EVALUATING RECTANGULAR DOMINEERING GAMES

MIHIR KOTBAGI

ABSTRACT. In this paper, we provide an exposition of some important theorems in the study of rectangular DOMINEERING games. We additionally discuss some of the computational methods that have been used to evaluate larger boards and propose techniques that could advance current algorithms.

1. INTRODUCTION

DOMINEERING is a normal-play partizan game frequently studied by combinatorial game theorists. It consists of a board that is some subset of a square lattice that two players place 2×1 dominoes on. The left player's dominoes are oriented vertically, and the right player's dominoes are oriented horizontally. These players place their dominoes until one of them cannot make a move; whoever can't move loses the game. All DOMINEERING games fall into one of the following outcome classes: $\mathcal{L}$, $\mathcal{R}$, $\mathcal{N}$, or $\mathcal{P}$.

This article discusses rectangular DOMINEERING games and presents an exposition of some theorems relevant for determining their outcomes. There currently exist no general formulas for determining the outcome classes of rectangular DOMINEERING games, so most research into larger boards has focused on developing algorithms that can compute their outcome classes. Consequently, we also describe existing computational approaches to evaluating DOMINEERING games and propose future research in that area, inspired by similar work on chess and Go.

2. EVALUATING RECTANGULAR BOARDS

Definition 2.1. $m \times n$ refers to a rectangular grid with m rows and n columns where m, n are nonnegative integers.

Definition 2.2. $[m \times n]$ refers to the value of a DOMINEERING game played on a $m \times n$ board.

To begin evaluating DOMINEERING boards, we have to come up with some way to use known results for smaller boards to compute the outcome classes of larger boards. It intuitively makes sense that we should be able to do this, as each move in DOMINEERING shrinks the available playing area and effectively transposes to a new board.

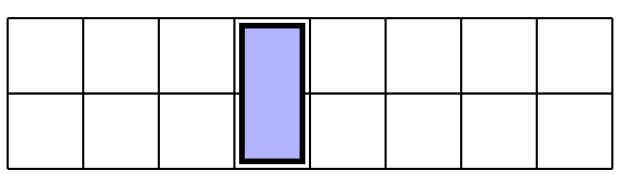

Figure 1. Splitting the 2×8 game into a 2×3 game and a 2×4 game.

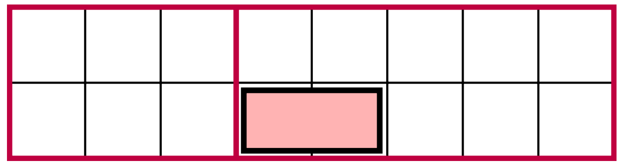

Figure 2. Splitting the 2×8 game into a 2×3 game and a 2×5 game.

We start by considering games with 2 rows to motivate an important observation regarding Domineering on rectangular boards. The following theorems were proven in [LMR00].

Theorem 2.3 (Vertical Board Splitting theorem)**.** *$[2 \times (m+n+1)] \,\|\!> 0$ if $[2 \times m] \geq 0$ and $[2 \times n] \geq 0$ for some $m, n \in \mathbb{N}$.*

Proof. Left is able to split the board into two when they move first, as seen in figure 1. This means that, if the width equals $m + n + 1$, left can effectively play on a $2 \times m$ board and a $2 \times n$ board. If $[2 \times a] \in \mathcal{L}$ and $[2 \times b] \in \mathcal{L}$, we clearly have that left wins in $[2 \times n]$. If one is $\mathcal{L}$ and the other is $\mathcal{P}$, left wins by responding in the $\mathcal{P}$ section after winning the $\mathcal{L}$ game. Finally, if both are $\mathcal{P}$, left wins by responding in both sections. ■

Theorem 2.4 (Horizontal Board Splitting theorem)**.** $[2 \times (m+n)] \leq [2 \times m] + [2 \times n]$

Proof. We now consider how right can choose to split (or not split) the board when they move first. Left can very explicitly divide a $2 \times n$ board into distinct sections, and right can do the same, albeit in a somewhat less obvious way. Note that in figure 2, right has transformed the 2×8 board into the sum of a 2×3 board and 2×5 board. Left is unable to cross the border between these two boards. Thus, right can play the $2 \times (m+n)$ board as if it were the sum of the $2 \times m$ and $2 \times n$ boards. If $[2 \times m] \in \mathcal{N}$ and $[2 \times n] \in \mathcal{P}$ or $\mathcal{R}$ (or vice versa), right wins by first winning the $\mathcal{N}$ section and the responding in the other board. If one section is $\mathcal{R}$ and the other is $\mathcal{R}$ or $\mathcal{P}$ right clearly wins. The only situations where right doesn't necessarily win going first are when both sections are $\mathcal{N}$ or $\mathcal{P}$.

Table 1 provides an overview of the aforementioned outcomes.

