Buttons & Scissors is NP-Complete

Harrison Gregg * Jody Leonard † Aaron Santiago‡ §Aaron Williams

Abstract

Buttons & Scissors is a popular single-player puzzle. A level is played on an n-by-n grid, where each position is empty or has a single coloured button sewn onto it. The player's goal is to remove all of the buttons using a sequence of horizontal, vertical, and diagonal scissor cuts. Each cut removes all buttons between two distinct buttons of the same colour, and is not valid if there is an intermediate button of a different colour. We prove that deciding whether a given level can be completed is NP-complete. In fact, NP-completeness holds when only horizontal and vertical cuts are allowed, and each colour is used by at most 7 buttons. Our framework was also used in an NP-completeness proof when each colour is used by at most 4 buttons, which is best possible.

Keywords: *NP-completeness, pencil-and-paper puzzle.*

1 Introduction

Buttons & Scissors is a single-player puzzle by KyWorks that is available as a free iOS and Android app. The goal is to remove all buttons from an $n \times n$ grid using a series of *scissor cuts* that have the following properties:

- a cut is a straight-line segment whose endpoints are centers of distinct buttons and which is horizontal, vertical, or diagonal at a 45° or −45° angle;
- a cut's line segment touches at least two buttons of the same color, and no buttons of another color;
- a cut removes all buttons on its line segment.

Figure 1 illustrates a sample level and solution.

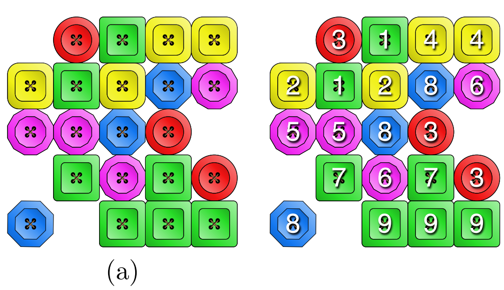


Figure 1: (a) Level 7 in Buttons & Scissors, and (b) a solution using nine cuts.

*hgregg11@simons-rock.edu
†jleonard11@simons-rock.edu
‡asantiago11@simons-rock.edu
§haron@uvic.ca

Buttons & Scissors is reminiscent of many grid-based pencil-and-paper puzzles that have been popularized by the Japanese company Nikoli. To analyze the computational complexity of an individual puzzle, it is necessary to generalize certain aspects of the puzzle, including its grid size. For example, the generalized version of Sudoku involves an n^2-by-n^2 grid with blocks of size n and integers from 1 to n^2 (with the standard version having $n = 3$). The book *Games, Puzzles, and Computation* by Hearn and Demaine [2] provides hardness results for many generalized grid games. We consider the following decision problem.

Decision Problem 1 B&S(B)
Input*: An n-by-n board B.*
Output*: True if B has a solution, and False otherwise.*

We also consider a cut-constrained version of Buttons & Scissors in which diagonal cuts are not allowed.

Decision Problem 2 B&S+(B)
Input*: An n-by-n board B.*
Output*: True if B has a solution using only horizontal and vertical cuts, and False otherwise.*

Notice $B\&S(B)$ is True and $B\&S{+}(B)$ is False for the board in Figure 1. In general, $B\&S{+}(B) \Rightarrow B\&S(B)$. However, *a priori*, there is no relationship between the difficulty of deciding $B\&S$ and $B\&S+$. In this article we prove that both problems are NP-complete. In fact, we achieve slightly stronger results that also constrain the number of times each distinct colour can be used.

Theorem 1 *B&S and B&S+ are both NP-complete. Furthermore, both problems are NP-complete when each colour is used by at most $F = 7$ buttons.*

Section 2 describes our Buttons & Scissors gadgets, and Section 3 formalizes the version of 3-SAT that we use for Theorem 1. Section 4 provides our reduction and Section 5 proves that it is correct. Open problems and further results appear in Section 6, including an improvement of Theorem 1 to $F = 4$ for the $B\&S$ puzzle.

