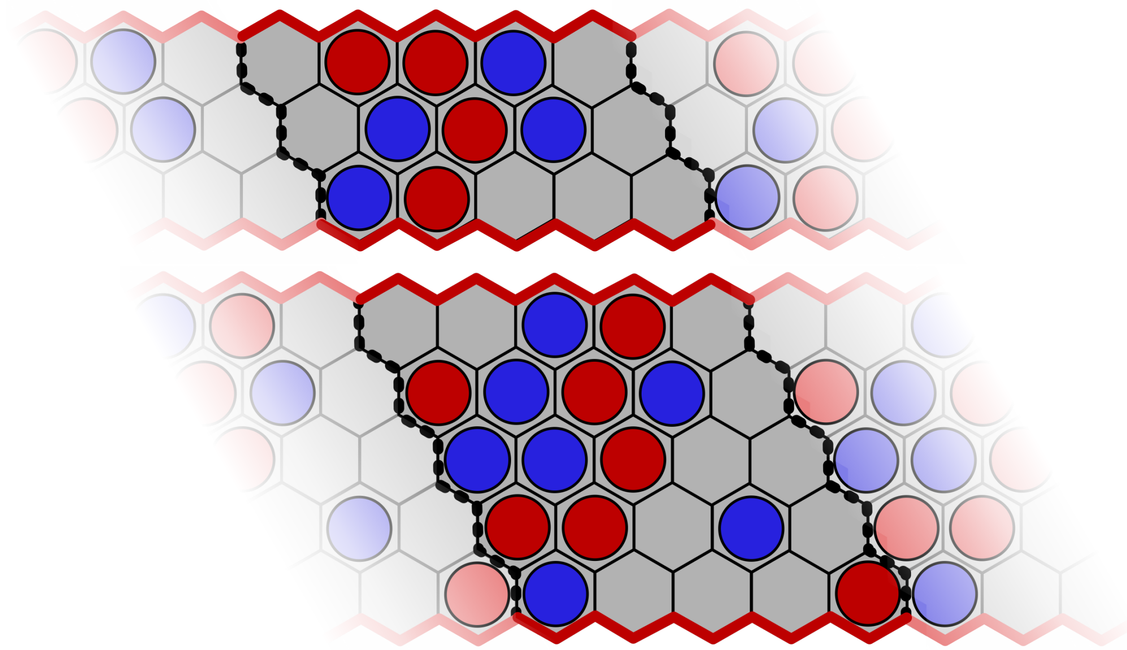
Cylindrical Hex and its Strategies: Introducing an Optimal Strategy for Red on m×n (>n) Boards, and an Improved Strategy for 5×n Boards
Fitzgerald, Kaitlyn

K.M. Fitzgerald

Cylindrical Hex and its Strategies

Introducing an Optimal Strategy for Red on $m \times n$ ($m > n$) Boards, and an Improved Strategy for $5 \times n$ Boards



Master thesis

January 12, 2026

Thesis supervisor: Prof. Dr. F.M. Spieksma

Leiden University
Mathematical Institute

Contents

Introduction

Hex may not be the most popular board game in the world, but it certainly is a well-loved one. It seems quite simple: Throughout the game, two players take turns placing coloured stones on the board — a parallelogram made of hexagonal cells — until one of them has created a path from one side of the board to the other — either horizontally or vertically, depending on the colour of their stones. Yet despite this simplicity, people have been looking for the best strategy to use ever since the game was first created in 1942.

In this thesis, we will mostly focus on a specific variation of Hex, called Cylindrical Hex: a version of the game in which the normally parallelogram-shaped board has been turned into a cylinder. Based on the known optimal strategies for this version of the game — which currently exist for boards that have either three, or an even number of columns — it is generally conjectured that the player whose goal it is to make a vertical path should be able to win on any non-trivial Cylindrical Hex board.

The initial goal of this thesis was to further improve an already existing, but not yet optimal strategy for Red on the first type of board for which no optimal strategy exists yet: the $5 \times n$ Cylindrical Hex board. The resulting algorithm is unfortunately also not quite optimal either. However, we will see that it does perform significantly better, having won 99.5% of the simulated games, compared to its predecessor's 87.08%. In the remaining 0.5% of games, the loss was always caused by the same type of Blue strategy, which we will also discuss in this thesis, along with some other ways in which one might further improve this algorithm in the future.

While finding a strategy for $5 \times n$ Cylindrical Hex was our main focus, we also became quite intrigued by Martin Gardner's strategy for asymmetrical Hex boards. After providing what seems to be the first rigorous proof of this strategy, we will also show that it can even be used by Red in $m \times n$ Cylindrical Hex if $m > n$. We will compare the performance of this new Cylindrical Hex strategy to that of a pre-existing optimal Cylindrical Hex strategy.

In more detail, the structure of this thesis is as follows.

We begin by briefly discussing the ways in which the game of Hex came to be in Chapter 1.

In Chapter 2 we will introduce the rules of Hex, as well as those of two variants of the game: Cylindrical and Torus Hex. For each variant, we also discuss how various aspects of the game can be represented mathematically. Additionally, we look which winning positions in one variation translate into winning positions in another.

Chapter 3 briefly discusses some known facts about the number of winners in a game of Hex.

In Chapter 4 we discuss several existing optimal strategies for various types of Hex and Cylindrical Hex boards. We pay special attention to the strategy for Hex on asymmetrical Hex boards suggested by Gardner [5] and provide a formal proof for this strategy. In addition, we briefly discuss which of the optimal Cylindrical Hex strategies discussed in this section could work in Torus Hex as well.

Chapter 5 introduces and proves a new optimal Cylindrical Hex strategy for Red on wide boards, based on the strategy suggested by Gardner [5]. We also discuss why this strategy will not work for Blue in Cylindrical Hex, or for either player in Torus Hex.

In Chapter 6 we look at various positions (namely, bridges, overlapping bridges, double-bridge set-ups and provisional paths) that may appear during a game of Hex. These positions will be useful when discussing the two algorithms introduced in Chapters 7 and 8.

Chapter 7 discusses the $5 \times n$ Cylindrical Hex strategy created by Van den Broek [13]. We analyse the ideas behind this strategy and the points where it can still be improved.

Chapter 8 introduces the new $5 \times n$ Cylindrical Hex algorithm that was created for this thesis based on the algorithm discussed in Chapter 7. The reasons behind the changes made to the earlier algorithm are explained in detail.

In Chapter 9 various algorithms discussed in this thesis are compared in terms of efficiency and, where relevant, effectiveness based on the results of simulated games. We also take a closer look at how the new algorithm for $5 \times n$ Cylindrical Hex operates in practice.

Finally, in Chapter 10 we briefly discuss a few ways in which the new $5 \times n$ Cylindrical Hex algorithm could still be improved, based on the shortcomings found during simulated games.

This thesis also has several appendices. Appendix A contains several lists of the moves made during particularly relevant simulated games. Appendix B gives an impression of which $m \times n$ Cylindrical Hex boards currently have a known optimal strategy. It contains a list of which strategies can be used on each board with $m, n \leq 10$, with the new strategies introduced in this thesis listed in bold.

Chapter 1

A Brief History of a Polygon-based Game

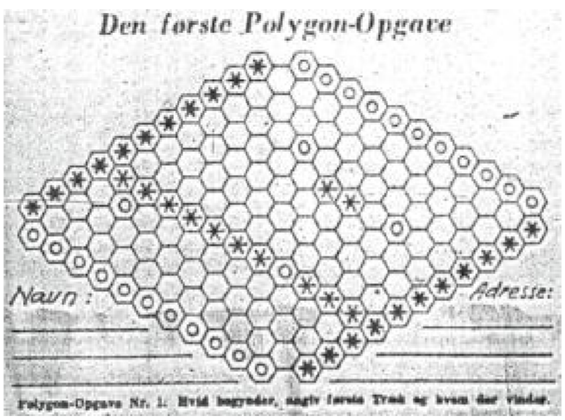


Figure 1.1: *"Would you like to learn Polygon?"* In this Polygon position, a single move by White (circle) is enough to guarantee that White will win from their opponent, Black (asterisk).
Piet Hein's first challenge to his readers was to determine where this move should be played. [9] The solution to this problem can be found in Appendix C.

The first to come up with the game we now call Hex was Danish mathematician and poet Piet Hein. The inspiration for the game came from the Four Colour Theorem — which at the time, in 1942, was still the Four Colour Conjecture. Piet Hein noted that, when considering four countries laid out in a ring, only one pair of 'opposite' countries could possibly share a border in the middle. Based on this observation he designed a game he called Polygon, where two players try to connect their own pair of 'opposite' countries on a board of hexagons — any other shape, he found, would allow more than three cells to meet at a corner and could therefore lead to a 'deadlock' situation resulting in a draw.

Piet Hein introduced the game to the public in a series of articles in the Danish newspaper Poletiken, in which he analysed the game and challenged readers to solve problems — such as the one in Figure 1.1 — based on the game.

The game was later invented independently by John Nash in 1948 at Princeton. Nash created the game in a (succesful) attempt at finding an example of a game that could be proven to be a first player win, without using, or indeed knowing, a general winning strategy for said game. The game became known as Nash, after its creator, and was quite popular among students and teachers at Princeton: In 1952 Nash himself noted that several variations on his game had already come about among players and genuine attempts were being made at finding effective strategies for the game —

including a game-playing machine built by C.E. Shannon and E.F. Moore, which played the game “better than many humans” according to its creators [12], and “fair, but not perfect” according to Nash himself [10].

The game only received the name it is most commonly referred to today in 1952, when Parker Brothers released it under the name Hex.

For a more complete discussion of the game’s history, especially regarding Piet Hein’s invention of and thoughts on the game, I highly recommend consulting [9].

Chapter 2

The Rules of Hex

In this chapter, we will discuss the rules of the traditional game of Hex, as well as the rules of two variations on the game: Cylindrical Hex and Torus Hex. For each of these games, we will also discuss how one can create a mathematical representation of the game — which can be used to more easily determine whether the game has been won, and by which player — and how winning positions in each of these games relate to equivalent positions in the other variations of Hex.

2.1 Hex

Hex is a board game played between two players, who are most commonly referred to as Red and Blue.

The board used for Hex consists of hexagonal cells which are organised along m (slanted) columns and n rows to create a parallelogram. Traditionally, a board will satisfy $m = n$, but boards where $m \neq n$ can just as easily be used to play the game.

Throughout the game, Red and Blue take turns placing a stone, coloured red or blue respectively, in an empty cell. The first move may be made by either player.

The objective for both players is to create a path of adjacent stones in their own colour from one side of the board to the opposite side. Red intends to create such a path from the top of the board to the bottom; Blue from the left side to the right. The game ends when one of the two players has created such a path and thereby wins the game.

The final position of a possible round of Hex is shown in Figure 2.1. In this case, Blue has won.

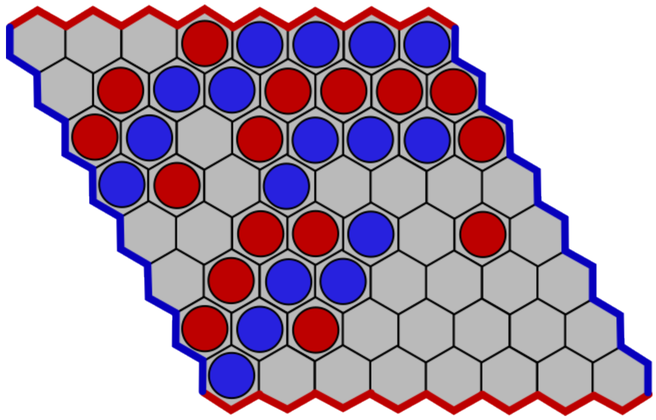


Figure 2.1: The final position of a Hex game on an 8×8 playing board. Blue has won this game, as there is a path of adjacent blue stones from the left side of the board to the right.

2.1.1 Mathematical Representation of Hex

The board used for Hex can easily be represented using a graph. To create such a graph based on an existing board, the hexagonal cells are taken as vertices. An edge is added between each pair of vertices of which the corresponding cells on the board are adjacent — i.e. each pair of hexagons that share a face. This ensures that vertices are adjacent to each other if and only if they were already adjacent on the original board. This process of creating a graphical representation of a Hex playing board can be seen in Figure 2.2.

The resulting graph, regardless of its size, has a consistent pattern to it: Its edges form a grid with added diagonal lines, which span from the top-right corner of a square to the bottom-left. This consistency makes it easy to formally define a Hex graph.

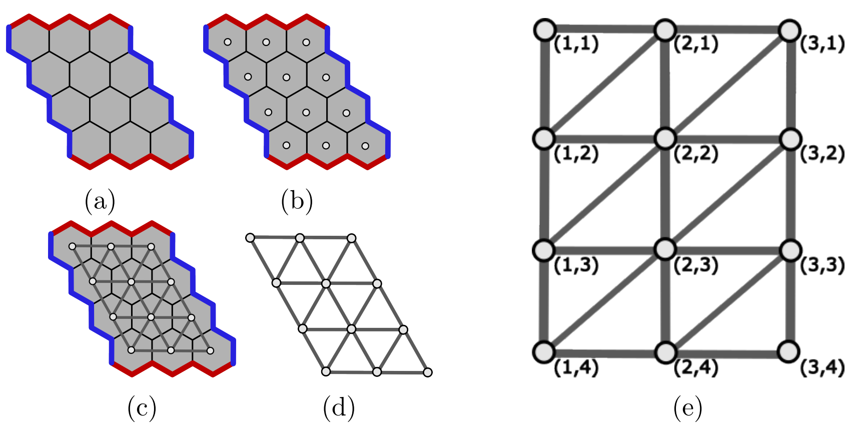
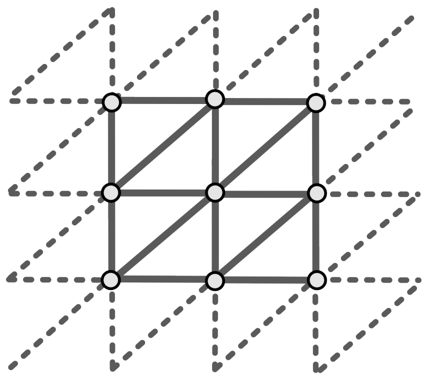
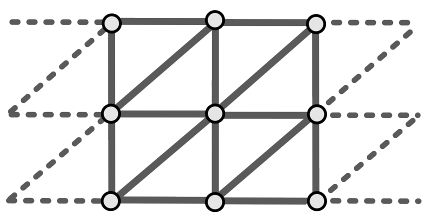


Figure 2.2: The process of creating the graph $H_r(3,4)$, based on a 3×4 Hex board: Given a Hex board (a), we introduce a vertex for each cell of the board (b). For each pair of adjacent cells, we add an edge between the corresponding vertices (c). We can then remove the original board, leaving only the vertices and edges (d), and label each of the vertices according to their column and row (e).



(a) S



(b) S_3

Figure 2.3: The infinite graph S and its subgraph S_3, which has a finite height 3, but infinite width.

We can define the Hex graph $H_r(m, n)$ in a few steps, using the following definitions:

Definition 2.1.1. *The graph S is an infinite graph such that the following hold:*

- *the set of vertices of the graph S is*

$$V = \{(i, j) : i, j \in \mathbb{Z}\}.$$

- *a pair of vertices $z = (z_1, z_2)$, $w = (w_1, w_2) \in V$ is adjacent in the graph S if and only if $z \neq w$, $|z_1 - w_1| \leq 1$, $|z_2 - w_2| \leq 1$ and $|(z_1 + z_2) - (w_1 + w_2)| \leq 1$.*

Note that S is the representation of a Hex board of infinite size. This would not be a very fun board to play on: Since there are no ends to this board, neither player can ever satisfy their win condition. We will, however, find it very useful in creating representations of more realistic boards to play various versions of Hex on.

Definition 2.1.2. *For $n \in \mathbb{N}_{>0}$, the graph S_n is the restriction of S to the vertices*

$$V_n = \{(i, j) : i, j \in \mathbb{Z}, 1 \leq j \leq n\} \subset V.$$

Note that S_n is the representation of a Hex board with n rows and infinitely many columns. Since there is now only a finite number of rows, it is now possible for Red to satisfy their win condition, but not yet for Blue. Again, while not a very fun board to play on, S_n will be useful to us later.

Parts of the graphs S and S_n are shown in Figures 2.3a and 2.3b, respectively.

Definition 2.1.3 (Hex Graph)**.** *For $m, n \in \mathbb{N}_{>0}$, the $m \times n$ Hex graph $H_r(m, n)$ is the restriction of S_n to the vertices*

$$V_{m,n} = \{(i, j) : i, j \in \mathbb{Z}, 1 \leq i \leq m, 1 \leq j \leq n\}$$

The graph $H_r(m, n)$ is the graphical representation of a Hex board with m columns and n rows. The Hex graph $H_r(3, 4)$ is shown in Figure 2.2e.

The moves made by Red and Blue can be represented by colouring the vertices of the graph: Whenever Red places a stone on a cell of the board, we colour the corresponding vertex red. Likewise, any cells that Blue plays on will be coloured blue. Doing this leads to a colouring of the graph $H_r(m, n)$. We can distinguish between complete and partial colourings.

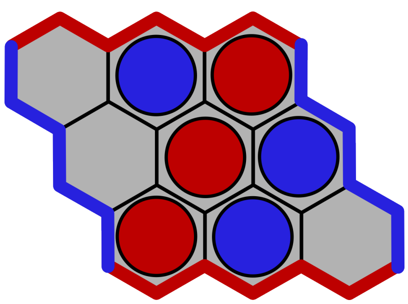
Definition 2.1.4 (Colourings on a graph)**.** *Given a graph $G = (V, E)$ and a set of colours $\mathcal{C}$.*

A complete colouring $C : V \to \mathcal{C}$ on G assigns a colour in $\mathcal{C}$ to each vertex $v \in V$.

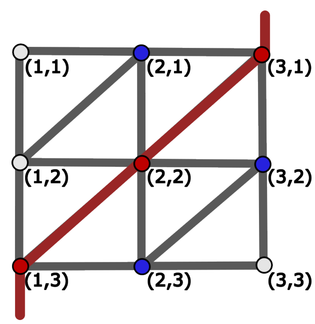
A partial colouring $C : W \to \mathcal{C}$ on G assigns a colour in $\mathcal{C}$ to each vertex $w \in W \subsetneq V$.

Note that, if we colour the vertices in the manner described above, a complete colouring on $H_r(m, n)$ will only occur once every cell has been played on. This can only occur at the end of a game. However, a final position of a game of Hex does not have to lead to a complete colouring: Since we end the game as soon as a winning path exists, there also exist final positions with unplayed cells, which must be represented using a partial colouring. Any intermediary position of the game can also be represented with a partial colouring.

A colouring can be used to determine whether a game has been won yet and, perhaps more importantly, by who: Red's objective — to create a red path from the top of the board to the bottom — is equivalent to creating a red path from any vertex in the first row of the corresponding Hex graph to any vertex in the last row. Similarly, Blue's objective — creating a blue path from the left to the right of the board — is equivalent to creating a blue path from any vertex in the first column of the graph to any vertex in the last column.

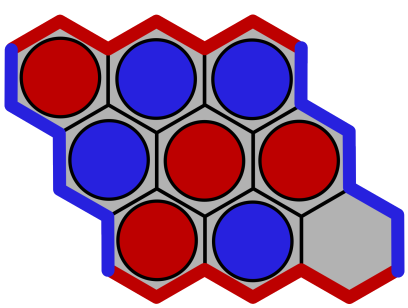


(a) A position on a 3×3 Hex board

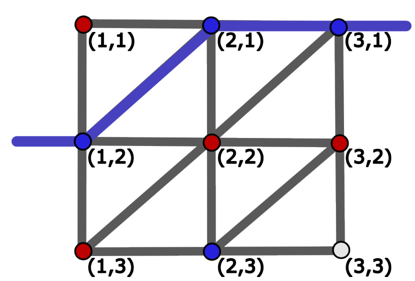


(b) A colouring of the graph $H_r(3, 3)$

Figure 2.4: An example of the same Red winning path shown on a Hex board (a) and on a partially coloured Hex graph (b).



(a) A position on a 3×3 Hex board



(b) A colouring of the graph $H_r(3, 3)$

Figure 2.5: An example of the same Blue winning path shown on a Hex board (a) and on a partially coloured Hex graph (b).

A formal definition of a winning path can be made as follows:

Definition 2.1.5 (Winning Paths for Hex)**.** *A C-winning path on $H_r(m,n)$ exists in a colouring $C : V_{m,n} \to \{red, blue\}$ if one of the following holds:*

- *There exists a C-winning path for Red, i.e. there exists at least one red path on $H_r(m,n)$ from some $(z_1, 1) \in V_{m,n}$ to some $(w_1, n) \in V_{m,n}$ in the colouring C.*
- *There exists a C-winning path for Blue, i.e. there exists at least one blue path on $H_r(m,n)$ from some $(1, z_2) \in V_{m,n}$ to some $(m, w_2) \in V_{m,n}$ in the colouring C.*

An example of a winning position for Red (resp. Blue) on the original Hex board, alongside the corresponding Hex graph and its colouring can be found in Figure 2.4 (resp. Figure 2.5).

2.2 Cylindrical Hex

In this thesis, we will pay the most attention to a variation of the normal Hex game called Cylindrical Hex.

The board used in Cylindrical Hex is created by attaching the left and right sides of a normal Hex board to each other, as depicted in Figure 2.6. As the name of this version of the game implies, this creates a cylinder.

When playing on this cylindrical board, the rules are the same as the original game: Two players take turns placing stones on the cells of the new board, until one of them has reached their objective. For Red the objective also remains the same: create a path from the top of the board to the bottom. However, since this new board has no left or right side, Blue's win condition has to change: To win in Cylindrical Hex, Blue's path must fully encircle the cylinder on which they are playing.

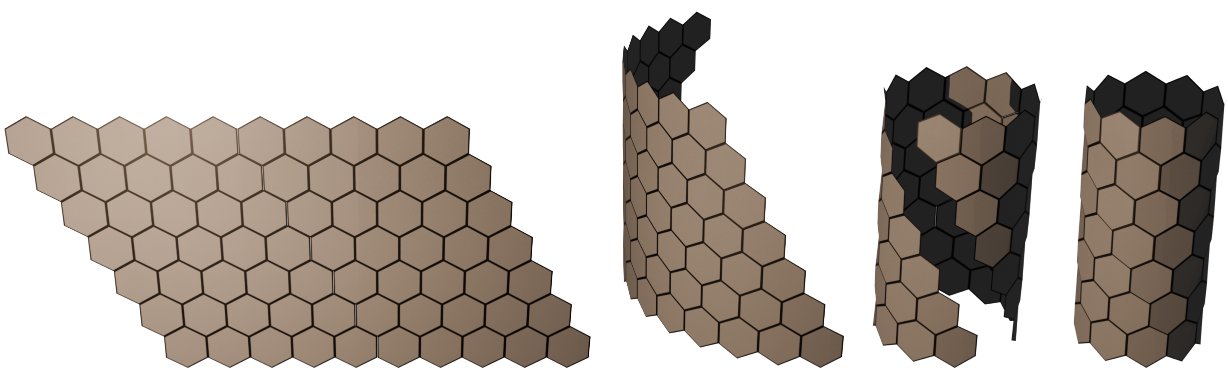


Figure 2.6: A normal 10×7 Hex playing board being turned into a 10×7 Cylindrical Hex playing board, by attaching the left and right sides of the original board to each other.

2.2.1 Mathematical Representation of Cylindrical Hex

As mentioned, the board used for a game of $m \times n$ Cylindrical Hex can be created by taking a normal $m \times n$ Hex playing board and attaching the left and right sides of this board to each other. By doing this, we create new connections between the cells of the first and last columns of the original board, which can be seen in Figure 2.7. In the graphical representation, these connections will become edges of the following form:

- Horizontal edges between vertices on the same row, i.e. edges between cells $(1, j)$ and (m, j) for $j = 1, ..., n$.
- Diagonal edges in the same orientation as the diagonal edges of $H_r(m, n)$, i.e. edges between cells $(1, j)$ and $(m, j + 1)$ for $j = 1, ..., n - 1$

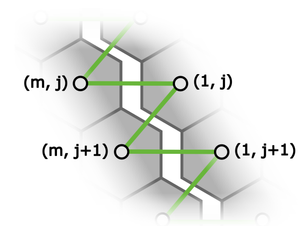


Figure 2.7: The extra connections made when creating a Cylindrical Hex playing board.

It is possible to create a Cylindrical Hex graph by simply adding these edges to $H_r(m, n)$. However, it will be more convenient — especially when looking for winning paths — to instead define the Cylindrical Hex graph using the graph S_n.

Definition 2.2.1 (Cylindrical Hex Graph)**.** *Define the map $\phi : V_n \to V_{m,n}$ on S_n as follows:*

$$\phi(z_1, z_2) = (z_1 \mod m,\ z_2).$$

For $m, n \in \mathbb{N}_{>0}$, the $m \times n$ Cylindrical Hex graph $H_c(m, n)$ is the graph with vertices $V_{m,n}$, such that two vertices $z, w \in V_{m,n}$ are adjacent in $H_c(m, n)$ if and only if there are vertices $z', w' \in S_n$ such that z' and w' are adjacent in S_n, $z = \phi(z')$ and $w = \phi(w')$.

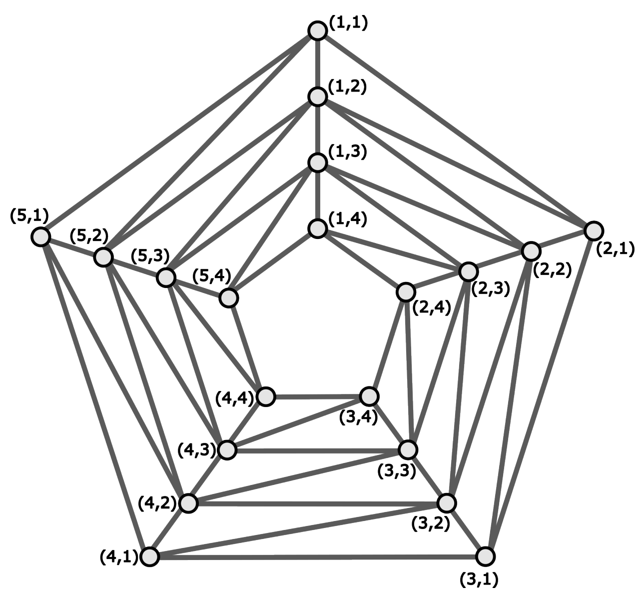


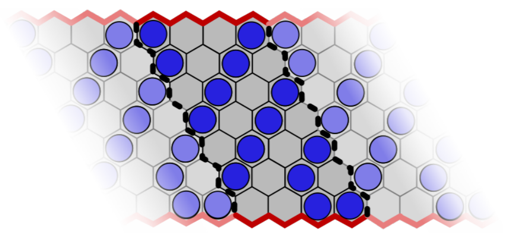
Figure 2.8: The Cylindrical Hex graph of a 5×4 playing board: $H_c(5, 4)$.

The graph $H_c(m, n)$ is the graphical representation of a Cylindrical Hex Board with m columns and n rings. The Cylindrical Hex graph $H_c(5, 4)$ is shown in Figure 2.8.

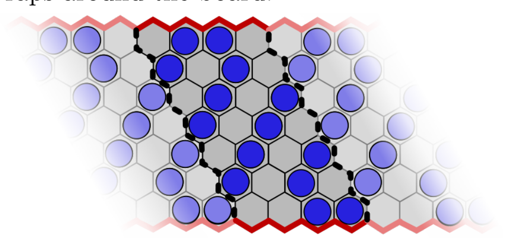
We can represent the moves made throughout a round of Cylindrical Hex in the same manner as we did for the original game of Hex. The resulting (partial) colouring of $H_c(m, n)$ can then be used to determine whether a game has been won yet and by who.

Since Blue's objective has changed, the definition of a winning path must change with it. An initial attempt at representing this new objective may be the following: "Blue must create a blue path from some vertex $(1, a)$ in the 'first' column of the board to some vertex (m, b) in the 'last' column, such that this path covers each column of the cylinder and there exists a newly added edge between these two vertices."

However, this definition would not be strict enough: As can be seen in Figure 2.9a, this can lead to a cycle that only wraps around the playing board — without fully encircling it — even if it does indeed cover every column.



(a) Not a blue winning path: The path only wraps around the board.



(b) A blue winning path: The path fully encircles the board.

Figure 2.9: Two blue paths on a 4×7 Cylindrical Hex board.

To determine whether a path fully encircles the board's cylinder, instead of merely wrapping around it, we can use the map ϕ to create a lifting of the colouring on $H_c(m, n)$ to S_n. Doing so will create a horizontally repeating pattern of our graph colouring. If there is indeed a blue cycle in $H_c(m, n)$ that encircles the cylinder, it should be possible to follow this same path in S_n from one specific vertex to another vertex in the same row, but m columns further along. Due to the repeating pattern on S_n, it will even be possible to extend this path to cover every column of S_n. Vice versa, any blue path on S_n that covers every column of S_n corresponds to a blue winning cycle on $H_c(m, n)$.

With this knowledge, it is easier to see what went wrong in the case of Figure 2.9a: While it is possible to follow the shown path from a specific vertex to *some* vertex m columns further along, it is not possible to follow the path from a specific vertex to the vertex *in the same row*, m columns along: On the graph S_n, this path connects back to its original starting vertex, instead of to the 'copy' of that vertex on the next board.

In Figure 2.9b, however, it is possible to follow the shown path from some vertex to the copy of that vertex. And indeed, should we imagine this path on an actual cylinder, we do find that it truly encircles the board.

Unlike Blue, Red's objective remains unchanged. In fact, it is still possible to find a red winning path in the same manner as before — by looking for a red path in the colouring on $H_c(m, n)$ itself, instead of looking for it in the lifting to S_n. In the formal definition, however, we stick to the latter option, as this allows us to use the same graph to find a winning path of either colour.

