Further Results on the Game of Synchronized Triomineering

By Tao Cao

A thesis submitted to
School of Information Science,
Japan Advanced Institute of Science and Technology,
in partial fulfillment of the requirements
for the degree of
Master of Information Science
Graduate Program in Information Science

Written under the direction of
Professor Hiroyuki Iida

February, 2012

Further Results on the Game of Synchronized Triomineering

By Tao Cao (1010034)

A thesis submitted to
School of Information Science,
Japan Advanced Institute of Science and Technology,
in partial fulfillment of the requirements
for the degree of
Master of Information Science
Graduate Program in Information Science

Written under the direction of
Professor Hiroyuki Iida

And approved by
Professor Hiroyuki Iida
Professor Akira Shimazu
Assistant Professor Kokolo Ikeda

February, 2012

Abstract

Keywords: Combinatorial Game, Synchronized Triomineering, decompose checkerboard.

Combinatorial Game Theory (CGT) is a branch of mathematics devoted to studying the optimal strategy in games defined as two-player, no random moves and zero-sum finite games with perfect information.

The idea of synchronism has been introduced recently in combinatorial games and so far it does not exist a general theory to study these games. The analysis of simple synchronized games is the first step toward this goal.

In Synchronized Combinatorial Games, both players play simultaneously, so it will be fair to both players. Cincotti *et al.* have been applied this idea to solved Combinatorial Games.

Synchronized Triomineering has been studied to understand if it is possible to create games with interesting outcome (G=VD, G=HD, G=VHD) in a rectangular board with triominoes. The remaining problem is to solve a general n by m rectangular board of Synchronized Triomineering using a mathematical approach.

In this thesis, the results of $n \times 7$ and $n \times 8$ checkerboard will be presented, and also with the $m + n = 13$ and $m + n = 14$ checkerboard situations. The main characteristic of the game is the possibility to decompose a board in many sub-boards making the analysis easier. To complete the study of Synchronized Triomineering and other games will help to get general theoretical results concerning synchronized combinatorial games.

In Chapter 1, we introduces the background and purpose of the game of Synchronized Triomineering, and the structure of this thesis.

In Chapter 2, we present the game of Triomineering and Synchronized Triomineering. We explain and analyze the relation and the differences between these two games in detail. Then we give some examples of the game of Synchronized Triomineering. At the end of this chapter, some results concering Synchronized Triomineering are presented.

In Chapter 3, the techniques of decomposing the checkerboard have been presented. we use this method to get the results with the $n \times 7$ and $n \times 8$ boards.

In Chapter 4, we use a program to get some results for the $m \times n$ boards with $m + n =$

13 and m + n = 14. And these experimental results will be also helpful to make the right hypothesis when we try to get general results for an arbitrary n × m board.

Finally, in Chapter 5, the conclusions of my research are presented.

Acknowledgement

First and foremost, I would like to express my deepest gratitude to my supervisor - professor Hiroyuki Iida, for his guidance through the course of study in JAIST, Japan. His invaluable advices and support gave me great help. My search could not be completed without his help.

Secondly, I would like to express my heartfelt gratitude to Assistant Professor Alessandro Cincotti. During my research, I got a lot of assistance from him, for his advice and constant guidance I finished my thesis successfully.

Thirdly, I would like to thank Associate Professor Kokolo Ikeda, for his encouragement and helpful comments.

I also would like to thank my seniors - Junichi Hashimoto who gave me a lot of useful advices and guidances not only in this thesis but also during my daily life in Japan.

Also I would like to thank my whole lab members, I spent a very interesting two years with you.

At last, I would show my gratitude to my family and all my friends, you gave me great courage to finish my research. I love you all!

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background and Purpose

Combinatorial Game Theory (CGT) is a branch of mathematics devoted to studying the optimal strategy in games defined as follow:

1. There are only two players who take turns moving alternately.
2. There are no random moves such as rolling dice or shuffling cards. So it is deterministic.
3. Both players know the result of his/her opponent's move. It is a game with perfect information.
4. The rules of the game ensure that it will end after a finite sequence of moves.
5. Under normal play the last player to move wins.

In Synchronized Combinatorial Games, both players play simultaneously, so it will be fair to both players. Nakamura et al. [10] proposed the idea of synchronized games in order to revive the solved games, Cincotti et al. [12] have been applied this idea to solved Combinatorial Games, Komori et al. [13] have been designed computer players for these games.

Synchronized Triomineering has been studied to understand if it is possible to create games with interesting outcome in a rectangular board with triominoes. In synchronized games players play simultaneously therefore it is impossible to establish the winner under perfect play. These interesting games never appear in the game of Synchronized Domineering where only games like G=V, G=H, and G=D seems to appear [1,7]. The remaining problem is to solve a general n by m rectangular board of Synchronized Triomineering using a mathematical approach.

In this research, the target will be to get some further results on Synchronized Triomineering. The study of the outcome classes and the techniques used in the analysis of the game of Synchronized Triomineering also can be used to other games.

The main characteristic of the game is the possibility to decompose a board in many sub-boards making the analysis easier. To complete the study of Synchronized Triomineering and other games will help to get general theoretical results concerning synchronized combinatorial games.

1.2 Structure of the Thesis

In Chapter 2 of this thesis we describe the game of Triomineering and Synchronized Triomineering. We explain and analyze the relation and differences between these two games in detail. At the end of this chapter, some results concerning Synchronized Triomineering are presented.

In Chapter 3, we solve the $n \times 7$ and $n \times 8$ checkerboard problems.

In Chapter 4, we use a computer program to do some experiments for the $m \times n$ boards with $m + n = 13$ and $m + n = 14$.

Finally, in Chapter 5, the conclusions of my research are presented.

Chapter 2

Synchronized Triomineering

This chapter presents the game of Synchronized Triomineering. Section 2.1 describes the game of Triomineering. Section 2.2 describes the game of Synchronized Triomineering. Section 2.3 gives some examples of the game of Synchronized Triomineering. Section 2.4 presents the results for Synchronized Triomineering played on small boards.

2.1 Triomineering

Around 1973, the game of Domineering was invented by Göran Andersson [7,8,9]. This is a two-player game. Rules are defined as follows:

a. In Domineering two players, usually denoted by Vertical and Horizontal, take turns in placing dominoes (2×1 tile) on a checkerboard.
b. Vertical is only allowed to place its dominoes vertically and Horizontal is only allowed to place its dominoes horizontally on the checkerboard.
c. Dominoes are not allowed to overlap and the first player that cannot find a place for one of its dominoes loses.

Berlekamp [2] have computed the results for $2 \times n$ board for odd n. The 8×8 board and many other small boards were recently solved by Breuker et al. [4] using a computer search. Then Lachmann et al. [11] solved the problem for boards of width 2, 3, 5, 7 and other specific cases. Finally, Bullock [5] solved the 10×10 board.

Blanco and Fraenkel [3] create a new combinatorial game in 2004, they substitute the domino by a "straight" 3×1 tile triomino. They call this game Triomineering. It has the same rules with Domineering. In Triomineering, the two players, Vertical and Horizontal, take turns in placing "straight" triominoes (3 × 1 tile) on a checkerboard. Triominoes are not allowed to overlap and the first player that cannot find a place for one of its triominoes loses.

