Morpion Solitaire

Erik D. Demaine
MIT Computer Science and Artificial Intelligence Laboratory
edemaine@mit.edu

Martin L. Demaine
MIT Computer Science and Artificial Intelligence Laboratory
mdemaine@mit.edu

Arthur Langerman
Langerman Diamonds
arthur@langerman.net

Stefan Langerman[*]
Université Libre de Bruxelles, Département d'informatique
stefan.langerman@ulb.ac.be

Abstract

We study a popular pencil-and-paper game called *morpion solitaire*. We present upper and lower bounds for the maximum score attainable for many versions of the game. We also show that, in its most general form, the game is NP-hard and the high score is inapproximable within $n^{1-\varepsilon}$ for any $\varepsilon > 0$ unless $P = NP$.

1 Introduction

The classic game of *morpion solitaire* starts with some configuration of points drawn on the intersections of a square grid, typically the cross shown on the left of Figure 1. In this game, the player makes a sequence of moves. Each move consists of placing a new point at a grid intersection and drawing a new line segment connecting 5 consecutive points that include the new one. The line can be drawn in any of the four directions: horizontal, vertical, or either diagonal. Moves are further constrained by one of two constraints. In the *disjoint model*, line segments with the same direction cannot share a point. However, line segments with different directions are always permitted to share points. In the *touching model*, line segments with the same direction are permitted to overlap

[*]Chercheur qualifié du FNRS.

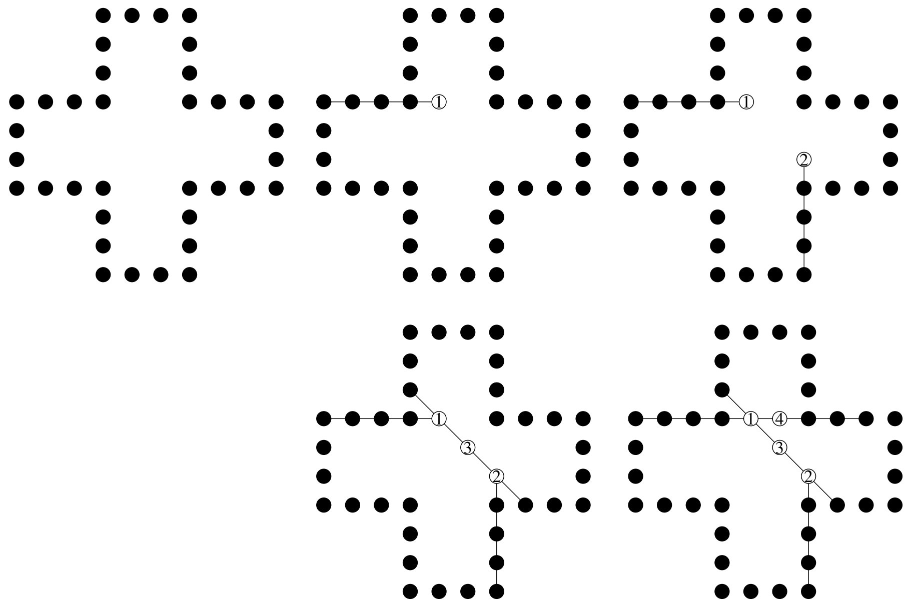

Figure 1: The cross starting configuration A_4 for $k = 4$, and four sample moves. The last move is permitted only in the touching model.

just slightly, at a common endpoint, but cannot share more than one point. In other words, the touching model allows point overlap but disallows positive-length overlap of the line segments. The game is over when no further moves can be made. The goal of the game is to maximize the number of moves before the game ends.

The morpion solitaire game is famous in several European countries (mainly in Belgium and France), where every elementary-school student is required to have graph paper in her/his schoolbag. The game is also commonly called "connector", "petites croix" ("little crosses"), or "Malta cross". The touching model is probably the most popular of the two models. The first published reference we could find about the game is in the magazine *Jeux & Stratégie* from September 1982 [4]. The article shows a solution of 164 moves and claims a record of 170 moves by Charles-Henri Bruneau without actually displaying it. The following two issues of the magazine mention that they have received a large number of proposed solutions, but those solutions have not been published either. Since then, several webpages have been dedicated to finding better solutions to the touching version of the game [1, 8, 9], and games of 170 moves due to Denis Excoffier, Charles-Henri Bruneau, and JB Bonté (bearing the date of January 15, 1982) have been published and verified. The game has also been used as a test case for an evolutionary algorithm by Hugues Juillé [7]. His program found a game of 122 moves.

The disjoint model is the one that appears under the name Connector in the excellent book by Walter Joris [6]. The book describes a two-player variant as well. A webpage maintaining the high scores for the disjoint-model solitaire game has been maintained by the fourth author since 1996 [8]. High scores have alternated between Stefan Schmieta, who used an implementation of a random-sampling algorithm with local search, and the third author, who used exclusively pencil and paper. The current record of 68 moves is held by the third author.

In this article, we consider combinatorial and computational issues for several variations on both the touching and disjoint variants of morpion solitaire. We first generalize the game so that, at every move, the drawn line segment joins $k+1$ points, rather than 5, for some specified value k, and scaling the initial cross configuration accordingly. We also consider the more general case (mentioned in [2]) where the starting configuration can be any given set of points. We present lower and upper bounds for the largest number of moves in all versions of the game, in particular partially characterizing when the number of moves can be infinite.