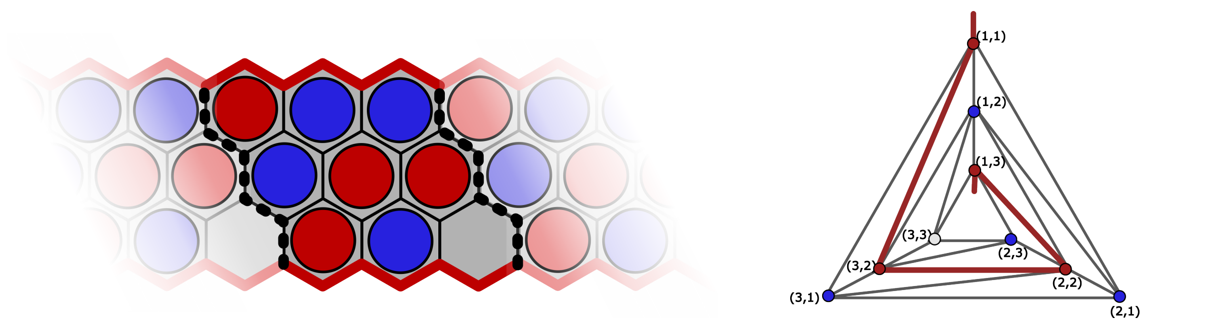
Definition 2.2.2 (Winning Paths for Cylindrical Hex)**.** *Let $C : V_{m,n} \to \{red, blue\}$ be a colouring on $H_c(m,n)$ and let $C^* : V_n \to \{red, blue\}$ be a lifting of C to S_n by $C^*(z) = C(\phi(z))$ for $z \in V_n$.*

A red C-winning path or blue C-winning cycle on $H_c(m,n)$ exists if one of the following holds:

- *There exists a C^*-winning path for Red on S_n, i.e. there exists at least one red path on S_n from some $(z_1, 1) \in V_n$ to some $(w_1, n) \in V_n$ in the lifting C^* of C.*
- *There exists a C^*-winning path for Blue on S_n, i.e. there exists at least one blue path on S_n from some $(1, z_2) \in V_n$ to some $(m+1, w_2) \in V_n$ in the lifting C^* of C, such that $z_2 = w_2$.*

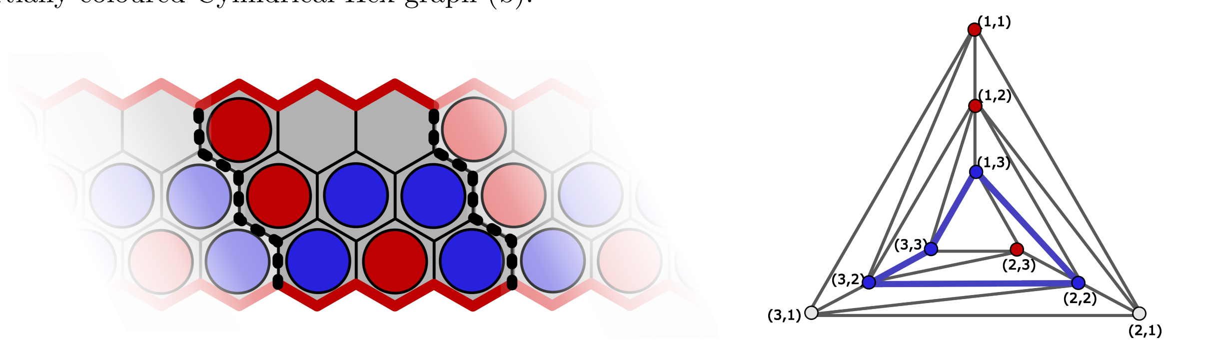
For any C-winning path P^* on S_n, the corresponding C-winning path P on $H_c(m,n)$ is found by $P = \phi(P^*)$.

An example of a winning position for Red (resp. Blue) on the Cylindrical Hex board (displayed as a part of the lifting to S_n), alongside the corresponding Hex graph and its colouring can be found in Figure 2.10 (resp. 2.11).



(a) A position on a 3×3 Cylindrical Hex Board. (b) A colouring of the graph $H_c(3,3)$.

Figure 2.10: An example of a Red winning path shown on a Cylindrical Hex board (a) and on a partially coloured Cylindrical Hex graph (b).



(a) A position on a 3×3 Cylindrical Hex board. (b) A colouring of the graph $H_c(3,3)$

Figure 2.11: An example of a Blue winning path shown on a Cylindrical Hex board (a) and on a partially coloured Cylindrical Hex graph (b).

2.2.2 Winning Hex Positions in Cylindrical Hex

Note that if a red winning path exists in a particular colouring C of $H_r(m,n)$, then this same path will still be a red winning path if we apply this same colouring to $H_c(m,n)$. In other words, a position of regular Hex in which Red has won will still be a winning position for Red in Cylindrical Hex.

The same can not be said for blue winning paths. While a blue winning path in a colouring C of $H_r(m,n)$ will still exist as a path if we apply the same colouring to $H_c(m,n)$, it is no longer a winning path. This is because we now require a winning cycle for Blue to win. In some cases, there may still be a winning cycle for Blue in the colouring on $H_c(m,n)$ — for example, if the beginning and ending of the winning path we found in $H_r(m,n)$ are adjacent to each other in $H_c(m,n)$ as a result of one of the new edges — but due to the updated requirements of Blue's new objective, this is not necessarily the case. In fact, we may instead find a red winning path in $H_c(m,n)$ that uses one of the newly added edges and was therefore not present in $H_r(m,n)$.

An example of this situation can be found by comparing Figures 2.5 and 2.10: Here both games end in the same position and therefore have an equivalent colouring, but a different player wins in each case.

In other words, a position of regular Hex in which Blue has won is not always a winning position for Blue in Cylindrical Hex — it may instead be a position in which Red has won or in which there is no winner yet.

Finally, while the added edges in Cylindrical Hex may turn an unfinished Hex game into a Cylindrical Hex win for Red, the same can not be said for Blue. In order to fully encircle the board, Blue still needs to have a path from column 1 to column m that does not use these new edges. As such a path is not already present in an unfinished Hex position, Blue can not be the winner if that position is placed on a Cylindrical Hex board.

We can conclude that Cylindrical Hex is more advantageous for Red than the original game was: Red still wins in any position in which they would have in the original game and some positions in which they would have originally lost, or in which the game has not yet ended, have now become a winning or still winnable position.

2.3 Torus Hex

The game can be taken one step further by attaching the top and bottom of the Cylindrical Hex Board to each other, creating a torus. This version is called Torus Hex.

Again, the rules are the same as they were in previous iterations of the game: Two players take turns placing stones on the new board, until one of them has reached their goal. This time, however, both players have a different objective from the one in the original game: Like in Cylindrical Hex, Blue wants to encircle the board — in this case, in the poloidal direction. Red now has a similar goal: encircling the board in the toroidal direction.

Figure 2.12: A 20×14 Cylindrical Hex playing board being turned into a 10×14 Torus Hex playing board, by attaching the top and bottom sides of the original board to each other.

2.3.1 Mathematical Representation of Torus Hex

As mentioned, the board used for a game of $m \times n$ Torus Hex can be created by taking an $m \times n$ Cylindrical Hex board and attaching the top and bottom of the cylinder to each other. By doing this, we create new connections between the cells of the first and last rings of the Cylindrical Hex board. In the graphical representations, these connections will become edges of the following form:

- Vertical edges between vertices of the the same column, i.e. edges between cells $(i, 1)$ and (i, n) for $i = 1, ..., m$.
- Diagonal edges in the same orientation as the diagonal edges of $H_r(m, n)$ and $H_c(m, n)$, i.e. edges between cells $(i, 1)$ and $((i \mod m) + 1,\ n)$ for $i = 1, ..., m$.

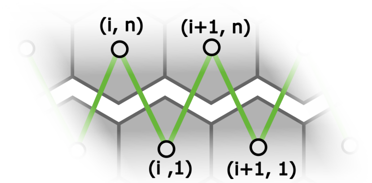


Figure 2.13: The extra connections made when creating a Torus Hex playing board.

The graph $H_t(m, n)$ is the graphical representation of a Torus Hex board with m columns and n rings. The Torus Hex graph $H_t(5, 4)$ can be seen in Figure 2.14.

Like before, one could consider simply adding these edges to $H_c(m, n)$, but it will again prove more fruitful to define the Torus Hex graph using the graph S.

Definition 2.3.1 (Torus Hex Graph)**.** *Define the map $\psi : V \to V_{m,n}$ on S as follows:*

$$\psi(z_1, z_2) = (z_1 \mod m,\ z_2 \mod n).$$

For $m, n \in \mathbb{N}_{>0}$, the $m \times n$ Torus Hex graph $H_t(m, n)$ is the graph with vertices $V_{m,n}$, such that two vertices $z, w \in V_{m,n}$ are adjacent in $H_t(m, n)$ if and only if there are vertices $z', w' \in S$ such that z' and w' are adjacent in S, $z = \psi(z')$ and $w = \psi(w')$.

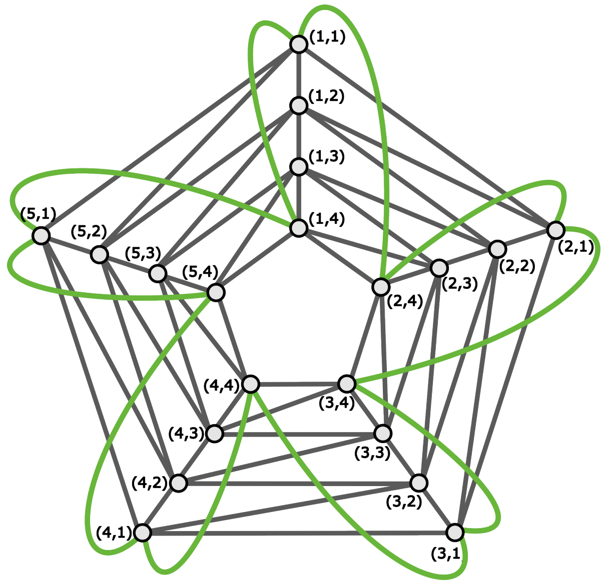


Figure 2.14: The Cylindrical Hex graph of a 5×4 playing board: $H_t(5,4)$. The extra edges compared to $H_c(5,4)$ are shown in green.

We can represent the moves made throughout a round of Torus Hex in the same manner as we did for the previous versions of the game. The resulting (partial) colouring of $H_t(m,n)$ can then be used to determine whether a game has been won yet, and by who.

This time, it is Red's objective that has changed. To encircle the board, Red's path now not only has to lead from the first ring of the graph to the last: It also has to connect its beginning and end vertices, using one of the newly made connections.

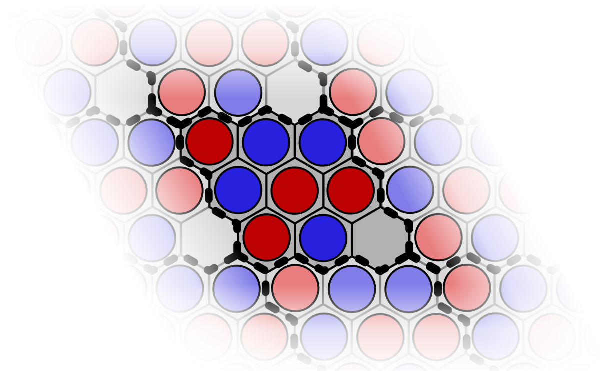
To make sure this path truly encircles the torus and doesn't only wrap around it, we again create a lifting, this time of the colouring on $H_t(m,n)$ to S. This will create a horizontally and vertically repeating pattern of our graph colouring. If there is a red path in $H_t(m,n)$ that encircles the torus in the right direction, it should be possible to follow this path in S_n from one specific vertex to another vertex in the same column, but n rows further along. In fact, a red winning path will pass through every row of S.

Definition 2.3.2 (Winning Paths for Torus Hex)**.** *Let $C : V_{m,n} \to \{red, blue\}$ be a colouring on $H_t(m,n)$ and let $C^\dagger : V_n \to \{red, blue\}$ be a lifting of C to S by $C^\dagger(z) = C(\psi(z))$ for $z \in V$.*

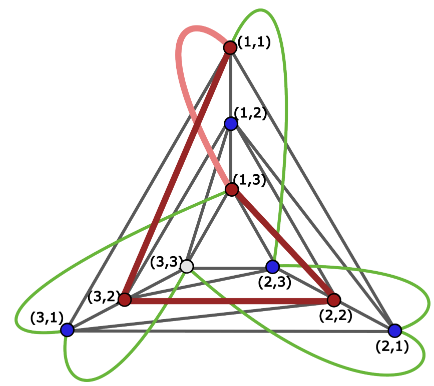
A C-winning cycle on $H_t(m,n)$ exists if one of the following holds:

- *There exists a $C^\dagger$-winning path for Red on S, i.e. there exists at least one red path on S from some $(z_1, 1) \in V$ to some $(w_1, n+1) \in V$ in the lifting $C^\dagger$ of C, such that $z_1 = w_1 \mod m$.*
- *There exists a $C^\dagger$-winning path for Blue on S, i.e. there exists at least one blue path on S from some $(1, z_2) \in V$ to some $(m+1, w_2) \in V$ in the lifting $C^\dagger$ of C, such that $z_2 = w_2 \mod n$.*

For any C-winning path P^* on S_n, the corresponding C-winning path P on $H_c(m,n)$ is found by $P = \phi(P^*)$.

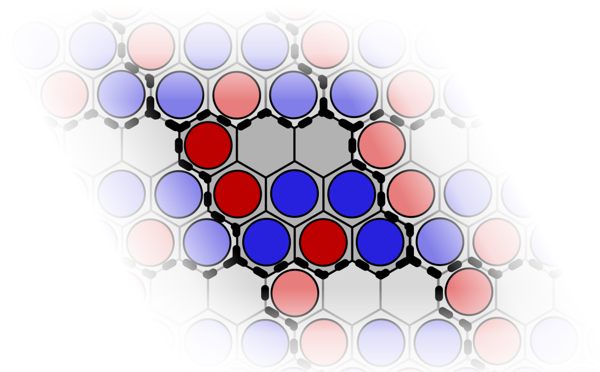


(a) A position on a 3×3 Torus Hex Board.

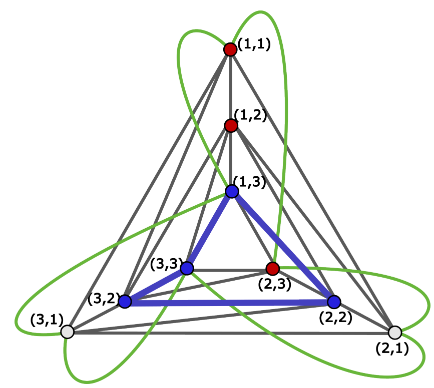


(b) A colouring of the graph $H_t(3,3)$.

Figure 2.15: An example of a Red winning path shown on a Torus Hex board (a) and on a partially coloured Torus Hex graph (b).



(a) A position on a 3×3 Torus Hex board.



(b) A colouring of the graph $H_t(3,3)$

Figure 2.16: An example of a Blue winning path shown on a Torus Hex board (a) and on a partially coloured Torus Hex graph (b).

An example of a winning position for Red (resp. Blue) on the Cylindrical Hex board (displayed as a part of the lifting to S), alongside the corresponding Hex graph and its colouring can be found in Figure 2.15 (resp. 2.16 and 2.17).

2.3.2 Winning Cylindrical Hex Positions in Torus Hex

Note that if a blue winning cycle exists in a particular colouring C of $H_c(m,n)$, then this cycle will still be a blue winning path if we apply this same colouring to $H_t(m,n)$. In other words, a position of Cylindrical Hex in which Blue has won will also be a winning position for Blue in Torus Hex.

The same can not be said for red winning paths. While a red winning path in a colouring C of $H_c(m,n)$ will still exist if we apply the same colouring to $H_t(m,n)$, it is no longer a red winning path. This is because Red now needs a winning cycle to win. In some cases, there may still be a winning cycle for Red in the colouring on $H_t(m,n)$ — for example, if the beginning and ending of the winning path we found in $H_c(m,n)$ are adjacent to each other on the Torus Hex board as a result of one of the newly added edges in $H_t(m,n)$ — but due to the updated requirements of Red's new objective, this is not always the case. In fact, we may instead find a new blue winning cycle that uses one of these new edges.

An example of this situation can be found by comparing the Cylindrical and Torus graphs in Figure 2.17. While the position on the board — and thus the colouring of $V_{m,n}$ — remains the same throughout, we find that Red wins in Cylindrical Hex, while Blue wins in Torus Hex.

In other words, a position of Cylindrical Hex in which Red has won will not necessarily be a winning position for Red in Torus Hex — it may instead be a position where Blue has won or in which there is no winner yet.

Finally, while the added edges in Torus Hex may turn an unfinished Cylindrical Hex game into a Torus Hex win for Blue, the same can not be said for Red. In order to fully encircle the board, Red still needs to have a path from row 1 to column n that does not use these new edges. As such a path is not already present in an unfinished Cylindrical Hex position, Red can not be the winner if that position is placed on a Torus Hex board.

We can conclude that Torus Hex is more advantageous for Blue than Cylindrical Hex was: Blue still wins in any position in which they would have in Cylindrical Hex, and some positions in which they would have originally lost have now become a winning or still winnable position.

2.3.3 Winning Hex Positions in Torus Hex

When going from Hex to Cylindrical Hex, and from Cylindrical Hex to Torus Hex, only one of the winning conditions changed. This made it quite easy to pinpoint who gained an advantage from this change: The number of possible winning positions could not decrease for the player whose condition remained the same — any position in which they had won in the original game (i.e. Hex or Cylindrical Hex), would still be a winning position in the more complicated one (i.e. Cylindrical Hex or Torus Hex respectively).

When going from Hex directly to Torus Hex, both winning conditions change. This makes it hard to determine if one of the players gains a distinct advantage by changing the game. By combining what we discussed in Sections 2.2.2 and 2.3.2, we see that the winner (or lack thereof) of the original position is not enough of an indicator of the winner (or lack thereof) on the Torus Hex board.

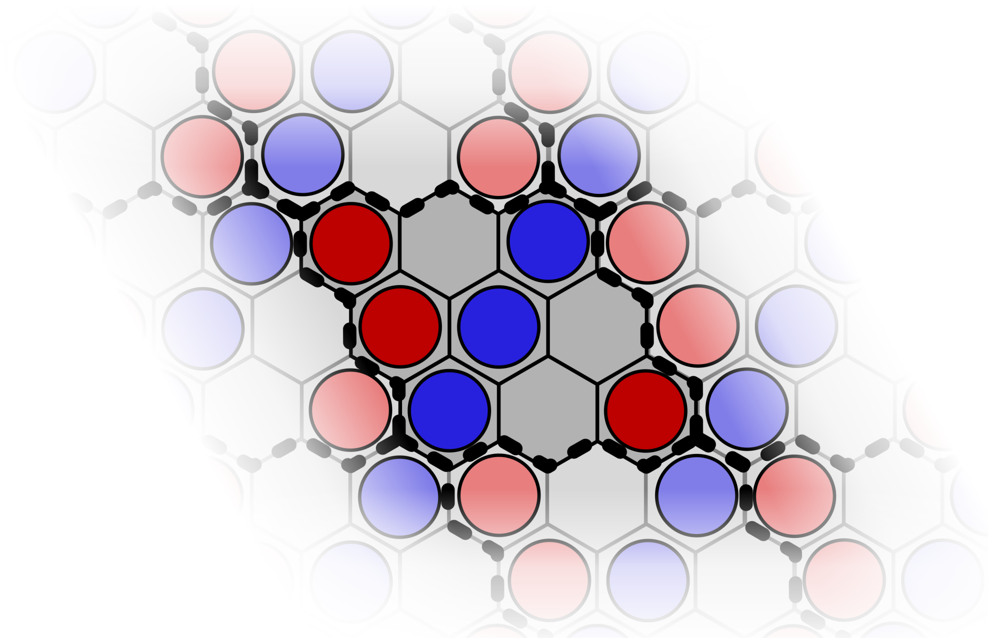
While we concluded that a position in Hex that was won by Red would remain a Red win in Cylindrical Hex, we also saw that this Red Cylindrical Hex win could have any outcome in Torus Hex. An unfinished Hex game could either be unfinished in Cylindrical Hex as well or be a Red win, and from there the outcome is just as uncertain in Torus Hex. Meanwhile, a Blue winning position in Hex already has a completely uncertain outcome in Cylindrical Hex.

Various example positions in this thesis show some of the myriad of possibilities in this regard. For example:

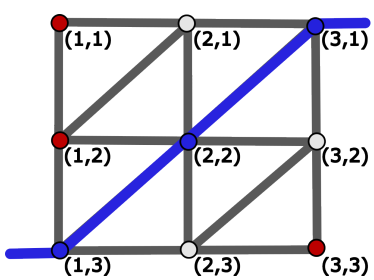
- Figures 2.4 and 2.11 both have the same winner — Red and Blue, respectively — regardless of which game was played.

- Figures 2.5, 2.10 and 2.15 each show the same position, but while Blue wins in the first game — a regular game of Hex — Red wins both other games.
- Figure 2.17 shows a position in which Blue wins in Hex, Red in Cylindrical Hex, and Blue wins again in Torus Hex.
- Figure 4.5 is unfinished in Hex, but a win for Red in Cylindrical and Torus Hex.
- Figure 4.7 is an unfinished game in Hex, a Red win in Cylindrical Hex, and a Blue win in Torus Hex.
- Conversely, Figure 5.2 is a Blue win in Hex, a Red win in Cylindrical Hex and unfinished in Torus Hex.

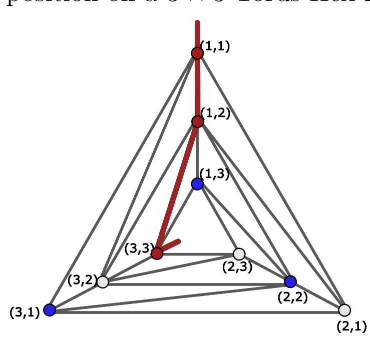
As the outcome of the game can change so unpredictably between Hex and Torus Hex, it is hard to say whether a specific player has more possible winning positions in one version of the game over the other. As such, we can not say whether either player gains an advantage by changing from Hex to Torus Hex.



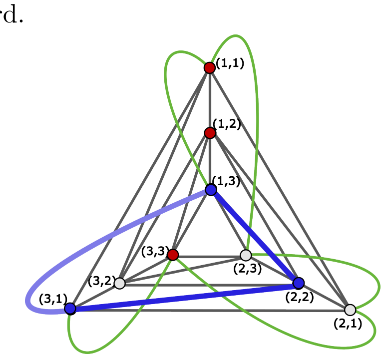
(a) A position on a 3×3 Torus Hex Board.



(b) A colouring of the graph $H_r(3,3)$, according to the above position.



(c) A colouring of the graph $H_c(3,3)$, according to the above position.



(d) A colouring of the graph $H_t(3,3)$, according to the above position.

Figure 2.17: An example of a colouring of $V_{m,n}$ that has a different winner depending on which version of Hex is being played. The position is shown on a Torus Hex board (a) and on partially coloured regular, Cylindrical and Torus Hex graphs ((b), (c) and (d) respectively).

Chapter 3

Winners of Hex

3.1 There is Always a Winner

An exciting characteristic of Hex is that the game can never end in a draw. Each game of Hex that is played will have a winner. In other words, each final position of a game of Hex — and thus, each complete colouring of a Hex board — contains a winning path.

As various proofs already exist of this fact, we will not discuss this matter in great depth. However, it would be remiss of me to discuss the game of Hex and not provide the knowledge of this fact, as well as sources that can be consulted to see its proof.

Theorem 3.1.1 (Weak Hex Theorem)**.** *For any complete colouring C of the $m \times n$ Hex graph $H_r(m, n)$, there exists a red or a blue C-winning path.*

The most commonly used proof of this theorem is the one described by Gale [4] and makes use of a path along the edges between the hexagonal cells of the Hex board. A somewhat different proof, in which the actual stones in the cells are recoloured to check if a winning path exists, was found by Van den Broek [13].

This fact does not only hold for the original game of Hex. It also holds for Cylindrical Hex:

Theorem 3.1.2 (Weak Cylindrical Hex Theorem)**.** *For any complete colouring C of the $m \times n$ Cylindrical Hex graph $H_c(m, n)$, there exists a red C-winning path or a blue C-winning cycle.*

As well as for Torus Hex:

Theorem 3.1.3 (Weak Torus Hex Theorem)**.** *For any complete colouring C of the $m \times n$ Torus Hex graph $H_t(m, n)$, there exists a red or blue C-winning cycle.*

Both of these theorems can be proven using the (Weak) Hex Theorem itself: These proofs can be found in [13].

In the case of the (Weak) Cylindrical Hex Theorem a different, direct proof is also possible, as shown by Van Hees [14]: This proof uses an algorithm based on the Depth-First-Search algorithm that recolours each red cell from which a path to the top of the cylinder exists to be orange instead and shows that in doing so exactly one of two situations should hold:

- There is an orange path to the bottom of the cylinder and so there clearly must have been a Red winning path originally.
- Within the set of cells adjacent to the orange cells, there must exist a Blue winning cycle. Informally, this can be seen by cutting away all orange cells. The top of the remaining cut-up cylinder will have a blue edge — after all, if one of those now-top edge cells had not been blue, it would have been recoloured instead — which forms the Blue winning cycle.

3.2 Is There Always Exactly One Winner?

We already know that it is impossible for a game of Hex to end without a winner, but is it possible for there to be more than one winner?

The answer is quite obvious: In a normal game of Hex — whether it is the original game, the Cylindrical version, or the Toroidal one — the game ends as soon as one of the players completes a C-winning path. Once this happens, we can conclude that their opponent did not have a C-winning path before the last move (the game would have already been over in that case) the last move did not complete a C-winning path for the opponent (the finishing player played a stone in their own colour, and could therefore not have brought a winning path of the opposing colour into existence) and the opponent will not be able to create such a path either (since the game is now finished). In normal play, it is therefore impossible for both players to have a C-winning path: Only one winner can exist and the question that forms the title of this chapter has already been answered.

Let's briefly consider a more difficult question instead: "In a complete colouring C of a given type of Hex board, is it possible for there to simultaneously be both a blue C-winning path and a red C-winning path?"

In this situation, we essentially consider what happens if we allow the players to continue making moves after a winner has been found until the board is completely filled up and no more moves can be made. One would hope that even then we only find one winner. In the case of the original game:

Theorem 3.2.1 (Strong Hex Theorem). *For any complete colouring C of the $m \times n$ Hex graph $H_r(m, n)$, there exists either a red or a blue C-winning path, but not both.*

For the original game of Hex, it feels quite obvious that this should be the case. Gale [4] offers an intuitive analogy to explain this: damming a river. We can interpret blue stones as a river and red stones as parts of a dam. In this interpretation of the game, Blue's goal is to ensure the river can flow from one side of the board to the other, while Red wishes to build a complete dam. Their goals are entirely opposed to the other — once the river reaches from one side to the other, no dam can be built to stop it and, conversely, once a dam has been built, there is no room for the river to flow around it. As such, we expect that only one player can ever succeed. Even when extending this to a colouring C that does not follow the traditional rules of Hex (for example, a colouring where one player was allowed to make moves more frequently than their opponent), this reasoning seems intuitively obvious.

Gale himself only mentions a combinatorial proof shown to him by D. Lichtenstein, of which unfortunately no published version could be found. Luckily, another proof has since been published: Hayward and Van Rijswijck [6] provide a proof using induction on a more general game called Y, of which Hex is simply a special case.

Theorem 3.2.2 (Strong Cylindrical Hex Theorem)**.** *For any complete colouring C of the $m \times n$ Cylindrical Hex graph $H_c(m,n)$, there exists either a red C-winning path or a blue C-winning cycle, but not both.*

For Cylindrical Hex, the Strong Hex Conjecture still seems like an obvious statement to make: While the board is more complicated, the intuitive reasoning discussed still holds. Even with the added edges, there is no way for Red to dam a completed river, nor is there a way for Blue to go around a completed dam.

And indeed, the proof provided by Van Hees [14], which was already mentioned in Section 3.1, also proves the Strong Cylindrical Hex Theorem.

One would hope that we can push this even further, but unfortunately a very simple counter example against a potential Strong Torus Hex Theorem exists, as shown in Figure 3.1.

It seems the additional edges added to create the Torus Hex board allow the C-winning paths of the two players to coil around each other, without interfering with each other. This coiling around each other results in cycles on the Torus Hex graph that encircles the board in both the poloidal direction and the toroidal one. Interestingly, any such cycle would satisfy the requirements to be either a red or a blue winning cycle — it could be used by either player to win.

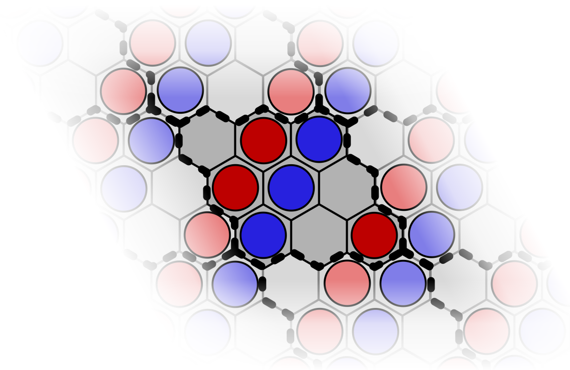
One may wonder what would happen if we were to restrict winning paths to only those that encircle the board in one direction. That is to say, blue (resp. red) winning paths must encircle the board in the poloidal (resp. toroidal) direction and, additionally, may not encircle the board in the toroidal (resp. poloidal) direction.

More formally expressed:

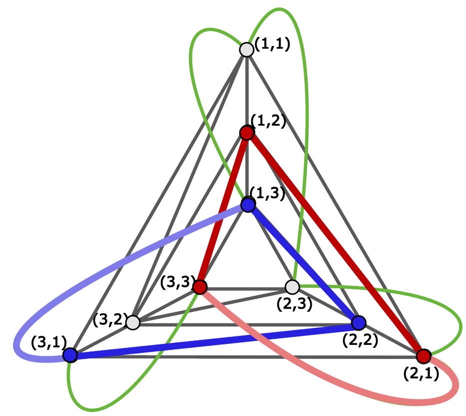
Definition 3.2.1 (Restricted Winning Paths for Torus Hex)**.** *Let the reversed colouring $C^r : V_{m,n} \to \{red, blue\}$ be a partial colouring on $H_t(m,n)$ such that: $C^r(x) = red \iff C(x) = blue$ and $C^r(x) = blue \iff C(x) = red$.*

A restricted C-winning cycle on $H_t(m,n)$ exists if one of the following holds:

- *There exists a C winning path for Red on $H_t(m,n)$, such that the same path is not a C^r winning path for Blue on $H_t(m,n)$.*
- *There exists a C winning path for Blue on $H_t(m,n)$, such that the same path is not a C^r winning path for Red on $H_t(m,n)$.*



(a) A position on a 3 × 3 Torus Hex Board.



(b) A colouring of the graph $H_t(3,3)$.

Figure 3.1: A counterexample that disproves a potential 'Strong Torus Hex Theorem'. Both a Red and a Blue winning path are shown in the same colouring on a Torus Hex board (a) and on a partially coloured Torus Hex graph (b).

Perhaps if we redefine our winning conditions in such a way, it is possible to guarantee that we have at most one winner. The river-and-dam intuition we used for regular and Cylindrical Hex certainly seems to hold in this case.

Unfortunately, even if this can be proven, the restricted version of Torus Hex comes with a different drawback: It instead becomes impossible to guarantee that there is at least one winner — we lose our Weak Torus Hex Theorem when we restrict our winning paths in this way. Figure 3.1 once again acts as a counterexample: Both of the cycles that exist in this position encircle the board in both the poloidal and toroidal position and are therefore not restricted winning paths. Additionally, one can easily check that any possible complete colouring of the uncoloured cells in this figure will not yield a restricted winning path either. Under the adjusted rules, the game will end in a draw.

Chapter 4

Existing Strategies

While it is interesting to know that there will be a winner, it is even more interesting to know who this winner will be. Or better yet — how you can ensure that you are the one to win. Luckily, a lot of research has been done into this subject for Hex, and in quite a few situations optimal strategies are known.

4.1 Strategies for Hex

In the case of the original game, it is useful to distinguish between symmetrical and asymmetrical board sizes

4.1.1 Hex Strategies on Symmetrical Boards

In the case of a symmetrical Hex board — i.e. an $n \times n$ board — it quickly became apparent that the first player to make a move would always be able to win the game [10].