2 Linear Gadgets

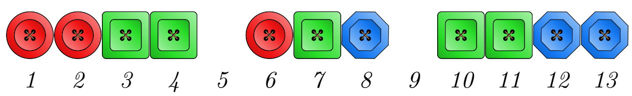
Our strategy is to prove the hardness of $B\&S$ and $B\&S+$ simultaneously using a single 3-SAT reduction. We do this by mapping instances of 3-SAT to Buttons

& Scissor boards in which no diagonal cuts are possible. More specifically, if two buttons have the same colour, then they will not lie on any common diagonal line. Towards this goal we construct gadgets whose buttons can fit on a single row or column. Linear gadgets for OR and AND are given in Sections 2.1 and 2.2, respectively.

2.1 OR Gadget

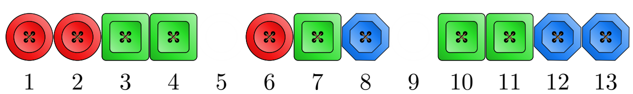
The following gadget has three Boolean inputs. Each input determines whether a given button has been removed from the board.

Definition 1 *Let* $\mathsf{OR}(\mathsf{X_1}, \mathsf{X_2}, \mathsf{X_3})$ *be the following Buttons & Scissors board,*

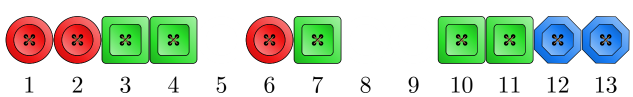


where buttons in positions $1, 2, 6$ *are one colour, those in positions* $3, 4, 7, 10, 11$ *are a second distinct colour, and those in positions* $8, 12, 13$ *are a third distinct colour. The buttons in positions 6, 7, 8 are absent if* $X_1 = \mathsf{T}$, $X_2 = \mathsf{T}$, $X_3 = \mathsf{T}$, *respectively.*

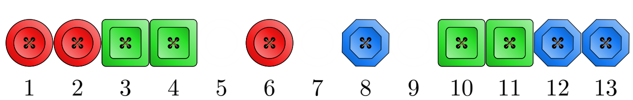
For example, when $X_1 = X_2 = X_3 = \mathsf{F}$, the buttons in positions $6, 7, 8$ are present, as shown below.



Notice that it is impossible to remove the button in position 7 from this board using any sequence of cuts. As another example, consider $X_1 = X_2 = \mathsf{F}$ and $X_3 = \mathsf{T}$, in which the button in position 8 is not present.



This board can be solved by successively cutting positions 3-4, then 1-6, then 7-11, then finally 12-13. For a third example, consider $X_1 = \mathsf{F}$, $X_2 = \mathsf{T}$, $X_3 = \mathsf{F}$, in which only the button in position 7 is removed.



This board can be solved by successively cutting positions 3-4, then 1-6, then 10-11, then finally 8-13.

More generally, this board can be solved if and only if at least one input is true.

Lemma 2 $B\&S(\mathsf{OR}(\mathsf{X_1}, \mathsf{X_2}, \mathsf{X_3})) \iff X_1 \vee X_2 \vee X_3$.

Proof. The following table provides a sequence of cuts to solve the board whenever $X_1 \vee X_2 \vee X_3 = \mathsf{T}$.

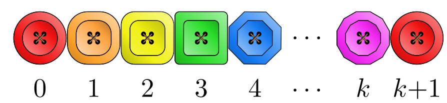
X_1	X_2	X_3	Cut Sequence
T	T	T	1-2, 3-11, 12-13
T	T	F	1-2, 3-4, 10-11, 8-13
T	F	T	1-2, 3-11, 12-13
T	F	F	1-2, 3-7, 10-11, 8-13
F	T	T	3-4, 1-6, 10-11, 12-13
F	T	F	3-4, 1-6, 10-11, 8-13
F	F	T	3-4, 1-6, 7-11, 12-13

When $X_1 = X_2 = X_3 = \mathsf{F}$ it is impossible to remove the button in position 6. □

2.2 AND Gadget

The following gadget has k Boolean inputs.

Definition 2 *Let* $\mathsf{AND}(\mathsf{X_1}, \mathsf{X_2}, \ldots, \mathsf{X_k})$ *be the following Buttons & Scissors board,*



where buttons in positions $1, 2, \ldots, k$ *have unique colours, and these colours are distinct from the common colour used by the button in position* 0 *and* $k{+}1$. *The button in position* i *is absent if* $X_i = \mathsf{T}$ *for each* $i = 1, 2, \ldots, k$.

