{wy} \cdot ya, wx \cdot \overrightarrow{xa} \cdot av_1, v_2 v_1 \cdot \overrightarrow{v_1 a} \cdot ax$. If $d_{G'}(w)$ is even, then we jump $xa \cdot \overrightarrow{av_2} \cdot v_2 v_1, av_1 \cdot \overrightarrow{v_1 v_2} \cdot v_2 a, aw \cdot \overrightarrow{wx} \cdot xa$. Either way, we can proceed as in Case 1, considering G' instead of G and w instead of v_1.

Case 2.2: $d_{G'}(w) = 1$. If P is not the unique longest path in G', then let y be some end-vertex of a longest path in G' but not of P. By choice of w we have $d_{G'}(y) = 1$, and, by our assumption above, $y \neq v_2$. Let z be the neighbour of y in G'. We jump $v_1 a \cdot \overrightarrow{av_2} \cdot v_2 v_1, zy \cdot \overrightarrow{ya} \cdot av_2, wx \cdot \overrightarrow{xa} \cdot av_1, v_2 v_1 \cdot \overrightarrow{v_1 a} \cdot ax$ and proceed in $(G' \setminus \{y\}) \vee K_1$ as before. If P is unique, then let y be the end-vertex of a longest path in G' starting with w. Now removing w or y might result in a graph containing a longest path with an end-vertex of degree larger than 1 where the hole is in the wrong place. If this is not the case, then we can continue as before (possibly exchanging w and y). Otherwise, let p be an end-vertex of a longest path in $G' \setminus \{w\}$ of largest degree and let q be its successor on a longest path in $G' \setminus \{w\}$ (choosing without loss of generality w here for reasons of simplicity as $w \neq v_2 \neq x$). We jump $v_1 a \cdot \overrightarrow{av_2} \cdot v_2 v_1, xw \cdot \overrightarrow{wa} \cdot av_2, qp \cdot \overrightarrow{pa} \cdot av_1, v_2 v_1 \cdot \overrightarrow{v_1 a} \cdot ap$ and proceed in $(G' \setminus \{w\}) \vee K_1$ as before.

Case 3: $d_G(v_1) = 2$. Let $y \neq v_2$ be the other neighbour of v_1 in G. If $w = v_2$, then we jump $v_3 a \cdot \overrightarrow{av_1} \cdot v_1 v_2, v_3 v_2 \cdot \overrightarrow{v_2 a} \cdot av_3, yv_1 \cdot \overrightarrow{v_1 v_2} \cdot v_2 a$ and are done by using the induction hypothesis on G', since a longest path in G' starting with v_2 has length $m-1$. Now consider $w \neq v_2$. If $d_{G'}(w) \geq 2$, then we denote by $z \neq x$ another neighbour of w in G' and, similarly as in Case 2.1, distinguish the degrees parity possibilities. If $d_{G'}(w)$ is even, then we jump

$v_1a \cdot \overrightarrow{av_2} \cdot v_2v_1, v_1y \cdot \overrightarrow{ya} \cdot av_2, v_1v_2 \cdot \overrightarrow{v_2a} \cdot ay, xw \cdot \overrightarrow{wa} \cdot av_2$. If $d_{G'}(w)$ is odd, then we jump $v_1a \cdot \overrightarrow{av_2} \cdot v_2v_1, yv_1 \cdot \overrightarrow{v_1v_2} \cdot v_2a, wx \cdot \overrightarrow{xa} \cdot av_1, v_1a \cdot \overrightarrow{aw} \cdot wx, zw \cdot \overrightarrow{wx} \cdot xa$. Either way, we can proceed as in Case 1, considering G' instead of G and w instead of v_1. If $d_{G'}(w) = 1$, then let p be an end-vertex of a longest path in $G' \setminus \{w\}$ of largest degree and let q be its successor on a longest path in $G' \setminus \{w\}$. Jump $wa \cdot \overrightarrow{av_1} \cdot v_1v_2, av_2 \cdot \overrightarrow{v_2v_1} \cdot v_1a, yv_1 \cdot \overrightarrow{v_1a} \cdot av_2, qp \cdot \overrightarrow{pa} \cdot aw, xw \cdot \overrightarrow{wa} \cdot ap$ and proceed in $(G' \setminus \{w\}) \vee K_1$ as before. □

Theorem 3.4. *Let G be a graph having at most $\frac{1}{2}|V(G)|$ isolated vertices. Then $G \vee K_1$ is ps-solvable.*