This fact can be shown through a simple strategy stealing argument: Assume the second player has a winning strategy. The first player can start the game by making an arbitrary move on the board. After this initial move they follow the second player's winning strategy wherever possible and, when asked to play on an occupied cell — a result of the first arbitrary move of the game — they make another arbitrary move. The first player can essentially act as if they are the second player, but with an extra stone on the board. Since an extra stone will never act as a handicap in Hex, this strategy will lead to a win for the first player: The assumption that the second player had a winning strategy was incorrect. A more detailed description of this proof can be found in [5].

Unfortunately, this proof of the existence of a winning strategy does not give us any indication as to what this strategy actually is. A general winning strategy for Hex has yet to be found.

That being said, optimal starting moves have been known for boards up to size 5×5 for over half a century [5], and mathematicians and computer scientists have certainly not given up since then: Boards up to size 8×8 have been solved over the years — smaller ones by hand and, more recently, the larger ones algorithmically [2].

4.1.2 Gardner's Hex Strategy on Asymmetrical Boards

On an asymmetrical board — i.e. an $m \times n$ board with $m \neq n$ — the winner no longer depends on who makes the first move: The player who has the shortest distance to cover can always win. In other words, if $m < n$, Blue has a winning strategy, and if $m > n$, Red has a winning strategy.

For the specific case of a board of size $k \times (k+1)$ (resp. $(k+1) \times k$), a strategy was presented by Gardner [5]: For this strategy, the board is interpreted as two equilateral triangles with edges of length k that mirror each other. Figure 4.1 illustrates this mirroring property through the labels on the 3×4 sub-board consisting of a green and a purple triangle: Each pair of cells with the same label are each others' mirrors.

These mirroring cells are the crux of the strategy: Whenever Red (resp. Blue) plays on one of these cells, Blue (resp. Red) should play on its mirror. Since Blue (resp. Red) has less distance to cover, this strategy allows them to create a winning path before their opponent does, regardless of who makes the first move.

It is easy to extend the strategy to any other board of size $m \times n$ with $m < n$ (resp. $m > n$): Blue (resp. Red) is guaranteed to create a full winning path before their opponent can even create a locally winning path on the sub-board made by the triangles of the original strategy — any moves outside of the sub-board will not form a threat to Blue (resp. Red). An example of the decomposition of the 3×6 Hex board can be seen in Figure 4.1: This board can be decomposed into two triangles with length 3 — forming the 3×4 sub-board on which Gardner's strategy can be used — and a slanted rectangle of size 3×2.

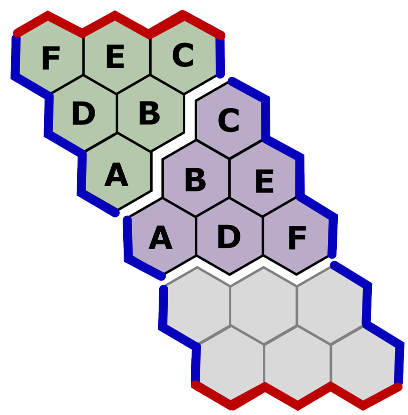


Figure 4.1: A 3×6 board split into three regions, according to the optimal strategy for an asymmetrical Hex board.
The labels on the triangle cells indicate how these cells mirror each other.

More formally, for player Blue, the strategy works as follows:

Algorithm 4.1.1 (Gardner's blue winning strategy for Hex on an $m \times n$ board with $m < n$)**.** *Consider the Hex Graph $H_r(m, n)$.*

Divide the vertices of this graph into three sections: Split the vertices of the first $m+1$ rows into two mirroring triangles, of which each side consists of m vertices. If $n > m+1$, the remaining $n-(m+1)$ rows form a slanted rectangle.

Label the vertices of the two triangles so that the labels mirror each other. We call the mirror of some cell (i, j), the mirror cell (i^*, j^*).

On their turn, Blue follows the first applicable rule:

I. If Red has not played yet, play on any cell in one of the two triangles.

II. If Red's latest move was on cell (i, j):

1. *If (i, j) was on one of the triangles, and its mirror, (i^*, j^*), has not yet been played: play on (i^*, j^*),*
2. *If (i, j) was on one of the triangles, and its mirror, (i^*, j^*), has already been played, play on any cell in one of the two triangles,*
3. *If (i, j) was on the slanted rectangle, play on any cell in one of the two triangles,*

For player Red, the algorithm works in much the same way. The only major difference is the way in which the board is decomposed:

Algorithm 4.1.2 (Gardner's red winning strategy for Hex on am $m \times n$ board with $m > n$)**.** *Consider the Hex Graph $H_r(m, n)$.*

Divide the vertices of this graph into three sections: Split the vertices of the first $n+1$ columns into two mirroring triangles, of which each side consists of n vertices. If $m > n+1$, the remaining $m-(n+1)$ rows form a slanted rectangle.

Label the vertices of the two triangles so that the labels mirror each other. We call the mirror of some cell (i, j), the mirror cell (i^*, j^*).

On their turn, Red follows the first applicable rule:

I. *If Blue has not played yet, play on any cell in one of the two triangles.*
II. *If Blue's latest move was on cell (i, j):*
    1. *If (i, j) was on one of the triangles, and its mirror, (i^*, j^*), has not yet been played: play on (i^*, j^*),*
    2. *If (i, j) was on one of the triangles, and its mirror, (i^*, j^*), has already been played, play on any cell in one of the two triangles,*
    3. *If (i, j) was on the slanted rectangle, play on any cell in one of the two triangles,*

An example of a final position of a game of Hex in which Algorithm 4.1.2 was used by Red can be seen in Figure 4.2.

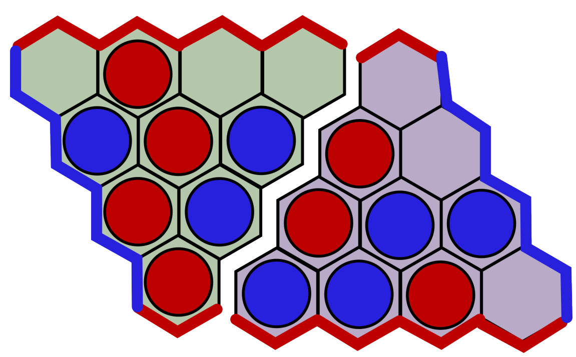


Figure 4.2: The final position of a game of regular Hex on a 5×4 board, during which Red used Algorithm 4.1.2. The first move was made by Blue and the game was won by Red.

Theorem 4.1.1 (Gardner's Hex Strategies). *Algorithms 4.1.1 and 4.1.2 are both optimal strategies for their respective players and situations, that is to say:*

- *If Blue follows Algorithm 4.1.1 on an $m \times n$ board where $m < n$, Blue will always win.*
- *If Red follows Algorithm 4.1.2 on an $m \times n$ board where $m > n$, Red will always win.*

Since no rigorous proof of this fact was found in the existing literature, we will prove the effectiveness of Algorithm 4.1.2 below. The proof of the effectiveness of Algorithm 4.1.1 works analogously and together the two prove that Theorem 4.1.1 does indeed hold.

Proof (Optimality of Algorithm 4.1.2). Assume that, at the end of a game on a $m \times n$ Hex board with $m > n$, Blue has at least one winning path, despite Red using Algorithm 4.1.2. Consider the shortest blue winning path.

Since this is a winning path on the full board, the restriction of this path to the first $n+1$ columns of the board must locally be a winning path as well. Note that this $(n+1) \times n$ sub-board consists of the two triangles that the algorithm relies on.

As can be seen in Figure 4.3, the deconstruction of the board means that the restricted path must start at some point a_B in the first column of the board — which is part of the green (left) triangle — and must end at some point b_B in column $(n+1)$ — which is part of the purple (right) triangle.

As the ends of this path lie in different triangles, there must be a point at which the path crosses from one into the other. Call the last cell before this (first) happens x_B and consider the sub-path from a_B to x_B on the green triangle.

Since Red used Algorithm 4.1.2 each mirror of a blue cell on the sub-board will be red, with one possible exception: the mirror of the last cell Blue played. Since the game ended after this move, Red was not able to mirror it. However, we know that this mirror cell can not have been blue either: If Blue had played this mirror cell earlier in the game, Red would have played on the original cell — the cell Blue played on to end the game. Since this is not possible, we know the mirror must be unplayed in the final position. As there is already a Blue winning path, this means that, by Theorem 3.2.1, Red can not also have a winning path, even if we complete the colouring. As such, we will also colour this cell Red for the time being — this will make it easier to discuss the rest of the proof.

Since Red used Algorithm 4.1.2, we also know that the blue path from a_B to x_B on the green triangle will have a mirrored red path on the purple triangle. This path starts from cell a_R — the mirror of a_B — and ends in cell x_R — the mirror of x_B.

We know that cell x_B is the last cell of the blue path on the green triangle. As we can see in Figure 4.3, this cell has only two adjacent cells on the purple triangle, namely $(x_{B_1}+1,\ x_{B_2})$ and $(x_{B_1},\ x_{B_2}+1)$. Since the next cell on the blue winning path, cell y_B, is the first cell of the path on the purple triangle, y_B cell must be one of these two adjacent cells. However, $(x_{B_1}+1,\ x_{B_2})$ is the mirror of x_B, which means that $x_R = (x_{B_1}+1,\ x_{B_2})$. As mentioned before, this cell must be Red as a result of Algorithm 4.1.2. The first cell of the blue winning path on the purple triangle must be $y_B = (x_{B_1},\ x_{B_2}+1)$.

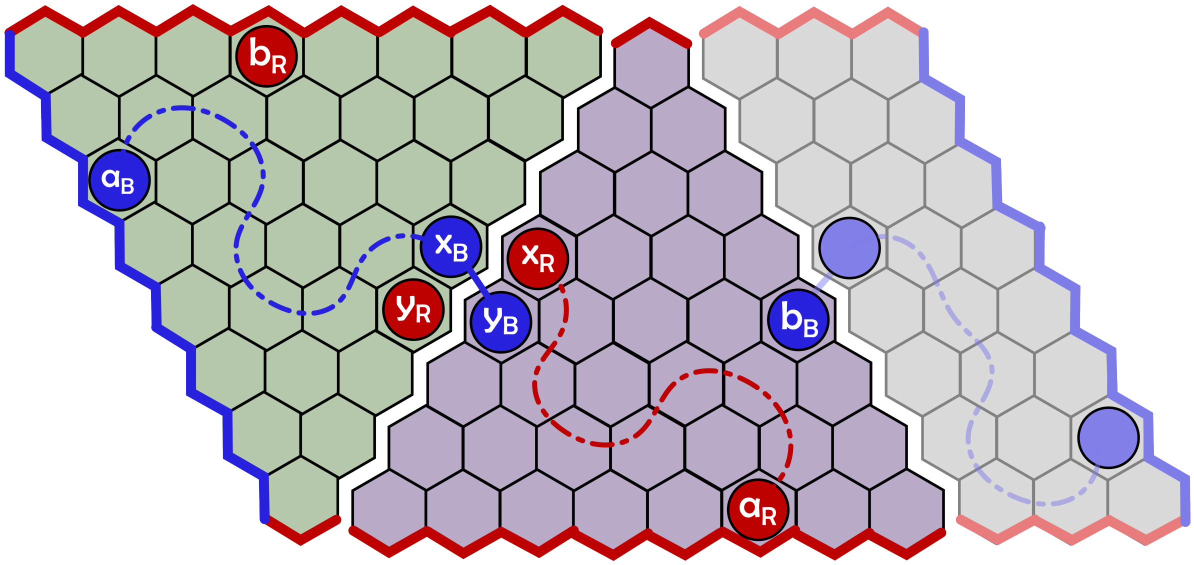


Figure 4.3: A sketch of the assumed situation from the proof of Theorem 4.1.1, which states that the strategy discussed by Gardner [5] is an optimal strategy on an asymmetrical board.

Figure 4.3 shows that the red path from a_R to x_R divides the purple triangle into two sections. The right-most column of the (sub-)board, on which b_B lies as the end of the (local) winning path, is in one of these sections, but y_B is in the other. Due to the blockade formed by the red path from a_R to x_R, it is impossible for Blue to connect y_B and b_B using only the cells in the purple triangle.

Note that this will be the case regardless of the exact situation, and is not simply a result of a conveniently made sketch:

- Due to the mirroring nature of the strategy, x_R will always be the node directly to the right of x_B. This forces Blue to continue their path via y_B — a move that forces the path downwards, into the bottom-left part of the purple triangle.

  The only time this doesn't happen is when x_B lies in row n: here, x_R blocks the only way for Blue to cross to the purple triangle. We thus find that, in this case, x_B could not have been the last cell of this path in the green triangle in the first place.

- a_R will always be in the bottom row of the purple triangle. This is because it is the mirror of a_B, which — being the start of the blue winning path — must lie in the first column of the board.

- Since the mirrored red path from x_R to a_R goes from the left edge to the bottom edge of the purple triangle, we can conclude the bottom-left part can not contain any cells from column $n+1$ — i.e. the rightmost column of the purple triangle. This column can only contain cells that are part of the right area of the triangle or are part of the red path.

- b_B, being the end of the blue (locally) winning path, must lie in column $n+1$. As such, it can never lie in the bottom-left part of the purple triangle and can thus not be reached by Blue from y_B without leaving this triangle.

If Blue tries to return to the green triangle to circumvent the blockade there, they will be forced to cross their own path to reach the other section of the right triangle — that is, the one in which b_B lies. This would mean that the circle blue's path has made through the purple triangle was unnecessary, and a shorter winning path therefore exists, which contradicts our earlier assumption.

Even if we were to ignore this contradiction, we would only be repeating our earlier arguments — whenever the blue path reaches the edge between the triangles, the mirrored red path forces Blue to enter a part of the purple triangle from which they can not reach b_B, or any other vertex on the right edge of the board, until there are no more ways for Blue to move to the purple triangle.

As a result of this contradiction, Blue can never make a winning path if Red follows this strategy. Since Hex will always have a winner — see Theorem 3.1.1 or Theorem 3.2.1 — we conclude that Red will always win if they use Algorithm 4.1.2. □

4.2 Strategies for Cylindrical Hex

There already exist strategies for the Red player in Cylindrical Hex that have been proven to work perfectly on specific board sizes.

4.2.1 A Cylindrical Hex Strategy on $2k \times n$ Boards

For Cylindrical Hex boards with an even number of columns, a very straightforward strategy exists. This strategy, discovered and proven by Alpern and Beck [1], simply tells Red to play the exact same move as the one their opponent just made, but on the opposite side of the cylinder.

Algorithm 4.2.1 (Red's winning strategy for Cylindrical Hex on a $2k \times n$ board)**.**
Let $m = 2k$ and consider the Cylindrical Hex graph $H_c(m, n)$. On their turn, Red follows the first applicable rule:

I. *If Blue has not played yet, play on any cell.*

II. *If Blue's latest move was on cell (i, j), play according to the first applicable rule:*

    1. *at cell $T(i, j) = (i + k \mod m, j)$,*
    2. *anywhere.*

An example of a final position of a game of Cylindrical Hex in which this strategy was used by Red can be seen in Figure 4.4.

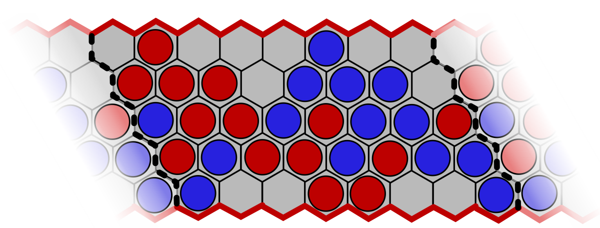


Figure 4.4: The final position of a game of Cylindrical Hex on an 8×5 board, during which Red used Algorithm 4.2.1. The first move was made by Blue and the game was won by Red.

4.2.2 A Cylindrical Hex Strategy on $3 \times n$ Boards

The optimal strategy for boards with an even number of columns relies on the existence of an exact opposite cell for each cell on the board. This strategy will therefore not work on a board with an odd number of columns [14]: A different approach is required instead.

For boards with 3 columns, an optimal strategy was found and proven by Huneke, Hayward, and Toft [8]. An alternative proof of this strategy was also found by Van den Broek [13].

The strategy focuses on creating Red locally winning paths — which will eventually connect to each other to form a global winning path — in and around the ring in which Blue has just played.

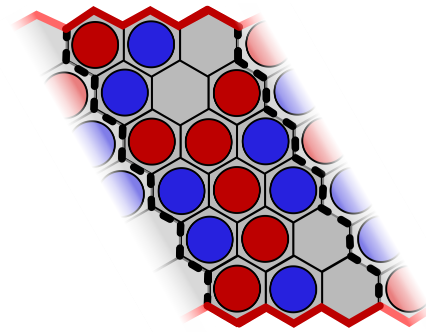


Figure 4.5: The final position of a game of Cylindrical Hex on a 3×6 board, during which Red used Algorithm 4.2.2. The first move was made by Blue and the game was won by Red.

Algorithm 4.2.2 (Red's winning strategy for Cylindrical Hex on a $3 \times n$ board)**.**
Consider the Cylindrical Hex graph $H_c(3, n)$. On their turn, Red follows the first applicable rule:

I. If Blue has not played yet, play on any cell.

II. If Blue's latest move was on cell (i, j), play according to the first applicable rule:

1. *in ring $j-1$, j or $j+1$, such that a red cell in ring j is touching a red cell in ring $j-1$ and one in ring $j+1$,*
2. *in ring j or $j+1$, such that a red cell in ring j is touching a red cell in ring $j+1$,*
3. *in ring $j-1$ or j, such that a red cell in ring j is touching a red cell in ring $j-1$,*
4. *in ring j,*
5. *anywhere.*

An example of a final position of a game of Cylindrical Hex in which this strategy was used by Red can be seen in Figure 4.5.

4.2.3 Cylindrical Hex Strategies on Other Cylindrical Hex Boards

In the previous section we saw an optimal strategy for Red on a $3 \times n$ Cylindrical Hex board, i.e. a $2k + 1 \times n$ board with $k = 1$. Unfortunately, this strategy does not work for boards with $k > 1$. On the $5 \times n$ board — for which $k = 2$ — for example, Red may end up in the situation shown in Figure 4.6 if they use the same strategy: The wider playing field allows Blue to slip through the cracks between Red's locally winning paths to create a winning cycle.

However, since Algorithm 4.2.1 is proven to be optimal for Red on any Cylindrical Hex board with an even number of columns, we expect that on any other non-trivial board with an odd number of columns, there also exists some strategy Red can use to win.

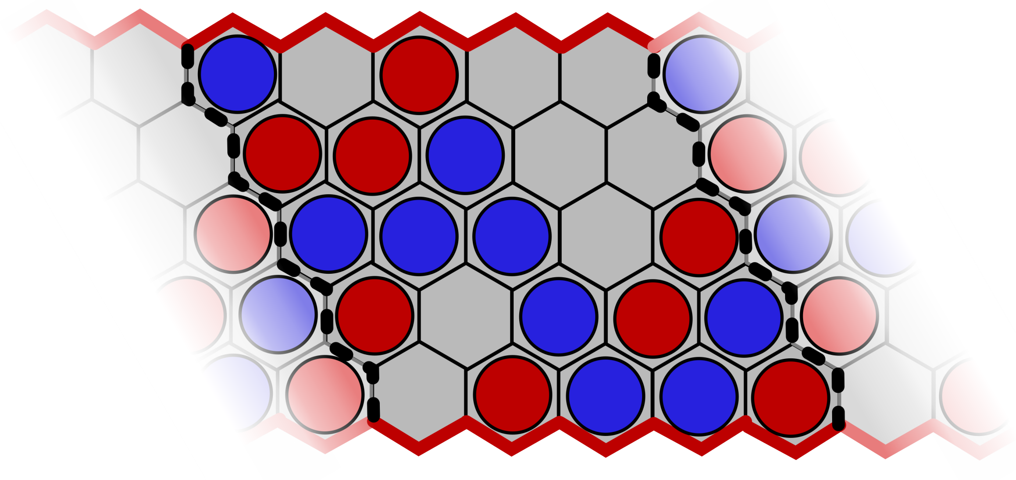


Figure 4.6: The final position of a game of Cylindrical Hex on a 5×5 board, during which Red used Algorithm 4.2.2. The first move was made by Blue and the game was won by Blue.

It seems likely that the only boards on which Red may not win are the trivial $1 \times n$ boards: On these boards all cells are winning moves for Blue. As such, Red can only win if they are the starting player and $n = 1$.

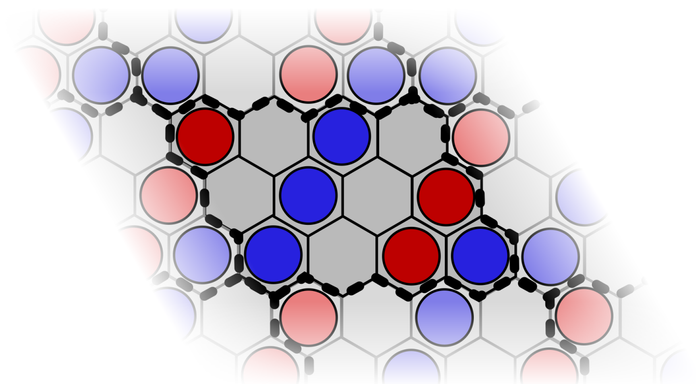
Attempts have already been made at finding an improved strategy for Red on a $5 \times n$ board. In Chapter 7 we will discuss one such strategy, which we then use as the jumping-off point for our own. However, to understand this algorithm, we must first become more familiar with some interesting positions one might encounter in a game of Hex, which will be discussed in Chapter 6.

While the original goal of this thesis was to find an improved strategy for Red on a $5 \times n$ board, we will also discuss a new optimal strategy for Red on a much more general type of board in Chapter 5, based on Gardner's strategies for asymmetrical Hex boards.

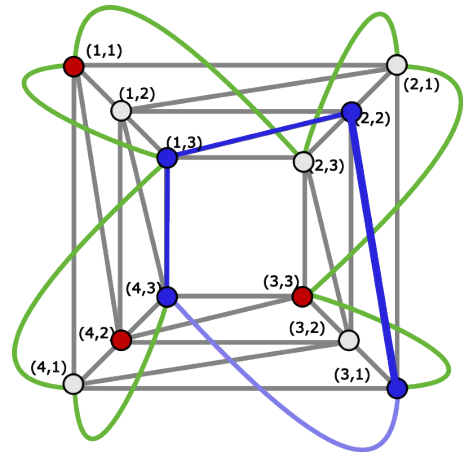
4.2.4 Cylindrical Hex Strategies on Torus Hex Boards

While Torus Hex is not the main focus of this thesis, it would, of course, also be nice to know how to win at that variation of the game. We will thus briefly discuss which of the Cylindrical Hex strategies discussed in Section 4.2 can also be used to guarantee a win in Torus Hex. Whether the Gardner strategy can be extended to Torus Hex will be discussed in Chapter 5.

Algorithm 4.2.2 (if adjusted to take into account the new connections on the board) is indeed optimal on the $3 \times n$ Torus Hex board. We will not prove this in detail in this thesis, as the proof would be almost an exact copy of either Huneke, Hayward, and Toft [8] or Van den Broek [13]'s proof of the optimality of the Cylindrical version of the strategy. Both of these proofs start by assuming that Blue has made a winning path, and then show, in different ways, that this could not have happened if Red used Algorithm 4.2.2, leading to a contradiction. As both proofs work on a 3×3 sub-board of the full $3 \times n$ board, they are easily translated to the toroidal board, without worrying about the newly added connections. Additionally, on the smaller boards (i.e. $n \leq 2$), where these connections could become an issue, the optimality of the strategy is almost trivial.



(a) A position on a 4×3 Torus Hex Board.



(b) A colouring of the graph $H_t(4, 3)$.

Figure 4.7: Example of a final position of a game of Torus Hex on a 4×3 board won by Blue, despite Red following an adjusted version of Algorithm 4.2.1.

Algorithm 4.2.1, on the other hand, is not guaranteed to be optimal in Torus Hex. This is because of the newly added connections, which allow Red and Blue to create paths that coil around each other, like the paths discussed at the end of Section 3.2. An example of a game in which Red lost a game of Torus Hex while using Algorithm 4.2.1 can be seen in Figure 4.7.

Chapter 5

Gardner-Based Strategies on Cylindrical and Toroidal Boards

When we introduced Cylindrical Hex in Section 2.2, we described how we could create the Cylindrical Hex board out of the regular one by attaching the sides of the board to each other. In Section 2.3, we created the Torus Hex board in a similar manner. Despite the additional connections, the original Hex board is still hidden within these new boards. It is only natural to wonder whether strategies that were effective on the original board could still be effective now.

In Section 4.1.2 we discussed the optimal strategy given by Gardner [5] for asymmetrical Hex boards. In this section, we will discuss whether this particular strategy can also be used for the other variations of Hex.

5.1 A New, Optimal Red Cylindrical Strategy on $m \times n$ Boards when $m > n$

In Section 2.2 we saw that we can lift the colouring of the Cylindrical Hex graph $H_c(m, n)$ to the graph S_n to determine whether there is a winning path. S_n by itself can also be seen as a regular Hex graph with infinitely many columns — a graph on which Red could win Hex using Algorithm 4.1.2.

This does not immediately mean that Red can always use this strategy to win Cylindrical Hex, of course. When used as an alternative interpretation of the Cylindrical Hex board, S_n has a repeating pattern to its colouring. When the board is not wide enough — i.e. when $m \leq n$ — this pattern will repeat too quickly: If we try to place the triangles used by Gardner's strategy on such an S_n, they will overlap somewhere on the cylindrical board. This leads to two potential mirror cells, as shown in Figure 5.1. As Red can only play on one of these mirrors on their turn, they are unable to create a true mirrored path.

However, when this overlap doesn't occur (i.e. when $m > n$), Red can use the algorithm to prevent Blue from creating a path crossing the first $n + 1$ columns of the cylinder — which makes it impossible for Blue to fully encircle the cylinder and win.

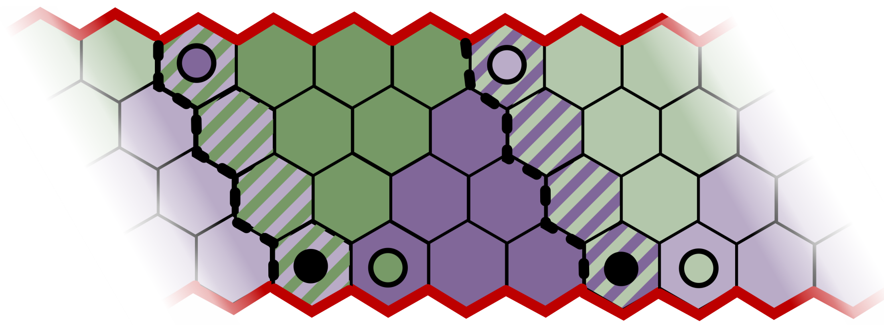


Figure 5.1: A 4×4 Cylindrical Hex board on which the green and purple triangles of the mirroring algorithm are shown to overlap in the first column of the cylinder. The purple and green circles indicate the two mirrors of the cell indicated by the black circle.

Algorithm 5.1.1 (Red's winning strategy for Cylindrical Hex on an $m \times n$ board with $m > n$)**.** *Consider the Cylindrical Hex Graph $H_c(m, n)$.*

Divide the vertices of this graph into three sections: Split the vertices of the first $n + 1$ columns into two mirroring triangles, of which each side consists of n vertices. If $m > n+1$, the remaining $m - (n + 1)$ rows form a slanted rectangle.

Label the vertices of the two triangles so that the labels mirror each other. We call the mirror of some cell (i, j), the mirror cell (i^*, j^*).

On their turn, Red follows the first applicable rule:

I. If Blue has not played yet, play on any cell in one of the two triangles.

II. If Blue's latest move was on cell (i, j):

1. If (i, j) was on one of the triangles, and its mirror, (i^*, j^*), has not yet been played: play on (i^*, j^*),

2. If (i, j) was on one of the triangles, and its mirror, (i^*, j^*), has already been played, play on any cell in one of the two triangles,

3. If (i, j) was on the slanted rectangle, play on any cell in one of the two triangles,

Theorem 5.1.1 (The Cylindrical Gardner Strategy)**.** *Algorithm 5.1.1 is an optimal strategy for Red on an $m \times n$ Cylindrical Hex board where $m > n$, that is to say: If Red uses this strategy on this board, they will always win.*

Proof (Optimality of Algorithm 5.1.1). Assume that, at the end of a game on an $m \times n$ Cylindrical Hex board with $m > n$, Blue has at least one winning path despite Red using Algorithm 5.1.1.

We can lift this winning path on the Cylinder to the graph S_n according to the lifting used to find winning paths in Cylindrical Hex.

For Blue to have a winning path in cylindrical Hex, the path must encircle the cylinder completely. This is equivalent with having a path that passes through every column of S_n in the aforementioned lifting. In particular, this path should pass through all columns of the subgraph $H_r(m, n)$ of S_n.

This subgraph is the regular Hex graph of the board that was used to create this Cylindrical Hex graph. Since the blue path we are examining is a path that passes through all columns of this graph, it must also be a winning path of a game of regular Hex on the board $H_r(m, n)$.

Note that if Blue makes the same moves in both games, the colouring of $H_r(m, n)$ based on the final position of a game of Cylindrical Hex in which Red used Algorithm 5.1.1 will be identical to the colouring of $H_r(m, n)$ in the final position of a game of Regular Hex in which Red used Algorithm 4.1.2. After all, the rules described in both strategies are identical.

Combining these facts, we have found that the restriction of our blue winning path to $H_r(m, n)$ must be a winning path for regular Hex and that this winning path can be formed by Blue in regular Hex, even when Red uses Algorithm 4.1.2 to respond. However, this is a contradiction: According to Theorem 4.1.1, it is impossible for Blue to create a winning path when Red uses this strategy.

Blue can therefore not have a winning path at the end of a game. Since Cylindrical Hex will always have a winner — see Theorem 3.1.2 or Theorem 3.2.2 — we conclude that Red will always win if they use Algorithm 5.1.1.

□

With the addition of Algorithm 5.1.1, we now have two optimal strategies that can be followed on Cylindrical Hex boards that have both an even number of columns and more columns than rows: namely Algorithms 4.2.1 and 5.1.1. We will compare the efficiency of these strategies in Section 9.2.

5.2 Blue Cylindrical Gardner is Not Optimal

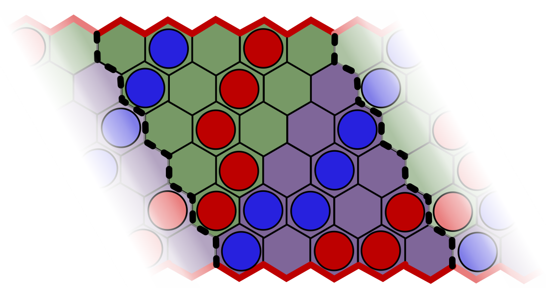


Figure 5.2: Example of a final position of a game of Cylindrical Hex on a 5×6 board won by Red, despite Blue following an adjusted version of Algorithm 4.1.1.

Since Gardner's algorithm still works on a cylinder for Red, it makes sense to wonder if the same might be true for Blue. Luckily for Red, this is not the case: Red can still win a game of Cylindrical Hex if Blue uses (an appropriately adjusted version of) Algorithm 4.1.1.

