Bust-a-Move/Puzzle Bobble is NP-Complete

Erik D. Demaine* Stefan Langerman†

Abstract

We prove that the classic 1994 Taito video game, known as Puzzle Bobble or Bust-a-Move, is NP-complete. Our proof applies to the perfect-information version where the bubble sequence is known in advance, and uses just three colors.

1 Introduction

Erik grew up playing the action platform video game *Bubble Bobble* (バブルボブル), starring cute little brontosauruses Bub and Bob, on the Nintendo Entertainment System. (The game was first released by Taito in 1986, in arcades [3].) Some years later (1994), Bub and Bob retook the video-game stage with the puzzle game *Puzzle Bobble* (パズルボブル), known as *Bust-a-Move* in the U.S. [4, 6]. This game essentially got Stefan through his Ph.D.: whenever he needed a break, he would play as much as he could with one quarter.

In Puzzle Bobble, the game state is defined by a hexagonal grid, each cell possibly filled with a *bubble* of some color. In each turn, the player is given a bubble of some color, which can be fired in any (upward) direction from the pointer at the bottom center of the board. The fired bubble travels straight, reflecting off the left and right walls, until it hits another bubble or the top wall, in which case it terminates at the nearest grid-aligned position. If the bubble is now in a connected group of at least three bubbles of the same color, then that group disappears ("pops"), and any bubbles now disconnected from the top wall also pop.

Here we study the perfect-information (generalized) form of Puzzle Bobble. We are given an initial board of bubbles and the entire sequence of colored bubbles that will come. The goal is to clear the board using the given sequence of bubbles. (The actual game has an infinite, randomly generated sequence of bubbles, like Tetris [1].) The game also has a falling ceiling, where all bubbles descend every fixed number of shots; and if a bubble hits the floor, the game ends. We assume that the resolution of the input is sufficiently fine to hit any discrete cell that could be hit by an (infinitely precise) continuous shot. (This assumption seems to hold in the original game, so it is natural to generalize it.)

Theorem 1 *Puzzle Bobble is NP-complete.*

Membership in NP is easy: specify where to shoot each of the n given bubbles. The rest of this paper establishes NP-hardness.

Our reduction applies to all versions of Puzzle Bobble. Viglietta [5] proved that Puzzle Bobble 3 is NP-complete, by exploiting "rainbow" (wildcard) bubbles. Our proof shows that this feature is unnecessary.

2 NP-Hardness

The reduction is from Set Cover: given a collection $\mathcal{S} = \{S_1, S_2, \ldots, S_s\}$ of sets where each $S_i \subseteq U$, and a positive integer k, are there k sets $S_{i_1}, S_{i_2}, \ldots, S_{i_k}$ whose union covers all elements of U?

Figure 1 shows the overall structure of the reduction. The bulk of the construction is in the central small square, which is aligned on the top side of an $m \times m$ square above the floor. By making the central square small enough, the angles of direct shots at the square are close to vertical (needed to solve most gadgets), and the rebound angles that hit the square are all approximately 45° (needed to solve the crossover gadget below). The player could do multiple rebounds (or destroy bubbles to cause the ceiling to lower prematurely) to make shot angles more horizontal, but this will only make it harder to solve the gadgets.

The sequence of bubbles given to the player is as follows. The very first color appears only k times, where k is the desired set-cover size. Each remaining color appears sufficiently many times ($\Theta(s|U|)$ times, which we will refer to as ∞). Unneeded bubbles can be discarded by forming isolated groups of size 3, 4, or 5 off to the side.

k blue ●,	∞ yellow ●,	∞ blue ●,	∞ red ●;
∞ blue ●,	∞ yellow ●,	∞ blue ●,	∞ red ●;
∞ blue ●,	∞ yellow ●,	∞ blue ●,	∞ red ●;
⋮	⋮	⋮	⋮
∞ blue ●,	∞ yellow ●,	∞ blue ●,	∞ red ●;
∞ red ●,	∞ red ●,	∞ red ●,	…

*MIT Computer Science and Artificial Intelligence Laboratory, 32 Vassar St., Cambridge, MA 02139, USA, edemaine@mit.edu

†Directeur de recherches du F.R.S.–FNRS, Départment d'Informatique, Université Libre de Bruxelles, Brussels, Belgium, stefan.langerman@ulb.ac.be

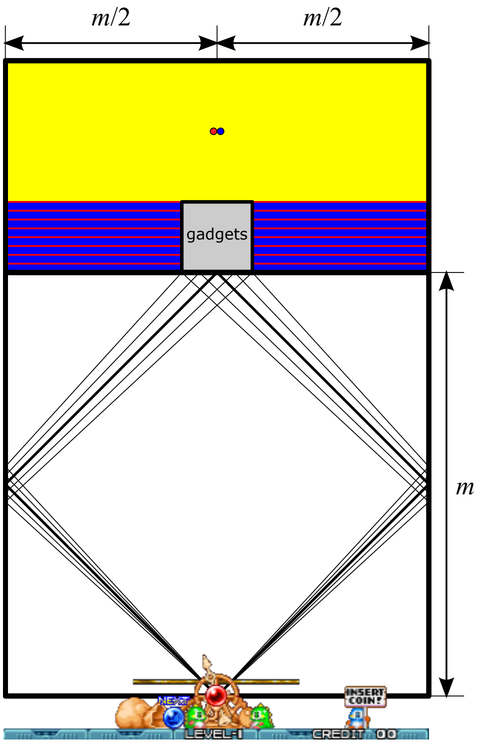

Figure 1: Overall structure of the reduction. All other gadgets lie within a small square at the top of an $m \times m$ square, where m is the width of the game. Red horizontal lines separate the gadgets into layers, with blue fill in between. At the top is a huge rectangle of yellow bubbles with one red bubble and one blue bubble in the middle.