After the magazine *Jeux & Stratégie* received a large number of solutions, they were faced with the computational problem of verifying them. In [2] they write "It is horribly difficult, or even impossible, to figure the order in which the line segments have been drawn, and thus to verify if the proposed game is valid. Indeed, after the 30th move, or even before, the addition of a new point allows 2, 3, or 4 alignment possibilities: which to choose? The number of possibilities grows as the game continues." (translated from French). In Section 4, we show that reconstructing a valid ordering from a drawing is not as difficult as it seems: we give a linear-time algorithm for this task. We then show that, on the other hand, determining the maximum number of moves that can be made from a given set of points is NP-hard and not approximable within $n^{1-\varepsilon}$ for any $\varepsilon > 0$ (unless $P = NP$).

2 Notation

Let $G_k(S)$ denote the maximum number of moves in a game starting with an initial set $S \subseteq \mathbb{Z}^2$ of points on the unit square grid, where at each step a line joining $k+1$ points is drawn through k existing points and a new one, and where two lines with the same direction cannot share a point (the disjoint model). Let $G'_k(S)$ be the maximum number of moves in the variant where two lines are allowed to share one point but not two (the touching model). Let A_k be the traditional initial set of $|A_k| = 12(k-1)$ points formed by a plus sign of thickness k. For example A_4 is the configuration shown on the left of Figure 1, $|A_4| = 36$, and $G_4(A_4)$ is the number of lines in the best possible solution of the original puzzle. We also write $G_k(n)$ and $G'_k(n)$ for the maximum value of $G_k(S)$ and $G'_k(S)$ respectively, over all sets S of n points, that is, the maximum number of moves possible if we are allowed to choose the position of the n starting points.

3 Combinatorial Results

In this section, we present upper and lower bounds on the values of $G_k(S)$, $G'_k(S)$, $G_k(A_k)$, and $G'_k(A_k)$. Because the touching model is less restrictive than the disjoint model, $G_k(S) \le G'_k(S)$ for any S and any k.

3.1 Potential Function

The following potential-function argument is partially described in [2].

A *drawing* D is any set of grid points and horizontal, vertical, and diagonal line segments, e.g., drawn during gameplay. Every point in the drawing can be seen as having 8 *slots* in the 8 different directions, which line segments may or may not connect to (overlap). We define the *potential* of a point in D to be the number of directions in which no line segment is connected, i.e., the number of empty slots. The potential $\phi(D)$ of the entire drawing D is the sum of the potential of all its points.

In the game $G_k(S)$, the potential at the beginning is $8|S|$. Each move adds a new point, which adds 8 to the potential, and a line which removes $2(k+1)$ from the potential. No further moves can be made when the potential is less than $2k$. This implies that $8|S| - (2(k+1) - 8)(G_k(S) - 1) \ge 2k$, and so when $k > 3$,

$$G_k(S) \le 1 + \frac{4|S| - k}{k - 3}.$$

For $G'_k(S)$, the situation is identical except that adding a line removes only $2k$ from the potential, and no further lines can be added when the potential is less than $2k - 1$. So, for $k > 4$,

$$G'_k(S) \le 1 + \frac{8|S| - 2k + 1}{2(k - 4)}.$$

For example, these potential arguments give the upper bound $G_4(A_4) \le 141$. Unfortunately, these simple arguments do not produce a bound for $G'_4(A_4)$. In fact, in that case, a move keeps the potential unchanged.

3.2 Boundary Bound for $G'_k(S)$

Let D be a drawing at some time in the game $G'_k(S)$, and let P be the set of points in D. Form a new drawing $\Gamma(P)$ by connecting every pair of points in P that are adjacent either horizontally, vertically, or diagonally, i.e., every pair of gridpoints with ℓ_∞ distance 1. Thus, $\Gamma(P)$ forms a superset of the drawing D, and $\phi(\Gamma(P)) \le \phi(D)$. By extension, we define the potential of any set of points Q: $\phi(Q) = \phi(\Gamma(Q))$. We assume that $\Gamma(P)$ is connected; this assumption can be later removed by considering each connected component separately.

Our goal is to bound the numder $e(P)$ of edges in $\Gamma(P)$, which will provide an upper bound on the number of moves in D. Brass [3] bounds the number of edges in unit-distance graphs in general norms $\|\cdot\|$, i.e., the number of edges of length 1 according to this norm, in terms of the number of vertices. His bound depends on $\lambda(\|\cdot\|)$, the length of the longest line segment on a circle of radius 1. In $\Gamma(P)$, we are measuring unit distances in the ℓ_∞ norm, so $\lambda(\|\cdot\|) = 2$. For that case, Brass proves

$$e(P) \leq \left\lfloor 4|P| - \sqrt{28|P| - 12} \right\rfloor .$$

(He also proves that this bound is tight in the worst case, even for our case of ℓ_∞.)