While Gardner's Algorithm is optimal for regular Hex, it doesn't quite work for Blue's new winning condition in Cylindrical Hex. Blue can still use the algorithm to guarantee a path from the first column of the board to the last — which is still a prerequisite for a blue winning path. However, this is no longer enough to win: The algorithm has no way to ensure the path ends up actually encircling the cylinder.

The algorithm also does not take into account the fact that Red can now use the extra connections between the first and last columns of the board. It is these connections that Red must use if they want to win against a Blue opponent using Gardner's algorithm. After all, if Red aims to create a path that does not use these connections, we may as well be considering a regular game of Hex — where we have already seen that Blue's strategy is optimal.

When Red does use the new connections, however, they do stand a chance against Blue. This can be seen in Figure 5.2: While Blue was able to create a path from the first to the fifth column by using Gardner's algorithm — which would have won them the game in regular Hex — their path does not fully encircle the cylinder. Red's path, meanwhile, connects the top and bottom of the cylinder using one of the new connections: Red has won.

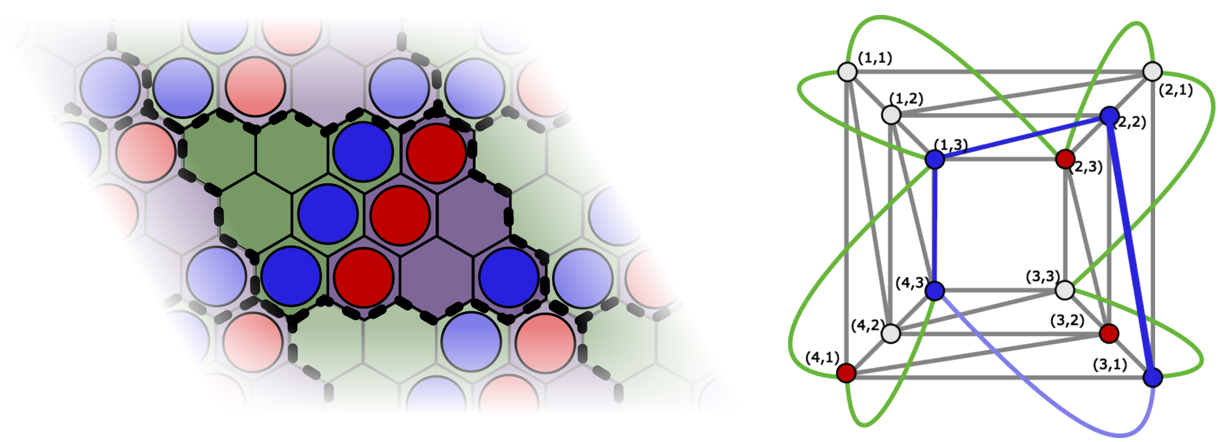
5.3 Torus Gardner is Not Optimal

Having seen some success on the cylindrical board, we may also wonder if we can further extend the strategy to Torus Hex.

For Blue, this is not possible. As we have seen in the previous section, the strategy already fails in Cylindrical Hex. Introducing the extra connections present in Torus Hex will not fix the issues causing this failure.

In fact, the counterexample used to prove Blue can no longer use Gardner's strategy in Cylindrical Hex — shown in Figure 5.2 — would also work as a counterexample for Torus Hex. Since Blue's winning condition does not change in Torus Hex, the Blue path in this figure is still not a winning path. The Red path, on the other hand, is still a winning path, even according to Red's new winning condition: The two ends of the path on the cylindrical board would line up perfectly on the toroidal board, creating a red winning cycle.

Unlike Blue, Red was still able to use Gardner's strategy in Cylindrical Hex. Unfortunately, this does not mean that we can extend its use further. When playing on a toroidal board, Red faces the same issues that prevent Blue from using this strategy on a cylindrical board: The algorithm does not take into account that the ends of a path now need to connect to each other, and also doesn't consider that Blue may use the newly added edges between the first and last ring to avoid Red's blockade.



(a) A position on a 4×3 Torus Hex Board.

(b) A colouring of the graph $H_t(4, 3)$.

Figure 5.3: Example of a final position of a game of Torus Hex on a 4×3 board won by Blue, despite Red following an adjusted version of Algorithm 4.1.2.

By using the new connections on the board, Blue is able to beat Red in Torus Hex, even if Red uses Gardner's strategy. This can be seen in Figure 5.3: By using Gardner's algorithm, Red does make a path from the lowest to the highest ring in the first three moves of the game, but as the ends of this path do not connect to each other on the torus, they have not won yet. Instead, Blue has the chance to make one more move, which connects the ends of the blue path using one of the board's new edges. Thus, Blue has achieved their win condition, before Red has the chance to connect the ends of their own path.

Chapter 6

Positions of Interest

We will now discuss various interesting positions that can occur on the board throughout a game of Hex. The first of these positions, so-called bridges, play a very important role in the algorithm by Van den Broek [13] — which we will discuss in Chapter 7 — while the other positions will be used to further improve this algorithm in Chapter 8.

6.1 Bridges

Bridges are an incredibly useful strategy for Hex. While their usefulness in Hex was already known by Piet Hein [9], they were first exploited in an algorithm for Cylindrical Hex by Van den Broek [13], which we will discuss in Chapter 7. Bridges can be defined as follows:

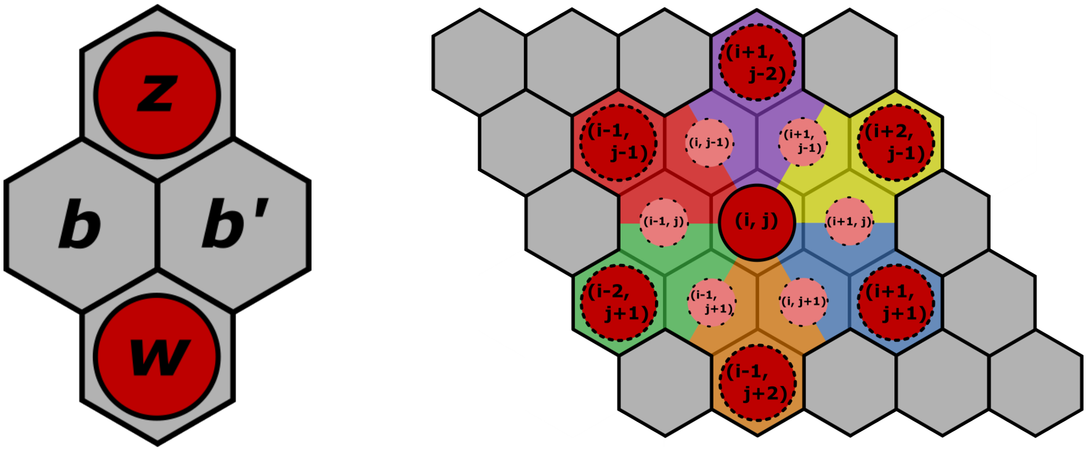
Definition 6.1.1 (Bridges)**.** *Two non-adjacent vertices $z, w \in V_{m,n}$ of the same colour are connected by a bridge if there are two uncoloured vertices $b, b' \in V_{m,n}$, such that the following all hold:*

- *b is adjacent to both z and w.*
- *b' is adjacent to both z and w.*
- *$b \neq b'$.*

b and b' are the bridge vertices connecting the bridge's end vertices z and w.

Bridges allow a player to extend an existing path more quickly than they ever could using only adjacent vertices. Care should be taken in the placement of these bridges — as we will discuss in more detail in Section 6.2 — but if used correctly, these bridges can create relatively safe *indirect connections* between vertices.

Consider a red bridge connecting the vertices z and w, like the one in Figure 6.1a. By definition, the bridge gives Red two disjoint ways of directly connecting z and w in the future; namely by playing on either of the bridges vertices, b and b'. While z and w are therefore not directly connected by a path of adjacent nodes, their indirect connection is still quite safe: If Blue plays on either one of



(a) An example of a bridge on a Hex board.

(b) All 6 possible bridges that can be made from the cell (i, j). Cells with a red stone are the end vertices of the bridges, while cells with a smaller pink stone are the bridge vertices.

Figure 6.1: Bridges, depicted on (portions of) a Hex board

the bridge vertices, Red can immediately 'save' the bridge by playing on the other and creating a direct connection.

The only time Blue can completely 'destroy' a red bridge is when Red decides not to save it — for example, when Red has more pressing matters to attend to elsewhere on the board, or when multiple bridges are threatened at once (see Section 6.2).

While making direct connections to neighbouring cells will always remain the *safest* option, being able to reach beyond the direct neighbours of a cell can be incredibly useful: Bridges allow a player to extend or connect existing paths more quickly than they could using only adjacent vertices.

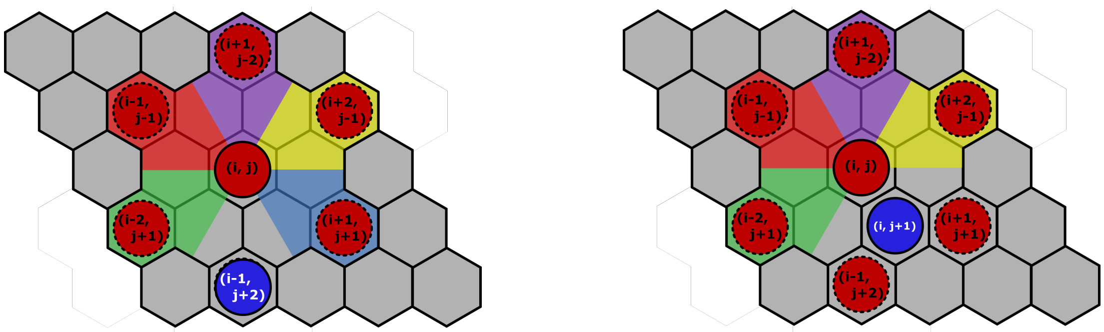
Consider the bridge from Figure 6.1a again. Using only direct connections, it would take Red 2 moves to connect an already red cell w to an uncoloured cell z: on move 1 Red plays on either b or b' and on move 2 Red plays on z, thereby connecting w to z directly using a red path. Meanwhile, using indirect connections, Red only needs one move to connect these vertices: Immediately playing on z creates a red bridge, and thereby an indirect connection, between the two cells.

This quicker method of reaching far-away vertices can be used to create 'provisional' paths — for example, between different existing paths or to the edge of the board — consisting of a chain of bridges. Since the bridges can not be destroyed in a single move, Red can focus on creating this provisional path and return later — when Blue attacks one of the bridges or when they have finished preparing their intended path — to fill in these bridges. We will discuss these types of paths in more detail in Section 6.4.

In the best-case scenario, utilising bridges may allow a player to halve the number of moves needed to reach a certain vertex from an already existing path. In actual play, it is very unlikely that a player will be able to perfectly halve the number of moves needed to reach a particular cell. Earlier moves may form an obstacle and an opponent — especially one that is also familiar with bridges — will may make moves that limit bridging options.

For example, if cell (i, j) is Red, Blue can prevent one potential red bridge by playing on any of the other red cells in Figure 6.1b. By playing on one of these cells, Blue prevents Red from using this cell as an end vertex and thus reduces the number of bridges Red can make by 1, as can be seen in Figure 6.2a.

Alternatively, Blue could play on any of the pink cells adjacent to (i, j): Since each of these cells could potentially become a bridge vertex for two different red bridges, playing on any of them reduces Red's potential bridges by two, as can be seen in Figure 6.2b. Playing on a cell that is adjacent to that of your opponent does also limit your potential number of bridges, however, so it may not (always) be the best strategy.



(a) Blue prevents one Red bridge by playing on a potential red end vertex.

(b) Blue prevents two Red bridges by playing on a potential red bridge vertex.

Figure 6.2: The two ways in which Blue can limit the number of bridges Red can make from cell (i, j).

6.2 Overlapping Bridges

While bridges can create a relatively safe indirect connection between two non-adjacent vertices, they can also carry risks if used recklessly. The point of a bridge is that they create indirect connections that can easily be saved if threatened. However, if bridges are placed in a certain way, it is possible for an opponent to threaten multiple bridges at once.

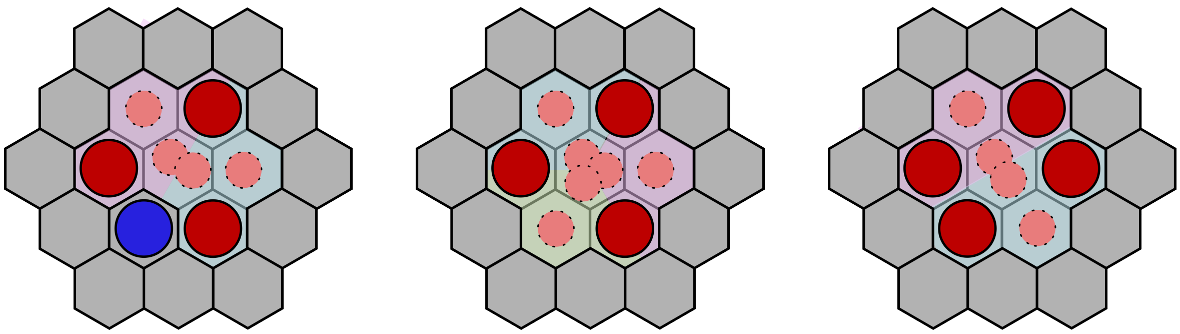
Definition 6.2.1 (Overlapping bridges)**.** *An overlapping bridge is any bridge that shares a bridge vertex with a different bridge of the same colour. The vertex that is part of both bridges is called the point of overlap.*

These overlapping bridges can occur in two ways:

1. *Two bridges share exactly one end-vertex.* This is the least stable type of overlapping bridge. If an opponent attacks the point of overlap in this case, saving one bridge will not save both. The player being attacked must choose which bridge they wish to save (if any) and the opponent will then have the opportunity to destroy the other bridge. Unless the opponent chooses not to destroy the other bridge, it is only possible to save one of these bridges. As a result, only two of the three involved end vertices can be connected to each other. An example of this type of overlapping bridge can be seen in Figure 6.3a.

In a special case of this type of overlapping bridge, shown in Figure 6.3b, it is also possible for the point of overlap to be part of three distinct bridges. This occurs when each of the three involved bridges shares each vertex with one of the other involved bridges. If an opponent attacks in this case, it is possible to save at most two out of three bridges, unless the opponent changes their plan partway through. Since this is enough to connect all involved cells, this special case is a lot less dangerous.

2. *The bridges are parallel to each other.* In this case, one need not be concerned: There must exist a direct connection between the two bridges on each end for this to be possible. As a result, if an opponent plays on the point of overlap, saving either bridge will connect all involved end vertices in one move. An example of this type of overlapping bridge can be seen in Figure 6.3c.



(a) Two bridges that share one of their end-vertices.

(b) Three bridges, each pair of which shares one of their end-vertices.

(c) Two parallel overlapping bridges.

Figure 6.3: The three ways in which overlapping bridges can occur.

6.3 Double-Bridge Set-Ups

We have seen that a single bridge can be enough to create a relatively safe path between three rows or columns using just two moves. If this bridge is connected to another bridge, we can reach even further: As shown by the white and black pairs of consecutive bridges in Figure 6.4, two consecutive bridges can already create a path between up to five rows or columns in just three moves.

Such a path of two consecutive bridges can be created in two different ways: A player can create the bridges one by one, or they can set up the board in such a way that they can create both bridges simultaneously.

The first of those methods is the most straightforward: The player can create a single bridge over the course of two moves and can use their third move to add the second bridge from either of the end vertices. This method is more likely to allow a player to adjust their strategy if their opponent interferes. For example, Red could try to play (i, j), $(i+1, j-2)$ and $(i+2, j-4)$, in that order, to create the white pair of bridges in Figure 6.4. If Blue interferes by playing $(i+2, j-4)$ after Red's second move, Red can still bridge to $(i, j-3)$ or $(i+3, j-3)$ instead, thereby still covering quite a large distance. Red could even decide to extend the bridge from the other end vertex of the existing bridge, by playing $(i-2, j+1)$, $(i+1, j+1)$ or $(i+1, j+2)$.

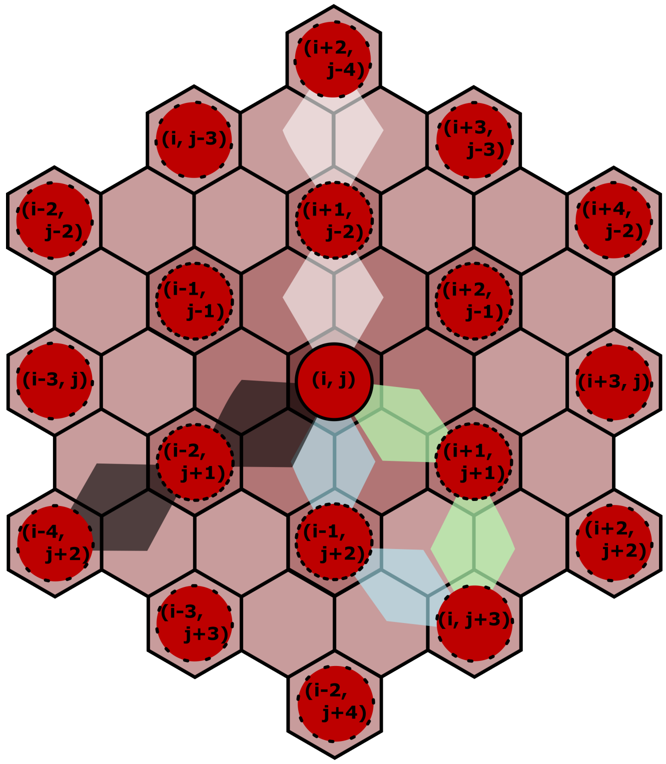


Figure 6.4: All possible cells that can be reached from the cell (i, j) using at most two bridges. Four possible double-bridges are highlighted: the white double-bridge covers five rows of the board; the black double-bridge covers five columns of the board; the green and blue double-bridges both end at the same cell, but use a different intermediary cell.

The second method requires a player to first play both end vertices of the intended path and then play on the correct cell between them, creating two bridges in one move. In the first two moves of this method, the player will have created the set-up needed the create these double-bridges.

Definition 6.3.1 (Double-Bridge Set-Ups)**.** *Two vertices $z, w \in V_{m,n}$ of the same colour form a double-bridge set-up with uncoloured intermediary vertex b_i if there are four uncoloured vertices $b_1, b_1', b_2, b_2' \in V_{m,n}$, such that the following holds:*

Colouring vertex b_i the same colour as vertices z and w creates two non-overlapping bridges: the first bridge connects z and b_i using bridge vertices b_1 and b_1' and the second bridge connects b_i and w and using bridge vertices b_2 and b_2'.

b_i is called the intermediary vertex and b_1, b_1', b_2, b_2' are the potential bridge cells.

By creating a set up like this, we lose some room for adjustments, especially when the end cells and intermediary cell form a straight line. For example, Red could also try to create the white bridges by playing (i, j), $(i + 2, j - 4)$ and $(i + 1, j - 2)$, in that order. However, if Blue interferes by playing $(i + 1, j - 2)$, or even any of the potential bridge cells, Red will no longer be able to make far-reaching connections on their third move.

The situation is slightly better when the end cells and intermediary cell do not form a straight line, as this means that there may be two possible intermediary cells. For example, Red could try to create the green bridges by playing (i, j), $(i, j+3)$ and $(i+1, j+1)$, in that order. If Blue interferes by playing $(i+1, j+1)$, Red could instead play $(i-1, j+2)$, thereby creating the blue bridges instead and still connecting the two end nodes.

This does not mean that such pairs of end cells do create a safe set-up. Given such a pair of potential double-bridges — for example, the green and blue bridges in Figure 6.4 — an opponent can still prevent both after the end vertices have been played, by playing on one of the two cells that are potential bridge cells for both intermediary cells.

However, despite the possibility of the double-bridge set-up being thwarted, it can still be a useful tool for a player, for example when trying to connect two separate paths.

Consider Figure 6.5. Given the choice between the two cells indicated by the smaller circles — both lengthening the upper path so that it covers one extra ring — the one on the right is a reasonable choice for Red's next move. Looking at the board, we can clearly see that it brings the end of the upper path closer to the lower path, making it easier to connect the two in the future. While this is easy to see when looking at the board, the creation of a double-bridge set-up also suggests that this is the case in a slightly more tangible way: Because of the set-up, we know the two paths are close enough that a single move could connect them — which is certainly not the case for the left move.

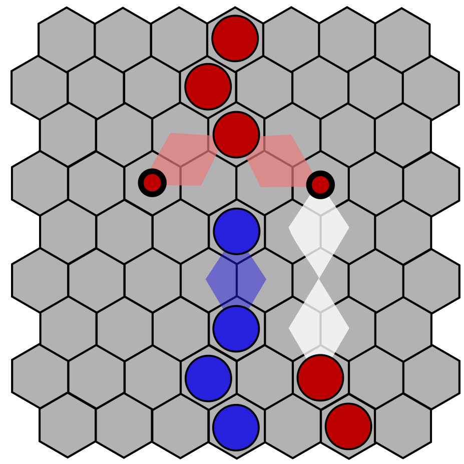


Figure 6.5: A situation that could occur during a game of Hex, with two potential future red moves and their resulting bridges and double-bridge set-ups highlighted.

In this situation, the actual goal of the move is to lengthen an existing path, which both moves (as well as either of the cells between the highlighted moves) achieve. Looking at which move would create a double-bridge set-up can be of use in deciding between the available options: The creation of this set-up — or, more accurately, the accompanying confirmation that the move brings two paths closer to each other and the possibility of connecting them sooner rather than later — acts as a point in favour for the right-most option.

6.4 Provisional Winning Paths

As mentioned in Section 6.1, we can use bridges to create a provisional path, like the one in Figure 6.6. Such provisional paths consist of at least two normal (i.e. directly connected) paths on the board, and one or more bridges that create indirect connections between the vertices of these paths.

Definition 6.4.1 (Provisional Paths)**.** *There exists a provisional path P of some colour c between two vertices a and b on the board if there exists a collection of k c-coloured paths $P_1, ...P_k$, and $k-1$ bridges — of which the pairs of end vertices can be written as $(z_1, w_2), ..., (z_{k-1}, w_k)$ — such that*

- *a is an end vertex of path P_1,*
- *$z_i \in V_{m,n}$ is an end vertex of path P_i for all $i \in \{1, ..., k-1\}$,*
- *$w_i \in V_{m,n}$ is an end vertex of path P_i for all $i \in \{2, ..., k\}$,*
- *b is an end vertex of path P_k.*

The provisional path P consists of paths $P_1, ..., P_k$ and the $k-1$ bridges between these paths.

A provisional path can be seen as the sketch for a player's intended path. Once each of the bridges along this sketch has been played on, the path will become a normal path.

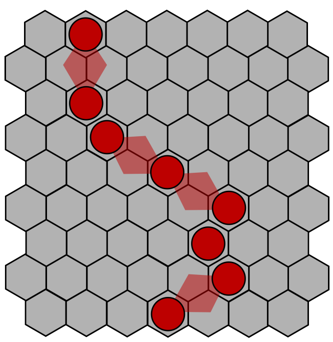


Figure 6.6: An example of a red provisional path. As none of the bridges overlap, this would be a true provisional winning path on a board of 9 rows or rings.

The only way for the opponent to prevent this intended path is by destroying one or more of the bridges along the provisional path. However, we have already seen that non-overlapping bridges can always be saved. This means that while a player may decide to abandon a threatened provisional path if they have other matters to attend to, they will always be able to save it.

A provisional winning path is especially interesting if it has the possibility of becoming a winning path.

Definition 6.4.2 (Provisional Winning Paths)**.** *A provisional winning path is a provisional path that will become a winning path once the owner of the path has played on at least one of each of the bridges' bridge vertices.*

If a provisional winning path does not contain any pairs of bridges that share a bridge vertex, it is called a true provisional winning path.

Note that once a player has created a true provisional winning path, their winning strategy from that point onwards is very simple: If their opponent plays on the bridge vertex of any of the bridges along this path, the player should play on the other bridge vertex of that same bridge. As there are no overlapping bridges, the opponent can only attack one bridge of this path at the same time. By saving this particular bridge, the owner of this path thus also saves the path itself. Should the opponent play anywhere other than on one of these bridges, the owner of the path can play on any of the bridge vertices along the provisional path, thereby slowly turning it into a winning path.

Chapter 7

An Earlier Attempt at Creating an Optimal $5 \times n$ Cylindrical Hex Strategy

An optimal strategy for Red on a $5 \times n$ Cylindrical Hex board has not been found yet for all values of n.

While Algorithm 5.1.1 does work for Cylindrical Hex boards made of up to $n = 4$ rings, it does not guarantee a win on longer boards. A version of Algorithm 4.2.1, adjusted for boards with an uneven number of columns — that is to say, a version wherein one plays on the cell k columns further along than Blue's latest move, for some fixed value of k — does not work either, as was shown by Van Hees [14]. Likewise, Figure 4.6 showed us that Algorithm 4.2.2 is also not optimal on boards that have five columns instead of three.

The most effective algorithm that has been found so far is the one by Van den Broek [13]. In this section, we will discuss this particular algorithm in more depth and look at ways the algorithm could still be improved.

7.1 The Algorithm Suggested by Van den Broek [13]

Algorithm 7.1.1 given below — the algorithm put forward by Van den Broek [13] — is largely based on Algorithm 4.2.2 — the algorithm for the $3 \times n$ board that was discussed in Chapter 4 — but with some added features. Most of these features take advantage of the bridges we discussed in Section 6.1.

Rules 1 to 3 of Algorithm 4.2.2 focus on creating direct connections between the ring on which the most recent move was made and the ring directly above and/or below it. Algorithm 7.1.1 expands this idea by using bridges to reach even further: In rules 3 to 8 of Algorithm 7.1.1, the goal has become creating direct *or indirect* connections between the most recently played on ring and *up to two rings* directly above and/or below. In addition, rules 9 to 12 are added to lengthen existing paths that are just barely out of reach of these preceding rules — doing so in such a way that, on Red's next turn, a bridge covering the most recently played on ring can be made.

With this new bridge-oriented strategy, it also becomes important to protect existing bridges when they are attacked (rule 2) and to fill in bridges when there is no path left to lengthen (rule 18).

In addition to taking advantage of bridges, Algorithm 7.1.1 also introduces steps to lengthen an existing path (rules 14 to 17), and encourages Red to immediately end the game when a winning move exists (rule 1).

Finally, during a game in which Red begins, the algorithm requires Red to make their first move in the most central ring possible instead of anywhere on the board. Playing on such a central position should give Red a more favourable starting position than any other cell would.

Formally, the algorithm can be described as follows:

Algorithm 7.1.1 (Van den Broek [13]'s Red strategy for Cylindrical Hex on a $5 \times n$ board)**.** *Consider the Cylindrical Hex graph $H_c(5, n)$. On their turn, Red follows the first applicable rule:*

I. *If Blue has not played yet, play in the middle of the board.*

II. *If Blue's latest move was on cell (i, j), play according to the first applicable rule:*

    1. *if there is a cell (x, y) that is a winning move: in cell (x, y),*
    2. *if cell (i, j) was part of an existing red bridge: on the other bridge cell of this bridge,*
    3. *in ring $j-2$, $j-1$, j, $j+1$ or $j+2$, such that these rings become (indirectly or directly) connected,*
    4. *in ring $j-1$, j or $j+1$, such that these rings become (indirectly or directly) connected,*
    5. *in ring j, $j+1$ or $j+2$, such that these rings become (indirectly or directly) connected,*
    6. *in ring $j-2$, $j-1$ or j, such that these rings become (indirectly or directly) connected,*
    7. *in ring j or $j+1$, such that these rings become (indirectly or directly) connected,*
    8. *in ring $j-1$ or j, such that these rings become (indirectly or directly) connected,*
    9. *in cell $(x, j-1)$, such that rings $j-2$ and $j-1$ become indirectly connected and $(x, j-1)$ has a possibility of forming a bridge with ring $j+1$,*
    10. *in cell $(x, j+1)$, such that rings $j+1$ and $j+2$ become indirectly connected and $(x, j+1)$ has a possibility of forming a bridge with ring $j-1$,*
    11. *in cell $(x, j-1)$, such that $j-3$ and $j-1$ become indirectly connected and $(x, j-1)$ has a possibility of forming a bridge with ring $j+1$,*
    12. *in cell $(x, j+1)$, such that $j+1$ and $j+3$ become indirectly connected and $(x, j+1)$ has a possibility of forming a bridge with ring $j-1$,*
    13. *in ring j,*
    14. *if ring k is the closest end of the existing path encompassing ring j, in ring $k+2$, such that these rings become indirectly connected,*
    15. *if ring k is the closest end of the existing path encompassing ring j, in ring $k-2$, such that these rings become indirectly connected,*

16. *if ring k is the closest end of the existing path encompassing ring j, in ring $k+1$, such that these rings become (indirectly or directly) connected,*
17. *if ring k is the closest end of the existing path encompassing ring j, in ring $k-1$, such that these rings become (indirectly or directly) connected,*
18. *on a bridge vertex of any existing red bridge,*
19. *anywhere.*

7.2 Shortcomings of Algorithm 7.1.1

We will now discuss a few shortcomings in the algorithm introduced above. Some of these were already noted as points of improvement by Van den Broek [13].

7.2.1 Gaps Between Existing Local Paths

In her own work, Van den Broek [13] already notes some of the limitations of this algorithm. Most notably, she points out that the algorithm does not actively try to connect existing local paths. Moves are made based only on how useful they are locally, which means that while the algorithm may connect disjoint local paths when they happen to be close enough to each other, no effort is put into making this possibility more likely.

Blue can use this fact to their advantage: By encouraging Red to create two separate local paths, Blue can ensure that there is a gap between these two paths through which they could make their winning path.

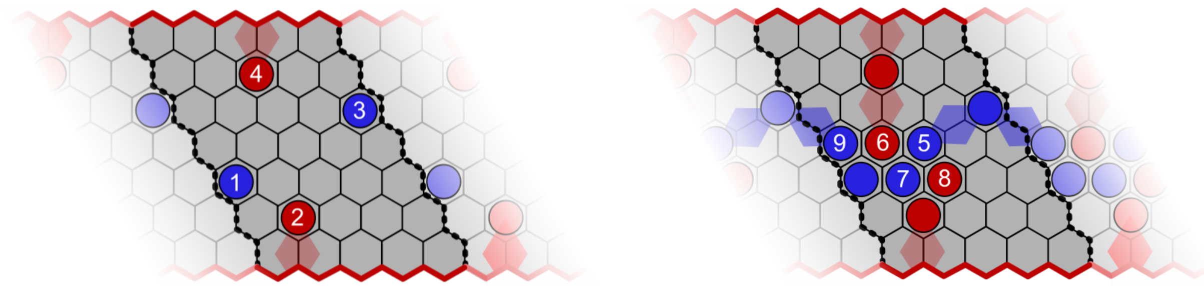
This situation may even be exacerbated by rules 9 to 12. When these moves are used, we may end up in a situation where ring j — i.e. the ring which Blue most recently played in — both does not contain a red cell and is not covered by a red bridge. Under the right circumstances, this could create an even bigger gap for Blue to slip through (compared to the gap that would be created if rule 13 was used instead).