2.2 Synchronized Triomineering

The idea of synchronized games has been introduced by Nakamura et al. [10] in order to study combinatorial games where players make their moves simultaneously. This concept of synchronism was also introduced in the game of Cutcake [8] and Maundy Cake [9] and Domineering [7].

The idea of synchronism has been introduced recently in combinatorial games and so far it does not exist a general theory to study these games. The analysis of simple synchronized games is the first step toward this goal.

In synchronized games, both players play simultaneously, therefore it does not exit any unfair advantage due to the turn to move [10].

As a result, in the synchronized versions of these games there exist no zero-games, i.e., games where the winner depends exclusively on the player that makes the second move. Moreover, there exists the possibility of a draw, which is impossible in a typical combinatorial game. In this work, we continue to investigate synchronized combinatorial games by focusing our attention on Triomineering [10].

In the game of Synchronized Triomineering, a general instance and the legal moves for Vertical and Horizontal are defined exactly in the same way as defined for the game of Triomineering. There is only one difference: Vertical and Horizontal make their legal moves simultaneously, therefore, triominoes are allowed to overlap if they have a 1 × 1 tile in common. We note that 1 × 1 overlap is only possible within a simultaneous move. At the end, if both players cannot make a move, then the game ends in a draw, else if only one player can still make a move, then he/she is the winner.

In Synchronized Triomineering, for each player there exist three possible outcomes:

Winning Strategy (*ws*): The player has a winning strategy (*ws*) independently of the opponent's strategy, or

Drawing Strategy (*ds*): The player has a drawing strategy (*ds*), i.e., he/she can always get a draw in the worst case, or

Losing Strategy (*ls*): The player has a losing strategy (*ls*), i.e., he/she does not have a strategy for winning or for drawing.

Table 2.1 shows all the possible cases.

Table 2.1: The possible cases in Synchronized Triomineering

	Horizontal *ls*	Horizontal *ds*	Horizontal *ws*
Vertical *ls*	G = V H D	G = H D	G = H
Vertical *ds*	G = V D	G = D	–
Vertical *ws*	G = V	–	–

It is clear that if one player has a winning strategy, then the other player has neither a winning strategy nor a drawing strategy. Therefore, the cases *ws* – *ws, ws* – *ds*, and *ds* – *ws* never happen.

As a consequence, if G is an instance of Synchronized Triomineering, then we have 6 possible legal cases:

- **G = D** if both players have a drawing strategy, and the game will always end in a draw under perfect play, or
- **G = V** if Vertical has a winning strategy, or
- **G = H** if Horizontal has a winning strategy, or
- **G = VD** if Vertical can always get a draw in the worst case, but he/she could be able to win if Horizontal makes a wrong move, or
- **G = HD** if Horizontal can always get a draw in the worst case, but he/she could be able to win if Vertical makes a wrong move, or
- **G = VHD** if both players have a losing strategy and the outcome is totally unpredictable.

2.3 Examples of Synchronized Triomineering

Here we will show some small instance of Synchronized Triomineering:

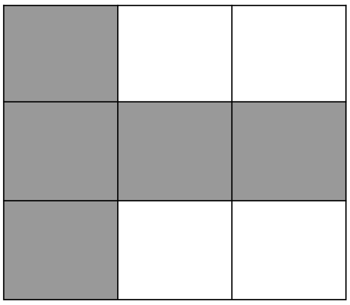


Figure 2.1 the example of G = D

The game in Figure 2.1 checkerboard always ends in a draw, therefore G = D.

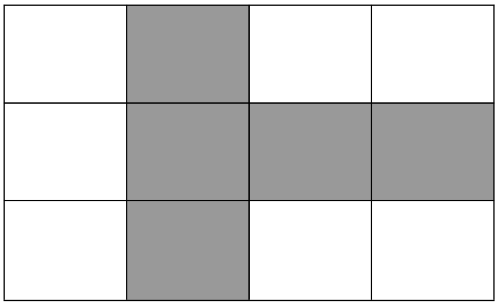


Figure 2.2 the example of G = V

In the game in Figure 2.2 checkerboard Vertical has a winning strategy moving in the second (or in the third) column, therefore G = V.

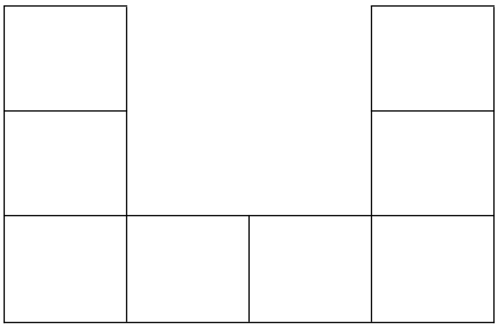
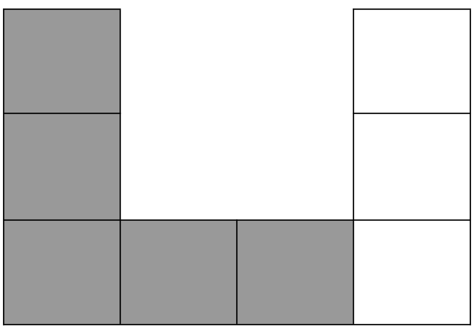


Figure 2.3(1) the example of G = VD

In the game in Figure 2.3(1), if Vertical moves in the first column, we have two possibilities:



or

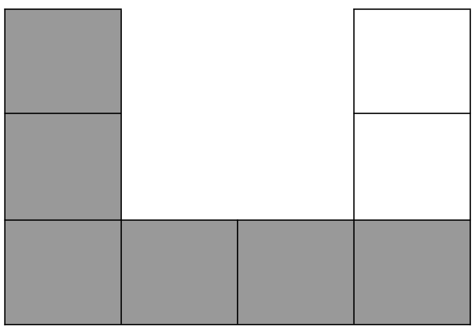
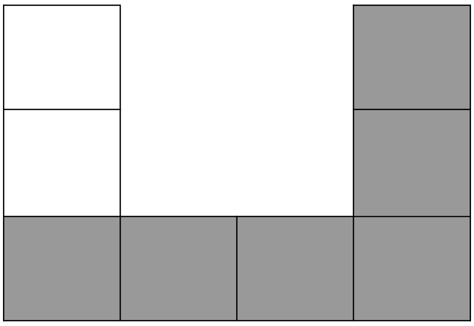


Figure 2.3(2) the example of G = VD

Therefore, either Vertical wins or the game ends in a draw. Symmetrically, if Vertical moves in the third column we have two possibilities:



or

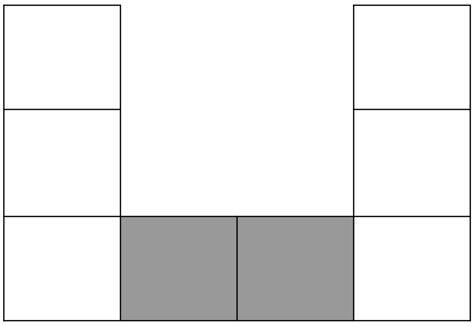


Figure 2.3(3) the example of G = VD

Therefore, either Vertical wins or the game ends in a draw. It follows G = VD.

