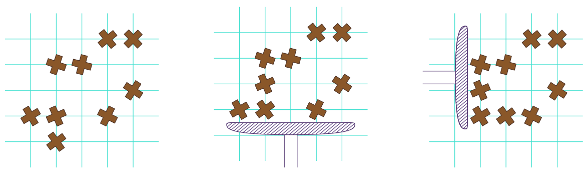
Compaction Games

Hugo Akitaya[1], Maarten Löffler[2], Anika Rounds[3], and Giovanni Viglietta[4]

[1]University of Massachusetts Lowell, MA, USA. hugo_akitaya@uml.edu
[2]Utrecht University, the Netherlands. m.loffler@uu.nl
[3]Amazon, Seattle, WA, USA. anika.rounds@gmail.com
[4]JAIST, Japan. johnny@jaist.ac.jp

1 Introduction

Let P be a set of n objects on a square grid. A *push* is a transformation of P that sweeps a horizontal or vertical line by one unit, starting from the hull of P. The sweep displaces objects that are in contiguous positions. For example, when pushing to the right, all the leftmost objects are displaced one unit to the right. This in turn displaces other objects further right. Refer to Figure 1.



(a) (b) (c)

Fig. 1. An example push sequence.

Given P, we look for a sequence of pushes that will produce a desired configuration. Akitaya *et al.* [1] showed that deciding whether P can be reconfigured into a square is NP-complete. In this note we introduce two *game* versions of the problem.

2 Game 1: LAST-MOVE-WINS

In *compaction games*, two players take turns pushing; in LAST-MOVE-WINS, the last player to make a valid move wins the game. We need one additional rule:

Rule 1. *A push that does not reduce the size of the bounding box of P is not allowed.*

Without Rule 1, the game never ends. With Rule 1 in place, there is a simple upper bound on the number of moves that can be made: the minimum number of empty cells in any row (within the bounding box) plus the minimum number of empty cells in any column. We call this number the *potential*. For example, the configuration in Figure 1(a) has potential 4. Figure 2 shows an endgame configuration, where the potential is 0.

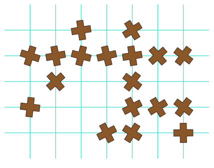


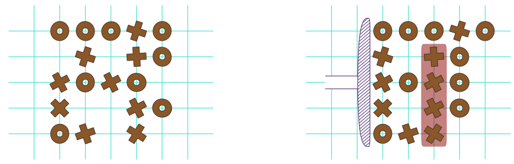
Fig. 2. An endgame situation in LAST-MOVE-WINS.

The potential is not trivial to control. However, trying to make it an even number seems to be a good strategy.

Open Problem 2. *Is* LAST-MOVE-WINS *PSPACE-complete?*

3 Game 2: k-IN-A-ROW

In this variant, the input is partitioned into two sets: $P = X \cup O$. The goal of player 1 is to arrive at a configuration where there are k objects from X on k consecutive grid points (horizontally or vertically), whereas the goal of player 2 is to get k objects from O on k consecutive grid points. When both conditions are met simultaneously, the game ends in a tie. Figure 3 shows a possible winning move in this game.



(a) (b)

Fig. 3. (a) An example configuration in a 4-IN-A-ROW compaction game. (b) A winning move for player 1.

k-IN-A-ROW is playable with or without Rule 1. Without the rule, playing optimally may result in infinitely long games, as shown in Figure 4. In this 2-IN-A-ROW game, the only non-losing move for player 1 is to push to the left, and vice versa. This causes the game to "oscillate" between the two indicated configurations; it would seem reasonable to declare such a situation a tie.

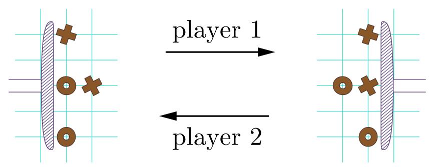


Fig. 4. Under optimal play, this 2-IN-A-ROW game never ends.

It is not clear whether similar examples with larger periods exist.

Open Problem 3. *Is it possible for a k-IN-A-ROW game (without Rule 1) to necessarily result in a loop of period greater than 2, under optimal play?*

Note that k-IN-A-ROW is only meaningful when $k \geq 2$, and already for $k = 2$ the game is interesting.

Open Problem 4. *Is 2-IN-A-ROW PSPACE-complete?*

Acknowledgments. This work was initiated at the 30th Bellairs Winter Workshop on Computational Geometry.

References

[1] H. Akitaya, G. Aloupis, M. Löffler, and A. Rounds. Trash compaction. In *Proc. 32nd European Workshop on Computational Geometry*, pages 107–110, 2016.