An example of such a large gap is shown in Figure 7.1a. In this case, Blue has taken advantage of rules 9 to 12 twice — once in the top half of the board and once in the bottom half. The resulting gap between the two red provisional paths is not only three rings wide; it also contains both of Blue's moves. The continuation of this game will discussed in Section 7.2.3.

7.2.2 Not Anticipating Blue's Strategy

In addition to pointing out the gaps between disjoint paths, Van den Broek [13] notes that Algorithm 7.1.1 does not take into account that Blue may also make use of bridges in their strategy. This statement can even be made more general: the algorithm does not take into account *any* connections that Blue has made, nor does it anticipate what Blue may do next.

In a vacuum, the algorithm seems well-suited for creating a local wall to block Blue's path. If rule 3 is used after Blue has played on cell (i, j), the existence of a Red provisional path that connects rings $j-2$ up to $j+2$ ensures that, on their next move, Blue will not be able to create a path that goes around this red path from their last-played cell: Any bridge they make from that cell can at



(a) The board after the initial four moves:
1B: $(1,5)$, *2R:* $(2,6)$ *(rule 12)*,
3B: $(5,3)$, *4R:* $(3,2)$ *(rule 11)*.

(b) The board five moves later:
5B: $(3,4)$, *6R:* $(2,4)$ *(rule 5)*,
7B: $(2,5)$, *8R:* $(3,5)$ *(rule 8)*,
9B: $(1,4)$.

Figure 7.1: Two positions from a game on a 5×7 Cylindrical Hex board, in which Red follows Algorithm 7.1.1 and Blue uses a double-bridge set-up to take advantage of the resulting gap between two disjoint Red paths. Since Blue has a true provisional winning path by move 9, it is clear that the game will ultimately be won by Blue.

most reach up to ring $j-2$ or $j+2$, where the red local path still blocks the way. From that point, the algorithm will most likely extend this local path even further, once again blocking off Blue's future plans. It is therefore very sensible to first focus on at least connecting as many of the rings mentioned in rule 3 as possible, before moving on to lengthening an existing path in a more general way.

However, the algorithm does not take into account any other moves Blue might have made before. This creates a risk, as the existing blue cells could very well form a provisional path that allows Blue to circumvent the red local path. In the worst case scenario, Blue may even be about to create a winning path and the algorithm — which does not take this possibility into account — is not guaranteed to interfere.

7.2.3 Double-Bridges

As a special case which combines the first two issues we discussed, a double-bridge set-up can be a very threatening move against Red, especially on a board that is only 5 columns wide. After all, a double-bridge set-up can cover all columns of such a board, as can be seen from the black double-bridge in Figure 6.4.

If Blue creates such a set-up at the right place within a gap between disjoint Red paths, they are able to slip through the gap by playing on the intermediary cell before Red is able to defend against it. Blue will then suddenly only have a very small gap to cover, and, if done right, plenty of ways to cover this gap.

Looking at the game from Figure 7.1a again, we can see that such a set-up was also made in a (for Blue) very advantageous spot: The two blue cells form a double-bridge set-up contained entirely in the three ring wide gap between the red provisional paths. The game's continuation in Figure 7.1b shows how easily Blue can take advantage of such a position: After playing on the intermediary cell to create two bridges (move 5) and protecting a bridge that has been attacked by

Red (move 7), Blue is able to make a true provisional winning path and is thereby guaranteed a win if they continue to play optimally.

7.2.4 Overlapping Bridges

Since Algorithm 7.1.1 does not pay attention to whether bridges overlap or not, overlapping bridges occur more often than Red would like.

It seems that one particular way in which they are often formed is as a result of the way in which the algorithm tries to connect rings to each other. Figure 7.2 shows an example of how this may happen.

In the situation shown in Figure 7.2a, ring j is already connected to rings $j-2$ and $j+2$. While rings $j-1$ and $j+1$ are not truly connected yet, these rings seem to be quite safe for now: Both are already covered by a bridge, and will therefore become connected as soon as Blue tries to attack this bridge.

Despite this relative safety, Algorithm 7.1.1 will still try to create a connection to these rings using rule 7. This is not necessarily an issue, as this connection can be made in a safe manner: Two of the four moves satisfying rule 7 are on the bridge between rings j and $j+2$. While playing on either of these cells may not lengthen the provisional path, it would make the path more secure, as there is one less bridge that Blue could attack.

That being said, the other two valid moves, which are equally likely to be chosen by this algorithm, each create an overlapping bridge. The right-most move creates three overlapping bridges between three red vertices. While this is not ideal (especially when there are two moves that connect these rings without any bridges), it is also not the end of the world: As we briefly discussed in Section 6.2, Red will always be able to save two out of the three bridges involved in this configuration over the next few moves, which will be enough to maintain the connection between all end vertices.

It is the left-most move that can cause the most problems in this situation. In this case, ring $j+1$ becomes connected to ring j through a series of two bridges: the first being a new bridge from ring $j+1$ to ring $j+2$, the second being the existing bridge from ring $j+2$ to ring j. The resulting overlapping bridge is shown in Figure 7.2b. As discussed in Section 6.2, this particular type of overlapping bridge is a lot less stable, as only one of these bridges can be saved if Blue plays on the point of overlap.

This method of unintentionally creating an overlapping bridge is not limited to rule 7, of course. Any of rules 3 to 8 could lead to similar situations.

7.2.5 Provisional Paths

Algorithm 7.1.1 does implicitly try to create a provisional winning path: In addition to creating paths locally using rules 3 to 8 and rule 13, Red also uses rules 14 to 17 to extend existing paths. If Blue does not manage to interfere with Red's plans, these rules will eventually lead to the formation of a red provisional winning path and once such a path exists, several rules of this algorithm will also encourage Red to fill in the bridges along the path, to eventually turn this provisional winning path into a winning path. However, the algorithm is not guaranteed to do so in an efficient or safe manner.

(a) The four moves Red can make in response to Blue's move (i, j), according to rule 7. The two outermost moves would create an overlapping bridge

(b) The board after Red has made the left-most possible move in response to move (i, j). The move has created the least stable type of overlapping bridge.

Figure 7.2: Two positions from a game on a $5 \times n$ Cylindrical Hex board, in which Red may create an overlapping bridge by following Algorithm 7.1.

For example, if there is a bridge along the provisional winning path that covers 3 rings, then rules 3 to 8 could lead to Red playing on one of the bridge vertices — moves that are equivalent to the middle two options highlighted in Figure 7.2a. However, as we have already seen in Figure 7.2a, there is no guarantee that Red will play on either of these cells: As long as other options exist that satisfy this rule, Red is just as likely to play on one of these alternatives. These alternatives usually do not help in turning a provisional winning path into a winning path and may instead create a new overlapping bridge.

This new overlapping bridge is not necessarily an issue if a provisional winning path already exists: If Blue does play on its point of overlap, Red can still use rule 2 to save the provisional winning path by saving the bridge that was part of this provisional path initially. However, we again have to note that there is currently no guarantee that Red will pick the correct move in this situation — they may choose to save (one of) the other overlapping bridge(s) instead, which could cause them to lose their provisional winning path.

If there exists a provisional path and the local rings (i.e. rings $j - 2$ to $j + 2$) are all already directly connected, then we expect Red to play according to rule 13 unless all cells in ring j have been played already. This will also not help in turning the existing provisional winning path into a winning path and only wastes time.

In fact, only when all cells in ring j have been played on already, does this algorithm begin to actively fill in the remaining red bridges using rule 18. However, while actively filling in bridges is the most efficient move to make when a provisional winning path exists, we again have no guarantee that Red will make the most efficient choice in which bridges to fill in and how: Instead of prioritising the point of overlap of overlapping bridges — thereby turning the provisional winning path into a true winning provisional path and creating more security — or at least focusing on those bridges that are part of this provisional path, Red may end up wasting time by playing on other bridges on the board.

In short, while the rules in this algorithm can certainly help with the creation and filling in of a provisional winning path, they are not designed to actively take this path into account and are therefore just as likely to get in the way and introduce unnecessary danger to these paths.

Chapter 8

A New $5 \times n$ Cylindrical Hex Strategy

Based on the shortcomings discussed in Chapter 7.2, a new algorithm was created. While this algorithm is heavily inspired by Algorithm 7.1.1, a major change was made to its structure. In addition, various changes were made to mitigate or fix the shortcomings discussed in Chapter 7.2 and some further changes were made to take advantage of new possibilities introduced by other changes. We will discuss these changes in detail before introducing the new algorithm — Algorithm 8.4.1 — in Section 8.4.

Throughout this section, when talking about multiple consecutive rings on the board, we will use the following notation:

Definition 8.0.1. *Let $a \leq b$. We define the set of rings from ring a to ring b as follows:*

$$R(a, b) = \{ring\ c \mid \bar{a} \leq c \leq \bar{b},\ \bar{a} = \max\{a, 0\},\ \bar{b} = \min\{b, n+1\}.$$

8.1 Major Change to Algorithm Structure

While looking for situations in which Algorithm 7.1.1 lost or came close to losing, it became apparent that many dangerous situations could have been avoided, or at least mitigated, by choosing a different move that still satisfied the same rule. The situation from Figure 7.2 (discussed in Section 7.2) is an obvious example of this: Since each move in ring $j+1$ satisfies rule 7, Algorithm 7.1.1 considers them be equally valid, despite two of the options unnecessarily creating an overlapping bridge, which Blue could easily take advantage of.

To mitigate situations like these, the structure of the algorithm was changed quite significantly. While Red still plays according to the first applicable rule in a list of moves — a list that does largely resemble the one used by Algorithm 7.1.1 — this list is no longer the only consideration Red has. Instead of simply picking any move that satisfies the chosen rule at random, Red should consider each possible move satisfying that rule and determine which would be the most beneficial, based on a given hierarchy.

The preferences put forward by this hierarchy, and the reasons why these characteristics may be considered more advantageous for Red, will be expanded on in Sections 8.2 and 8.3.

In addition to adding a hierarchy to determine the best move, most rules of Algorithm 8.4.1 are to be skipped in favour of the next applicable rule if all moves that satisfy said rule are considered 'too dangerous'. The particular dangerous situations to be avoided are moves that create overlapping bridges of which the bridges share at least one end vertex (these overlapping bridges lack the security we would expect from a bridge) and moves that have five or six blue neighbours (which are certain dead ends for a Red path). The only rules exempt from being skipped over due to these concerns are those in which either Red is in such an advantageous position that they are no longer considered a threat (as is the case for rules 1 and 3), or those in which there is a more pressing threat for Red to deal with (as is the case for rules 2, 5, 6 and 7).

8.2 Changes Based on Shortcomings

Several changes were made based on the shortcomings that were explicitly discussed in Section 7.2.

8.2.1 Gaps Between Existing Local Paths

Three preferences in the newly added hierarchy are designed specifically to reduce the number and size of gaps between disjoint local paths as much as possible. Preference 4 does this by actively connecting disjoint paths, which closes existing gaps completely, while preferences 5 and 6 encourage Red to plan ahead and prepare ways to connect disjoint paths on their next move. Under the right circumstances, preference 3 may also encourage Red to close a gap between paths, namely when Red can somehow connect to distant path that would considerably lengthen the path through ring j.

In addition to closing gaps as much as possible through the hierarchy, rules 9 to 12 from Algorithm 7.1.1 were removed altogether. While there could certainly be situations in which these rules could be of use, it is far too easy for Blue to take advantage of these rules in the way described in Section 7.2.

8.2.2 Not Anticipating Blue's Strategy

To better anticipate and react to Blue's strategy, several new rules were added. Rule 2 encourages Red to prioritise preventing a blue winning move at all costs, as such a move would instantly end the game. Similarly, rules 5 and 6 were added to interfere with and prevent the creation of blue provisional winning paths, respectively.

In the case of rule 6, the algorithm initially tries to find a move that prevents Blue from creating a provisional winning path altogether, while giving preference to moves that simultaneously protect Red's own (overlapping) bridges. When it is impossible to fully prevent the creation of such a path, it will instead try to find the move that leaves Blue in the most inconvenient situation possible: one in which Blue has as many overlapping bridges and as few moves that create a provisional path as possible. The existence of overlapping blue bridges is especially important in this situation: If the only Blue provisional path contains an overlapping bridge, Red may still be able to prevent a loss by playing on its point of overlap later.

Even if Blue does have a true provisional winning path, rule 5 will still encourage Red to continue playing on bridges along this path. If Algorithm 8.4.1 is optimal, this subrule is unnecessary, as

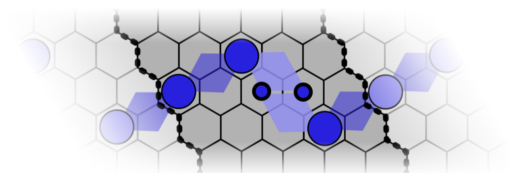
Red should never end up in this situation in the first place — an optimally playing Blue will always win when such a path exists, after all. However, since there is no guarantee that the algorithm in its current form is optimal, the subrule is still included to improve its performance as much as possible, as this is the best way to salvage such a position when playing against a sub-optimal opponent.

8.2.3 Double-Bridges

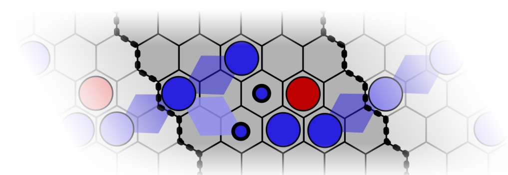
A double-bridge set-up can be made in many different ways — as shown in Figure 6.4 of Section 6.3 — but not all of them are likely to cause problems for Red. Once Blue has formed a set-up by playing on cells (i, j) and $(i + 3, j)$, for example, Red, while using Algorithm 7.1.1 or a similar algorithm, will usually have covered ring j by either playing on what would be the intermediary vertices or potential bridge cells of the set-up (preventing Blue from turning this set-up into a double-bridge) or on the cell between the two end cells of the set-up (greatly interfering with Blue's ability to turn the double-bridges into a winning path later). As another example, while a set-up made using cells (i, j) and $(i + 2, j - 4)$ could be useful for Blue in specific situations, there is a much larger distance for them to cover between the two ends of this set-up than there would be in some of the other options. Therefore, when considered on its own as the basis for a Blue winning path, this particular set-up doesn't seem all that threatening to Red.

Instead of taking each of these possible double-bridge set-up into account, Algorithm 8.4.1 will only consider the most threatening double bridges: those that will be easiest for Blue to turn into a winning path. The most threatening set-ups are the ones resembling the set-up from Figure 7.1b — those set-ups that can be made from some cell (i, j) to any of the cells $(i + 4, j - 2)$, $(i + 2, j - 2)$, $(i - 4, j + 2)$ and $(i - 2, j - 2)$.

The reason these particular set-ups are so threatening is that if any of these set-ups are completed on a $5 \times n$ board (i.e. if Blue plays on the intermediary vertex of the set-up to turn it into a double-bridge) and Red has not yet made any moves to interfere, Blue will be able to turn the double-bridge into a true provisional winning path in just a single move. As shown in Figure 8.1a, there are two cells in the ring of the intermediary cell that Blue can use to bridge the gap between the two ends of the double-bridge and create this path. We have already seen in Figure 7.1b that Blue may be able to create a provisional winning path using these moves even if Red has already made some moves in the surrounding rings.



(a) The moves that Blue can use to create a provisional winning path if Red does not interfere at all.



(b) If Red prevents both moves in Figure 8.1a, Blue can play on a cell such that they have two provisional winning moves on their next turn.

Figure 8.1: The ways in which Blue can turn their double-bridge into a true provisional winning path.

Even if Red plays on either of these cells on their next move — thereby interfering with both of these options — this only delays Blue's provisional winning path. On Blue's subsequent move, they can ensure that there are again two ways to create a blue true provisional winning path, as shown in Figure 8.1b. Since these new options do not overlap in any way, Red can no longer prevent both. Additionally, while one of these moves does create a blue overlapping bridge, the provisional winning path only uses one of these overlapping bridges. Since Blue will thus always be able to create a true provisional winning path on their next move, they are guaranteed to win once they've reached this position. It is therefore incredibly important for Red to interfere with this plan long in advance.

Of course, during a game, it will take Blue several turns to create these double bridges, giving Red time to interfere with either the creation of the double-bridge or the subsequent connection between its ends more than is shown in Figure 8.1. However, even with rules 9 to 12 from Algorithm 7.1.1 removed, the set-up can still form a threat to Red. As such, several options were considered for a new rule aimed at dealing with this type of set-up.

Playing on the intermediary cell of a double-bridge set-up, for example, would stop a double-bridge from being formed. However, as shown in Figure 8.2a, this method can only ever interfere with one double-bridge set-up at a time; if Blue creates multiple set-ups in one move, Red will have to pick one of these set-ups to interfere with, allowing Blue to complete the other. Additionally, as we may not have a cell in ring j yet when the set-up is made, this method could reintroduce the type of gaps between local paths that we tried to avoid by removing rules 9 to 12 from Algorithm 7.1.1.

Playing right next to (i, j) on one of Blue's future bridges instead can interfere with multiple set-ups — provided they are made in the same (horizontal) direction — and does not leave a gap in ring j. In addition, it will also give Red a bit of a foothold in the gap between the ends of the other possible set-ups, as can be seen in Figure 8.2b. This foothold, combined with the fact that Red hereby also prevents one of the two options from Figure 8.1a, already gives Red more of an advantage. However, this method still only considers a very limited set of options and doesn't take advantage of the current situation on the board.

In order to in order to take advantage of the current situation of the board, Red will need to expand their options. Knowing what Blue would like to do after completing the set-up, the area that would be used by Blue to connect the two ends of the double-bridges to each other seems like a promising candidate. While playing here does not interfere with the double-bridge set-up itself, it will make it harder for Blue to turn the double-bridge into a provisional winning path.

(a) Red has played on the intermediary vertex of one of the bridges, interfering with only one of the two double-bridge set-ups.

(b) Red has played on a potential bridge cell next to (i, j), interfering with one of the two double-bridge set-ups and limiting the ways in which Blue can connect the ends of the other double-bridge set-up.

Figure 8.2: Two ways in which Red can try to interfere with two double-bridge set-ups that were created simulataneously by Blue playing on their shared end cell, (i, j).

Formally, we can define the area between the two ends of the set-up as follows:

Definition 8.2.1 (Connection Area)**.** *On a* $5\times n$ *Cylindrical Hex board, the connection area between the two ends* $z, w \in V$ *of a double-bridge set-up is the set of vertices*

$$CA(z,w) = \{(i,j) \mid i \in \{z, w, \tfrac{z+w}{2}\},\ j \in \{z,w\}\}.$$

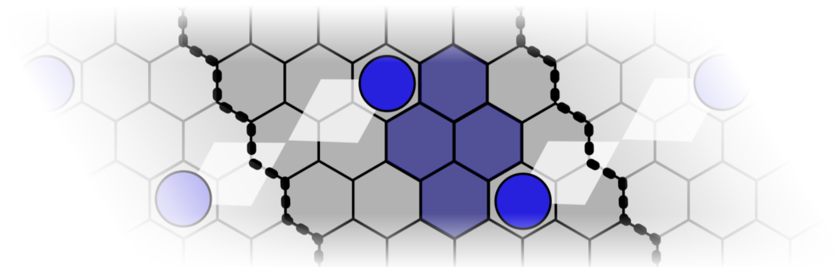


Figure 8.3: The connection area of a double-bridge set-up.

The connection area should not be considered the only option for Red, however. While it does indicate a set of cells that are likely to help Red interfere with Blue's plans, it can sometimes be more useful to play outside a connection area than it would be to play within the area — for example, if playing outside the connection area would allow Red to cover all of the rings involved in the set-up. The connection area will therefore be, at most, an additional preference unique to this specific rule.

Building a provisional path from ring j outwards remains more important to Red than playing in this specific area, as not doing so might reintroduce the gaps caused by Algorithm 7.1.1's rule 9 to 12. In addition, when there are set-ups in both (vertical) directions, as was the case in Figure 8.2b, ensuring that at least ring j is covered helps to interfere with both set-ups. On the other hand, when there is only a set-up in one direction, it becomes important to build the path outwards from ring j in the direction of this bridge.

Rule 7 bases Red's next move one these considerations. In the case of set-ups in both (vertical) directions, covering ring j and building a provisional path outwards is done in exactly the same way as in rules 10 to 14 of Algorithm 8.4.1, except this time with a preference for moves in the connection area.

When there is only a set-up in one direction, Red can still build their path in a similar way as in rules 10 to 14. However, a small change must be made for rules 12 and 13: For these rules Red must limit their search to only those moves that cover rings in the direction of the set-up, as this is where the current threat lies. Once a move has been found, Red can again limit their options to those in the connection area, should such moves exist.

If none of the found moves for a particular rule lie in the connection area, we can conclude that playing outside the connection area will allow Red to connect more rings — or a more beneficial set of rings — than playing inside the connection area. In this situation, Red will thus play one of these cells outside of the connection area instead.

Finally, if a set-up has just been made but the involved rings are already covered, Red can move on to rule 8 immediately. In this case there must already be a provisional path through the connection area, so even if Blue does complete the set-up, they will not be able to instantly connect the ends of the double bridges. The set-up is therefore deemed to not be a threat to Red at this point.

8.2.4 Overlapping Bridges

As mentioned in Section 8.1, the new algorithm will skip most rules if they can only be satisfied by creating an overlapping bridge. In addition to this, if a rule is not skipped, the first preference within the hierarchy tells Red to disregard any moves that do create an overlapping bridge.

Should an overlapping bridge be made in rules 1 to 3 or 5 to 7, rules 8 and 17, as well as preference 8 of the new algorithm will give priority to protecting or filling in this type of bridge over a normal bridge — where possible by playing on its point of overlap.

With these changes alone, the situation in Figure 7.2 would no longer causes any issues: Due to the hierarchy, only the middle two moves on the existing bridge would be considered. These moves will not create any overlapping bridges and instead fortify an existing bridge, making the provisional path more secure.

However, while added security can be beneficial, it is not necessarily the most useful move in this situation. Since we do not create an overlapping bridge, Blue will not be able to destroy the existing bridge unless Red allows it. Furthermore, there is already a provisional path from ring $j-2$ to ring $j+2$. While rings $j-1$ and $j+1$ are not truly connected themselves, they are, in a way, both already covered by a secure bridge. If the 'cover' given by the bridges is acknowledged, Red could move on to lengthening the existing path — bringing them one step closer to creating a provisional winning path — instead of securing an already quite secure situation.

Rules 10 to 14 of Algorithm 8.4.1 were modified from Algorithm 7.1.1's rules 3 to 8 to acknowledge the cover afforded by both existing and future bridges. We define this cover as follows:

Definition 8.2.2 (Covering Rings)**.** *A set of rings $R(a, b)$ with $1 \leq a, b \leq n$ is covered if there exists a (provisional) path from ring a^* to ring b^*, with $a^* \in \{a-1, a\}$ and $b^* \in \{b, b+1\}$.*

The change from connecting rings to covering them would greatly reduce the number of overlapping bridges created by Red, even if the new algorithm did not disregard moves that will create overlapping bridges. For example, when faced with a situation like the one in Figure 7.2, a Red player following a version of Algorithm 7.1.1 modified to cover rings instead of connecting them would immediately be able to move on to that algorithm's rule 13 — and for later moves even to rules 14 to 18 — without ever considering a move that might create an overlapping bridge.

In addition to introducing the idea of covering a set of rings, rules 10 to 14 also consider more moves to achieve this goal. Now that covering these rings is enough, the rings directly above and below the rings Red wants to cover can be of use as well. For example, if there is a path from ring $j-2$ to ring $j+1$, then using Algorithm 7.1.1, Red would have make a move in ring $j+2$ to satisfy rule 3. However, now that Red wants to cover rings $R(j-2, j+2)$ instead of connecting them, Red could also consider making a bridge from ring $j+1$ to ring $j+3$, which would have the added benefit of resulting in an even longer provisional path.

8.2.5 Provisional Paths

Explicit rules are introduced in the new algorithm that encourage Red to make a provisional path (rule 4) or, when one already exists, to fill bridges along the shortest existing one (rule 3).

In rule 3, priority is given to saving bridges that were just attacked by Blue to ensure the provisional path is not destroyed. In addition, filling in overlapping bridges to turn a provisional winning path into a true provisional winning path is also given priority, as this will guarantee Red's win.

8.3 Further Changes to the Algorithm

The changes discussed above have also introduced new opportunities to further improve the algorithm in other ways.

8.3.1 Delaying Arbitrary Preferences

Algorithm 7.1.1 has an arbitrary preference for creating a path towards the 'lower' parts of the board as opposed to the 'higher' parts. When it comes to rules 5 and 6 for example, this algorithm looks at moves to create a path 'downwards' from ring j (to ring $j + 2$), before considering the options 'upwards' (to ring $j - 2$).

Whether we create this path upwards or downwards is, of course, a decision that needs to be made eventually. If Red continues to make this decision immediately, however, they miss plenty of opportunities. Several games where Red followed Algorithm 7.1.1 were observed in which there was a move satisfying rule 6 (or rule 8) that was clearly more beneficial than those satisfying rule 5 (or rule 7) — for example, because it closed a gap or lengthened the path more — but it was not even considered by the algorithm. Simply reversing the order of these rules would, of course, not make much of a difference, as we could instead miss out on more beneficial moves satisfying rules 5 or 7.

With the introduction of the hierarchy, however, it becomes possible to put this decision off for longer. Rules 12, 13 and 16 of Algorithm 8.4.1 take advantage of this: They combine rules from the previous algorithm — rules 5 and 6, rules 7 and 8, and rules 14 to 18, respectively — , allowing Red to consider all similar moves at the same time, even if they are mirrors of each other. All of these moves are then simultaneously evaluated according to the hierarchy, which only shows a preference for which ring or column a move lies in near the very end, in preferences 10 and 11, respectively.

8.3.2 Further Encouragement for Lengthening a Path

Several changes were introduced to encourage Red to lengthen their provisional paths as much as possible. We have already seen one such change to rules 10 to 14 of the new algorithm in Section 8.2, when discussing overlapping bridges: Red now considers moves one ring further than before in order to cover the local rings.

While the rules themselves encourage Red to consider moves in more distant rings, the hierarchy encourages Red to actually use these new options whenever possible. This is done by preference 3. If there is a way to lengthen the path through ring j by connecting to another path, this preference will also prioritise this move.

In addition to lengthening a path as much as possible, the hierarchy also tries to set up as many vertical bridges as possible through item 7. This will allow Red to more easily lengthen a path in future moves.

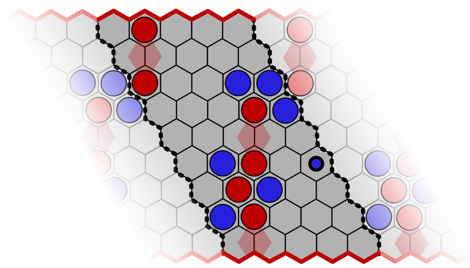


Figure 8.4: A position on a 5×9 Cylindrical Hex board. If Blue plays on the cell $(5, 6)$, indicated by a smaller stone, Red is not able to extend the existing path through that ring, due to Blue's blockade in ring 3. Red could try to evade this blockade by playing in ring 4 again, somewhat closing the gap between the two provisional paths.

Finally, Algorithm 7.1.1's rules 14 to 17, which are aimed at increasing the length of an existing path beyond the local rings, were modified as well. These rules together are equivalent to Algorithm 8.4.1's rule 16. This rule looks for ways to extend existing paths from either end of the path simultaneously, instead of only looking at the side that has the nearest end. Additionally, if it is not possible to safely extend an existing path at either end, this rule encourages Red to play in the same ring as one of the ends of the path, in an attempt to go around whichever blue path is blocking the way.

An example of a position in which going around would be useful is shown in Figure 8.4. If Blue were to play at $(5, 6)$, for example, Red would not be able to extend the Red path through ring 6, due to the blockade formed by the blue stones on cells $(4, 3)$ and $(5, 3)$. Instead of moving on to rule 18 by playing on one of the bridge cells, it would be more beneficial to play on a cell in ring 4 again in this case. This will help bridge the gap between the two existing provisional paths and could thereby allow Red to lengthen this path on a future turn.

8.3.3 Avoiding Unnecessary Moves

A few more preferences were added to the hierarchy in order to prevent Red from taking unnecessary risks. Both preferences 2 and 9 encourage Red to avoid dead ends by not playing on cells that have 5 or 6 neighbours (blue or red, respectively).

At the same, item 9 also encourages Red to play on cells that have more red neighbours than other moves: While looking at games resulting from Algorithm 7.1.1, moves that created bridges tended to have fewer neighbours than those that made direct connections. By preferring moves that have more neighbours, Red should therefore make more direct connections and less bridges. As this is one of the last preferences listed in the hierarchy, it will only be used if the remaining moves are considered equally good in all other aspects — i.e. if making a bridge can be considered unnecessary. In other words, all else being equal, Red will prefer the more secure move.

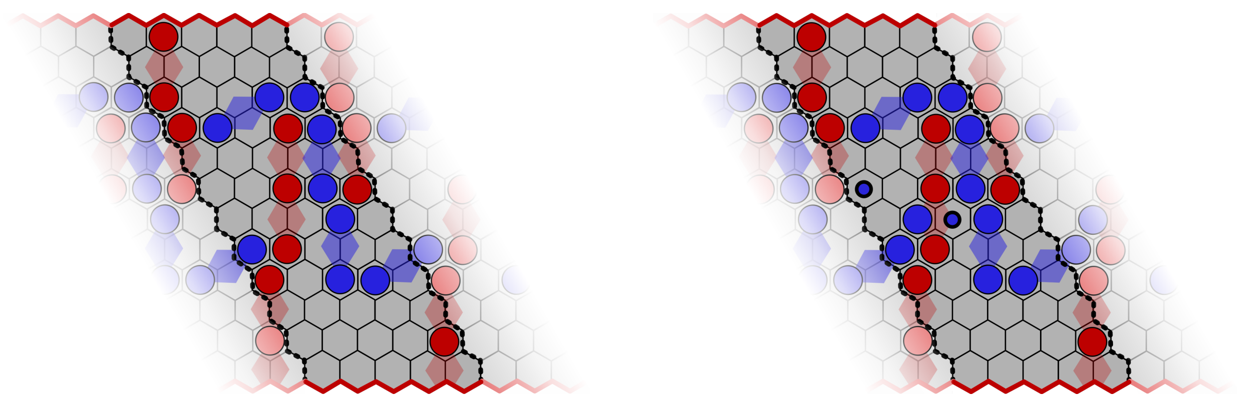
8.3.4 Strengthening Paths In Popular Areas

While the new algorithm usually lengthens existing paths by covering as many rings as possible, there is one notable situation in which Red will choose to strengthen an existing provisional path instead: If ring j has already been covered, but has not been played in yet by Red, Red will hold off on trying to extend the path through ring j beyond rings $R(j-2, j+2)$ and play on a bridge cell in ring j instead.

This is because Blue has just played in an area that is already somewhat covered by Red. After all, for this situation to occur, there must be a red bridge from ring $j-1$ to ring $j+1$ — otherwise either Red would have followed rule 14 instead, or Blue just made a winning path using all cells of ring j, thereby winning the game. Rings $R(j-1, j+1)$ — and likely (but not necessarily) more, as rules 10 to 12 must have been disregarded for some reason — are thus already covered, yet Blue continues to play here. Since this could be an indication that Blue is planning something in this area, it is better to strengthen the provisional path in this area before lengthening the path further. Rule 15 does this by playing on the bridge that must exist between rings $j-1$ and $j+1$.