Lemma 3 $B\&S(\mathsf{AND}(\mathsf{X_1}, \mathsf{X_2}, \ldots, \mathsf{X_k})) \iff X_1 \wedge X_2 \wedge \cdots \wedge X_k$.

Proof. If $X_1 \wedge X_2 \wedge \ldots \wedge X_k$ is true, then the only buttons in $\mathsf{AND}(\mathsf{X_1}, \mathsf{X_2}, \ldots, \mathsf{X_k})$ are those in positions 0 and $k{+}1$. These buttons have the same colour and can be removed with one cut, 0-$k{+}1$. If $X_1 \wedge X_2 \wedge \ldots \wedge X_k$ is false, then $\mathsf{AND}(\mathsf{X_1}, \mathsf{X_2}, \ldots, \mathsf{X_k})$ contains at least one button with a unique colour that cannot be cut. □

3 Conventions

This section describes the version of 3-SAT that we utilize in our reduction, and our conventions for describing Buttons & Scissors boards.

3.1 3-SAT

Let $\mathcal{S}$ denote the set of 3-SAT instances that have distinct clauses and exactly three literals of distinct variables in each clause. It is easy to see that this version of 3-SAT is NP-complete, even though it varies from Karp's original formulation [3].

An arbitrary 3-SAT instance $S \in \mathcal{S}$ has variables $V_1, V_2, \ldots, V_n$ and clauses $C_1, C_2, \ldots, C_m$. We write an arbitrary clause C_x and its literals as

$$L_{i,x} \vee L_{j,x} \vee L_{k,x} \text{ where } i < j < k \text{ and } \begin{aligned} L_{i,x} &\in \{V_i, \neg V_i\} \\ L_{j,x} &\in \{V_j, \neg V_j\} \\ L_{k,x} &\in \{V_k, \neg V_k\}. \end{aligned}$$

In other words, $L_{a,b}$ is the literal of variable V_a that appears in clause C_b. We refer to each $L_{a,b}$ as a *literal instance*. For example, if S is the following,

$$(\mathsf{V}_1 \vee \mathsf{V}_2 \vee \mathsf{V}_3) \wedge (\mathsf{V}_1 \vee \neg\mathsf{V}_2 \vee \neg\mathsf{V}_3) \wedge (\neg\mathsf{V}_1 \vee \neg\mathsf{V}_3 \vee \neg\mathsf{V}_4),$$

then $C_1 = (\mathsf{V}_1 \vee \mathsf{V}_2 \vee \mathsf{V}_3)$ and so $L_{2,1} = V_2$. Similarly, $C_3 = (\neg\mathsf{V}_1 \vee \neg\mathsf{V}_3 \vee \neg\mathsf{V}_4)$ and so $L_{4,3} = \neg V_4$.

3.2 Buttons & Scissors Boards

Let $\mathcal{B}$ denote the set of Buttons & Scissors boards. We will use different shapes for buttons that play different roles in each $B \in \mathcal{B}$ that we construct:

- *Clause buttons* are circular ◯ (Section 4.1);
- *AND buttons* are octagonal ⯃ (Section 4.1).
- *Variable buttons* are trapezial ⏢ (Section 4.2);
- *Literal instance buttons* are rectangular □ (Section 4.3);

Each button will be uniquely identifiable by (a) its shape, (b) its label in the shape, and (c) its subscript. Our convention is that buttons have the same colour if and only if they have (a) the same shape, and (b) the same label in the shape. For example, $\textcircled{3}_L$ and $\textcircled{3}_R$ are the same colour. Similarly, $\boxed{1{,}2}_L$ and $\boxed{1{,}2}_R$ are the same colour. However, $\backslash 3/_L$ and $\backslash 4/_L$ are different colours.

4 Reduction

This section describes our reduction $r : \mathcal{S} \rightarrow \mathcal{B}$ that maps an instance of 3-SAT $S \in \mathcal{S}$ to a Buttons & Scissors board $B = r(S) \in \mathcal{B}$. Sections 4.1–4.3 describe buttons in B resulting from clauses, variables, and literal instances, respectively. The layout of the entire board is then discussed in Section 4.4, and various properties of the constructed board are given in Section 4.5.

Figure 2 contains both a high-level view of our reduction, as well as a specific example.