Proof. We assume $|V(G)| \geq 3$ from now on as the other cases are trivial. Let $u_1, u_2, \ldots, u_k$ denote the isolated vertices of G, note $k \leq \frac{1}{2}|V(G)|$, and let $C_1, C_2, \ldots, C_\ell$ be the non-trivial connected components of G. Furthermore, let a denote the vertex of K_1. If $|V(C_i)| = 2$ for all $i \in [\ell]$, then we are done using Proposition 2.6. Hence we find some component of G, say C_1, on at least three vertices. Pick a longest path $v_1v_2 \ldots v_m$ of C_1, note that $m \geq 3$ holds, and start with a hole in v_1v_2. If $k = 0$, then we can use Theorem 3.3 iteratively on $C_i \vee K_1$, clearing a solved component completely and preparing the next one by the jumps indicated in Figure 5. Otherwise, we distinguish two cases.

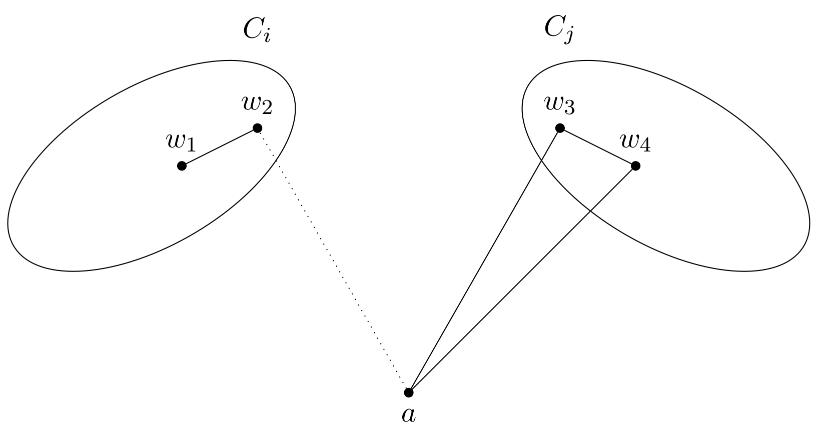

Figure 5: The transfer jump for Theorem 3.4. Perform the jumps $w_4w_3 \cdot \overrightarrow{w_3a} \cdot aw_2, w_1w_2 \cdot \overrightarrow{w_2a} \cdot aw_3$. Note that w_3 is an end-vertex of a longest path in C_j.

Case 1: If v_1 has degree $d_G(v_1) > 1$, then execute $u_1a \cdot \overrightarrow{av_2} \cdot v_2v_1$ and jump from some edge incident to v_1 and some vertex other than a, v_2 over v_1v_2 into v_2a. This removes a stick from the edge u_1a and from some edge in C_1. Note that, since v_1 is at the end of a longest path, C_1 without that edge is still connected. Proceed in this way (using $u_2, u_3, \ldots$) until there is only one stick in an edge incident to v_1, namely in av_1. Note that if we have to stop short because of $d(v_1) > k + 2$, then we are again done using Theorem 3.3. If there are still some sticks in edges incident to some u_i, then we proceed as in Case 2. For reasons of simplicity, we redefine G to be the graph obtained by deleting the vertices $u_1, u_2, \ldots, u_{d(v)-2}$ and the empty edges of C_1 (expect for v_1v_2). After renaming the other vertices u_i (u_i will become $u_{i-d(v)+2}$) and redefining C_1 (previous C_1 without the deleted edges), we consider $G \vee K_1$. Note that (the new) G now has even less than $\frac{1}{2}|V(G)|$ isolated vertices.

Case 2: Consider $d_G(v_1) = 1$ now.

Case 2.1: If v_2 is adjacent to some $v \neq v_i$ for all $i \in [m]$, then we jump $u_1a \cdot \overrightarrow{av_1} \cdot v_1v_2, vv_2 \cdot \overrightarrow{v_2a} \cdot av_1, av_1 \cdot \overrightarrow{v_1v_2} \cdot v_2a$. Since u_1, v_1 are incident only to empty edges, we delete them from G, redefining everything as before. Note that $vv_2v_3 \ldots v_m$ is also a longest path in C_1 and that we have a hole in vv_2. Hence after changing the name of v to v_1 (and also redefining C_1 by deleting the previous v_1), we can proceed either with Case 1 or continue with Case 2.2.