Symmetrically, in the game:

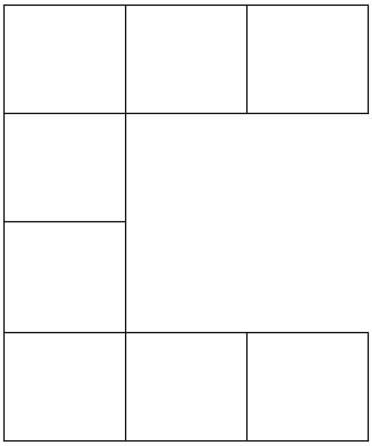


Figure 2.3(4) the example of G = VD

either Vertical wins or the game ends in a draw. It follows G = VD.

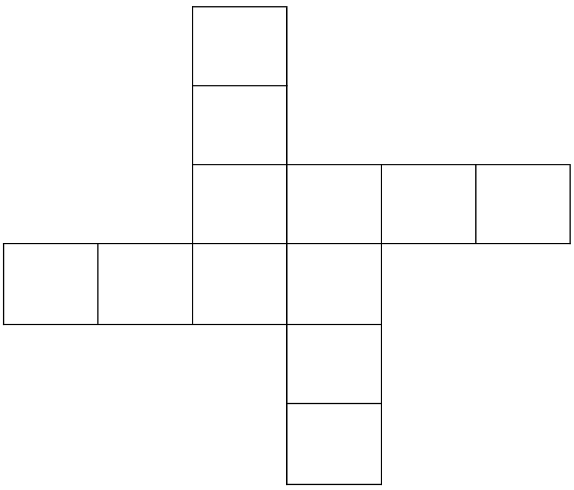


Figure 2.4 the example of G = VHD

In the game in Figure 2.4, each player has 4 possible moves. For every move of Vertical, Horizontal can win or draw (and sometimes lose); likewise, for every move by Horizontal, Vertical can win or draw (and sometimes lose). As a result it follows that G = VHD.

2.4 Results for Synchronized Triomineering

The results of this game for small checkerboard are presented in [12]. Table 2.2 shows the results for n × m boards with $n + m \leq 12$. In the same paper the general problems for n × 4 boards and n × 5 boards are solved [12].

Table 2.2 Results for Synchronized Triomineering on the n × m board [12]

	1	2	3	4	5	6	7	8	9	10	11
1	D	D	H	H	H	H	H	H	H	H	H
2	D	D	H	H	H	H	H	H	H	H	
3	V	V	D	V	V	D	V	V	D		
4	V	V	H	D	V	H	H	HD			
5	V	V	H	H	D	H	H				
6	V	V	D	V	V	D					
7	V	V	H	V	V						
8	V	V	H	VD							
9	V	V	D								
10	V	V									
11	V										

Chapter 3

Further Results on Synchronized Triomineering

In this chapter, the solution for the n × 7 and n × 8 boards problems are presented. This chapter is an update of

- T. Cao, A. Cincotti, H. Iida: New Results for Synchronized Triomineering, Proceeding of the International Multiconference of Engineer and Computer Scientists, March, 2012.

3.1 New Results of n × 7 Boards

In this section, we will give the proof of n × 7 board of this game.

Theorem 1: Let G be a n × 7 board Synchronized Triomineering with $n \geq 27$. Then, Vertical has a winning strategy.

Proof: In the beginning, Vertical always move into the third and the fifth column of the board, i.e., (k, c), (k + 1, c) and (k + 2, c); (k, e), (k + 1, e) and (k + 2, e), where $k \equiv 1 \pmod 3$, as shown in Figure 3.1.

When Vertical cannot move anymore into the third and the fifth column, we divide the main rectangle into 3 × 7 sub-rectangles starting from the top of the board (by using horizontal cuts). Of course, if $n \neq 0 \pmod 3$, then the last sub-rectangle will be of size either 1 × 7 or 2 × 7, and Horizontal will be able to make respectively either two more move or four more moves. The Figure 3.1 shows the Vertical strategy on n × 7 board.

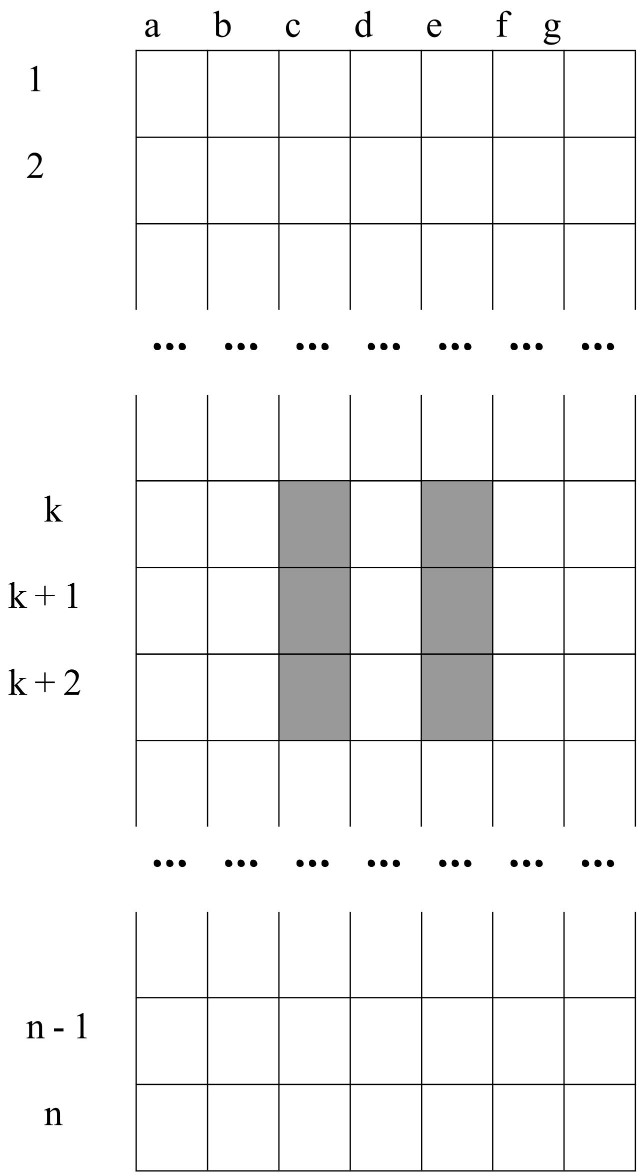


Figure 3.1 Vertical strategy on n × 7 board

We can classify all these sub-rectangles into 13 different classes according to:

• The number of vertical triominoes already placed in the sub-rectangle (vt),

• The number of horizontal triominoes already placed in the sub-rectangle (ht),

• The number of moves that Vertical is able to make in the worst case, in all the sub-rectangles of that class (vm),

• The number of moves that Horizontal is able to make in the best case, in all the sub-rectangles of that class (hm).