For every point in P, each of its 8 incident edges is counted either in $e(P)$ (if the edge connects to another point in P) or in $\phi(P)$ (otherwise). The former edges are counted only once in $e(P)$ but twice if we consider each point in P separately. Thus $2e(P) + \phi(P) = 8|P|$, so

$$\phi(P)/2 \geq \left\lceil \sqrt{28|P| - 12} \right\rceil .$$

But $\phi(P)$ is even, so we can drop the ceiling:

$$\phi(P)/2 \geq \sqrt{28|P| - 12}.$$

Therefore we obtain the following bound on the size of a point set in terms of its potential:

$$|P| \leq (\phi(P)^2/4 + 12)/28 = (\phi(P)^2 + 48)/112. \tag{1}$$

If m is the number of moves performed, we have $|P| = |S| + m$, and in the game $G'_k(S)$,

$$\phi(P) \leq \phi(D) = 8|S| + m(8 - 2k).$$

Therefore,

$$|S| + m \leq \frac{(8|S| + m(8 - 2k))^2 + 48}{112}.$$

In particular, when $k = 4$, the maximum number of moves is bounded by

$$m \leq \frac{4}{7}|S|^2 - |S|$$

which implies that the original $G'_4(A_4) \leq 704$. This upper bound applies to any starting set of 36 points.

In the remainder of this section, we consider each value of k individually and describe the best upper and lower bounds we know for the two games, in particular $G_k(A_k)$ and $G'_k(A_k)$.

3.3 $k = 1$

Starting with one point, the game $G_1(S)$ can continue indefinitely, as shown in Figure 2. Thus, for any S with $|S| > 0$, $G_1(S) = G'_1(S) = \infty$.

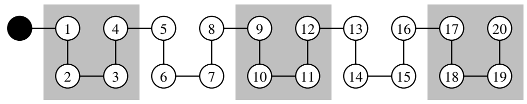

Figure 2: $G_1(1) = G'_1(1) = \infty$.

3.4 $k = 2$

Two starting points allow no more than one move: $G_2(2) = G'_2(2) = 1$. But there is a starting set of three points from which one can play indefinitely; see Figure 3. So, $G_2(3) = G'_2(3) = \infty$ and in particular, $G_2(A_2) = G'_2(A_2) = \infty$.

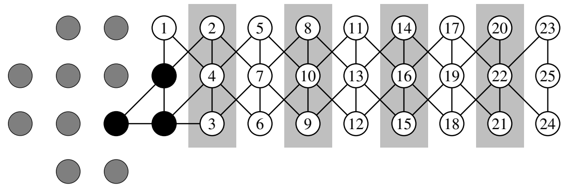

Figure 3: $G_2(A_2) = G'_2(A_2) = G_2(3) = G'_2(3) = \infty$.

3.5 $k = 3$

The case $k = 3$ is the first interesting one. The potential argument does not help because, for $G_3(S)$, the potential remains unchanged after a move. Moreover, there exists a starting set of 7 points from which one can play indefinitely; see Figure 4. So, $G_3(7) = G'_3(7) = \infty$.

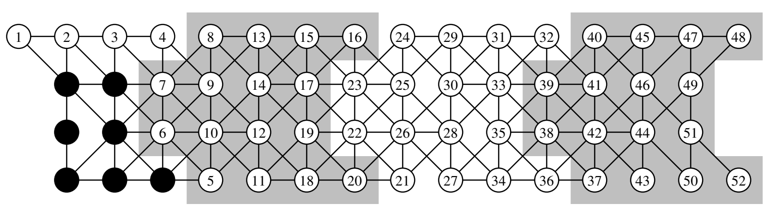

Figure 4: $G_3(7) = G'_3(7) = \infty$.

Nevertheless, we can show that both $G_3(A_3)$ and $G_3(A_3)$ are bounded. Assume that the bottom leftmost point of A_3 has coordinates $(1, 1)$. We claim that every point of A_3 has at least one of its coordinates odd, so no point with both coordinates even can ever be played during the game. To see this claim, just notice that any segment of length 4 incident to one (even,even) point has to be incident to exactly two (even,even) points. The claim reduces the number of slots available at every point: an (odd, odd) point has 4 slots and an $(\text{odd}, \text{even})$ point has 6 slots available. So the starting potential is 120.

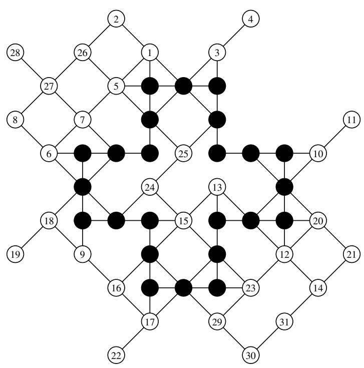

Figure 5: $G_3(A_3) \geq 31$.

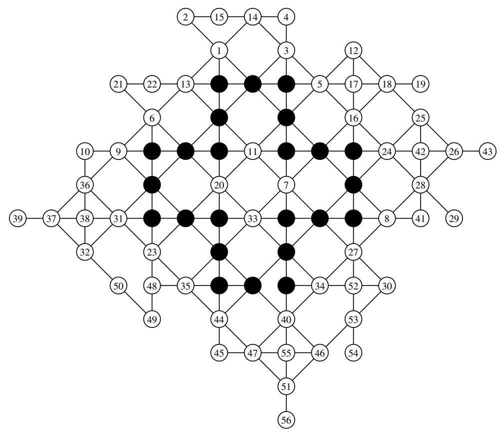

Figure 6: $G_3'(A_3) \geq 56$.