An example of how Blue might try to win by attacking a red bridge is shown in Figure 8.5. In this game, Red's moves are based on an earlier version of the improved algorithm. In Figure 8.5a, Blue has already played in ring 7 and the surrounding rings often enough for Red to have formed a rather long provisional path in response. While the bridges along this path individually are quite secure, Blue can still win by attacking one of them: By playing on $(2, 7)$, as shown in Figure 8.5b, blue not only attacks one of the red bridges, but also creates two ways in which they can create a true provisional path on their next turn — by playing on $(1, 6)$ if Red does protect the attacked bridge, or by playing on $(3, 7)$ if they do not. Doing so guarantees a Blue win.

While this exact position will probably not appear in games where Red uses the version of the algorithm introduced in Section 8.4, similar situations may still occur. Encouraging Red to occasionally strengthen bridges that might be a target for similar strategies, as is done by rule 15, could therefore be a useful preventative measure.



(a) A position during a game. Blue has only played in rings 3 to 9.

(b) Blue has made a move on $(2, 7)$, giving them two disjoint ways to make a true provisional winning path on their next move.

Figure 8.5: A position on the 5×12 Cylindrical Hex board, in which Blue was the starting player and Red used an earlier attempt at an earlier version of the new algorithm.

8.4 The New Algorithm

Having made all of the changes discussed above, the algorithm works as follows:

Algorithm 8.4.1 (Suggested Red strategy for Cylindrical Hex on a $5 \times n$ board)**.**
Consider the Cylindrical Hex graph $H_c(5, n)$. Add rings 0 and $n+1$ to the board and colour both rings entirely red.

On their turn, Red does the following:

I. *If Blue has not played yet: play in the middle of the board.*

II. *Else, if Blue's latest move was on cell (i, j): set $r = 1$. While $r \leq 19$:*

- *If rule r, as described in the list of rules given below, can not be satisfied: $r = r + 1$.*
- *Else, if rule r can be satisfied, $r \in \{4\} \cup R(8, 19)$ and all moves that satisfy rule r create an overlapping bridge or have at least 4 blue neighbours: $r = r + 1$.*
- *Else, list all moves that satisfy rule r in the list $\mathcal{L}$ and set $h = 1$. While $|\mathcal{L}| \neq 1$:*[1]
  - *Determine all moves $(x, y) \in \mathcal{L}$ that satisfy preference h from the hierarchy given below and list them in the list $\hat{\mathcal{L}}$,*
    * *If $|\hat{\mathcal{L}}| = 0$: $h = h + 1$.*
    * *Else: $\mathcal{L} = \hat{\mathcal{L}}$ and $h = h + 1$.*

Note that throughout the algorithm — including in the rules and hierarchy items listed below — (i, j) refers to the cell that has just been played on by Blue, while (x, y) and (a, b) are used to refer to some other cell on the board.

The Rules Used by Algorithm 8.4.1

1. if there is an unplayed cell (x, y) that is a winning move for Red: on (x, y),
2. if there is an unplayed cell (x, y) that is a winning move for Blue:
   (a) if there exists a blue winning move (x^*, y^*), such that Blue no longer has a winning move if Red plays on this cell: on (x^*, y^*),
   (b) else, on some winning move (x, y),
3. if there existed a Red provisional winning path before move (i, j):
   (a) if (i, j) was part of an overlapping bridge of which the bridges share an end vertex along the shortest provisional winning path:
      i. if the point of overlap of this overlapping bridge has not been played on by either player: on this point of overlap,

[1] By narrowing down the list of potential moves $\mathcal{L}$ in the way described here, Red is guaranteed to eventually be left with one single move. This is due to the last two hierarchy items, which pick a move based on the ring and column the remaining moves lie in.

ii. else, on another bridge cell of this overlapping bridge,

(b) else, if (i, j) was part of a bridge along the shortest provisional winning path: on the other bridge cell of this bridge,

(c) else, if the shortest provisional winning path contains at least one overlapping bridge of which the bridges share an end vertex: on the point of overlap of an overlapping bridge along this path,

(d) else, on one of the bridges along the shortest provisional winning path,

4. if there is an unplayed cell (x, y), such that playing on this cell a provisional winning path for Red: on cell (x, y),

5. if there exists a Blue provisional winning path:

   (a) if there exists a Blue overlapping bridge of which the bridges share an end vertex, such that Blue will no longer have a provisional winning path if Red plays on its point of overlap: on a point of overlap of one of these bridges

   (b) else, one one of the cells of any of the bridges along this path (including overlapping bridges that do not satisfy rule 5(a)),

6. if there are unplayed cells (x, y), such that a blue provisional winning path will be created if Blue plays on this cell next:

   (a) if cell (i, j) was part of a red overlapping bridge of which the bridges share an end vertex, and playing on its point of overlap prevents Blue from creating a provisional winning path on their next move: on the point of overlap,

   (b) else, if cell (i, j) was part of an existing red bridge (including overlapping bridges that do not satisfy 6(a)) and playing the other bridge cell of this bridge prevents Blue from creating a provisional winning path on their next move: on this other bridge cell,

   (c) else, if playing on (1) one of the blue winning moves (x, y), (2) the point of overlap of a blue overlapping bridge of which the bridges share an end vertex, (3) any bridge cell of a blue bridge or (4) a potential blue bridge cell — i.e. the bridge cell of a blue bridge that will be created if Blue were to play on cell (x, y) — prevents Blue from creating a provisional winning path on their next move: on any of these cells,

   (d) else, one one of the following: (1) the points of overlap of a red overlapping bridge of which the bridges share an end vertex that was just attacked, (2) the other bridge cell of a red bridge that was just attacked, (3) the blue provisional winning moves (x, y), (4) the points of overlap of any blue overlapping bridges of which the bridges share an end vertex, (5) the bridge cells of all blue bridges and (6) all potential blue bridge cells. If there are multiple moves that satisfy this subrule, limit your options to:

      i. those moves that will leave Blue with the highest number of blue overlapping bridges,

      ii. among the above moves, those moves that will leave Blue with the least provisional winning moves (x, y),

7. if the move on cell (i, j) created a double-bridge set-up to $(i+4, j-2)$, $(i+2, j-2)$, $(i-4, j+2)$ or $(i-2, j-2)$:

   (a) if the rings covered by the double-bridge set-up(s) created by move (i, j) are already covered by a red path: move according to the next applicable rule,

   (b) else, if move (i, j) created double-bridge set-ups covering $R(j-2, j+2)$: play according to rules 10-14 of this algorithm. If the resulting list of moves contains at least one move within the connection area, on one of the moves within this area,

   (c) else, if move (i, j) created a double-bridge set-up covering rings $R(j-2, j)$: play according to rules 10-14, but limit rules 12 and 13 to only covering rings $R(j-2, j)$ and $R(j-1, j)$, respectively. If the resulting list of moves contains at least one move within the connection area, on one of the moves within this area,

   (d) else, (i.e. if move (i, j) created a double-bridge set-up covering rings $R(j, j+2)$): play according to rules 10-14, but limit rules 12 and 13 to only covering rings $R(j, j+2)$ and $R(j, j-1)$, respectively. If the resulting list of moves contains at least one move within the connection area, on one of the moves within this area

8. if cell (i, j) was part of an existing red overlapping bridge of which the bridges share an end vertex:

   (a) if the point of overlap has not been played on by either player: play on this point of overlap,

   (b) else, on a remaining bridge cell of these overlapping bridges,

9. if cell (i, j) was part of an existing red bridge: on the other bridge cell of this bridge,

10. in a ring from $R(j-3, j+3)$ such that rings $R(j-2, j+2)$ become covered,

11. in a ring from $R(j-2, j+2)$, such that rings $R(j-1, j+1)$ become covered,

12. in a ring from $R(j-3, j+1)$, such that rings $R(j-2, j)$ become covered, or in a ring from $R(j-1, j+3)$, such that rings $R(j, j+2)$ become covered,

13. in a ring from $R(j-2, j+1)$, such that rings $R(j-1, j)$ become covered, or in a ring from $R(j-1, j+2)$, such that rings $R(j, j+1)$ become covered,

14. in a ring from $R(j-1, j+1)$, such that ring j becomes covered,

15. if there is no red cell in ring j: play on a bridge cell in ring j,

16. Determine the nearest end(s) (a, b) of all paths through ring j in terms of the distance $|j-b|$ in both directions.

   (a) if there exists and unplayed (a^*, b^{**}) — where $b^{**} = b+2$ when $b > j$ and $b^{**} = b-2$ when $b < j$ — such that playing on (a^*, b^{**}) will create a bridge from ring b to ring b^{**}: on (a^*, b^{**}),

   (b) if there exists and unplayed (a^*, b^*) — where $b^* = b+1$ when $b > j$ and $b^* = b-1$ when $b < j$ — such that either playing on (a^*, b^*) will create a bridge from ring b to ring b^* or (a, b) and (a^*, b^*) are neighbours: on (a^*, b^*),

(c) else, in ring b,

17. if there exists at least one red overlapping bridge of which the bridges share an end vertex: on a point of overlap of one of the red overlapping bridges,
18. if there exists at least one red bridge: on a bridge cell of one of the red bridges,
19. on any unplayed cell.

The Hierarchy Used by Algorithm 8.4.1

1. The move does not create an overlapping bridge,
2. The move has at most 4 blue neighbours,
3. The move increases the length of a path that covers ring j the most,
4. The move connects the most disjoint paths to each other,
5. The move will allow Red to connect the most disjoint paths to each other on their next move,
6. The move will allow Red to connect to disjoint paths in the largest number of ways on their next move,
7. The move will give Red the most options for making vertical bridges from this cell on their next move,
8. If such a move remains, the move on the point of overlap of a red overlapping bridge. Else, the move on any bridge cell,
9. The move has the most (but less than 5) Red neighbours,
10. The move lies in the lowest ring out of all remaining moves — i.e. the move (x, y) that has the lowest value of y out of all remaining moves,
11. The move lies in the left-most column out of all remaining moves — i.e. the move (x, y) that has the lowest value of x out of all remaining moves.

Chapter 9

Simulated Games

In this section we will compare the effectiveness and efficiency of several algorithms discussed in this thesis. We will do this by looking at the results of simulations of games in which Red uses these algorithms. The methodology of these simulations is discussed in Section 9.1.

In Section 9.2 we will compare the efficiency of two algorithms that are proven to be optimal on certain boards — Algorithm 4.2.1 and Algorithm 5.1.1.

In Section 9.3 we compare the effectiveness of two algorithms specifically designed for a $5 \times n$ board — Algorithm 7.1.1 and Algorithm 8.4.1.

We also spend some time looking at other results from the simulated games that use Algorithm 8.4.1 in Section 9.4.

9.1 Simulation Methodology

To study the effectiveness of these algorithms, we will look at how Red fares in simulations where they use a particular algorithm against a Blue Monte Carlo opponent.

A Monte Carlo player will, on their turn, evaluate each possible move by simulating a given number of playouts to determine which move they should make. Each of these playouts is a simulated game in which both players make random moves until a winner has been found.

In the simulations that were done for this thesis, the number of playouts per possible move by the Monte Carlo player, $\bar{n}$, was recalculated at the beginning of each turn. Each time, this number was chosen such that, based on the current position on the board, the estimated winrate would have a margin of error $\epsilon = 0.05$ and confidence interval 95%, based on methods described by Owen [11].

The decision to recalculate $\bar{n}$ each turn, as opposed to choosing a fixed value, such as $\bar{n} = 500$, was made in order to reduce the required computation time: Even when taking into account the extra playouts needed to calculate the new value, the total number of playouts needed to determine the best move was lower, without compromising on accuracy. By reducing the computation time in this way, it was possible to simulate more different boards and look at a wider range of data.

Throughout the simulations, Blue thus chose their moves according to the algorithm below.

Algorithm 9.1.1 (Blue Monte Carlo move).
Given an incomplete colouring C on a Cylindrical Hex Board $H_c(m,n)$, Blue decides their next move by estimating the best move to make as follows:

1. *Make an estimate n for the minimum number of simulations needed for an accurate estimate of the red winrate based on the current position on the board, as follows:*
   (a) *Do 500 playouts starting from the current position as given by the colouring C. In each of these simulations, both players, starting with Blue, make random moves until a winner is found.*
   (b) *Calculate the variance estimate of the fraction of wins s^2, based on the results of these playouts.*
   (c) *Calculate the estimated minimum number of simulations needed $n = s^2 \cdot \frac{1.96^2}{0.05^2}$.*
2. *Set the number of playouts to be done for each unplayed cell to be $\bar{n} = 10 \cdot \lceil \frac{n}{10} \rceil + 50$.*
3. *For each unplayed cell $(x,y) \in V_{m,n}$ (i.e. for each move Blue could make in the current position):*
   (a) *Do $\bar{n}$ playouts starting from the current position as given by the colouring C, and playing (x,y) for the first move as Blue. After this first move, both players make random moves until a winner is found.*
   (b) *Calculate Red's winrate of move (x,y) based on these playouts:*

$$r_w(x,y) = \frac{\textit{Playouts won by Red}}{\bar{n}}.$$

4. *Choose a move (x^*,y^*), such that $r_w(x^*,y^*) = \min_{(x,y)}\{r_w(x,y)\}$*

In practice, an exception to this method was made if Red, while playing according to Algorithm 8.4.1, didn't have any overlapping bridges and had just used rule 3 or 4. In this situation Red's provisional winning path — which, at that point, is guaranteed to exist due to the use of rule 3 or 4 — can not have any overlapping bridges, and must therefore be a true provisional winning path. In Section 6.4 we noted that once a player has created such a path, they are guaranteed to win if they play optimally. Since rule 3 of Algorithm 8.4.1 is specifically designed to play optimally in this situation, Blue can no longer win in this case, regardless of which moves they make. As such, Blue was made to play randomly whenever this situation occurred, to further save on computation time.

The cylindrical board $H_c(5,n)$ was simulated as a finite subgraph of S_n. To ensure that any winning path on the cylindrical board could also be found by only looking at this particular subgraph, regardless of how many times the path encircled the cylinder, enough copies of the board needed to be simulated. As such, the subgraph contained $c = \max\{3, \lceil \frac{n}{2} \rceil + 1\}$ copies of the board.

Whenever a move was made, the corresponding vertex and all of its copies were coloured simultaneously. Because of this, playing on a given vertex yielded the same result as playing on its copy. When deciding on their next moves, the players thus do not need to consider these copies as separate options. For Blue, in particular, this means that Algorithm 9.1.1 only needs to calculate the winrate for the original cell, and not for each individual copy of this cell.

While Red also did not consider copies as individual options in most cases, occasional exceptions were made for convenience. While these exceptions did not influence the choice made by Red — and therefore did not influence the outcome of the game — they may have slightly influenced the results regarding how often each hierarchy item was used, which we will discuss in Section 9.4.2. As this only influenced the rules that were used the least (as we will see in Section 9.4.1), the discrepancies should be minimal. but it may be useful to understand how this discrepancy came about.

One situation in which copies were treated as a separate option, occurred as a result of how bridges were handled: Throughout each game, Red kept a list of all Red bridges that existed on the board at that moment. However, to ensure that bridges that went over the 'edge' separating the original board from its copy were listed as well, each copy of the bridge was listed individually. Most rules where bridges are relevant only considered one copy of the bridge to make a move[1], but for rules 17 and 18 each bridge vertex was treated as a separate option. A similar approach was used for rule 16.

For each algorithm discussed here, 100 simulations were run per starting player on several differently sized boards. In addition to the winner of each game, data such as the number of moves made and the length of the red winning path (whenever it existed) was also recorded.

9.2 Comparison: Algorithm 4.2.1 vs. Algorithm 5.1.1

The first two algorithms that we will compare are two of the optimal algorithms discussed in this thesis: Algorithm 4.2.1 — the $2k \times n$ strategy, which is optimal on boards with an even number of columns — and the newly introduced Algorithm 5.1.1 — the Gardner strategy, which is optimal on boards with more columns than rings. Since both of these algorithms are already optimal on specific boards, we will compare their efficiency in terms of both the length of the game and the length of the red winning path on five differently sized boards.

9.2.1 Minimal Size Boards

We will first look at the results on 3 boards of the form $(n+1) \times n$ (i.e. the thinnest type of board on which Algorithm 5.1.1 is optimal), with n equal to 3, 5 and 7. In other words, the boards 4×3, 6×5 and 8×7.

The average number of moves made during a game and the average length of the shortest winning path on these boards are shown in Figure 9.1.

We can easily see that Red tends to win faster and create shorter winning paths when they are the starting player, regardless of which algorithm is used — though Red's starting advantage is generally smaller when using the Gardner strategy.

[1] In the case of rule 9, for example, Red would check the list of bridges to see whether the first copy of Blue's latest move was a red bridge vertex. If this was the case, then the recommended move was the other bridge vertex of this particular copy. By looking at the first copy, it was guaranteed that bridges on both the left and right edge of the original board were considered as well.

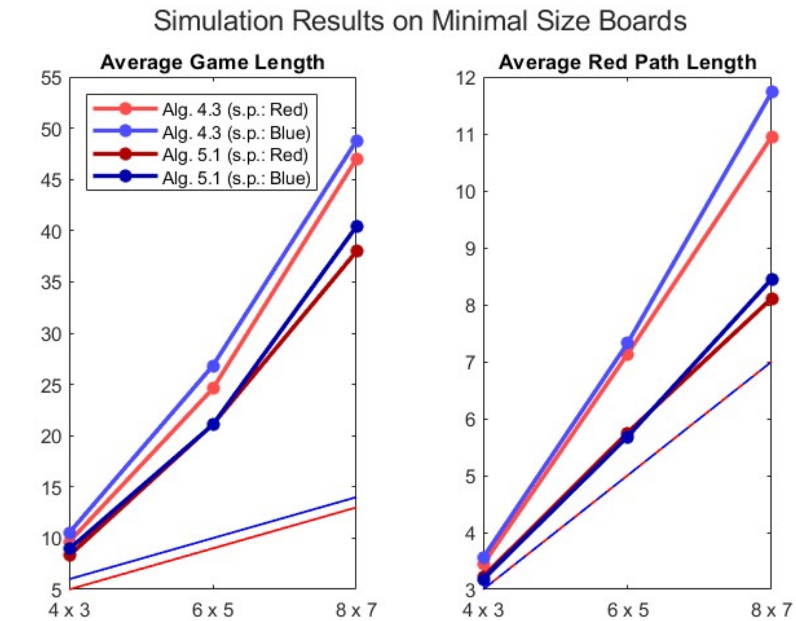


Figure 9.1: The results of the simulations done on various minimal size boards using Algorithm 4.2.1 (shown in lighter shades of red and blue) and Algorithm 5.1.1 (shown in darker shades).
Two additional lines indicate the minimum number of moves made in a game, and the minimum length of a red winning path, on a particular board.

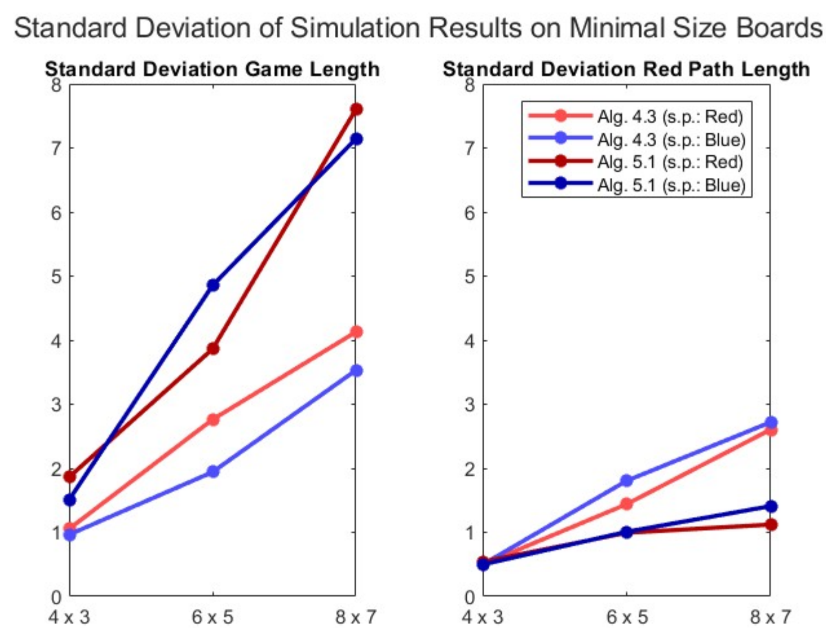


Figure 9.2: The standard deviation of the results shown in Figure 9.1.

It is not surprising that being able to make the first move is more advantageous to Red: Red's behaviour throughout a game is essentially the same, regardless of which player begins. The only difference is the existence of an extra Red stone throughout the game, which can only be advantageous to Red.

Figure 9.1 also shows a more interesting result: Using the Gardner strategy consistently leads to games that, on average, are shorter and result in a shorter winning path than using the $2k \times n$ strategy would. In other words, the Gardner strategy is more efficient on these minimal size boards.

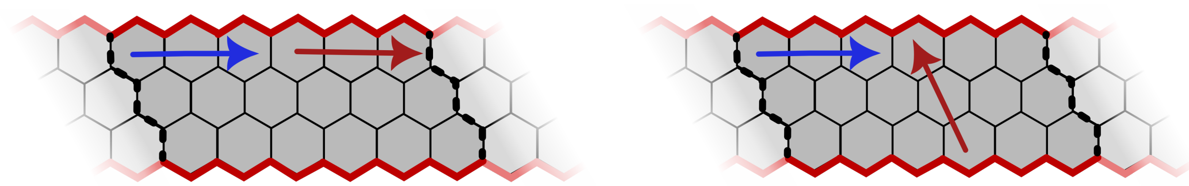
Not only is Gardner more efficient, the difference in efficiency also seems to increase as the board becomes larger. While this holds for both metrics, it is perhaps most obvious for the increasing gap between the winning path lengths.

It is not surprising that the Gardner strategy is the more efficient of the two. After all, due to their winning condition, Blue will want to work horizontally: They are more likely to stick to a limited set of rings in an attempt to be as efficient as possible in making their winning path and will only extend to other rings when they feel they don't have another choice.

Since Algorithm 4.2.1 translates Blue's moves horizontally to determine Red's next move, Red will also be working horizontally. This not only makes it more time-consuming for satisfying Red's winning condition — which ultimately requires a vertical red path — but is also more likely to lead to more zig-zagging in the final winning path.

Algorithm 5.1.1, on the other hand, makes its moves by mirroring Blue in a specific way. The resulting red path is almost perpendicular to the one Blue is trying to make and is therefore likely to be more vertical — leading to both a shorter game and a shorter path.

This difference in direction is illustrated in Figure 9.3.



(a) The $2k \times n$ strategy's horizontal translation (b) The Gardner strategy's mirroring

Figure 9.3: The directions in which the algorithms in this section tend to work, as a result of Blue trying to form a path horizontally.

Figure 9.2 shows the standard deviations of the results of the simulations done on these boards. From these graphs we can see that the length of the winning path is also more consistent when Red uses the Gardner strategy than when they use the $2k \times n$ strategy. Interestingly, the opposite is true for the length of the game.

9.2.2 Widening boards

Next we will look at the results on 3 boards of the form $m \times 5$ with m equal to 6, 8 and 10. In other words, the boards 6×5, 8×5 and 10×5. This will show us what we happens when we take one of the minimum size boards discussed previously and add extra columns.

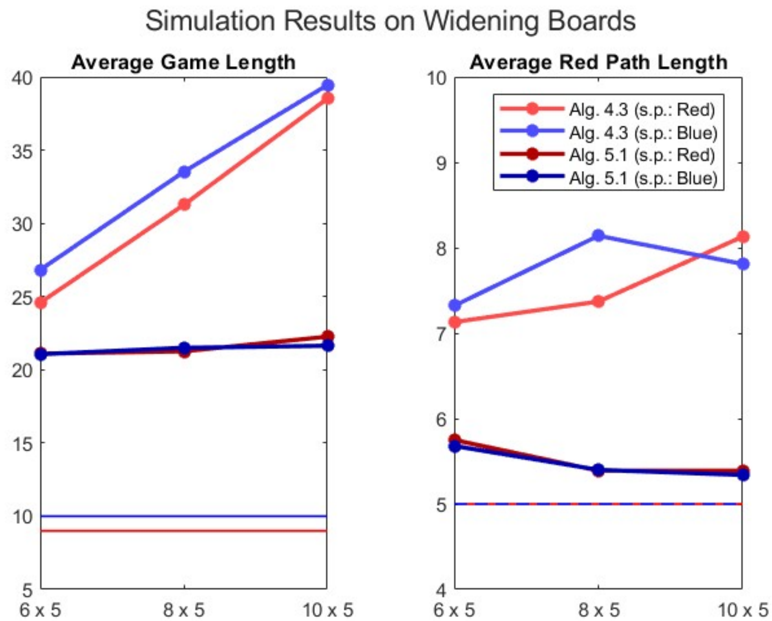


Figure 9.4: The results of the simulations done on various widening boards using Algorithm 4.2.1 (shown in lighter shades of red and blue) and Algorithm 5.1.1 (shown in darker shades).
Two additional lines indicate the minimum number of moves made in a game, and the minimum length of a red winning path, on a particular board.

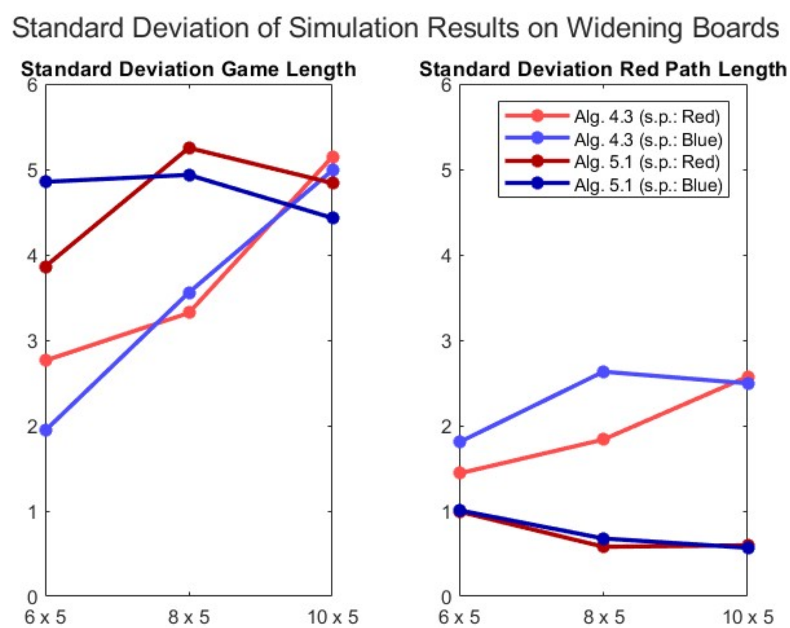


Figure 9.5: The standard deviation of the results shown in Figure 9.4.

The average number of moves made during a game and the average length of the shortest winning path on these boards are shown in Figure 9.4.

It is harder to make conclusions regarding the advantage of being the starting player based on the results of these simulations. As shown in Figure 9.4, Red does generally seem to maintain this advantage as the board becomes wider when using the $2k \times n$ strategy — the only exception being the average winning path length on a 5×10 board. For the Gardner strategy, the results are less consistent and remain incredibly close.

Based on graphs in Figure 9.4, we can also see that the Gardner strategy remains more efficient than the $2k \times n$ strategy even as the board widens. While the game and winning path length when using the $2k \times n$ strategy tend to increase noticeably as extra columns are added to the board, the same is not quite true for the Gardner strategy: Here, the average game length only increases very slightly, while the winning path length even decreases as the board widens.

It is not surprising that the average game length of the Gardner strategy remains more steady than that of the $2k \times n$ strategy. While using Algorithm 5.1.1, Red never plays outside of a 6×5 sub-board, as they know that is enough to cut off Blue's attempts to create a winning path. Even if Blue plays outside of this area a few times, the game will likely be shorter due to Red playing more efficiently.

In fact, we can use this observation to find an upper bound on the length of games in which Red uses the Gardner strategy. The longest possible game using this strategy would involve Red making as many moves as possible on the sub-board without creating a winning path, before making their winning move. On an $(n+1) \times n$ sub-board, this maximum number of moves would be $(n+1)(n-1)+1$: Red uses $(n+1)(n-1)$ moves to completely fill in all rings except one on the sub-board, and then makes one additional move that finishes the winning path.

In the meantime, Blue would have to make moves that do not interfere with Red playing in this manner: They can only play on the n remaining cells of the sub-board and on the $(m-(n+1)) \cdot n$ cells outside of the sub-board. These two areas combined will thus need to be large enough to accommodate the total number of moves Blue has to make throughout this game.

Since the Gardner strategy is proven to be optimal, we know that Blue is guaranteed to lose. If Red begins this means that, at the end of the game, Blue will have made one move less than Red. For this type of longest possible game to occur, the board thus needs to be large enough that Blue can make $(n-1)(n+1)$ moves in their designated area of the board. In other words, we require $(n-1)(n+1) \leq n+(m-(n+1)) \cdot n$, which simplifies to $2n - \frac{1}{n} \leq m$. Similarly, if Blue begins, they will make the same number of moves as Red, and we require $(n-1)(n+1)+1 \leq n+(m-(n+1)) \cdot n$, which simplifies to $2n \leq m$.

If the relevant equation is not satisfied, Blue is forced to play on the sub-board more than n times due to a lack of space outside of the sub-board. Red is therefore not able to make the $(n-1)(n+1)+1$ moves needed for this longest possible game, as Blue has already played on some of the necessary cells. In this situation, we can find a more accurate upper bound on the length of a game in which Red uses the Gardner strategy by simply looking at the maximum length of any game won by Red on a $m \times n$ board, which depends on the size of the full board and whether its has an odd or even number of cells.

The upper bound on the total length of a game in which Red uses the Gardner strategy and is the starting player, $U_{G,R}(m,n)$, can thus be written as follows:

$$U_{G,R}(m,n) = \begin{cases} m \cdot n & \text{if } m < 2n - \frac{1}{n} \text{ and } m \cdot n \text{ is odd,} \\ m \cdot n - 1 & \text{if } m < 2n - \frac{1}{n} \text{ and } m \cdot n \text{ is even,} \\ 2 \cdot (n+1)(n-1) + 1 & \text{if } m \geq 2n - \frac{1}{n}. \end{cases}$$

Meanwhile, the upper bound on the total length of a game in which Red uses the Gardner strategy and is not the starting player, $U_{G,B}(m,n)$, can be written as follows:

$$U_{G,B}(m,n) = \begin{cases} m \cdot n - 1 & \text{if } m < 2n \text{ and } m \cdot n \text{ is odd,} \\ m \cdot n & \text{if } m < 2n \text{ and } m \cdot n \text{ is even,} \\ 2 \cdot ((n+1)(n-1) + 1) & \text{if } m \geq 2n. \end{cases}$$

For the boards discussed in this section, where $n = 5$, the tipping point for this upper bound, when continuing to add more columns, occurs when $m = 10$. For $m^* \geq 10$, the upper bound no longer depends on m and thus stays at a fixed value: $U_{G,R}(m^*, 5) = 49$ and $U_{G,B}(m^*, 5) = 50$. As the upper bound thus stays the same from this point, we can expect the average length of a game to continue to not increase much when widening the board further and will probably see it approach some fixed value.