Table 3.1 shows all these 13classes.

Table 3.1 The 13 classes for 3 × 7 sub-rectangles

	vt	ht	vm	hn
A	2	0	5|A|	0
B	2	1	3|B|	0
C	2	2	|C|	0
D	1	1	2|D|+Ceil(|D|/2)	2|D|
E	1	2	Ceil(|E|/2)	2|E|
F	1	3	0	|F|
G	1	4	0	0
H	0	1	2|H|	2|H|
I	0	2	Ceil(|I|/2)	4|I|
J	0	3	0	3|J|
K	0	4	0	2|K|
L	0	5	0	|L|
M	0	6	0	0

The appendix 1 shows the examples of all these 13 classes.

When vertical cannot move anymore into the third and the fifth column, both Vertical and Horizontal have placed the same number of triominoes, therefore

$2|A| + 2|B| + 2|C| + |D| + |E| + |F| + |G| = |B| + 2|C| + |D| + 2|E| + 3|F| + 4|G| + |H| + 2|I| + 3|J| + 4|K| + 5|L| + 6|M|$

We can get the equation below:

$2|A| + |B| = |E| + 2|F| + 3|G| + |H| + 2|I| + 3|J| + 4|K| + 5|L| + 6|M|$……………………………………………………………………………....(1)

Then let us prove by contradiction that Vertical can make a large number of moves than Horizontal. Assume therefore *moves*(V) ≤ *moves*(H) using the data in Table 3.1.

$5|A| + 3|B| + |C| + 2|D| + \mathrm{Ceil}(|D|/2) + \mathrm{Ceil}(|E|/2) + 2|H| + \mathrm{Ceil}(|I|/2) \leq 2|D| + 2|E| + |F| + 2|H| + 4|I| + 3|J| + 2|K| + |L| + 4$

In this case the last sub-rectangle will be the size of 2 × 7, and Horizontal will be able to make 4 more moves.

And using equation (1),

$|A| + |B| + |C| + \mathrm{Ceil}(|D|/2) + \mathrm{Ceil}(|E|/2) + 3|F| + 6|G| + 2|H| + \mathrm{Ceil}(|I|/2) + 3|J| + 6|K| + 9|L| + 12|M| \leq 4$

We know that:

|A| + |B| + |C| + |D| + |E| + |F| + |G| + |H| + |I| + |J| + |K| + |L| + |M| = Floor(n/3)

Therefore when n ≥ 27, *moves*(V) ≤ *moves*(H) does not hold and consequently *moves*(V) > *moves*(H).

In particular when the size of the last sub-rectangle is 1 × 7, and Horizontal will be able to make 2 more moves and the inequality *moves*(V) > *moves*(H) holds for n ≥ 16.

When n ≡ 0(3), then *moves*(V) > *moves*(H) is satisfied for n ≥ 3.

3.2 New Results of n × 8 board

In this section, we will give the proof of n × 8 Board of this game.

Theorem 2: Let G be a n × 8 board Synchronized Triomineering with n ≥ 15. Then, Vertical has a winning strategy.

Proof: In the beginning, Vertical always move into the third and the sixth column of the board, i.e., (k, c), (k + 1, c) and (k + 2, c); (k, f), (k + 1, f) and (k + 2, f), where k ≡ 1 (mod 3), as shown in Figure 3.2.

When Vertical cannot move anymore into the third and the fifth column, we divide the main rectangle into 3 × 8 sub-rectangles starting from the top of the board (by using horizontal cuts). Of course, if n ≠ 0 (mod 3), then the last sub-rectangle will be of size either 1 × 8 or 2 × 8, and Horizontal will be able to make respectively either two more move or four more moves. The Figure 3.2 shows the Vertical strategy on n × 8 board.

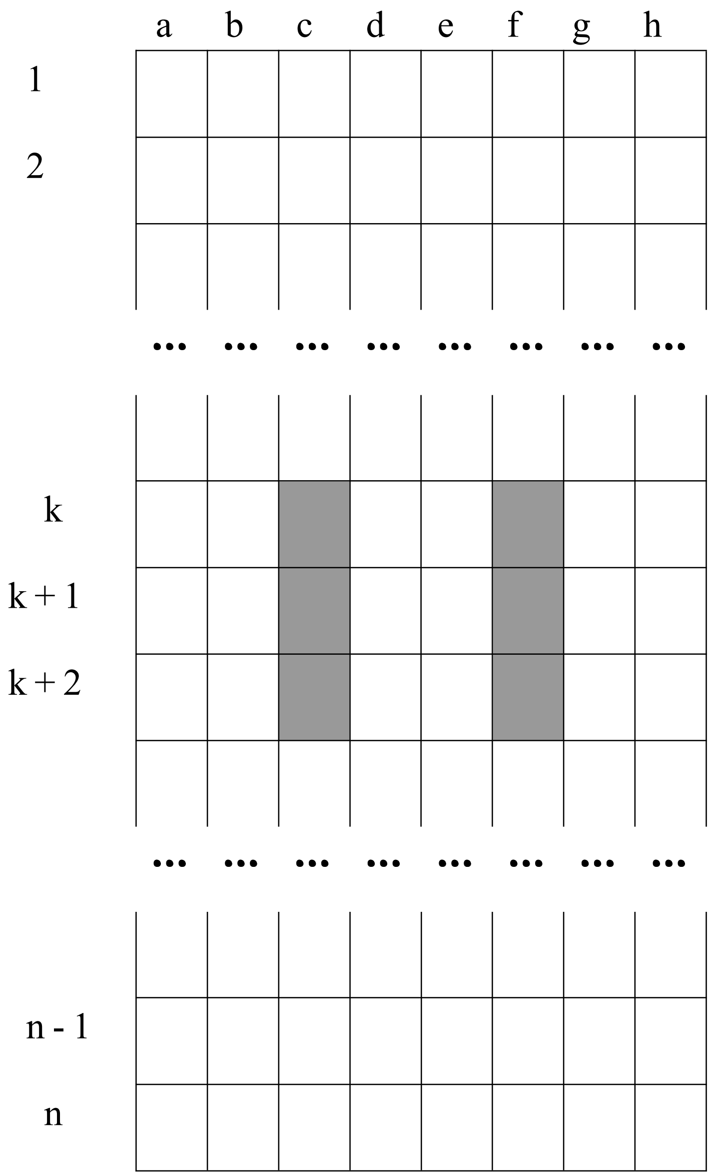


Figure 3.2 Vertical strategy on n × 8 board

In the same way we can also classify all these sub-rectangles into 12 different classes according to:

• The number of vertical triominoes already placed in the sub-rectangle (vt),

• The number of horizontal triominoes already placed in the sub-rectangle (ht),

• The number of moves that Vertical is able to make in the worst case, in all the sub-rectangles of that class (vm),

• The number of moves that Horizontal is able to make in the best case, in all the sub-rectangles of that class (hm).

Table 3.2 shows all these 12classes.