Case 2.2: If v_2 is adjacent only to vertices on the longest path, then we jump $u_1a \cdot \overrightarrow{av_1} \cdot v_1v_2, v_3v_2 \cdot \overrightarrow{v_2a} \cdot av_1, av_1 \cdot \overrightarrow{v_1v_2} \cdot v_2a$. Again, we delete v_1 and u_1 from G and redefine everything. Note that the redefined C_1 keeps being connected and we keep having a hole in the first edge of a longest path of C_1. Furthermore, the new G has at most $\frac{1}{2}|V(G)|$ isolated vertices.

If we proceed in this manner, then we either reach a state where the current graph is ps-solvable by Theorem 3.3 or C_1 is reduced to a graph on two vertices. In the second case, we use, if $k > 1$, the jumps indicated in Figure 6 and proceed with C_2 instead of C_1.

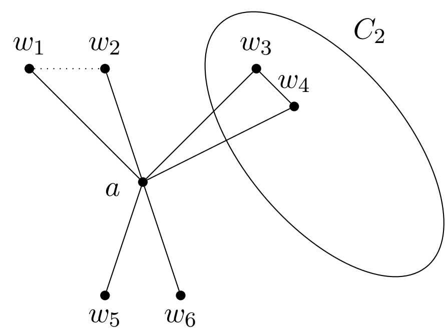

Figure 6: The transfer jump for Theorem 3.4, Case 2.2. Perform the jumps $w_5a \cdot \overrightarrow{aw_1} \cdot w_1w_2, w_1w_2 \cdot \overrightarrow{w_2a} \cdot aw_1, w_4w_3 \cdot \overrightarrow{w_3a} \cdot aw_2, w_6a \cdot \overrightarrow{aw_1} \cdot w_1w_2, w_1w_2 \cdot \overrightarrow{w_2a} \cdot aw_3$.

For $k = 1$, we also have two cases to consider. If some $i \in [2, \ell]$ with $|V(C_i)| \geq 3$ exists, then solve $C_1 \vee K_1$, use the transfer jump in Figure 5 to get to C_i, and proceed as before. The stick from u_1a will be removed in the process and we finish using Theorem 3.3. Otherwise, we have a windmill which is ps-solvable, with the current stick situation, using Proposition 2.6. Since in the above process we have always deleted at least as many non-isolated vertices as isolated vertices, the statement follows. □

Joins on one hand have the disadvantage of "adding" a lot of edges, and hence sticks, but on the other hand this can be helpful since many extra jumping possibilities appear. Cartesian products do not have this advantage and, therefore, appear to be less accessible (and less usable for the construction of ps-solvable graphs). Nevertheless, due to the nice symmetry, we can prove that the Cartesian product of two path graphs is ps-solvable.

Proposition 3.1. *For every positive integer n, the graph $P_n \,\square\, P_2$ is ps-solvable.*

Proof. The cases $n = 1, 2$ follow from Proposition 2.1. Let $u_1, u_2, \ldots, u_n$ be the vertices of P_n, where for every $i \in [2, n-1]$ the vertex u_i is adjacent to u_{i-1} and u_{i+1}. Furthermore, let v_1, v_2 denote the vertices of P_2. Start with a hole in $(u_1, v_1)(u_1, v_2)$ and jump $(u_3, v_1)(u_2, v_1) \cdot \overrightarrow{(u_2, v_1)(u_1, v_1)} \cdot (u_1, v_1)(u_1, v_2)$, $(u_1, v_2)(u_2, v_2) \cdot \overrightarrow{(u_2, v_2)(u_2, v_1)} \cdot (u_2, v_1)(u_1, v_1)$, $(u_1, v_2)(u_1, v_1) \cdot \overrightarrow{(u_1, v_1)(u_2, v_1)} \cdot (u_2, v_1)(u_3, v_1)$. All vertices except (u_1, v_1) and (u_1, v_2) induce a graph isomorphic to $P_{n-1} \square P_2$ with a hole in $(u_2, v_2)(u_2, v_1)$ in the current configuration. By induction, this is ps-solvable. □

A very similar argument, which is visualised in Figure 7, yields the following result.