We split the potential into three parts: ϕ_{oo} is the sum of all horizontal and vertical free slots at (odd, odd) points, initially 48; ϕ_{oe} is the sum of all horizontal and vertical free slots at (odd, even) points, initially 24; and ϕ_d is the sum of all diagonal free slots (which never appear at (odd, odd) points), initially 48. Let m_{oo} be the number of moves placing an (odd, odd) point and drawing a horizontal or vertical line, m_{oe} the number of moves placing an (odd, even) point and drawing a horizontal or vertical line, and m_d the number of diagonal moves (which place only (odd, even) points). The potentials can be expressed by the following equations:

$$\begin{aligned}
\phi_{oo} &= 48 - 4m_{oe}, \\
\phi_{oe} &= 24 - 2m_{oe} - 4m_{oo} + 2m_d, \\
\phi_d &= 48 + 4m_{oe} - 4m_d.
\end{aligned}$$

Solving the linear program of maximizing $m_{oo}+m_{oe}+m_d$ subject to nonnegativity constraints $\phi_{oo}, \phi_{oe}, \phi_d, m_{oo}, m_{oe}, m_d \geq 0$, we obtain $m_{oo} = 12, m_{oe} = 12, m_d = 24$, which imply that $G_3(A_3) \leq 48$. Figure 5 shows that $G_3(A_3) \geq 31$.

For the second variant $G_3'(A_3)$, the potentials can be expressed as follows:

$$\begin{aligned}
\phi_{oo} &= 48 - 3m_{oe} + m_{oo}, \\
\phi_{oe} &= 24 - m_{oe} - 3m_{oo} + 2m_d, \\
\phi_d &= 48 + 4m_{oe} - 2m_d.
\end{aligned}$$

Solving the linear program of maximizing $m_{oo}+m_{oe}+m_d$ subject to nonnegativity constraints $\phi_{oo}, \phi_{oe}, \phi_d, m_{oo}, m_{oe}, m_d \geq 0$, we obtain $m_{oo} = 60, m_{oe} = 96, m_d = 36$, which imply that $G_3'(A_3) \leq 192$. Figure 6 shows that $G_3'(A_3) \geq 56$.

3.6 $k = 4$

The case $k = 4$ is the original game. The potential-function argument from Section 3.1 shows that $G_4(n) \leq 4n - 3$, in particular, $G_4(A_4) \leq 141$. Figure 7 shows that $G_4(A_4) \geq 68$.

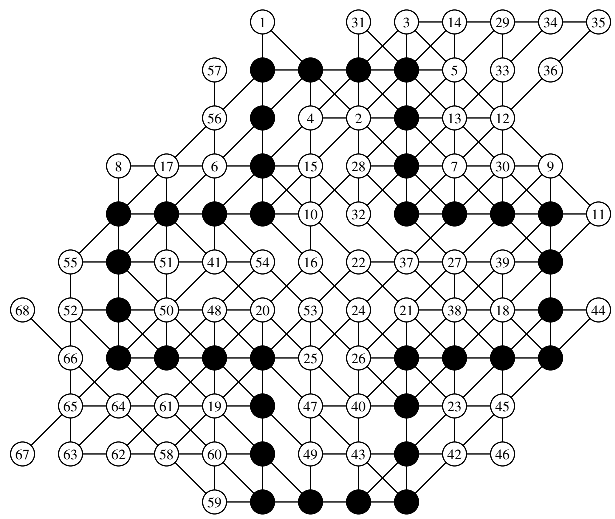

Figure 7: $G_4(A_4) \geq 68$.

The boundary argument from Section 3.2 shows that $G'_4(n) \leq \frac{4}{7}n^2 - n$, in particular, $G'_4(A_4) \leq 704$. Several lower bounds on $G'_4(A_4)$ are described in the introduction; the best known is $G'_4(A_4) \geq 170$, as shown in Figure 8.

In independent work, Flammenkamp [5] claims a tighter upper bound of $G'_4(A_4) \leq 324$, but we have been unable to verify his proof sketch.

3.7 $k \geq 5$

Figure 9 shows all the new points that can be generated from A_k in the case $k \geq 5$. The addition of those points does not form any line of sufficient density to perform another move. Furthermore, only 12 of those 24 points can appear simultaneously in a game. This shows that $G_k(A_k) = G'_k(A_k) = 12$ for $k \geq 5$.