4.1 Clauses

Each clause in the 3-SAT instance is translated into its own row of buttons in the board B (see Figure 2). In particular, $C_x = L_{i,x} \vee L_{j,x} \vee L_{k,x}$ with $i < j < k$ contains the following OR gadget

$$\boxed{\text{i}}_{L_1}\ \boxed{\text{i}}_{L_2}\ \boxed{\text{j}}_{L_1}\ \boxed{\text{j}}_{L_2} \quad \boxed{\text{i}}_{M}\ \boxed{\text{j}}_{M}\ \boxed{\text{k}}_{M} \quad \boxed{\text{j}}_{R_1}\ \boxed{\text{j}}_{R_2}\ \boxed{\text{k}}_{R_1}\ \boxed{\text{k}}_{R_2}$$

Each of these rectangular literal buttons has the additional label $\boxed{,\text{x}}$ which is not shown for space reasons. Also, buttons corresponding to negative literals have an overline which is not indicated above (see Figure 2). Notice that there are three distinct button colours above and they follow the OR gadget pattern. The subscripts for each literal instance colour are Left (twice for i and j), Middle (once each), and Right (twice for j and k). The horizontal positioning of the middle buttons is below its respective variable positive/negative column as discussed in Section 4.2. An additional pair of Cleanup buttons for each colour are placed above each middle button.

The row of buttons for clause C_x contains one additional pair of clause buttons as bookends.

$$\textcircled{x}_L\ \boxed{\text{i}}_{L_1}\ \boxed{\text{i}}_{L_2}\ \cdots\ \boxed{\text{k}}_{R_1}\ \boxed{\text{k}}_{R_2}\ \textcircled{x}_R$$

The subscripts for these clause buttons are Left and Right. These are the only two buttons of this colour on the entire board B. Therefore, by Lemma 2 we have the following remark.

Remark 1 *The clause buttons $\textcircled{x}_L$ and $\textcircled{x}_R$ can only be removed after all of the buttons in the OR gadget for clause C_x are removed.*

We place all of the Left clause buttons within an AND gadget as follows

$$⯃_1\ \ \textcircled{1}_L\ \textcircled{2}_L\ \textcircled{3}_L\ \ \cdots\ \ \textcircled{m}_L\ \ ⯃_2$$

which appears as a single vertical column in B (see Figure 2). The AND buttons ⯃$_1$ and ⯃$_2$ are the only such pair on B, and they share a unique colour. Therefore, the previous remark and Lemma 3 imply the following.

Remark 2 *The AND buttons ⯃$_1$ and ⯃$_2$ can only be removed after all of the clause buttons are removed.*