Table 3.2 The 12 classes for 3 × 8 sub-rectangles

	vt	ht	vm	hn
A	2	0	6|A|	0
B	2	1	4|B|	0
C	2	2	2|C|	0
D	1	1	3|D|	2|D|
E	1	2	|E|	2|E|
F	1	3	0	|F|
G	1	4	0	0
H	0	2	Ceil(3|H|/4)	4|H|
I	0	3	0	3|I|
J	0	4	0	2|J|
K	0	5	0	|K|
L	0	6	0	0

The appendix 2 shows the examples of all these 12 classes.

When vertical cannot move anymore into the third and the sixth column, both Vertical and Horizontal have placed the same number of triominoes, therefore

$2|A| + 2|B| + 2|C| + |D| + |E| + |F| + |G| = |B| + 2|C| + |D| + 2|E| + 3|F| + 4|G| + 2|H| + 3|I| + 4|J| + 5|K| + 6|L|$

We can get the equation below:

$2|A| + |B| = |E| + 2|F| + 3|G| + 2|H| + 3|I| + 4|J| + 5|K| + 6|L|$……………………….....(1)

Then let us prove by contradiction that Vertical can make a large number of moves than Horizontal. Assume therefore *moves*(V) ≤ *moves*(H) using the data in Table 3.2.

$6|A| + 4|B| + 2|C| + 3|D| + |E| + \text{Ceil}(3|H|/4) \leq 2|D| + 2|E| + |F| + 4|H| + 3|I| + 2|J| + |K| + 4$

In this case the last sub-rectangle will be the size of 2 × 8, and Horizontal will be able to make 4 more moves.

And using equation (1),

$2|A| + 2|B| + 2|C| + |D| + |E| + 3|F| + 6|G| + \text{Ceil}(3|H|/4) + 3|I| + 6|J| + 9|K| + 12|L| \leq 4$

We know that:

$|A| + |B| + |C| + |D| + |E| + |F| + |G| + |H| + |I| + |J| + |K| + |L| + |M| = \text{Floor}(n/3)$

Therefore when n ≥ 15, *moves*(V) ≤ *moves*(H) does not hold and consequently *moves*(V) > *moves*(H).

In particular when the size of the last sub-rectangle is 1×8, Horizontal will be able to make 2 more moves and the inequality *moves*(V) > *moves*(H) holds for $n \geq 10$.

When $n \equiv 0(3)$, then *moves*(V) > *moves*(H) is satisfied for $n \geq 3$.

By symmetry the following two theorems hold.

Theorem 3: Let G be a $7 \times n$ board Synchronized Triomineering with $n \geq 27$. Then, Horizontal has a winning strategy.

Theorem 4: Let G be a $8 \times n$ board Synchronized Triomineering with $n \geq 15$. Then, Horizontal has a winning strategy.

Chapter 4

The Experimental Results

In this chapter I will use an exhaustive search program to do some experiments. Results for small $n \times m$ boards with $n + m \leq 14$ are presented.

These experimental results are useful to make the right hypothesis when we try to get general results for an arbitrary n by m board.

The board 7×7 is still unsolved but a lot of games played on this board end in a draw.

Table 4.1 shows the new results for Synchronized Triomineering with $n + m \leq 14$.

The appendix 3 shows the exhaustive search program.

Table 4.1 New results for Synchronized Triomineering

	1	2	3	4	5	6	7	8	9	10	11
1	D	D	H	H	H	H	H	H	H	H	H
2	D	D	H	H	H	H	H	H	H	H	H
3	V	V	D	V	V	D	V	V	D	V	V
4	V	V	H	D	V	H	H	HD	H	H	
5	V	V	H	H	D	H	H	H	H		
6	V	V	D	V	V	D	V	V			
7	V	V	H	V	V	H	?				
8	V	V	H	VD	V	H					
9	V	V	D	V	V						
10	V	V	H	V							
11	V	V	H								

Chapter 5

Conclusion

The idea of synchronism has been introduced recently in combinatorial games and so far it does not exist a general theory to study these games. The analysis of simple synchronized games is the first step toward this goal. Then the game of Synchronized Triomineering has been proposed, therefore the study of the outcome classes and the techniques used in the analysis of Synchronized Triomineering will be very useful to deal with this kind of games.

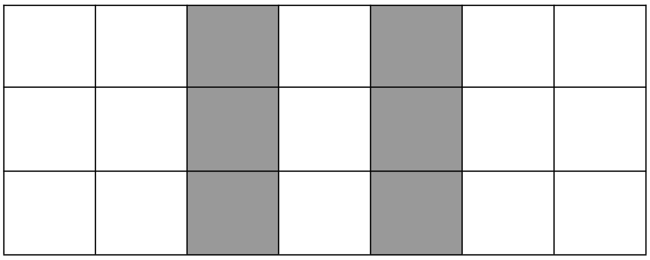
In Chapter 3, the techniques of decomposing the checkerboard have been presented. we not only got the results with the $n \times 7$ and $n \times 8$ boards, but also provide a method to deal with this kind of problems.

In Chapter 4, we use a program to get some results, and these experimental results will be also helpful to make the right hypothesis when we try to get general results for an arbitrary $n \times m$ board.

Synchronized Triomineering have been proposed recently to understand if it is possible to create games with interesting outcomes ($G = VD$, $G = HD$, $G = VHD$) in a rectangular board with triominoes, the remaining problem is to solve a general $n \times m$ rectangular board of Synchronized Triomineering using a mathematical approach.

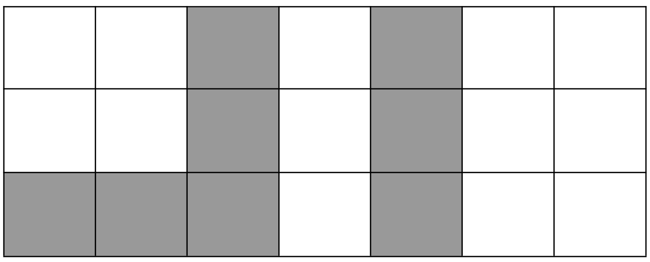
Appendix 1 Examples of classes for n × 7 board

Class A. Vertical is able to make five more moves in each sub-rectangle of this class.

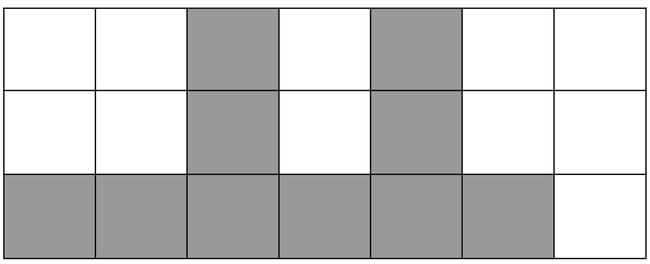


In this class there are 2 vertical triominoes already placed in the sub-rectangle, (vt = 2); no horizontal triominoes is placed in the sub-rectangle, (vh = 0); in each sub-rectangle Vertical is able to make 5 moves in the worst case. We denote with |A| the number of sub-rectangles in the A class. Therefore vm = 5|A|. Horizontal is not able to make any move, therefore hm = 0.