4 Algorithmic Results

4.1 Verifying a Drawing

In this section, we present algorithms for verifying a drawing without an ordering on the added points. We use a simple greedy algorithm: at every step, find a line in the drawing that covers k existing points and one point not yet played. Play that point and line, and repeat. If all lines have been played, report a success

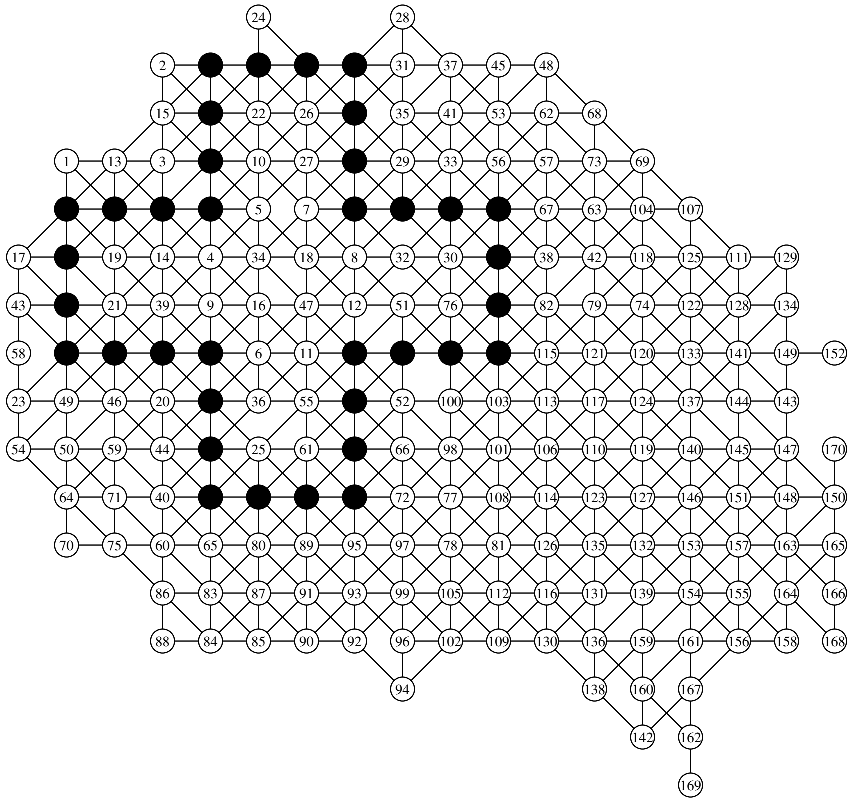

Figure 8: $G'_4(A_4) \geq 170$. This solution is by JB Bonté [9].

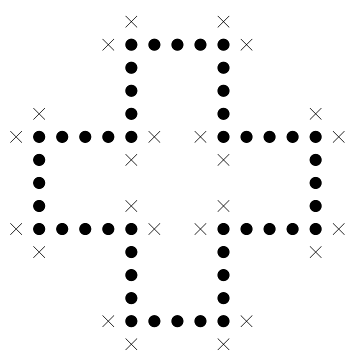

Figure 9: $k \geq 5$.

and the ordering of the lines. Otherwise, if no playable line exists, report a failure.

Lemma 1 *The drawing is valid if and only if the greedy algorithm succeeds.*

Proof: Because the greedy algorithm obeys the rules of the game, if it is successful, then the drawing is valid. So we just have to show that if the drawing is valid, then the greedy algorithm will succeed. So suppose there is a valid ordering $\ell_1, \ell_2, \ldots, \ell_n$ for drawing the lines, that the greedy algorithm has already drawn $\ell_1, \ldots, \ell_i$, and let ℓ_j be a drawable line chosen by the greedy algorithm. We just have to show that $\ell_1, \ldots, \ell_i, \ell_j, \ell_{i+1}, \ldots, \ell_{j-1}, \ell_{j+1}, \ldots, \ell_n$ is also a valid ordering. If this is not the case, it would mean that some line $\ell_{j'}$, $i+1 \leq j' \leq j-1$ cannot be drawn because of the presence of ℓ_j. It cannot be because ℓ_j collides with $\ell_{j'}$ (i.e., they share a point in the same direction in the disjoint model, or they share two points in the touching model), because the collision is independent of the order in which the lines are drawn. The only other possible reason is that when drawing ℓ_j, we drew the point that was supposed to be drawn for $\ell_{j'}$. But then drawing $\ell_{j'}$ before ℓ_j would produce the same problem, contradicting that $\ell_1, \ell_2, \ldots, \ell_n$ is a valid ordering. $\square$

Theorem 1 *Given S and a set of n lines, it is possible to verify whether those n lines are a solution for either $G_k(S)$ or $G'_k(S)$ in $O(n + |S|)$ time and space, and if so, report an order in which the lines can be drawn.*

Proof: By the previous lemma, all we need to do is to be able to find drawable lines quickly. For this task, we preprocess the drawing, creating for each line a doubly linked list of its points, and pointing each drawn point to its position in the lists of the ≤ 8 lines it is covered by. Whenever we draw a point, we remove it from the lists of all the lines pointed to by the point. If a list contains only one point, we put it in a *drawable* queue. First, the algorithm draws the points from the starting configuration. Then, at every step, a line is taken from the drawable queue, and drawn with its last remaining point. The running time for the preprocessing is $O(|S| + n)$, and every of the n steps takes $O(1)$ time. $\square$

4.2 General Dot Patterns are Hard

In this section we prove that maximizing the number of lines played starting from a general dot pattern is hard, even to approximate, in both versions of the game. We first notice that the associated decision problem is in NP for $k \geq 4$: using the bounds from Sections 3.1 and 3.2, we know that the size of a solution is polynomial in $|S|$, and can be verified in linear time using Theorem 1. The same argument does not work for $k = 2$ and $k = 3$, and it remains open whether those cases are in NP, or even in P for $k = 2$. For $k \geq 3$, the problem is NP-hard and inapproximable as shown in the remainder of this section.