$[2 \times (m+n)]$	$\mathcal{N}$	$\mathcal{P}$	$\mathcal{R}$
$\mathcal{N}$	$\mathcal{N}, \mathcal{P}, \mathcal{L}, \mathcal{R}$	$\mathcal{N}, \mathcal{R}$	$\mathcal{N}, \mathcal{R}$
$\mathcal{P}$	$\mathcal{N}, \mathcal{R}$	$\mathcal{P}, \mathcal{R}$	$\mathcal{R}$
$\mathcal{R}$	$\mathcal{N}, \mathcal{R}$	$\mathcal{R}$	$\mathcal{R}$

Table 1. $[2 \times (m+n)]$ depending on $[2 \times m]$ and $[2 \times n]$

Theorem 2.4 simply results from summarizing this table. ■

Because of theorem 2.4, we can state that right benefits by splitting the board (crossing a vertical boundary).

The generalized form of 2.4 (e.g. for boards that don't have 2 rows) presented below was proven by Lachmann et al., but its proof is omitted here [LMR00]. We include it due to its usefulness in extending the values of rectangular boards.

Theorem 2.5 (General Board Splitting theorem)**.**

$$[m \times (n_1 + n_2)] \leq [m \times n_1] + [m \times n_2]$$

$$[(m_1 + m_2) \times n] \geq [m_1 \times n] + [m_2 \times n]$$

3. Evaluating Square Boards

We now consider the more specific case of square DOMINEERING games.

Definition 3.1. A square DOMINEERING game is a DOMINEERING game played on an $n \times n$ board.

It is known that such games are always either $\mathcal{N}$ or $\mathcal{P}$ due to the fact that the grid is diagonally symmetric (the effect of placing a vertical domino in some position is the same as the effect of placing a horizontal domino there). In other words, $[n \times n] + [n \times n] = 0$ for all n.

Remark 3.2. The problem of determining the outcome classes for $n \times n$ boards for $n > 5$ is also given in the CGT Book as problem 22 in chapter 5.

It is currently conjectured that all unsolved square games are $\mathcal{N}$, as all $[n \times n]$ games with $6 \leq n \leq 11$ have been found to be $\mathcal{N}$ (only $n = 1$ and $n = 5$ are second-player wins, and the outcomes of boards up to $n = 11$ have been computed).

Theorem 3.3 is particularly useful in propagating the values of square boards, and it was proven in [LMR00].

Theorem 3.3 (Square Addition theorem)**.**

$$\text{If } [n \times n] \in \mathcal{N}, \text{ then } [n \times kn] \in \begin{cases} \mathcal{P} \text{ or } \mathcal{R} & \text{for even } k > 1, \\ \mathcal{N} \text{ or } \mathcal{R} & \text{for odd } k > 1, \end{cases}$$

$$\text{If } [n \times n] \in \mathcal{P}, \text{ then } [n \times kn] \in \mathcal{P} \text{ or } \mathcal{R} \text{ for all } k > 1.$$

Proof. To prove the case where $[n \times n] \in \mathcal{N}$, we utilize the fact that $[n \times n] + [n \times n] = 0$ for all n. From this, we can state that combining an even number $[n \times n]$ boards has an outcome class of $\mathcal{P}$. Combining this with theorem 2.5, we have that $[n \times kn] \leq 0$ for even $k > 1$. Thus, $[n \times kn] \in \mathcal{P}$ or $\mathcal{R}$ when k is even. When k is odd, $[n \times kn] \leq [n \times n]$, so $[n \times kn] \in \mathcal{N}$ or $\mathcal{R}$.

For the case where $[n \times n] \in \mathcal{P}$, we have that $[n \times kn] \leq 0$ for all k using the same reasoning as above. Thus, $[n \times kn] \in \mathcal{P}$ or $\mathcal{R}$ for all $k > 1$ when $[n \times n] \in \mathcal{P}$. ■

Remark 3.4. The analogous statements for $[kn \times n]$ DOMINEERING boards also hold due to symmetry.

As an example of how theorems 2.5 and 3.3 can be applied, consider the 5×10 board in figure 3. Theorem 3.3 states that, since $[5 \times 5] \in \mathcal{P}$, $[5 \times 10] \in \mathcal{P}$ or $\mathcal{R}$. We also know that $[5 \times 4] \in \mathcal{R}$ and $[5 \times 6] \in \mathcal{R}$ and, so by theorem 2.5, $[5 \times 10] \in \mathcal{N}$ or $\mathcal{R}$. The only outcome that satisfies both of the above statements is $\mathcal{R}$, so we must have that $[5 \times 10] \in \mathcal{R}$.

4. Computing Game Outcomes Algorithmically

The outcome classes for larger DOMINEERING boards have largely been determined algorithmically due to the difficulty of extending rules by hand (e.g. $[10 \times 10] \in \mathcal{N}$ and $[11 \times 10] \in \mathcal{R}$ have been found using computers, but there's no straightforward way to compute $[21 \times 10]$ from that) [Col14, Uit16]. Primarily brute-force approaches sufficed for boards up to 8×9, but they were unsuitable for larger boards [BUv00]. Breuker et al.'s program DOMI included some basic rules, but it still had to search millions of nodes for even relatively small boards (e.g. over 4.4×10^8 nodes for the 8×8 board).