Figure 9.5 shows the standard deviations of the results of the simulations. From these graphs we can see that, in addition to becoming shorter, the length of the winning path also becomes more consistent on wider boards when using the Gardner strategy — which does not seem to be the case for the $2k \times n$ strategy. While it is harder to make conclusive statements about the length of the games based on the standard deviations of these simulations, it does seem that while the $2k \times n$ strategy leads to a more consistent length initially, its standard deviation's steady increase does eventually overtake that of the Gardner strategy. Since the upper bound on the game length of the Gardner strategy is a fixed value when $m \geq 10$, the standard deviation will likely not increase much on wider boards either.

9.3 Comparison: Algorithm 7.1.1 vs. Algorithm 8.4.1

Now, we will look at the two algorithms discussed in this thesis that were specifically made for Cylindrical Hex on a $5 \times n$ board: Algorithm 7.1.1 — created by Van den Broek [13]— and Algorithm 8.4.1 — introduced in this thesis. While these algorithms can be used on other board sizes as well (with appropriate adjustments to, for example, the definition of the connecting area used in Algorithm 8.4.1), this section will focus on the boards these algorithms were designed for. We will compare results from board of size $5 \times n$ with $n = 4, 5, 6, 7, 9, 12$.

9.3.1 Effectiveness

As neither of these algorithms are proven to be optimal, our first question will be which is the most effective of the two. In other words, with which strategy is Red more likely to win?

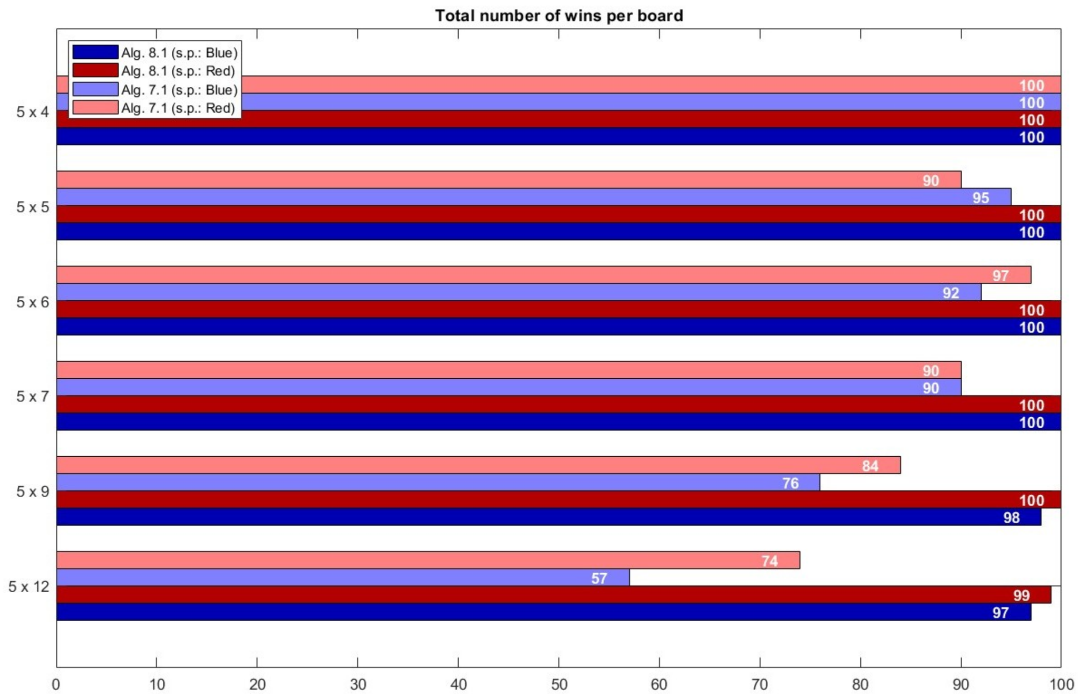


Figure 9.6: Number of won games out of the 100 games simulated per board for each possible combination of starting player and algorithm used by Red.

Overall, Algorithm 7.1.1 won 1045 games out of the 1200 simulated — i.e. about 87.08%.

As can be seen in Figure 9.6, the effectiveness of this algorithm depends on the size of the board: While Red won every game on the 5×4 board, the number of wins slowly decreases as more rings are added. This is not surprising as a longer board will require a longer red winning path. Since it will take longer to make this path, Blue will have more opportunities to interfere with Red's strategy and create their own winning path instead.

In general, while using Algorithm 7.1.1, games in which Red started were won at least as often as those in which Blue won. In fact, the number of won games seems to decrease faster when Blue is the starting player. The only outlier in the simulated data occurs on the 5×5 board, where Red lost a few more games when they were the starting player.

In comparison, out of 1200 games total simulated using Algorithm 8.4.1, 1194 were won — i.e. 99.5%.

Like with Algorithm 7.1.1, this strategy seems to work best on boards with less rings and is most effective if Red can make the first move. Figure 9.6 shows that the only games that were lost were played on the two largest boards — one game was lost on the 5×9 board and five on the 5×12 board — and overall only one game in which Red was the starting player was lost. We will discuss these games, and the strategy that allowed Blue to win them, in more detail in Section 10.1.

While both algorithms become less effective on larger boards, Algorithm 8.4.1 is clearly the more effective of the two: While Algorithm 7.1.1 already loses a few games on the 5×5 board, Algorithm 8.4.1 maintains a perfect record up to and including the 5×7 board and continues to win significantly more games than Algorithm 7.1.1 on larger boards. Additionally, once Red does start losing games using Algorithm 8.4.1, the number of won games seems to decrease at a much slower rate than when using Algorithm 7.1.1.

9.3.2 Efficiency

In Section 9.3.1, we saw that while neither strategy is optimal, Algorithm 8.4.1 is noticeably more effective at winning the game than Algorithm 8.4.1. Here, we will also compare the two algorithms in other ways, namely in terms of the overall length of all simulated games and the length of the winning red paths created in games won by Red.

The average number of moves made by both players during the simulated games are shown in Figure 9.7.

The length of a game doesn't differ much based on which algorithm is used. However, Algorithm 8.4.1 does lead to slightly shorter games on smaller boards. In fact, the average game length while using this strategy on a 5×4 board is 7.04 when Red makes the first move and 8.1 when Blue does — incredibly close to the minimum lengths of a game in which Red wins on this board, which are 7 and 8 respectively. Despite Algorithm 8.4.1 initially producing shorter games, this does not seem to hold true when the board becomes longer. On the 5×12 board, it is Algorithm 7.1.1 that produces shorter games, regardless of starting player. More data would be needed to confirm whether Algorithm 7.1.1 remains the faster algorithm on even longer boards.

As we can see from Figure 9.8, the length of a game is generally also more consistent when using Algorithm 8.4.1. When Red is the starting player, the only exception is the 5×12 board. This may indicate that Algorithm 7.1.1 is more consistent on bigger boards, but more data would be needed to confirm this. When Blue makes the first move, the only board on which Algorithm 7.1.1's game length is more consistent in the simulated games is the 5×7 board.

The average length of the shortest red winning path in simulated games won by Red is shown in Figure 9.7.

Like the game length, the average length of a winning path also does not differ much between the two algorithms. That being said, Algorithm 8.4.1 generally produces the shorter winning paths, with the only exception being the 5×9 board. There, Algorithm 7.1.1's winning paths in the simulated games were 0.0212 cells shorter on average when Blue was the starting player.

As we can see from Figure 9.10, the length of the Red winning path when Red is the starting player is also more consistent when using Algorithm 8.4.1. The only exception to this is the 5×12 board, which again may indicate that Algorithm 7.1.1 becomes the more consistent one in this regard as the board becomes taller. The more consistent strategy in terms of red winning path length when Blue is the starting player is far less consistent, making it hard to draw conclusions on the matter.

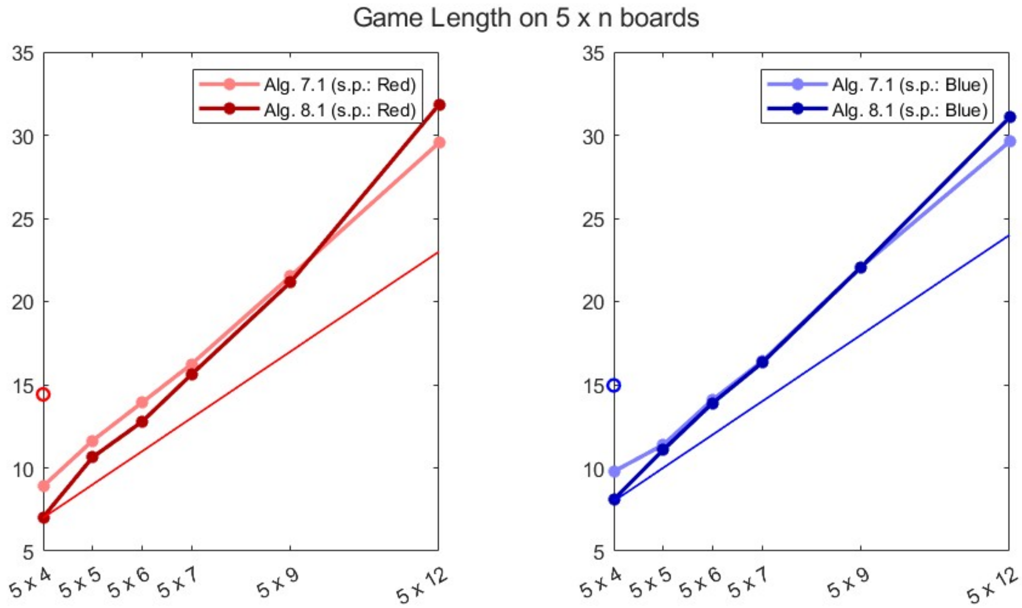


Figure 9.7: The results of the simulations done on various $5 \times n$ boards using Algorithm 7.1.1 (shown in lighter shades of red and blue), Algorithm 8.4.1 (shown in darker shades) and, on the 5×4 board, Algorithm 5.1.1 (indicated by a circle). The data on the Gardner strategy is limited to the only simulated board on which it is optimal.
Two additional lines indicate the minimum number of moves made in a game in which Red wins on a particular board.

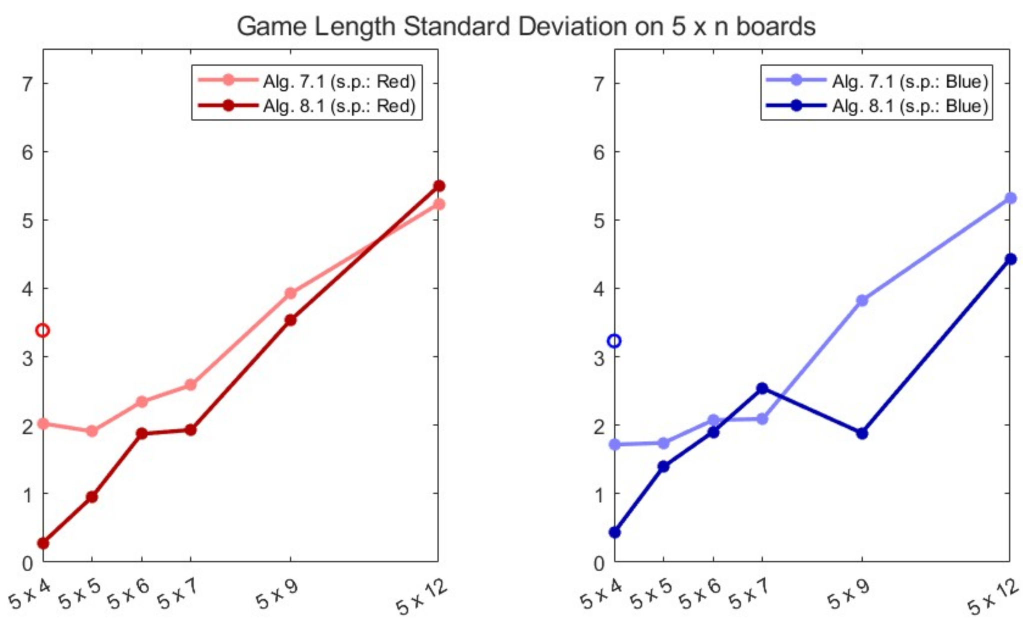


Figure 9.8: The standard deviation of the results shown in Figure 9.7.

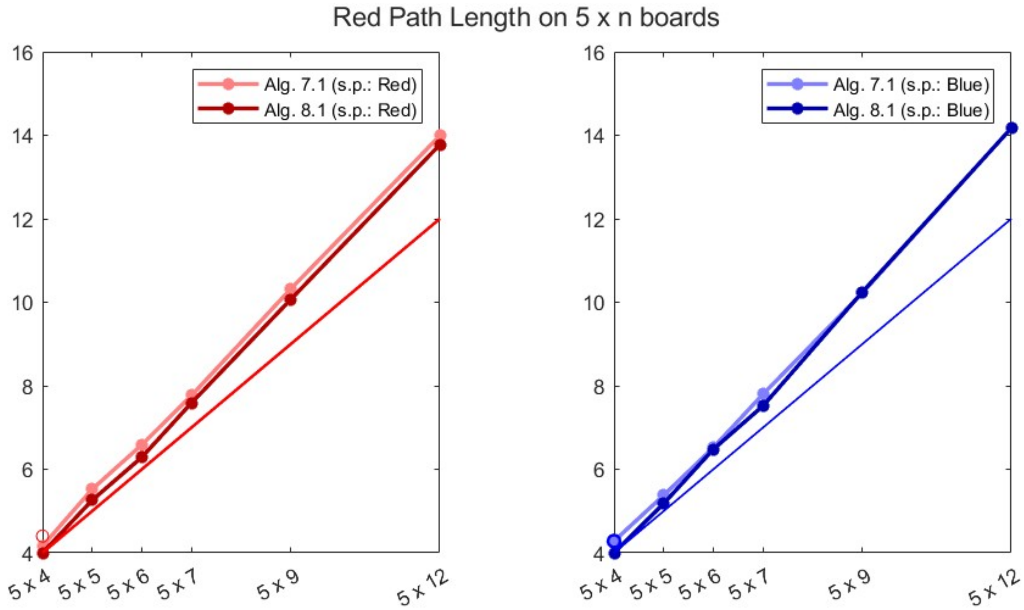


Figure 9.9: The results of the won simulations done on various $5 \times n$ boards using Algorithm 7.1.1 (shown in lighter shades of red and blue), Algorithm 8.4.1 (shown in darker shades) and, on the 5×4 board, Algorithm 5.1.1 (indicated by a circle). The data on the Gardner strategy is limited to the only simulated board on which it is optimal.
Two additional lines indicate the minimum length of a red winning path on a particular board.

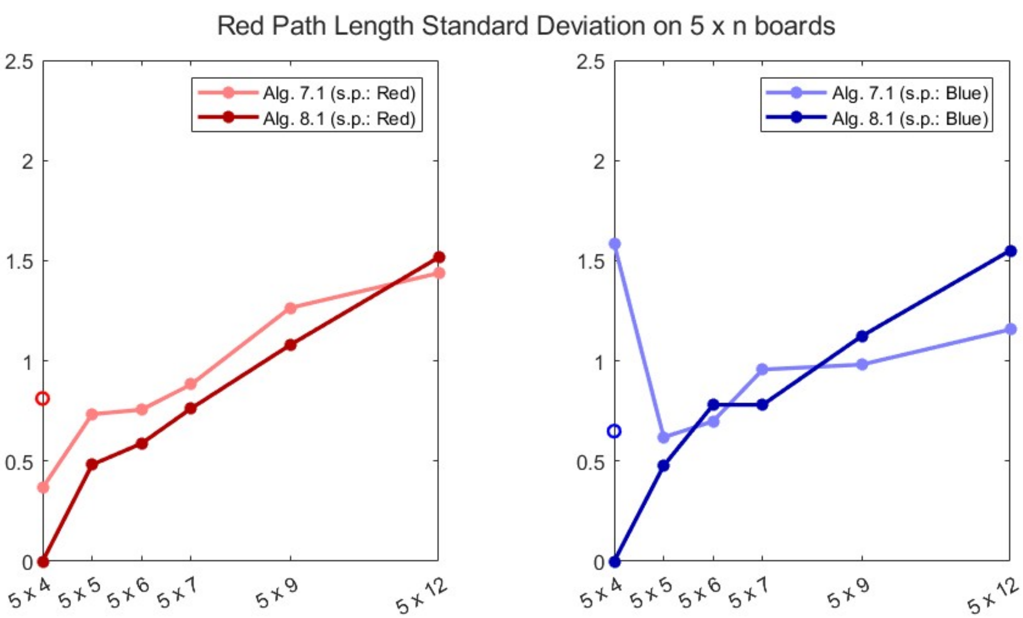


Figure 9.10: The standard deviation of the results shown in Figure 9.9.

9.4 Notes on Algorithm 8.4.1

The simulated games also give some insight into how Algorithm 8.4.1 is used by Red. We will discuss how frequently each rule and hierarchy item was used in these games in Sections 9.4.1 and 9.4.2, respectively.

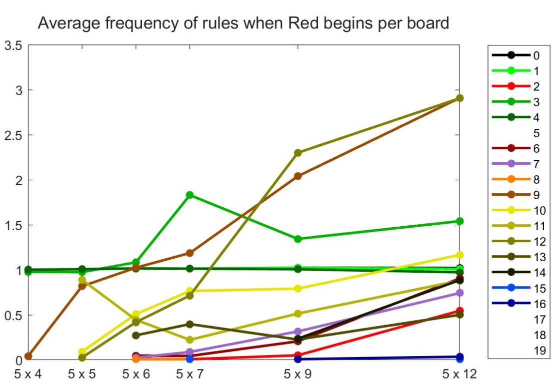
We also briefly look at how the algorithm behaves on smaller boards in Section 9.4.3.

9.4.1 Rules Used in Algorithm 8.4.1

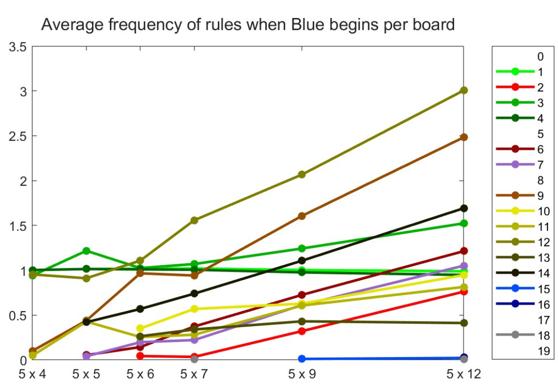
For the new algorithm introduced in this thesis — Algorithm 8.4.1 — the number of times a rule was used per game was also tracked. These results are shown in Figures 9.11 and 9.12 and largely lead to the same conclusions regardless of the starting player.

As one would expect, rule 1 is used exactly once per game on every board on which Red always wins. Once Red starts losing, this rule understandably becomes used slightly less often. This is easiest to see in Figure 9.12a, by comparing it to the use of “rule 0”: the first move Red makes when they are the starting player, which is always played exactly once in these games.

While rule 4 is initially also used once per game on average, it becomes less frequently much sooner, and does so at a slightly faster rate than rule 1 does. This indicates that while Red always creates a provisional winning path before making their winning move on smaller boards (i.e. boards of up to and including 7 rings if the starting player is Red and 5 rings if it is Blue), this is not always the case on larger boards. While on the largest boards, this is partially due to the lost games, this can not be the only cause for the decreased usage of rule 4. Instead some of the games on these boards must have been won by coincidentally making a provisional winning path while using rule 2, or without ever creating a provisional winning path at all. Despite this, the average number of moves used to fill in a winning path throughout a game — indicated by the use of rule 3 — does increase as the frequencies of rules 1 and 4 start to diverge from each other, likely as a result of the longer distance Red has to cover to create a winning path.



(a)



(b)

Figure 9.11: The average number of times Algorithm 8.4.1 used each rule during a single game in the simulations, per starting player. These results are also shown in Figure 9.12, separated into three graphs per player, to provide a clearer overview.

As the board becomes longer, we see that the average usage of rules 2, 6 and 7 each increase. Since these rules are all defensive measures against Blue's possible strategies, their increased usage indicates that these longer boards give Blue more opportunities to set these strategies up, likely as a result of Red now needing more time to create their own winning path.

Interestingly, a related defensive move — rule 5 — was never used at all. This indicates that whenever Blue did succeed in creating a provisional winning path, Blue could complete it in just one move and rule 2 had to be used to interfere instead.

One of the most noticeable increases in usage out of all rules occurs for rule 9. From this increase we can conclude that Blue attacks significantly more red bridges per game on larger boards. This increase is at least partially the result of Red creating more bridges on these boards throughout the game, as they have a larger distance to cover.

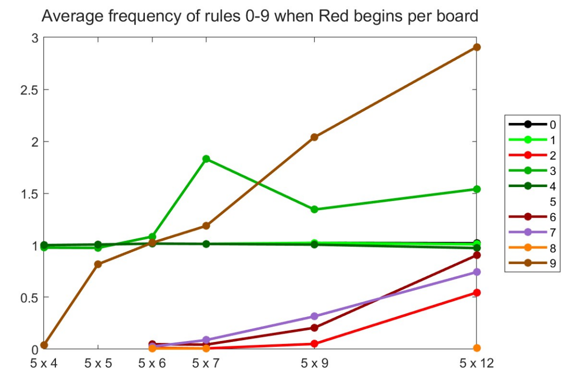
Rule 8, which protects overlapping bridges specifically, is only used when Red is the starting player, and even then only rarely, with the highest average use being 0.001 moves per game. Whether overlapping bridges were made at any point during a game was unfortunately not tracked during the simulations, so it is uncertain if this means that no such bridges were made at all when Blue was the starting player, or if, whenever a red overlapping bridge existed, only rules 1 to 7 were used.

The usage of rules 10 to 14, all of which extend red paths locally, tends to increase steadily with the length of the board. Out of these 5 rules, 12 is generally the most popular rule regardless of starting player — especially on larger boards. In fact, on larger boards, the starting player does not seem to have a large impact on which of rules 10 to 13 is used most and least frequently.

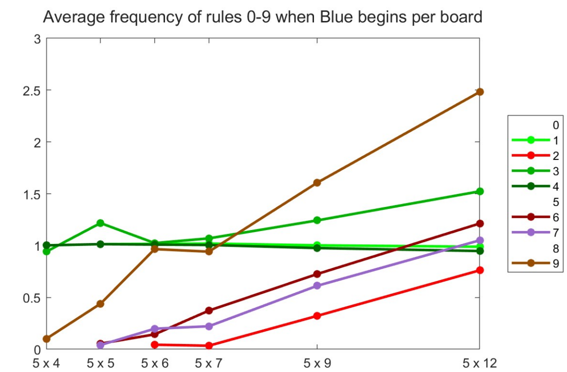
Only rule 14 seems to have a drastic change in importance: It is the second-most popular rule when Blue begins and is used on most boards, but when Red begins, it is the second-least popular on the two boards it is used on. The most likely cause of this phenomenon is red's initial move as the starting player. This starting move is made in the middle of the board, making it more likely that they will be able to use rules 10 to 13 to connect to this ring later and thereby leaving less incentive to use rule 14. On the other hand, when Blue is the starting player, Red is forced to use rule 14 if Blue plays on the rings in the middle of the board early on in the game: Since these rings are far away from both the top and the bottom of the board, and there is no longer an extra red cell in the middle of the board to connect to, it is far less likely that Red will be able to connect multiple rings to each other in this situation, leading to the use of rule 14. When Red's starting move is added to the number of times rule 14 is used, as is done in Figure 9.12c, we see that the importance of this combination of rules more closely resembles the importance of rule 14 in Figure 9.12d.

Initially, only a few of rules 10 to 14 are used. In fact, on the 5×4 board, none of these rules are used at all when Red is the starting player. We will briefly discuss this particular phenomenon in Section 9.4.3. On the other small boards, the top and bottom are close enough to each other that most moves Red can make will result in a bridge to one of these two sides. This makes rules that only cover one or two rings much less likely to be used. As the board becomes longer, however, we reach a point where, initially, there are no red cells near enough to create a large cover, and other rules are therefore utilised as well.

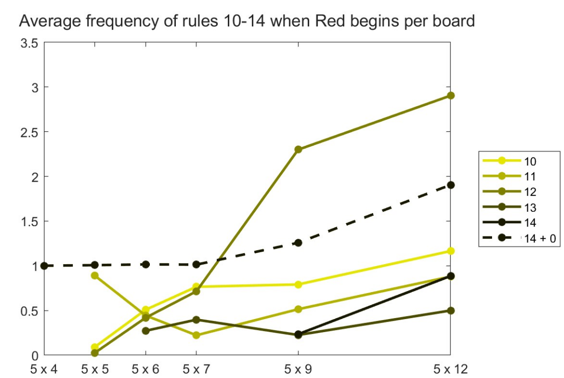
The remaining rules are used very infrequently, if at all. Rules 15, 16 and 18 are used very rarely and only on some of the largest boards — though not necessarily on consecutive ones in terms



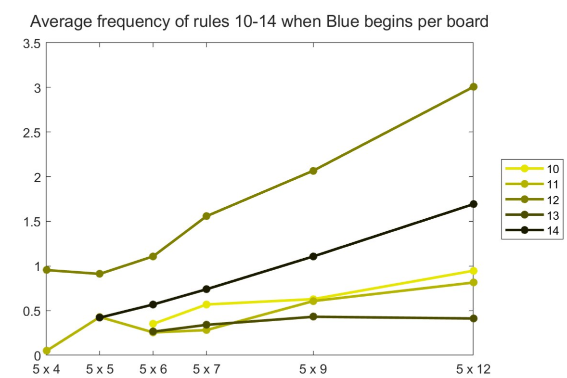
(a)



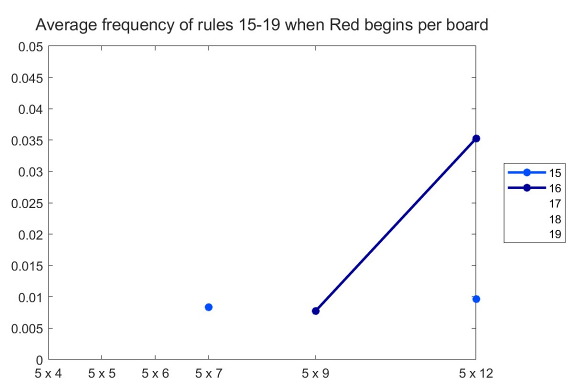
(b)



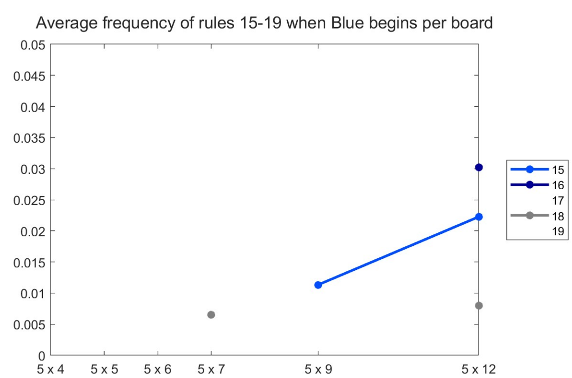
(c)



(d)



(e)



(f)

Figure 9.12: The average number of times Algorithm 8.4.1 used each rule during the simulations, displayed across three graphs, per starting player.

of length. Interestingly enough, rule 18 in particular was only used when Red was the starting player. Whenever these rules were used on multiple boards, there did appear to be an increase in the frequency at which they are used as the board lengthens.

Other than rule 5 — which was discussed earlier — only rules 17 and 19 were never used in the simulated games.

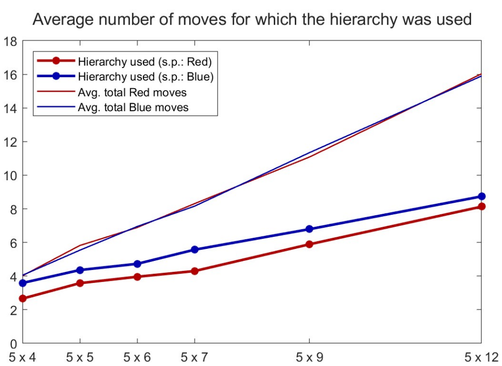
9.4.2 Hierarchy Items Used in Algorithm 8.4.1

In addition to how often each rule was used in Algorithm 8.4.1, usage of the hierarchy items was also tracked.

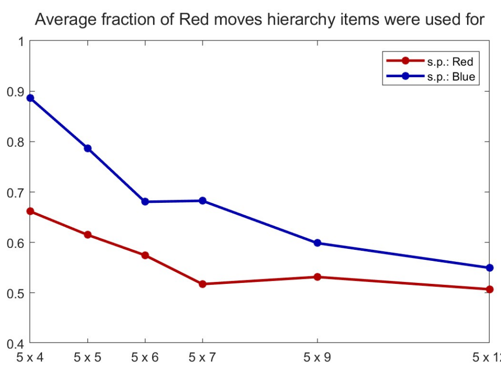
As already mentioned in Section 9.1, when discussing the way the games were simulated, these results may differ slightly from their 'true' value. For a few rules from Algorithm 8.4.1, Red considered copies of moves as if they were separate options. As such, the hierarchy may occasionally have been used to when it wasn't truly necessary — for example, when only copies of one single move remained: The chosen move would have been the same regardless of which copy was chosen, but as each copy was considered a separate move, the hierarchy was still used until only one copy remained.

This did not influence the outcome of the games and was only done for a few rules — rules 16, 17 and 18 — , each of which was used very infrequently, as we have seen in Section 9.4.1. Most rules did not consider each copy as its own separate move and as such, the inaccuracies in these results should be negligible.

Figure 9.13a shows the average number of moves for which the hierarchy was used. In other words, the graph shows the number of times Red had several moves they could make to satisfy the chosen rule and had to eliminate moves using the hierarchy. Figure 9.13b shows the frequency at which the hierarchy was used as a fraction of the total number of moves made by Red.



(a) The average number of moves for which the hierarchy was used, compared to the average total number of moves Red made during a game, per starting player.



(b) The average fraction of Red moves for which the hierarchy was used, per starting player.

Both figures show that the hierarchy is used more frequently when Blue is the starting player. The total number of times the hierarchy is used, shown in Figure 9.13a, increases with the size of the board at about the same rate for each starting player. It does not, however, increase as quickly as the total number of Red moves per game does. It seems that as the board becomes longer, Red will find only a single option for the chosen rule more and more often. As a result, the fraction of moves for which the hierarchy is used, shown in Figure 9.13b, actually decreases on longer boards.

Figure 9.14 shows how many different hierarchy items were used to eliminate potential moves for each instance in which the hierarchy was needed. An item was counted as being used if it removed at least one option from the list, without removing all options. The number of items used fluctuates for both players as the board size changes, but overall, it seems to increase when Red is the starting player and decrease when Blue is the starting player.

Figure 9.15 shows how often each hierarchy item was used on average per game. These results seem to be largely the same for each starting player. While items 2 and 7 (regarding the number of blue neighbours, and the number of ways a vertical bridge can be made, respectively) were not used at all in the simulations, all other items seem to be used more frequently as the size of the board increases.