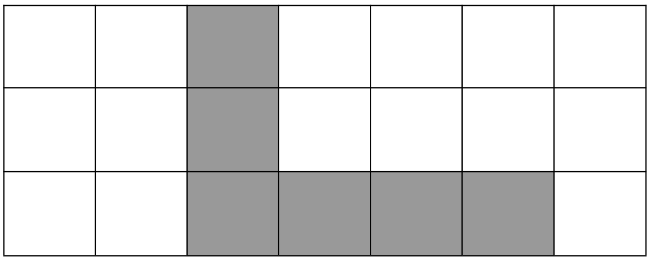
Class B. vt = 2; ht = 1; vm = 3|B|; hm = 0.



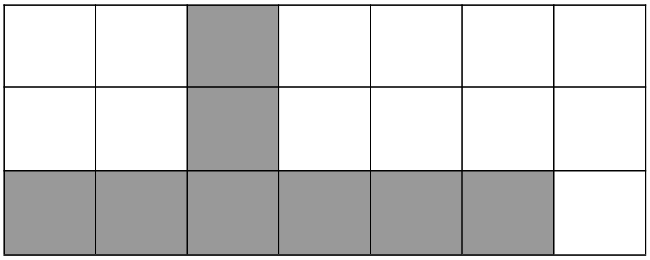
Class C. vt = 2; ht = 2; vm = |C|; hm = 0.



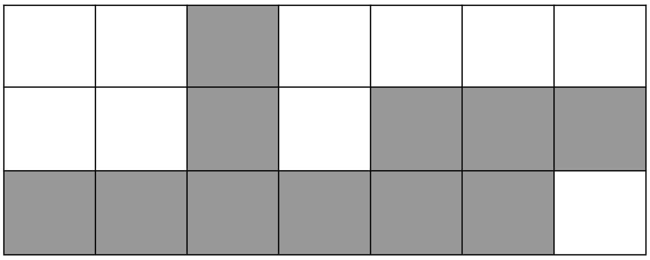
Class D. vt = 1; ht = 1; vm = 2|D| + Ceil(|D|/2); hm = 2|D|.



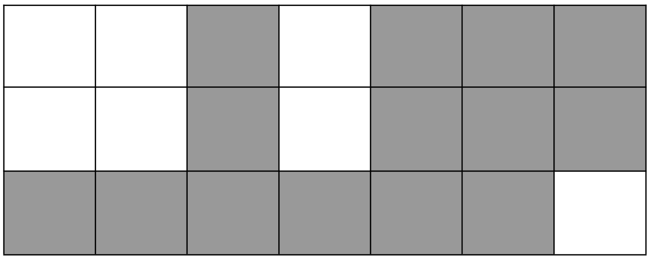
Class E. vt = 1; ht = 2; vm = Ceil(|E|/2); hm = 2|E|.



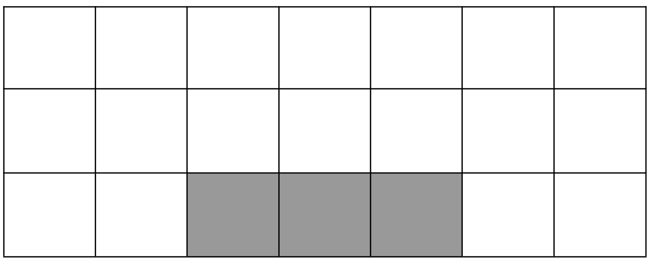
Class F. vt = 1; ht = 3; vm = 0; hm = |F|.



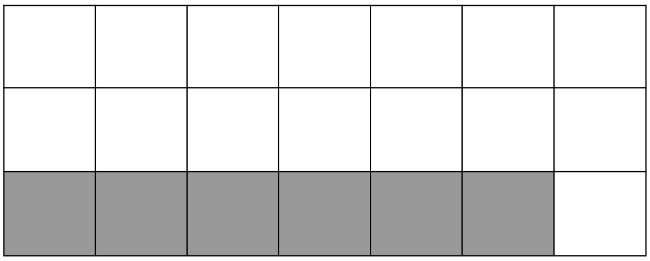
Class G. vt = 1; ht = 4; vm = 0; hm = 0.



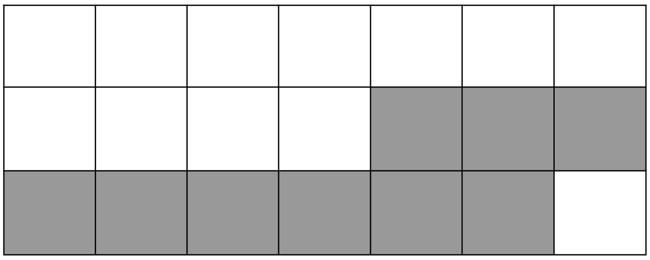
Class H. vt = 0; ht = 1; vm = 2|H|; hm = 2|H|.



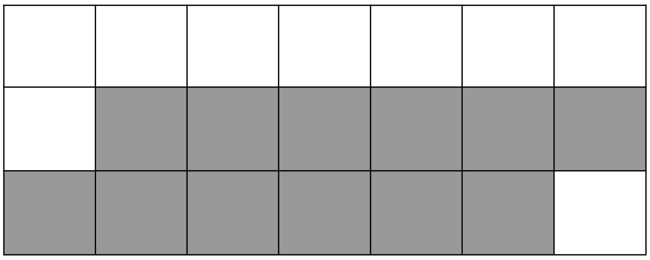
Class I. vt = 0; ht = 2; vm = Ceil(|I|/2); hm = 4|I|.



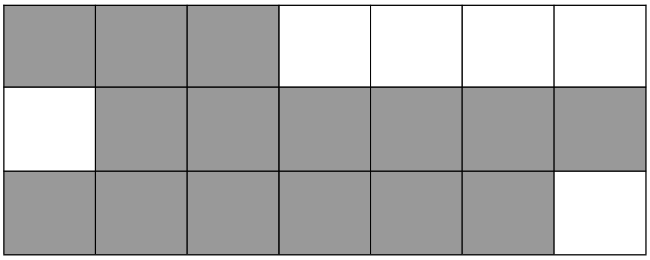
Class J. vt = 0; ht = 3; vm = 0; hm = 3|J|.



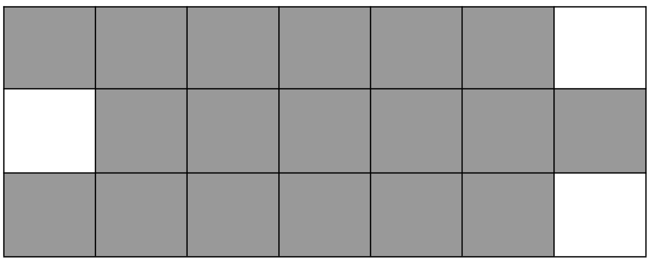
Class K. vt = 0; ht = 4; vm = 0; hm = 2|K|.



Class L. vt = 0; ht = 5; vm = 0; hm = |L|.