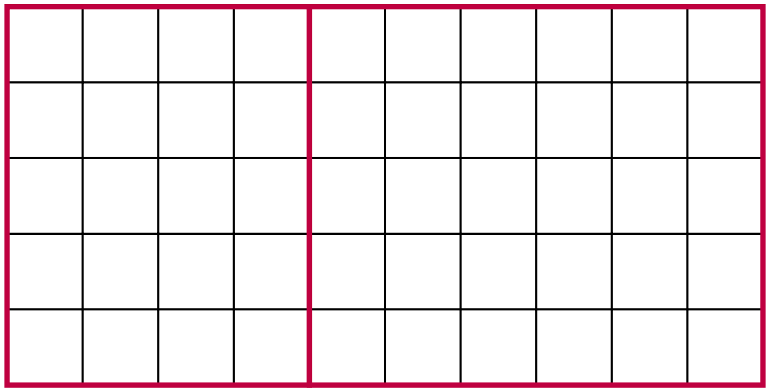

Figure 3. The 5×10 board can be thought of as a combination of the 5×4 and 5×6 boards.

More advanced rules-based approaches enabled the outcome classes for boards up to 11×11 to be determined, although still not particularly quickly [Bul02, UB15, Uit14]. To optimize these programs, α-β pruning, transposition tables, and more sophisticated game heuristics were implemented.

These techniques have resulted in programs such as MUDoS (Maastricht University Domineering Solver) being able to compute the outcomes of large boards much faster [Uit16]. MUDoS only needed to check 2.4×10^4 nodes for the 8×8 board, making it nearly 20,000 times faster than Domi.

Remark 4.1. The outcomes of many larger boards than those mentioned above (e.g. 3×150, 9×30) have been computed, but most research has focused on determining the outcome classes of roughly square boards. Propagating the theorems discussed in sections 2 and 3 prove to be particularly effective for $m \times n$ boards where $m \gg n$ (or vice versa).

5. Future Research

From a theoretical perspective, the aforementioned techniques are remarkably similar to those used for evaluating chess and Go positions. As these games have been studied far more, we draw inspiration from them to propose future areas of investigation.

When evaluating a combinatorial game like Domineering or chess, computation time depends on the number of nodes (positions) searched and on how fast those nodes can be evaluated. We suggest that Monte Carlo tree search and reinforcement learning could enable future programs to reduce the number of nodes searched by dramatically reducing how many nodes need to be considered [WBS10, SHS+18, GKS+12].

Furthermore, we propose that convolutional neural networks could be used to solve larger boards, as each move in Domineering on a $m \times n$ board can be thought of as flipping the values of two pixels. This means that the representation of the board as an array of bits changes smoothly from one move to the next. Convolutional neural networks have succeeded in other combinatorial games, and Domineering's structure appears to be particularly conducive to their implementation.

References

[Bul02] Nathan Bullock. Domineering: Solving large combinatorial search spaces. Master's thesis, University of Alberta, 2002.

[BUv00] D.M. Breuker, J.W.H.M. Uiterwijk, and H.J. van den Herik. Solving 8×8 domineering. *Theoretical Computer Science*, 230(1):195–206, 2000.

[Col14] Gabriel C Drummond Cole. An update on domineering on rectangular boards. *Integers*, 14:G3–1, 2014.

[GKS+12] Sylvain Gelly, Levente Kocsis, Marc Schoenauer, Michèle Sebag, David Silver, Csaba Szepesvári, and Olivier Teytaud. The grand challenge of computer go: Monte carlo tree search and extensions. *Commun. ACM*, 55(3):106–113, March 2012.

[LMR00] Michael Lachmann, Cristopher Moore, and Ivan Rapaport. Who wins domineering on rectangular boards?, 2000.

[SHS+18] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, Timothy Lillicrap, Karen Simonyan, and Demis Hassabis. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 362(6419):1140–1144, 2018.

[UB15] Jos W.H.M. Uiterwijk and Michael Barton. New results for domineering from combinatorial game theory endgame databases. *Theoretical Computer Science*, 592:72–86, 2015.

[Uit14] J.W.H.M. Uiterwijk. Perfectly solving domineering boards. In *Computer Games: Workshop on Computer Games, CGW 2013*, volume 408 of *Communications in Computer and Information Science*, United States, 2014. Springer.

[Uit16] Jos W. H. M. Uiterwijk. 11 × 11 domineering is solved: The first player wins. In Aske Plaat, Walter Kosters, and Jaap van den Herik, editors, *Computers and Games*, pages 129–136, Cham, 2016. Springer International Publishing.

[WBS10] Mark H. M. Winands, Yngvi Bjornsson, and Jahn-Takeshi Saito. Monte carlo tree search in lines of action. *IEEE Transactions on Computational Intelligence and AI in Games*, 2(4):239–250, 2010.