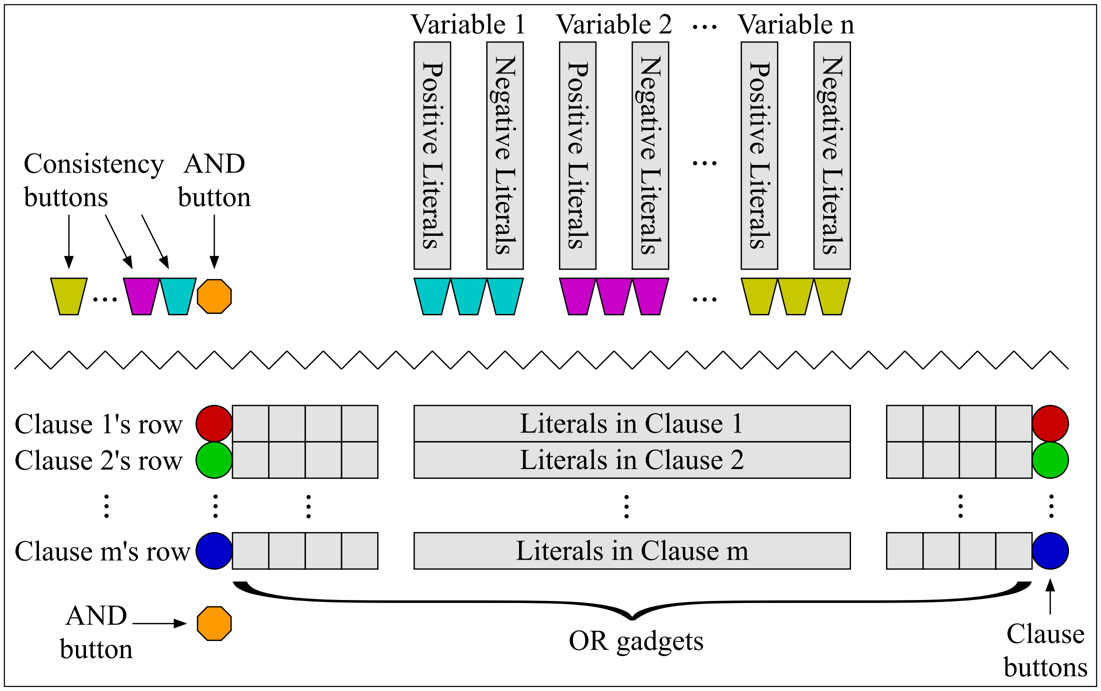
4.2 Variables

The following row of buttons appears in B

$$\backslash\text{n}/_C \cdots \backslash 2/_C\ \backslash 1/_C\ ⯃_1\ \backslash 1/_P\ \backslash 1/_D\ \backslash 1/_N\ \backslash 2/_P\ \backslash 2/_D\ \backslash 2/_N \cdots \backslash\text{n}/_P\ \backslash\text{n}/_D\ \backslash\text{n}/_N.$$

There are four buttons for each variable, with subscripts for Positive, Decision, Negative, and Consistency. Each quartet has its own distinct colour that is not used by any other buttons on B, and the consistency button is separated from the others by the first AND button.

As we will see in Section 5, this configuration ensures that a variable's decision button will either be cut with its positive button, or its negative button, and this choice will correspond to setting the variable equal to T or F, respectively. (If all four buttons are cut simultaneously, then the variable is 'free' and can be assigned either way in a satisfying assignment.)