Theorem 3.5. *Let n and m be integers greater than 1. The graph $P_n \square P_m$ is ps-solvable.*

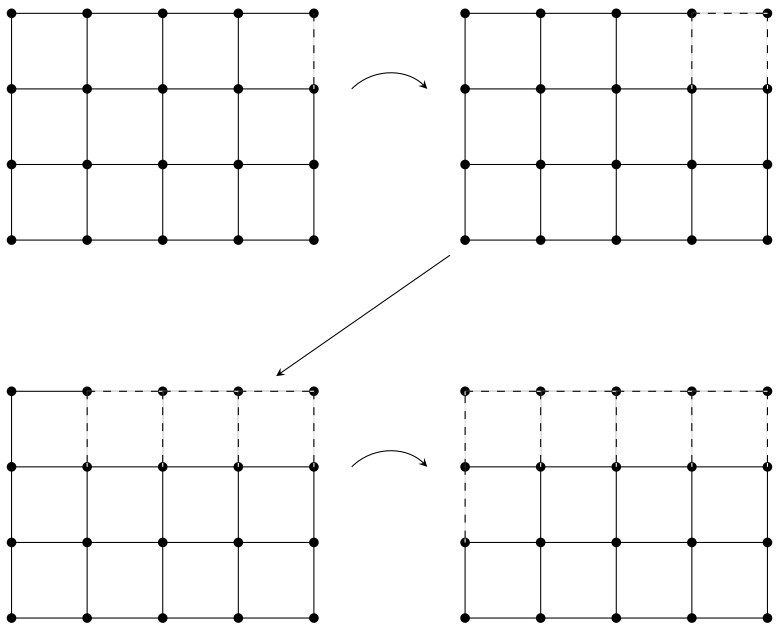

Figure 7: The elimination process for iteratively ps-solving the graph $P_n \square P_m$. Proceed until the remaining graph is $P_n \square P_2$ or $P_2 \square P_m$.

4 Open problems

Certainly many more questions/problems could be considered, such as the determination of $\mathrm{t}(G)$ for other classes of graphs. As for the original game of peg solitaire, it would be nice to have a characterisation of ps-solvable trees (a problem studied for peg solitaire in [3, 7, 11]) or at least for certain classes of trees, e.g., caterpillars. In addition, one might examine the game of Fool's path-stick solitaire.

It would be interesting to know more about the "gap" for ps-solvable joins as indicated by Corollary 3.2 and Theorem 3.4. Also, can more general results on the ps-solvability of Cartesian products be proved? Furthermore, since joins and Cartesian products introduce new sticks, it might be especially fruitful to investigate binary graph operations without that property, such as the identification of certain vertices of the involved graphs.

In [10], the authors define the number $\mathrm{ms}(G)$ to be the least number of edges that have to be added to make G peg solvable. Since complete graphs are ps-solvable, an analogue question arises for path-stick solitaire. But note that adding an edge does not necessarily increase the ps-solvability. Is it instead possible to lower $\mathrm{t}(G)$ by adding vertices in a suitable way? Might subdividing a graph help? For example, unsolvable path graphs can be made solvable by subdividing an edge.

Another approach would be the concept of contracting edges. Thus, one might ask if contracting certain edges always yields (after some iterations) a ps-solvable graph and, if so, how many edges have to be contracted to obtain a ps-solvable graph.

References

[1] R. A. Beeler and A. D. Gray. Extremal results for peg solitaire on graphs. *Bull. Inst. Combin. Appl.*, 77:30–42, 2016.

[2] R. A. Beeler and A. D. Gray. Double jump peg solitaire on graphs. *Bull. Inst. Combin. Appl.*, 91:80–93, 2021.

[3] R. A. Beeler, H. Green, and R. T. Harper. Peg solitaire on caterpillars. *Integers*, 17:Article G1, 2017.

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

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

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

[7] R. A. Beeler and C. A. Walvoort. Peg solitaire on trees with diameter four. *Australas. J. Combin.*, 63:321–332, 2015.

[8] G. I. Bell. Solving triangular peg solitaire. *J. Integer Seq.*, 11(4):Article 08.4.8, 2008.

[9] T. C. Davis, A. De Lamere, G. Sopena, R. C. Soto, S. Vyas, and M. Wong. Peg solitaire in three colors on graphs. *Involve*, 13(5):791–802, 2020.

[10] J.-H. de Wiljes and M. Kreh. Making graphs solvable in peg solitaire. *Electron. J. Graph Theory Appl.*, forthcoming, 2022.

[11] J.-H. de Wiljes and M. Kreh. Peg solitaire on banana trees. *Bull. Inst. Combin. Appl.*, 90:63–86, 2020.

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

[13] M. Kreh and J-H. de Wiljes. Peg solitaire on line graphs. *Preprint*, 2022.

[14] M. Kreh and J-H. de Wiljes. Peg solitaire on Cartesian products of graphs. *Graphs Comb.*, 37(3):907–917, 2021.

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