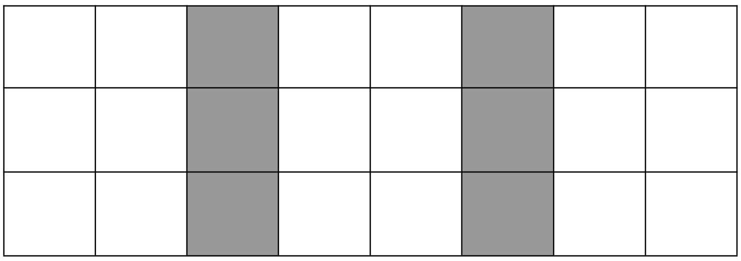


Class M. vt = 0; ht = 6; vm = 0; hm = 0.

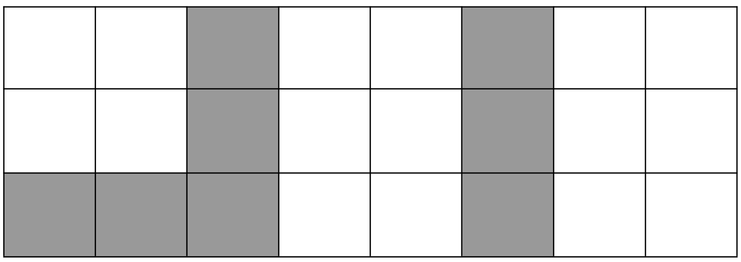


Appendix 2 Examples of classes for n × 8 board

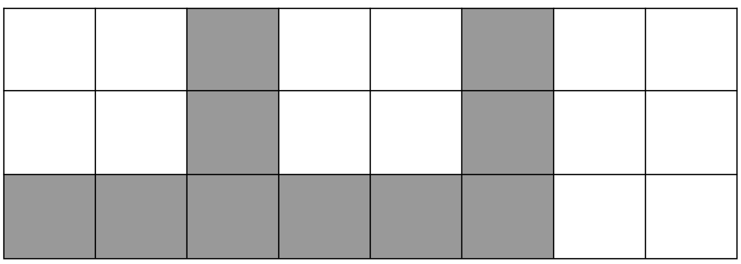
Class A: vt = 2; ht = 0; vm = 6|A|; hm = 0.



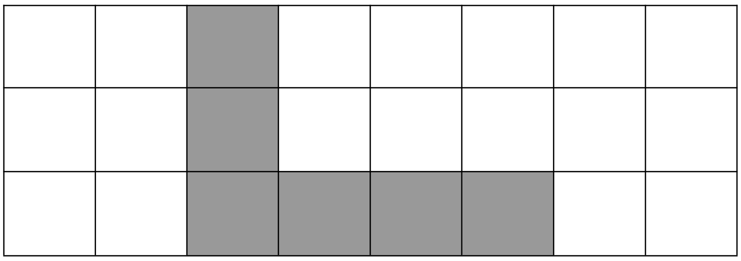
Class B: vt = 2; ht = 1; vm = 4|B|; hm = 0.



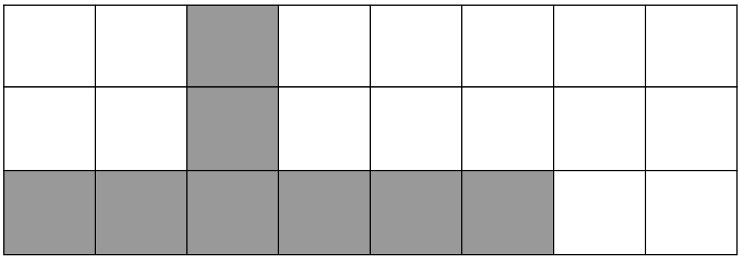
Class C: vt = 2; ht = 2; vm = 2|C|; hm = 0.



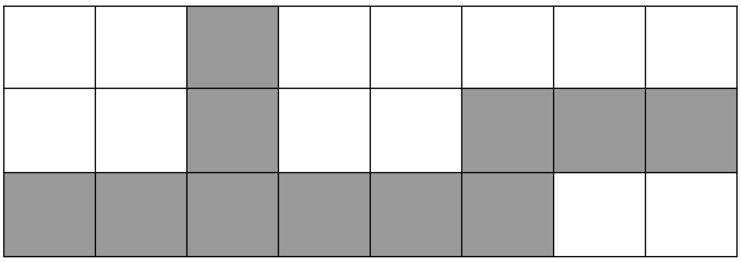
Class D: vt = 1; ht = 1; vm = 3|D|; hm = 2|D|.



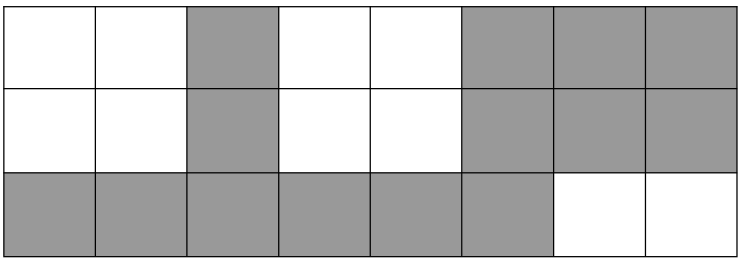
Class E: vt = 1; ht = 2; vm = |E|; hm = 2|E|.



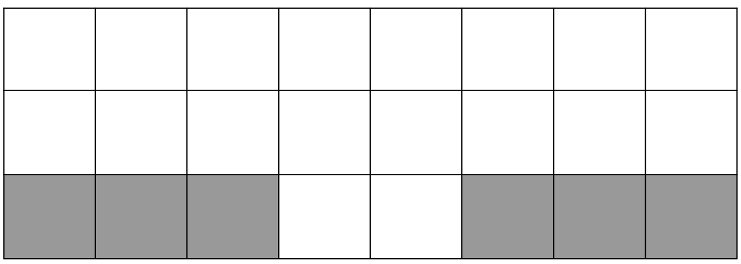
Class F: vt = 1; ht = 3; vm = 0; hm = |F|.



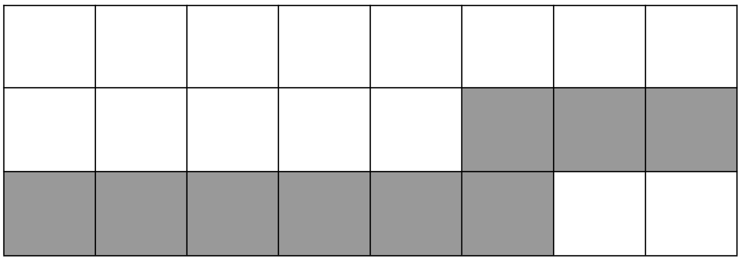
Class G: vt = 1; ht = 4; vm = 0; hm = 0.



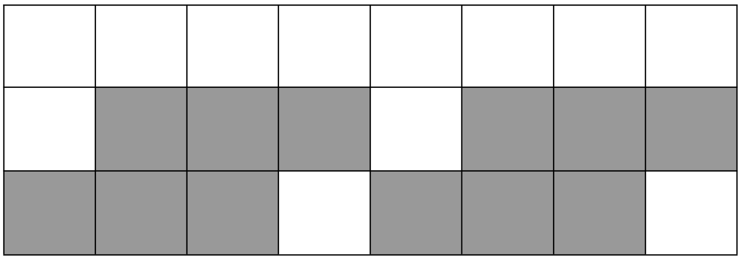
Class H: vt = 0; ht = 2; vm = Ceil(3|H|/4); hm = 4|H|.