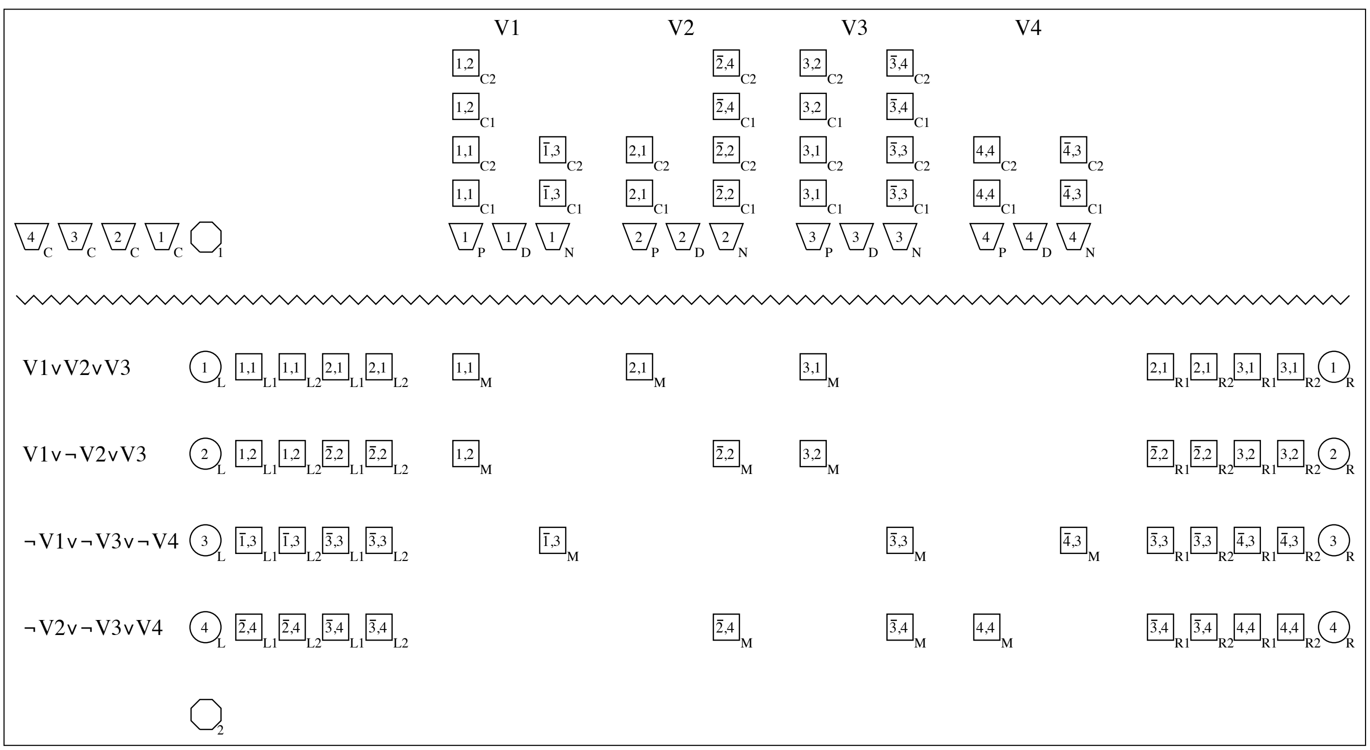


Figure 2: Top: An overview of our reduction. The gray squares contain buttons whose colours are not specified in the figure, and the gray rectangles contain blank spaces and buttons whose colours are not specified. Bottom: $(V_1 \vee V_2 \vee V_3) \wedge (V_1 \vee \neg V_2 \vee \neg V_3) \wedge (\neg V_1 \vee \neg V_3 \vee \neg V_4) \wedge (\neg V_2 \vee \neg V_3 \vee V_4)$. The zig-zags denote a series of empty rows.

4.3 Literal Instances

For each variable V_i, pairs of Cleanup buttons for each of its positive and negative literal instances are stacked vertically above its positive button $\backslash\text{i}/_P$ and negative button $\backslash\text{i}/_N$, respectively, as follows

$$\begin{array}{ccc} \vdots & & \vdots \\ \backslash\text{i}/_P & \backslash\text{i}/_D & \backslash\text{i}/_N. \end{array}$$

More precisely, if V_i appears as a positive literal in clauses $C_{p_1}, C_{p_2}, \ldots, C_{p_x}$ for $p_1 < p_2 < \cdots < p_x$, then the buttons above the variable's positive button appear as follows (with left-to-right being bottom-to-top)

$$\backslash\text{i}/_P\ \boxed{\text{i},p_1}_{C_1}\ \boxed{\text{i},p_1}_{C_2}\ \boxed{\text{i},p_2}_{C_1}\ \boxed{\text{i},p_2}_{C_2}\ \cdots\ \boxed{\text{i},p_x}_{C_1}\ \boxed{\text{i},p_x}_{C_2}$$

where $\backslash\text{i}/_P$ is at the bottom of this vertical stack. Similarly, if V_i appears as a negative literal in clauses $C_{n_1}, C_{n_2}, \ldots, C_{n_x}$ for $n_1 < n_2 < \cdots < n_x$, then the buttons above the variable's negative button appear from left-to-right as follows

$$\backslash\text{i}/_N\ \boxed{\text{i},n_1}_{C_1}\ \boxed{\text{i},n_1}_{C_2}\ \boxed{\text{i},n_2}_{C_1}\ \boxed{\text{i},n_2}_{C_2}\ \cdots\ \boxed{\text{i},n_x}_{C_1}\ \boxed{\text{i},n_x}_{C_2}$$

where $\backslash\text{i}/_N$ is at the bottom of this vertical stack.

These cleanup pairs appear in the same column and in opposite order as the literal instance buttons in their respective clauses (see Figure 2). Therefore, we have the following remark.

Remark 3 *If variable V_i's positive button $\backslash\text{i}/_P$ is removed from B, then all buttons in V_i's positive column can be removed. Similarly, if $\backslash\text{i}/_N$ is removed, then all buttons in V_i's negative column can be removed.*

4.4 Layout

In our construction, the only button colours that appear on more than one single row or column are the literal instance buttons $\boxed{\text{i,j}}$. More specifically, a button of colour $\boxed{\text{i,j}}$ appears both on clause C_j's row and in variable v_i's positive or negative literal column. To avoid possible diagonal cuts between these buttons, we include a series of blank rows between the clause rows and the variable row. By our construction, a total of $4n + 6$ blank rows ensures this property since the leftmost literal instance button in clause C_1's row is strictly below the bottom literal instance button in variable V_n's negative literal column, with respect to a 45° diagonal line.

4.5 Properties

We conclude this section by summarizing a number of simple properties of our reduction.