Theorem 2 *For any $k \geq 3$, it is NP-hard to find the longest play from a pattern of n dots, or even to find a play of length within $n^{1-\varepsilon}$ of the longest play, for any constant $\varepsilon > 0$. This result holds for both variants of the game.*

Proof: To prove this theorem we reduce from 3-SAT. The reduction is identical for both variants of the game. The construction is only slightly different for

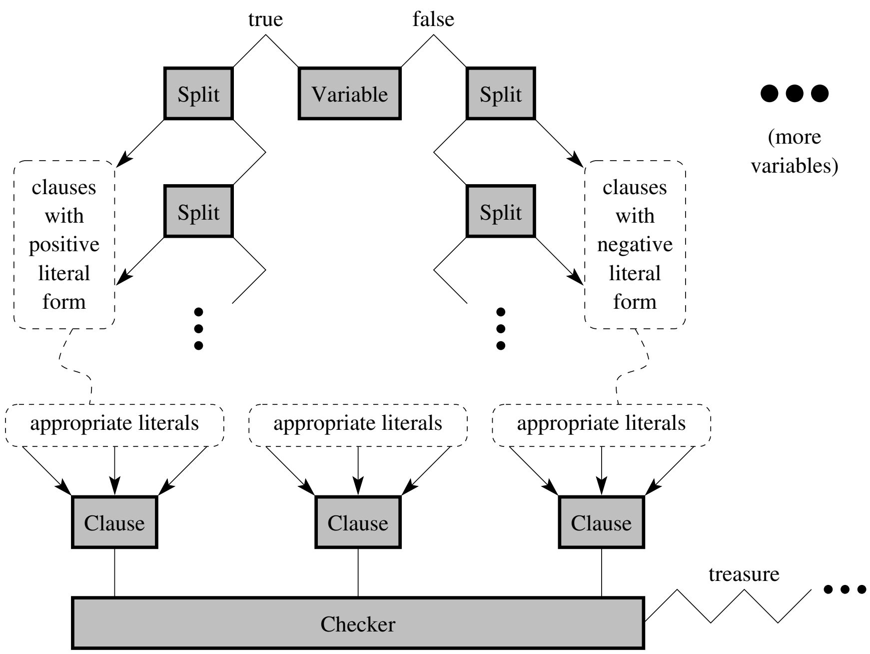

Figure 10: A schematic overview of our NP-hardness reduction from 3-SAT. This diagram illustrates most of the gadgets as black boxes, and ignores the use of crossover and shift gadgets.

different k; in the figures, we focus on $k = 3$. Figure 10 shows a schematic overview of our reduction.

Our construction represents boolean values by whether certain dots can be placed to make certain lines. The wire gadget in Figure 11 propagates this information across the construction. Specifically, one unit of a wire consists of $k-1$ dots diagonally in a row.[1] If a dot is placed on one side of these $k-1$ dots, then we can draw a line and create a dot on the other side of the $k-1$ dots. By arranging several of these $k-1$ repeats to share the blank spaces on their ends, we obtain a wire that propagates a single dot placement at one end to a dot placement at the other end. To allow for the disjoint model, we do not allow two $k-1$ repeats in a row to be collinear, but this restriction does not cause any difficulties in routing.

To start the wires with values corresponding to variables, we use the variable gadget shown in Figure 12. This gadget simply consists of k dots in a row instead of $k-1$. Thus the wire on either end can be started, but both wires cannot

[1] The routing of wires is entirely diagonal, but some other gadgets use horizontal or vertical connections to such wires. A single unit of horizontal or vertical wire works like a diagonal wire, but it is dangerous to combine them: for example, an east wire followed by a south wire followed by an east wire can be triggered without any extra dots by drawing a diagonal line. This subtle danger is why most wires in our construction are diagonal.

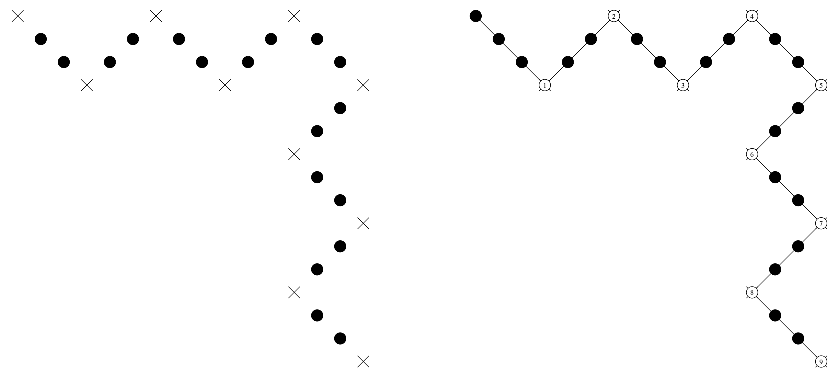

Figure 11: Wire gadget with a 90° turn. The X in the upper-left can be covered if and only if the X in the lower-right can be covered.