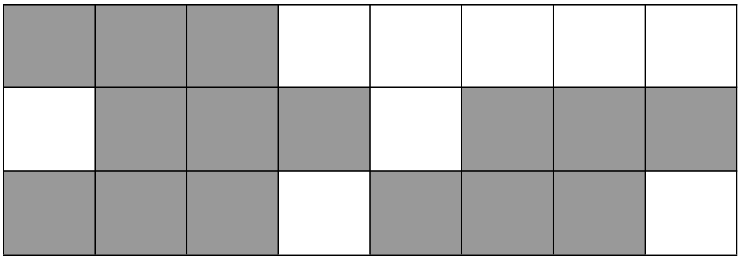
Class I: vt = 0; ht = 3; vm = 0; hm = 3|I|.



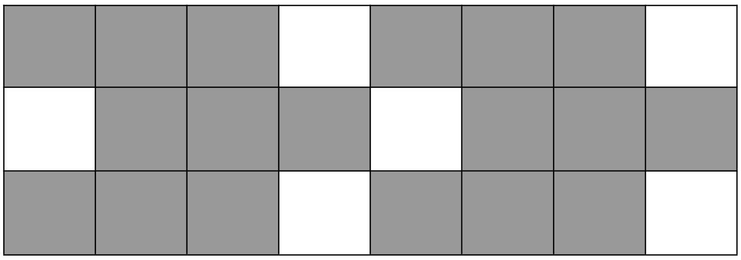
Class J: vt = 0; ht = 4; vm = 0; hm = 2|J|.



Class K: vt = 0; ht = 5; vm = 0; hm = |K|.



Class L: vt = 0; ht = 6; vm = 0; hm = 0.



Appendix 3 the search program of the game of Synchronized Triomineering

```
#include <stdio.h>
#include <stdlib.h>
#define R m
#define C n
short mat[R][C];
void clean() {
    int i,j;
    for (i=0; i<R; i++)
        for (j=0; j<C; j++)
        mat[i][j] = 0;
}
int compute(short mat[R][C]) {
    short mat1[R][C]; short res[R*C];
    int i, j, k,l, i1, j1, r, flag, t;
    r = -1;
    res[R*C-1] = -2;
    for (i=0; i<R*C-1; i++)
    res[i] = 2;
    for (i=0; i<R; i++)
        for (j=0; j<C-2; j++) {
            if ((mat[i][j] == 0) && (mat[i][j+1] == 0) && (mat[i][j+2] == 0))
        {
        r++;
        for (k=0; k<R-2; k++)
            for (l=0; l<C; l++)
```

```
            if ((mat[k][l] == 0) && (mat[k+1][l] == 0) && (mat[k+2][l] == 0))
            {
                for (i1=0; i1<R; i1++)
                for (j1=0; j1<C; j1++)
                    mat1[i1][j1] = mat[i1][j1];
                mat1[i][j] = 1;
                mat1[i][j+1] = 1;
                mat1[i][j+2] = 1;
                mat1[k][l] = 1;
                mat1[k+1][l] = 1;
                mat1[k+2][l] = 1;
                t = compute(mat1);
                            if (t < res[r]) {
                res[r] = t;
                }
            }
        if (res[r] > res[R*C-1]) {
            res[R*C-1] = res[r];
            if (res[R*C-1] >= 1) return 1;
            // H has a winning strategy
            //If res[R*C-1] == 2 then V does not have a legal move
          // else if (res[R*C-1] == 0) return 0;
            // Drawing strategy check
        }
        }
}
flag = 0;
for (k=0; k<R-2; k++)
for (l=0; l<C; l++)
    if ((mat[k][l] == 0) && (mat[k+1][l] == 0) && (mat[k+2][l] == 0))
    flag = 1;
if (r == -1) {
if (flag == 0) return 0;
// Neither H nor V has a legal move
else return -1;
```

```
    // H does not have a legal move but V has a legal move
    }
    else return res[R*C-1];
    // H and V have a legal move but H has not a winning strategy
}
int main() {
    int s;
    clean();
    s = compute(mat);
    if (s == 1)
    printf("\n H has a winning strategy.\n");
    else if (s == 0)
    printf("\n H has a drawing strategy.\n");
    else if (s == -1) printf("\n H has a losing strategy.\n");
    else printf("\n ERROR! \n");
    return 0;
```

References

[1] S. Bahri and C.P. Kruskal: New Solutions for Synchronized Domineering. In Proceedings of the Conference on Computers and Games 2010, 2011, pp. 211-229.

[2] E.R. Berlekamp: Blockbusting and Domineering. Journal of Combinatorial Theory Series A 49(1), 67-116, 1988.

[3] S.A. Blanco and A.S. Fraenkel: Triomineering, tridomineering, and L-tridomineering. Technical Report MCS04-10, Department of Computer Science and Applied Mathematics, Weizmann Institute of Science. Available: http://wisdomarchive.wisdom.weizmann.ac.il:81/archive/00000367/.

[4] D.M. Breuker, J.W.H.M. Uiterwijk, H.J. van den Herik: Solving 8 × 8 Domineering. Theoretical Computer Science 230(1-2), 195-206, 2000.

[5] N. Bullock: Domineering: Solving Large Combinatorial Search Spaces. ICGA Journal 25(2), 67-84, 2002.

[6] T. Cao, A. Cincotti, and H. Iida: New Results for Synchronized Triomineering. In Proceedings of the International Multiconference of Engineers and Computer Scientists March, 2012.

[7] A. Cincotti and H. Iida: The Game of Synchronized Domineering. In Proceedings of the Conference on Computers and Games 2008, 2008, pp. 241-251.

[8] A. Cincotti and H. Iida: The Game of Synchronized Cutcake. In Proceedings of the IEEE Symposium on computational Intelligence and Games, 2007, pp. 374-379.

[9] A. Cincotti and H. Iida: The Game of Synchronized Maundy Cake. In Proceedings of the 7th Annual Hawaii International Conference on Statistics, Mathematics and Related Fields, 2008, pp. 422-429.

[10] T. Nakamura, A. Cincotti, and H. Iida: The rebirth of solved games. In Proceedings of the 8th International Conference on Computer Science and Informatics, 2005, pp. 265-269.

[11] M. Lachmann, C. Moore, I. Rapaport: Who Wins Domineering on Rectangular Boards. In: R.J. Nowakowski, (ed.) More Games of No Chance, vol.42, pp. 307-315. Cambridge University Press, Cambridge, 2002.

[12] A. Cincotti, S. Komori, and H. Iida: The Game of Synchronized Triomineering and Synchronized Tridomineering. International Journal of Computational and Mathematical Sciences, 2(3), 143-148, 2008.

[13] S. Komori, Computer Player on Synchronized Games, 2009.