Remark 4 *If two buttons in B have the same colour, then they do not lie on the same diagonal line.*

Remark 5 *If $S \in \mathcal{S}$ has n variables and m clauses, then the board $B = r(S)$ is an $O(n+m)$-by-$O(n)$ grid.*

Remark 6 *Each colour is used by at most 7 buttons. In particular, variable buttons $\backslash i/$ appear 4 times, clause buttons j appear twice, AND buttons $\bigcirc$ appear twice, and instance literals buttons $\boxed{i,j}$ appear 5 or 7 times.*

5 NP-Completeness

In this section we prove Theorem 1. That is, $B\&S$ and $B\&S+$ are both NP-complete. First we demonstrate that the problems are in NP.

Lemma 4 *$B\&S$ and $B\&S+$ are in NP.*

Proof. Suppose $B \in \mathcal{B}$ is an n-by-n board. In both versions of the puzzle, a cut can be specified by two grid co-ordinates, and at most $O(n^2)$ cuts are necessary to clear B. Therefore, a sequence of cuts that solves B can be specified in polynomial-size with respect to B. It is also clear that we can verify if a sequence of cuts clears a board in polynomial-time. Therefore, a sequence of cuts provides a polynomial-size certificate that can be verified in polynomial-time when $B\&S(B) = \mathsf{T}$ or $B\&S{+}(B) = \mathsf{T}$. $\square$

Our reduction creates a polynomial-size board by Remark 5. Therefore, to complete the proof of Theorem 1 we need to prove that if $S \in \mathcal{S}$ and $B = r(S)$ then

$$S \text{ is satisfiable} \iff B\&S(B) \text{ and}$$
$$B\&S(B) \iff B\&S{+}(B).$$

By Remark 4 there are no diagonal cuts in B, so $B\&S(B) \iff B\&S{+}(B)$ has been established. We prove the first $\iff$ in the following two subsections.

5.1 Clearing the board

In this subsection we provide a solution for the Buttons & Scissors board given a satisfying assignment to the 3-SAT problem.

Lemma 5 *Suppose $S \in \mathcal{S}$ and $B = r(S)$. If S is satisfiable, then $B\&S(B) = \mathsf{T}$.*

Proof. Consider a fixed satisfying assignment for S. We now provide a sequence of cuts that clears B.

1. Perform the following for each $i = 1, 2, \ldots, n$:
   - If $V_i = \mathsf{T}$ in the satisfying assignment, then cut $\backslash\text{i}/_P$ and $\backslash\text{i}/_D$. Then cut all buttons in variable V_i's positive literal column by Remark 3.
   - Otherwise, cut $\backslash\text{i}/_N$ and $\backslash\text{i}/_D$. Then cut all buttons in variable V_i's negative column.

2. Perform the following for each $j = 1, 2, \ldots, m$:

- Remove every button in the OR gadget on clause C_j's row.
- Remove the clause buttons $\textcircled{j}_L$ and $\textcircled{j}_R$.

The first step is possible by Lemma 2 and the fact that we stared with a satisfying assignment. The second step is possible by Remark 1 .

3. Cut the AND buttons $\bigcirc_1$ and $\bigcirc_2$ by Remark 2.
4. Perform the following for $i = 1, 2, \ldots, n$ in order:
   - If $V_i = \mathsf{T}$ in the assignment, then cut $\overline{\backslash i/}_C$ and $\overline{\backslash i/}_N$. Then cut all buttons in variable V_i's negative literal column by Remark 3.
   - Otherwise, cut $\overline{\backslash i/}_C$ and $\overline{\backslash i/}_P$. Then cut all buttons in variable V_i's positive literal column by Remark 3.

The cuts remove all buttons, so $B\&S(B) = \mathsf{T}$. □

5.2 Satisfying the formula

Now we provide a satisfying assignment to the 3-SAT problem given that its equivalant board is solvable. Lemma will allow us to map a solution to the Buttons & Scissors board to a 3-SAT variable assignment.