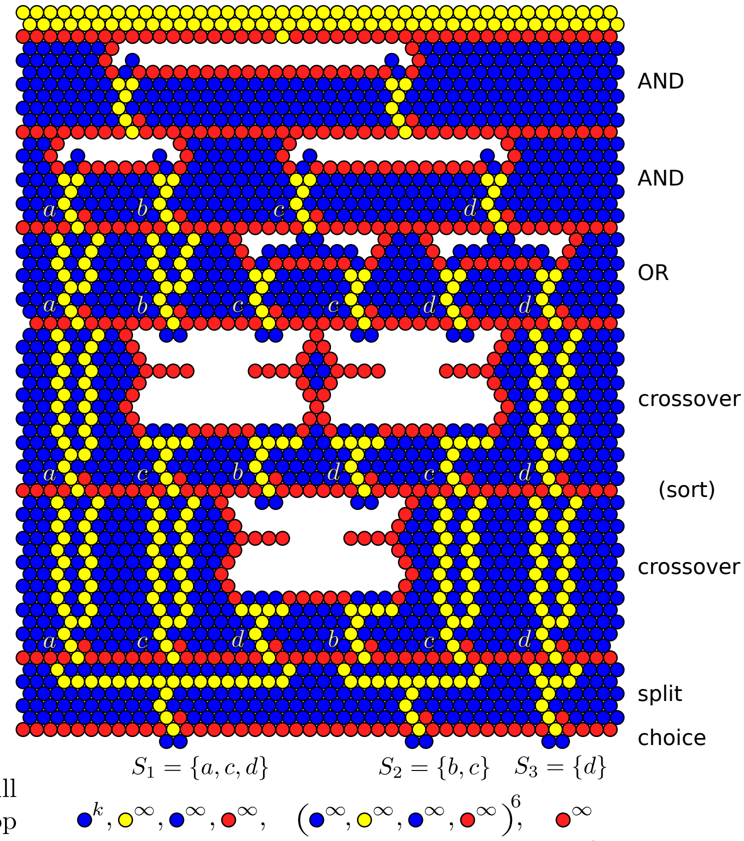

Figure 2: Example of the main construction (the gray box in Figure 1) with three sets and four elements. The sequence at the bottom can solve the puzzle for $k = 2$ and $k = 3$, but not for $k = 1$.

The rough idea is the following. Red bubbles separate vertical layers that unravel sequentially, as enforced by blue buffers. Blue and yellow bubbles form triggers to communicate signals into the next layers, alternately. Blue triggers cup yellow triggers in the next level, and vice versa.

Figure 2 illustrates the gadgets in a full example. First (at the bottom) we have one instance of the choice gadget, which allows triggering k sets (whichever the player chooses). Then we use several split gadgets, to split each trigger for set S_i into $|S_i|$ triggers. Then we use several crossover gadgets, to bring together all the triggers for element x, for every element x. Next, for each element, we merge all the triggers for that element (coming from sets that contain the element), using the OR gadget. Finally, we combine the element triggers using the AND gadget. We end up with one trigger indicating that all elements are covered, i.e., we found a set cover of size k. This trigger is connected to a huge ($n^{1-\varepsilon}$-area) rectangle of yellow bubbles at the top of the board, with one red bubble in the middle, as shown in Figure 1. Thus, even approximating the maximum number of poppable bubbles better than a factor of $n^{1-\varepsilon}$ is NP-hard (as in Tetris [1]).

See the full paper [2] for omitted details.

3 Open Problems

We have proved NP-hardness for just three colors. What about just two colors? Or even one color?

Acknowledgments. We thank Giovanni Viglietta for helpful discussions, in particular for pointing out bugs in earlier versions of this proof.

References

[1] R. Breukelaar, E. D. Demaine, S. Hohenberger, H. J. Hoogeboom, W. A. Kosters, and D. Liben-Nowell. Tetris is hard, even to approximate. *Int. J. Comput. Geom. Appl.*, 14(1–2):41–68, 2004.

[2] E. D. Demaine and S. Langerman. Bust-a-Move/Puzzle Bobble is NP-complete. arXiv: 1506.08409, 2015.

[3] The International Arcade Museum. Bubble Bobble. http://www.arcade-museum.com/game_detail.php?game_id=7222.

[4] The International Arcade Museum. Puzzle Bobble. http://www.arcade-museum.com/game_detail.php?game_id=9169.

[5] G. Viglietta. Gaming is a hard job, but someone has to do it! In *Proc. 6th Int. Conference on Fun with Algorithms*, pages 357–367, 2012.

[6] Wikipedia. Puzzle Bobble. http://en.wikipedia.org/wiki/Puzzle_Bobble.