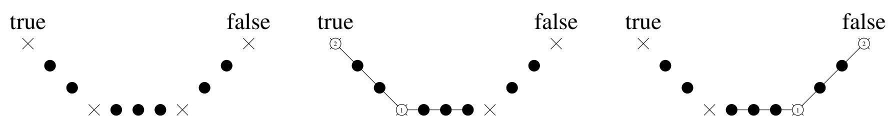

Figure 12: Variable gadget. The left or right wire can be triggered by this gadget, but not both.

be started from this gadget because of the nonoverlapping constraint. Thus, one wire represents the variable being true and the other wire represents the variable being false.

To route the value of a variable to multiple clauses, we need the split gadget shown in Figure 13. This gadget consists of joining three wire gadgets together. However, to avoid multiple wires joining collinearly, we need to use some horizontal wires. Any of the wires can be the effective "input" that triggers the other two "outputs".

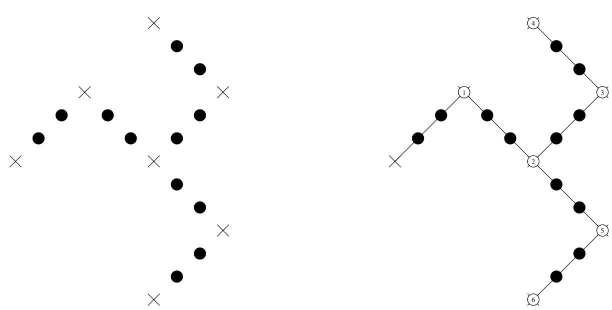

Figure 13: Split gadget. The X on the left can be covered if and only if either or both of the top and bottom X's can be covered.

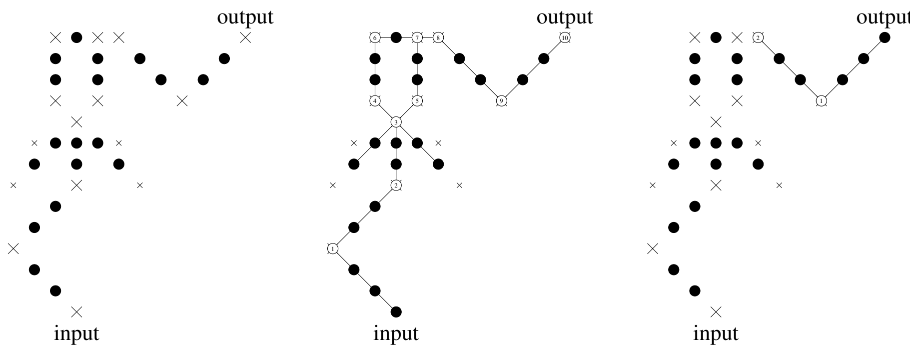

Figure 14: One-way gadget. If the X in the lower-left is covered, then the X in the upper-right can be covered, but there is no such implication in the reverse direction. Small X positions can be triggered but are irrelevant.

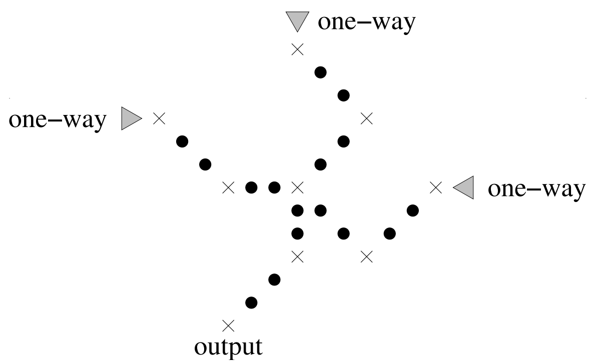

Figure 15: Clause gadget, which uses three one-way gadgets. The output wire on the bottom can be triggered if any of the three input wires can be triggered, but no other implications hold.

Before we can define the clause gadget, we need a one-way gadget that prevents information flow in one direction. Figure 14 shows such a gadget. The basic idea is to split the input wire into two so that two X's can be created in close proximity, enabling us to trigger the output wire. On the other hand, the output wire itself creates only one X, but the relevant row is lacking two X's before a line can be drawn. Thus the input wire cannot be triggered from this gadget even if the output wire is triggered.

The clause gadget is essentially three one-way input wires brought together, together with an output wire, as shown in Figure 15. Thus whenever any of the input wires is triggered, the output wire can be triggered, but the triggered input wire does not contaminate the other input wires.

We connect all of the output wires of clause gadgets to a final checker gadget, shown in Figure 16, which offers a large reward for setting all clause output wires