Lemma 6 *Consider a sequence of cuts that clears board $B = r(S)$. For each variable V_i, the variable buttons for V_i are cut in one of three ways:*

1. *First $\overline{\backslash i/}_P$ and $\overline{\backslash i/}_D$ are cut. Then $\overline{\backslash i/}_C$ and $\overline{\backslash i/}_N$ are cut after the two AND buttons are cut.*
2. *First $\overline{\backslash i/}_D$ and $\overline{\backslash i/}_N$ are cut. Then $\overline{\backslash i/}_C$ and $\overline{\backslash i/}_P$ are cut after the two AND buttons are cut.*
3. *All four buttons — $\overline{\backslash i/}_C$, $\overline{\backslash i/}_P$, $\overline{\backslash i/}_D$ and $\overline{\backslash i/}_N$ — are cut together after the two AND buttons are cut.*

Proof. The relative order of these buttons is

$$\overline{\backslash i/}_C \quad \bigcirc_1 \quad \overline{\backslash i/}_P \quad \overline{\backslash i/}_D \quad \overline{\backslash i/}_N.$$

Thus, the three cases follow immediately. □

We now complete our proof of Theorem 1.

Lemma 7 *Suppose $S \in \mathcal{S}$ and $B = r(S)$. If $B\&S(B) = \mathsf{T}$, then S is satisfiable.*

Proof. Let $c_1, c_2, \ldots, c_k$ be a sequence of cuts that clears B. By Lemma 6, we can create a variable assignment for S as follows:

- If the first case occurs, then set $V_i = \mathsf{T}$.
- If the second case occurs, then set $V_i = \mathsf{F}$.
- Otherwise, the choice is arbitrary and set $V_i = \mathsf{T}$.

We will prove that this assignment is satisfying.

Consider the cut c_a that removes the AND buttons $\bigcirc_1$ and $\bigcirc_2$. Prior to c_a, all clause buttons must have been removed by Remark 2. Therefore, by Remark 1 all of the OR gadget buttons must have been removed prior to c_a. Therefore, by Lemma 2 and the construction of B, at least one of the central buttons in each OR gadget must have been removed prior to c_a by some vertical cut. Therefore, we have the following prior to c_a for each clause C_j: There exists a variable V_i such that $V_i = \mathsf{T}$ and the literal V_i is in C_j and its button $\boxed{\text{i,j}}_M$ was removed, or $V_i = \mathsf{F}$ and the literal $\neg V_i$ is in C_j and its button $\boxed{\text{i,j}}_M$ was removed. Therefore, the variable assignment satisfies S. □

6 Additional Results and Open Problems

Buttons & Scissors has a number of natural variations including the following colour-constrained versions:

1. There are at most C distinct colours of buttons.
2. Each colour can be used by at most F buttons.

We proved $B\&S$ and $B\&S+$ are NP-complete for $F = 7$ in Theorem 1. We also conjectured NP-completeness for $F = 4$ in our initial submission. This was recently verified by a second research group for the $B\&S$ puzzle.

Theorem 8 ([1]) *$B\&S$ is NP-complete when each colour is used by at most $F = 4$ buttons.*

The proof of Theorem 8 uses our reduction with a new OR gadget (see Figure 2 in [1]). The new gadget uses colours less frequently but requires all four cut directions. Thus, the hardness of $B\&S+$ with $F = 4$ is still open. The $F = 4$ cases are particularly interesting because $B\&S$ and $B\&S+$ are polytime solvable when $F = 3$. To see this, notice that all buttons of a given colour must be removed by a single cut when $F = 3$. Furthermore, removing all buttons of a given colour cannot turn a solvable board into an unsolvable board. Thus, a simple greedy algorithm suffices.

Our initial submission also conjectured hardness when $C = 2$, and this was also recently verified in [1]. A full journal article with the authors of [1] is also planned.

An implementation of our reduction is available: http://jabdownsmash.com/button3sat/index.html.

References

[1] K. Burke, E. Demaine, R. Hearn, A. Hesterberg, M. Hoffman, H. Ito, I. Kostitsyna, M. L offler, Y. Uno, C. Schmidt, R. Uehara, and A Williams. Single-player and two-player buttons & scissors games. In *Proceedings of the 18th Japan Conference on Discrete and Computational Geometry and Graphs*, page 2 pages, 2015.

[2] R. A. Hearn and E. D. Demaine. *Games, Puzzles, & Computation.* A K Peters, 1st edition, 2009.

[3] Richard M. Karp. Reducibility among combinatorial problems. *Complexity of Computer Computations*, pages 85–103, 1972.