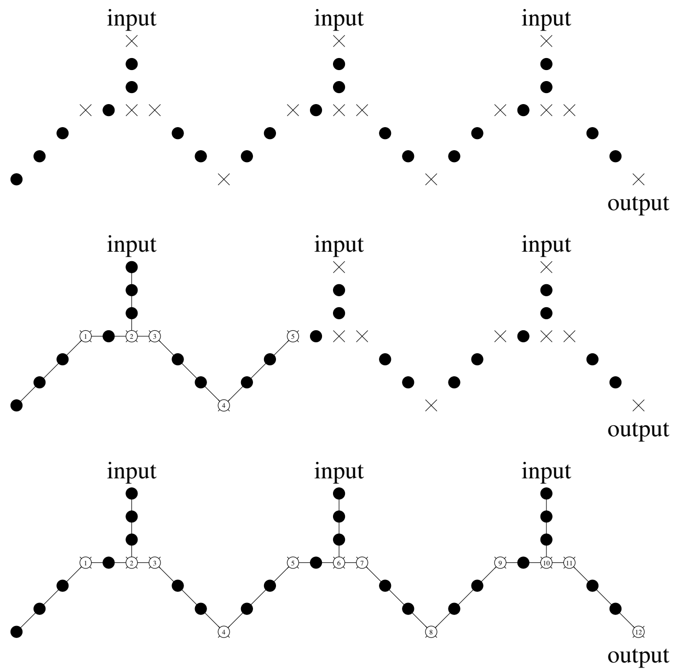

Figure 16: Checker gadget. The output wire can be triggered only if all input wires have been triggered.

correctly. The checker gadget is self-triggered by k dots in a row, but the trigger can continue at each stage only if another wire has triggered it. Thus the output wire in the lower-right can be triggered only if all clauses have been satisfied.

The output wire is connected to "treasure" which is a wire of length $n^{1/\varepsilon+O(1)}$. The reward of this treasure is so large compared to the $n^{O(1)}$ possible lines obtained elsewhere in the construction that even approximate solution to the instance requires solving the 3-SAT instance to gain the treasure.

Two technical issues not yet addressed in this construction are crossings and parity. Crossings in the wiring map can be handled with the crossover gadget in Figure 17. Parity issues arise when trying to connect gadgets whose sizes do not evenly divide each other. These issues can be resolved using the shift gadget in Figure 18, which moves a wire one step (modulo 3) in any desired direction. By repeating $O(1)$ shift gadgets, wires can be aligned horizontally or vertically to match any target gadget. □

Acknowledgments

We thank Walter Joris for introducing us to some of the previous work on morpion solitaire; he also independently discovered an infinite play for the $k = 3$

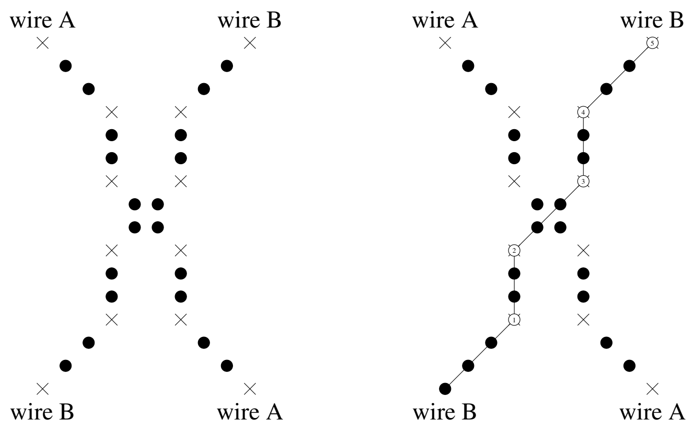

Figure 17: Crossover gadget. Wires A and B act as if they did not cross.

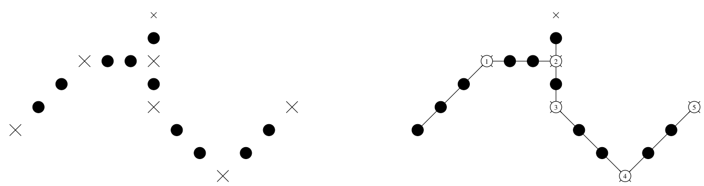

Figure 18: Shift gadget, shown here shifting a horizontal wire up by one step.

touching variant. We thank Stefan Schmieta for helpful discussions and his script for drawing game executions, on which our figures are based. We thank KHBO Spellenarchief and Jean-Dominique Quinet for providing copies of the *Jeux & Stratégies* articles. We thank Barry Cipra and the anonymous referee for helpful comments on the paper.

References

[1] Enigma of malta cross game. http://croix2malte.free.fr/indexGB.php.

[2] Post-scriptum au no. 17. *Jeux & Stratégie*, 18:101, Dec. 1982 and Jan. 1983.

[3] P. Brass. Erdős distance problems in normed spaces. *Comput. Geom. Theory Appl.*, 6:195–214, 1996.

[4] Michel Brassine. Le morpion solitaire. *Jeux & Stratégie*, 16:28–29, Aug.–Sept. 1982.

[5] Achim Flammenkamp. Le morpion solitaire. Manuscript, March 19 2003. http://wwwhomes.uni-bielefeld.de/achim/morpion.dvi.

[6] Walter Joris. *All Squared Up: Tic-Tac-Toe for Geniuses (100 Strategic Games for Pen and Paper)*. Carlton Books, 2002.

[7] Hugues Juille. Incremental co-evolution of organisms: a new approach for optimization and discovery of strategies. In *Proceedings of the Third European Conference on Artificial Life*, volume 929 of *LNCS*, Granada, Spain, June 1995.

[8] Stefan Langerman. A game. http://slef.org/jeu/.

[9] Jean-Charles Meyrignac. Morpion solitaire. http://euler.free.fr/morpion.htm.