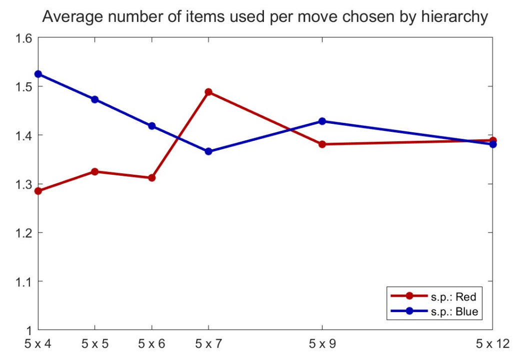
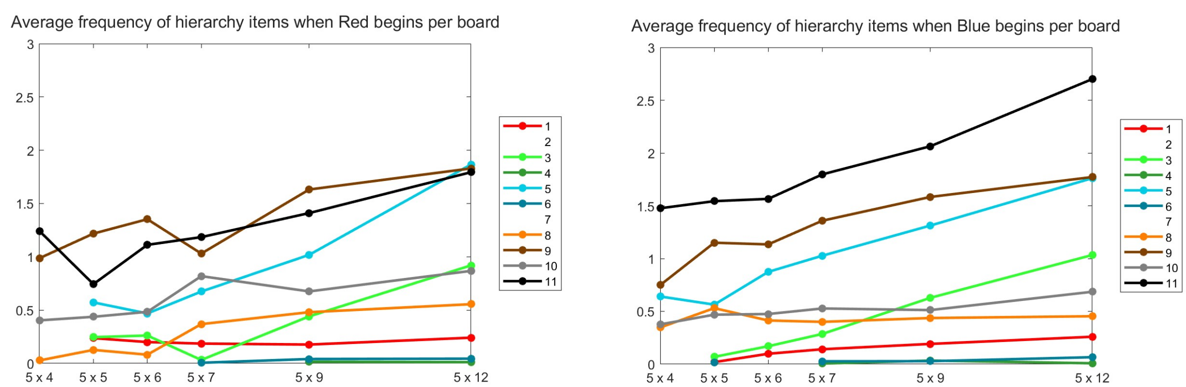


Figure 9.14: The average number of hierarchy items that were needed to determine the best move, whenever the hierarchy was used to decide which move to make.



(a) (b)

Figure 9.15: The average number of times Algorithm 8.4.1 used each hierarchical item during the simulations, per starting player.

Despite the order in which these items appear, the hierarchy is more likely to prioritise moves that will allow them to connect disjoint paths later (item 5) than moves that immediately lengthen a path or connect disjoint paths (items 3 and 4, respectively). It also prioritises moves with more red neighbours (item 9) surprisingly often.

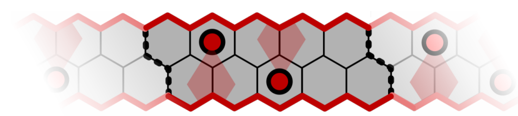
The most notable difference between starting players is the usage of item 11 (the final preference, which arbitrarily gives priority to the left-most column): This item is used noticeably less when Red makes the first move.

9.4.3 Algorithm 8.4.1 on Smaller Boards

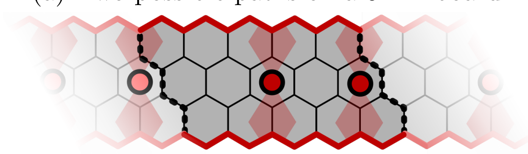
It should be noted that on $5 \times n$ boards, with $n \leq 3$, Algorithm 8.4.1 is optimal. In fact, Red will always be able to guarantee a win in just one move.

On a 5×1 board, this is trivial, as any move Red can make will immediately connect the top and bottom of the board and is therefore a winning move. Only rule 0 or 1 would be used on this board.

On 5×2 and 5×3 boards, Red will always be able to make a true provisional winning path on their first move, thereby guaranteeing a win. If Red makes the first move, playing according to the algorithm — i.e. in the middle of the board, which, in the case of the 5×2 board could, refer to either of the two rings — will immediately create this true provisional winning path on either board. Red will only use rules 0, 1, and 3 in this situation.



(a) Two possible paths on a 5×2 board.



(b) Two possible paths on a 5×3 board.

Figure 9.16: Two possible non-overlapping true provisional paths Red could make on two different empty boards.

If Blue is the starting player on either of these boards, Red can still always create a true provisional winning path in their first move by playing somewhere in a middle ring of the board: As shown in Figure 9.16, Red has at least two disjoint ways of creating such a path on an empty board. While Blue's starting move may interfere with one of these options, they will never be able to prevent both, because the resulting paths don't overlap, allowing Red to create a true provisional winning path, regardless of where Blue makes their first move. Red will only use rules 1, 3 and 4 in this situation.

If Red is the starting player on a 5×4 board, like the one that was used in the simulations, Red is still guaranteed to win quite quickly using Algorithm 8.4.1: Their first move, in the middle of the board, will create a bridge between either the top of the board (i.e. ring 0) and ring 2, or the bottom of the board (i.e. ring 5) and ring 3. Once this bridge has been made, Red has several ways to complete the provisional winning path in the future using just one move. As we can see in Figure 9.17, at least two of these paths do not overlap with each other at all, apart from the shared bridge made in the initial move of the game. This means that if Blue tries to interfere with one of the moves shown in Figure 9.17, or one of the bridges resulting from said moves, Red will still be able to make a provisional path using the other move. Red is therefore guaranteed a true provisional winning path in either two moves (if Blue plays their first move — i.e. the second move of the overall game — anywhere outside of the existing red bridge) or three moves (if Blue attacks the red bridge, and Red is therefore forced to defend this bridge before completing the path).

Because of this, Red will only use a few specific rules on this board if they are the starting player — rules 0, 1, 3, 4, and, if Blue attacks their first bridge, rule 9 — , which coincides with the data from the simulations on this board, shown in Figures 9.11a, 9.12a, 9.12c and 9.12e.

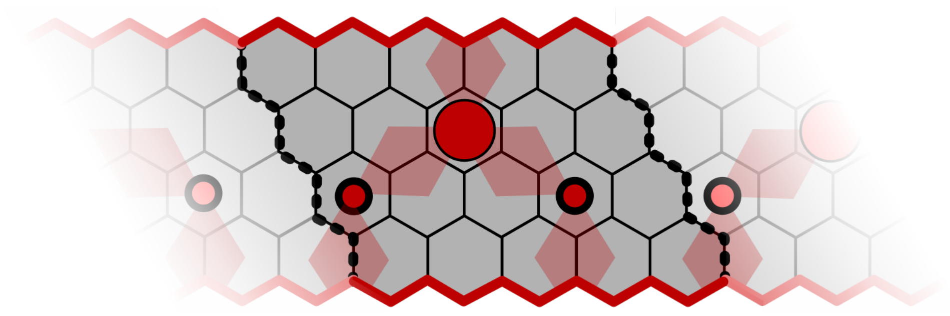


Figure 9.17: Red's first move as a starting player and two possible moves Red could make as their second (or third) move on a 5×4 Cylindrical Hex board to create a true provisional winning path. The resulting paths of both moves are also shown and do not overlap.

Chapter 10

Possible Future Improvements

In this section, we will briefly discuss some of the issues that Algorithm 8.4.1 still has. While no further improved algorithm will presented in this thesis, some ways in which these issues might be fixed in the future will be pointed out as well.

The most important of the issues we will discuss is how the algorithms responds to the type of strategy that Blue used in each of the simulated games lost by Red. This will be discussed in Section 10.1.

We will also discuss a situation in which rule 18 is used, and how this indicates where there is further room for improvement, in Section 10.2.

Finally, we will discuss the potential of using live and dead cells to help choose the next move in Section 10.3.

10.1 Preventing Blue's Current Winning Strategy

When we look at the moves made in the simulated games lost by Red in more detail, a pattern starts to emerge: Blue ends up using a similar strategy every single time. In this section, we will look at the way this strategy is used in one of the simulated games in detail to illustrate the idea behind the strategy more generally. The most relevant part of this game is shown in Figure 10.2 and the full list of moves made during the game can be found in Appendix A.1.1. The list of moves of all other lost games can also be found in Appendix A.1.

10.1.1 How Does Blue's Strategy Work?

There is a pattern of cells that often occurs in Blue's current strategy, shown in Figure 10.1. We find this same configuration in the position shown in Figure 10.2a, in rings 3 to 6, as well as in most of the other lost games — sometimes having been rotated 180 degrees.

That being said, variations of this layout are also possible, and Blue may also make use of cells beyond these rings: Both the second game listed in Appendix A.1.3 and the game listed in Appendix A.1.2 prepare their strategy in a slightly different way, and even the game we discuss in detail here ends up using an additional Blue cell — cell $(2, 7)$. While the positions in these games may

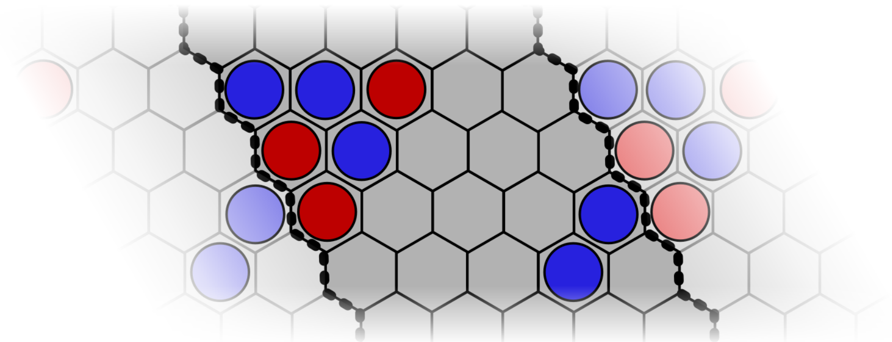


Figure 10.1: The configuration of cells that shows up in most of the simulated games won by Blue. In some of the games, the configuration was rotated by 180 degrees.

be slightly different, both the idea behind their strategies and the way Red could have prevented them are the same as the one described in this section.

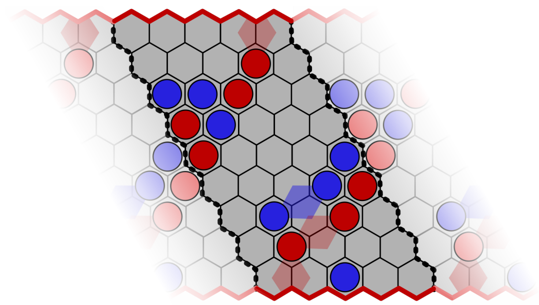
In the game shown in Figure 10.2, the strategy unfolds from the position shown in Figure 10.2a: From this position, Blue plays on $(3, 4)$, threatening to make a true provisional winning path on their next turn, in such a way that Red has two ways of preventing this from happening. Blue's move and Red's options to prevent a blue provisional winning path are both shown in Figure 10.2b.

Red will use rule 6 to determine their next move, and uses the hierarchy to decide which of the two possible moves to make, to prevent the blue provisional winning path. In this case, Red chooses to play on $(5, 3)$.

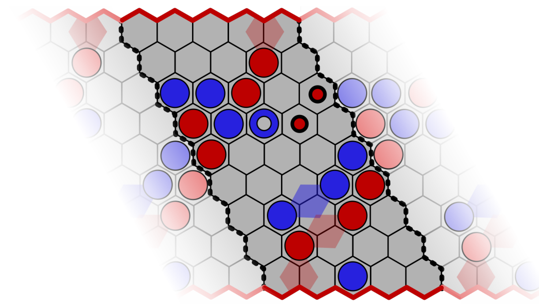
Blue is then able to play on $(5, 4)$, threatening to make a winning path on their next turn, while also threatening to make a provisional winning path on their next turn, such that only one of these paths can be prevented at the same time. Faced with this position, Red will attempt to prevent the winning path first — according to rule 2 — as that is the bigger threat. Both Blue's move and the move Red will be forced to make are shown in Figure 10.2c.

Red being forced to play on $(4, 4)$ leaves the opening Blue needs to make good on their other threat: creating the provisional winning path shown in Figure 10.2d by playing on $(2, 6)$. As this is a true provisional winning path, they are now guaranteed to win as long as they play optimally.

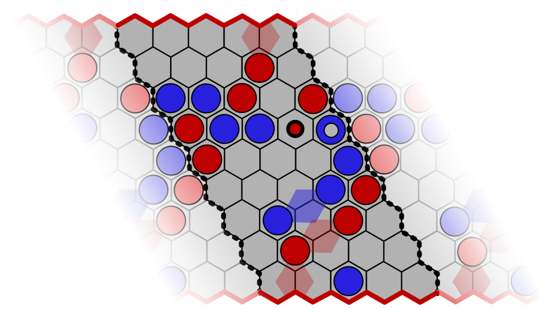
While there are differences in terms of the layout and exact execution of the strategy, the idea behind Blue's strategy remains the same in each of the simulated games that they won: Blue creates a position similar to the one shown in Figure 10.2a, before threatening to make a true provisional winning path on their next turn, such that Red can prevent this in one of two ways. If Red does not make the correct move in response, Blue is able to make a move with which they both threaten to make a winning path and a (true) provisional winning path on their next turn. With the way Blue has set this strategy up, Red can only prevent one of these paths on their turn, and thus, Blue will be able to guarantee their win, regardless of what choice Red makes.



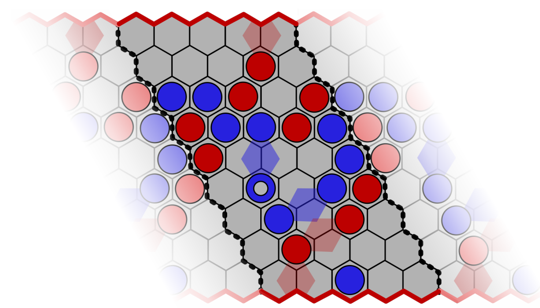
(a) A position in the latter half of the game.



(b) Blue has made a move on $(3,4)$, threatening to make a true provisional winning path on their next move.
Using rule 6, Red has two ways in which they can prevent this: $(5,3)$ and $(4,4)$.



(c) After Red plays on $(5,3)$ (chosen according to the hierarchy) Blue makes a move on $(5,4)$, threatening to make a winning path on their next move, while simultaneously threatening to make a provisional winning path.
Using rule 2, Red will have to play on $(4,4)$ to prevent the winning path.



(d) After Red plays on $(4,4)$, Blue can make a move on $(2,6)$, creating a true provisional winning path. If Blue plays optimally from this point onwards, they are guaranteed to win.
Note that while Blue uses the cell $(2,7)$ in this path, it is not strictly required: Playing on cell $(3,5)$ would have yielded the same outcome.

Figure 10.2: Some of the later positions found in one of the games simulated on a 5×9 Cylindrical Hex board. In this game Blue was the starting player and Red, while using Algorithm 8.4.1, lost the game.
The list of moves made throughout this game is listed as the first game in Appendix A.1.1.

10.1.2 How Can Blue's Strategy be Prevented?

As all of Blue's wins are the result of a variation of this strategy, it is important to find a way to prevent it in future attempts at an optimal algorithm for the $5 \times n$ board. This may be quite easy, as it seems that a single change by Red could have saved the game in each case: Their decision in response to the first threat, shown in Figure 10.2b.

This decision was made according to the hierarchy of Algorithm 8.4.1. This hierarchy was created to estimate what the best move would be among several potential moves that satisfy a given rule. However, it seems that in this case, the hierarchy chooses the wrong move: If Red chooses the other option instead, as shown in Figure 10.3, Blue's strategy falls apart.

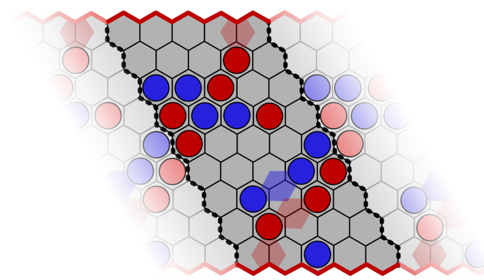


Figure 10.3: The position on the board if Red had chosen to play on $(4, 4)$, when faced with the position shown in Figure 10.2b. From this position, Blue can no longer make two disjoint threats at once.

From the position shown in Figure 10.3, Blue is no longer able to make two separate threats with only one move. This means that Red can now respond to each threat Blue might make individually, without leaving another opening for Blue to make a different (provisional) winning path. By changing this one decision, Blue is no longer able to win using this strategy. Ensuring that the right choice is made in response to these types of positions would thus be very beneficial in improving Algorithm 8.4.1 further.

Making a separate rule (or sub-rule) to recognise and deal with this strategy, as was done for the double-bridge set-up in this thesis, might be an option. However, while Blue used the same set-up for most of the simulated games in which Red lost, the lay-out was a bit different for the others. Dealing with this strategy by adding a separate rule would thus require careful thought regarding the best way to detect this strategy. It would likely involve looking a step further ahead at Blue's future moves, to find dilemma's such as this before Red is actually faced with them.

Adjusting the hierarchy — by changing its order or introducing a new item, for example — in such a way that the correct decision is made even in this situation, may also be an option. Of course, when using this method, it will be important to ensure that the best move is still chosen in other situations as well.

10.2 Eliminating Situations in which Rules 17 to 19 are Used

Rules 17 to 19 of Algorithm 8.4.1 are only used if Red can't find a way to (safely) make a useful move. In an optimal algorithm, we wouldn't expect to use them at all. However, rule 18 was used exactly once in the 1200 simulations of Algorithm 8.4.1. While the game in which it was used was ultimately won by Red, this is still an indication that there are situations in which Red doesn't really know what to do. Understanding why rule 18 was used could therefore point to some other ways in which the algorithm could still be improved.

10.2.1 When Was Rule 18 Used?

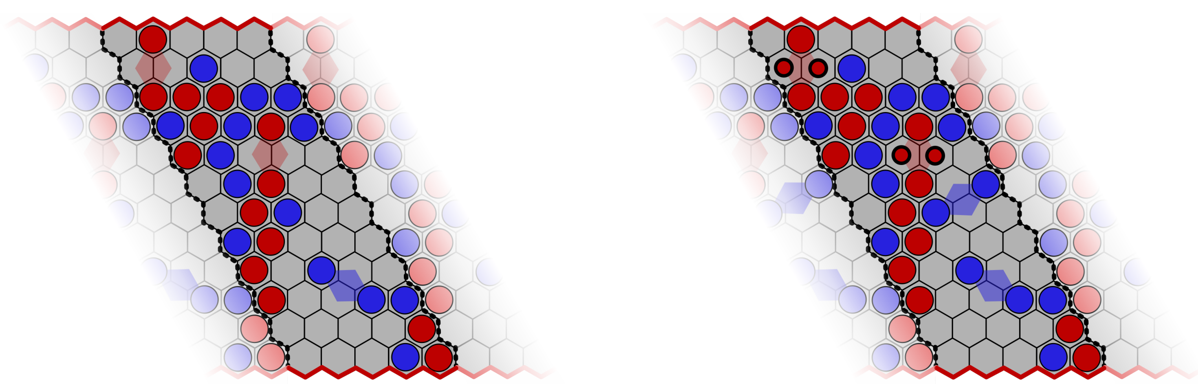
The full list of moves during the simulated game in which rule 18 was used can be found in Appendix A.2. Figure 10.4a shows the position on the board shortly before rule 18 was used. In this position, Red already has two provisional paths, which together already cover the full board — the sets of rings each path covers even overlap slightly. The two paths could even be turned into a true provisional winning path with just a single move — for example, by playing on $(4, 6)$.

When Blue plays on $(5, 6)$, however, the current version of the algorithm runs into a problem. Looking at Figure 10.4b, we can see that playing on $(4, 6)$ would no longer create a provisional winning path. While Red could still make a provisional winning path in this situation, by playing on $(5, 5)$, this would create an overlapping bridge between the cells $(3, 6)$, $(4, 4)$ and $(5, 4)$. As this is regarded as too much of a risk, this option is disregarded by the algorithm.

As rings 4 to 8 are already covered by one of the provisional paths, Red will eventually try to lengthen the path going through ring 6 according to rule 16. However, they are unable to do so. One end is already connected to the bottom of the board — there is no need to extend this side further. The other end of the path is stuck in a dead end. Blue has already formed a blockade around the red cell $(4, 4)$, leaving no room for Red to connect this cell to either ring 2 or ring 3. There is also no room for this path to go around the blockade by playing in the same ring — in addition to this cell being too surrounded by Blue to immediately connect this path to another cell in ring 4, all other cells in the ring have also already been played on.

Since rule 16 also can't be used, we are left with the last 3 rules. Red ends up having to play on one of the bridge cells, according to rule 18. In this case $(4, 5)$ will be chosen by the hierarchy, as this will somewhat close the gap between the two provisional paths.

This simulated game was ultimately won by Red after 5 more turns. However, Red did have to use rule 2 twice to prevent a Blue win before they were able to create a provisional winning path.



(a) A position in the latter half of the game, just before Blue makes their next move.

(b) Blue has made a move on $(5, 6)$. As Red is not able to use rules 1 to 17 without making an overlapping bridge, they are forced to play on one of their bridge cells, according to rule 18.

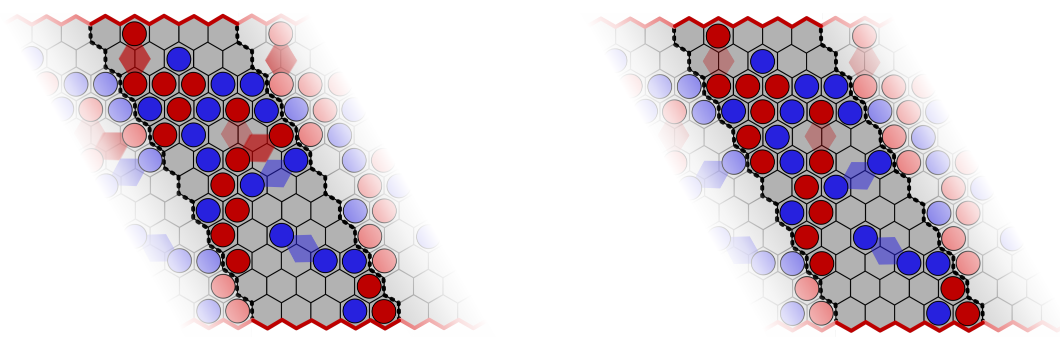
Figure 10.4: A position on the 5×12 Cylindrical Hex board, in which Blue was the starting player and Red used rule 18 from Algorithm 8.4.1.

10.2.2 What Might Have Been a More Beneficial Move?

There are several ways in which one could potentially improve Algorithm 8.4.1's response to this situation in a future iteration of the algorithm.

As we can see in Figure 10.5a, only one of the overlapping bridges formed by playing on $(5,5)$ would be used by the resulting provisional winning path. This means that while only one of these bridges can be saved if Blue were to play on the point of overlap, Red can always save the provisional path by choosing the correct bridge to protect: Despite the existence of the overlapping bridge, it is a true provisional path. By adjusting the algorithm to allow this, the game would end after two more turns, as Red would only need to fill in the two bridges along this path. Allowing Red to make an overlapping bridge in situations like this — provided that doing so creates a true provisional path — would not only shorten the game, but would also reduce the risk Red has to take in this situation.

Rule 16 might be improved to help deal with similar situations. While we have added a mechanism for trying to go around a blockade by playing in the same ring as the end of a path, it may, in some cases, be useful to backtrack further — though in this particular game, doing so would likely still lead to playing on $(4,5)$. Alternatively (or additionally), Red could consider paths that can be extended to ring j in a single move, in addition to paths that already cover this ring, when applying rule 16: In the case of Figure 10.4b, this would result in Red playing on $(1,6)$, as shown in Figure 10.5b, which would lengthen one of the paths and bring the two provisional paths closer together, without creating an overlapping bridge.



(a) Playing on $(5,5)$ would create an overlapping bridge, but only one of these bridges is part of the provisional winning path. Bridges that are part of this path are shown in a slightly darker shade of Red.

(b) Playing on $(1,6)$ would lengthen the other provisional path, while avoiding overlapping bridges and bringing the two paths closer together.

Figure 10.5: Two alternative moves Red might have made in response to Blue playing on $(5,6)$ from the position shown in Figure 10.4a.

10.3 Implementing Live and Dead Cells

Algorithm 8.4.1 may also be further improved by properly implementing the concept of live and dead cells. This concept can be used as an indication of which moves are actually useful to Red and which moves will not make a difference in the long run.

A cell is considered to be dead if it no longer influences the outcome of a game: On any colouring that can be obtained from filling all unplayed cells on the board, the winner of the game doesn't depend on the colour of this particular cell.

To determine which cells are live and which are dead at a specific moment in the game, we thus need to look at the completed colourings of this position.

Definition 10.3.1 (Completed colouring)**.** *A colouring C' is a completed colouring of the partial colouring C, if the following all hold:*

- *For all $x \in V_{m,n}$, such that $C(x) = red$, we also have $C'(x) = red$.*
- *For all $x \in V_{m,n}$, such that $C(x) = blue$, we also have $C'(x) = blue$.*
- *For all other $x \in V_{m,n}$, either $C'(x) = red$ or $C'(x) = blue$*

Formally, live and dead vertices can be defined as follows:

Definition 10.3.2 (Live and Dead Vertices)**.** *A vertex $x \in V_{m,n}$ is considered live in a partial colouring C on $H_c(m, n)$ if there exists at least one completed colouring C' of C, such that changing the colour $C'(x)$ changes the winner in C'.*

If a vertex is not live, it is considered dead.

This definition is based on the one given in [6]. An alternative way of defining live and dead cells, which uses sets of unplayed cells, can be found in [3].

This concept can be quite useful for Red. As a dead cell will not influence the outcome of the game, playing on one is never advantageous for Red — they would only be wasting a move by doing so.

Playing on live cells instead is far more beneficial. Limiting the moves Red will consider in a future iteration of Algorithm 8.4.1 to only live cells could thus improve its effectiveness and efficiency.

While it is possible to analyse every completed colouring of the current position to determine exactly which cells are live and which are dead, this can be quite time-consuming, especially on larger boards. A more efficient way of including this concept is by recognising common patterns that cause a cell to be dead within a colouring.

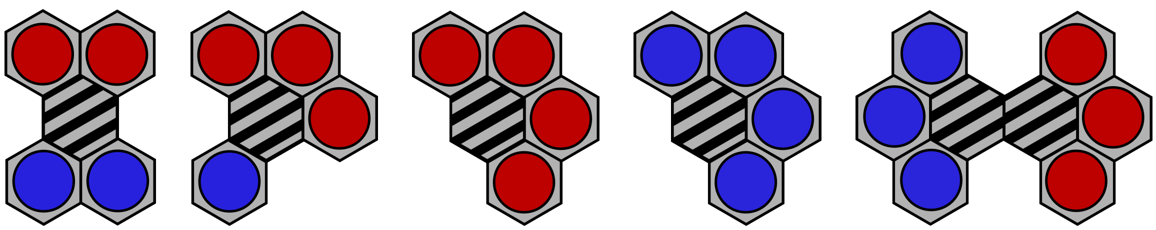
Algorithm 8.4.1 already tries to do this to some extent, through hierarchy items 2 and 9. Both of these items encourage Red to disregard cells that have been surrounded by 5 or 6 other cells of the same colour if other options are available: These cells are too surrounded to meaningfully contribute to a winning path.

However, the current algorithm does still allow these cells to be used for certain rules if there is no alternative. If a future iteration of this algorithm were to properly recognise these cells as dead cells, and therefore not advantageous to Red at all, they can be completely disregarded.

Additionally, more patterns that indicate the existence of dead cells could be included. This would further reduce the number of moves Red needs to consider by only allowing those that Red could truly benefit from. Some patterns that indicate the presence of a dead cell are mentioned by Fabiano and Hayward [3], and recreated here in Figure 10.6.[1]

The definition of live and dead vertices, as given in this thesis, can easily be applied to both unplayed and coloured vertices. This means that a vertex that was played on while it was live may become dead at a later point in the game. When this happens, we know that that vertex will never become part of a (shortest) winning path.

For an example of this happening, one could take some time look at the game discussed in Section 10.2 in more detail — the complete list of moves made during this game can be found in Appendix A.2. When Red played on the cell $(4, 4)$ during this game, this cell only had two Blue neighbours — cells $(3, 3)$ and $(5, 4)$. At that point, it was still possible to create a complete colouring of this position's partial colouring such that changing the colour of cell $(4, 4)$ would change the winner in that position — in other words, the cell was still live. However, by the time we reach the position in Figure 10.4a, Blue has created a blockade around this cell. This blockade is the same as the pattern shown in Figure 10.6d, rotated slightly, which indicates that $(4, 4)$ has, at some point after being played on, become a dead cell.



(a) (b) (c) (d) (e)

Figure 10.6: Several configurations of Red and Blue cells that indicate the existence of a dead cell, which is indicated by Black stripes

[1] Unfortunately, papers suggesting the use of dead cells to reduce the number of moves Red can consider were only found once a significant number of simulations had already been run using Algorithm 8.4.1. Adjusting the algorithm to account for these positions and redoing all of these simulations would have been too time-consuming. As such, it was not possible to integrate these positions into the algorithm for this thesis.

Discussion

Throughout this thesis we discussed the game Hex, as well as two variations on this game — Cylindrical Hex and Torus Hex — and some strategies that can be used in these games.

We paid special attention to two strategies in particular:

- **Algorithm 5.1.1 — The optimal Cylindrical Gardner strategy:**

  In addition to rigorously proving Martin Gardner's strategy for asymmetrical Hex boards (Algorithms 4.1.1 and 4.1.2), we showed that it remains an optimal strategy for Red in Cylindrical Hex if $m < n$.

  We also compared the performance of Algorithm 5.1.1 to that of Algorithm 4.2.1 on boards on which both are optimal using simulated games. In these games Algorithm 5.1.1 was noticeably more efficient in terms of both the game length and the winning path length, especially on wider boards. This is likely as the result of the upper limit on the game length, which does not depend on the number of columns the board has.

- **Algorithm 8.4.1 — A $5 \times n$ strategy:**

  After analysing an earlier attempt at creating an optimal strategy for Cylindrical Hex on a $5 \times n$ board (Algorithm 7.1.1), we introduced a new $5 \times n$ strategy. While this new strategy is based on the earlier strategy, several of its rules were added, removed or changed, and a hierarchy of moves was added, in an attempt at fixing the shortcomings we found.

  We then compared the performance of these two strategies using simulated games. While Algorithm 8.4.1 is not yet optimal, it does perform noticeably better than its predecessor — winning more games and maintaining a perfect record on most of the simulated boards. In terms of efficiency, the two algorithms closely matched each other.

  By looking at the games that Algorithm 8.4.1 lost and a game in which rule 18 was surprisingly used, we were able to find some more points for improvement, that may be useful for creating a future iteration of the algorithm. Additionally, we discussed how the theory of live and dead cells may also be beneficial in a future iteration of the algorithm.

The obvious avenue for future research would be further improving Algorithm 8.4.1. While a few shortcomings and possible solutions were already discussed in this thesis, we cannot say whether these solve the remaining issues without further research.

It would also be interesting to see how this algorithm (or, better yet, an improved version thereof) performs on other boards — for example on other $(2k + 1) \times n$ boards, or on a toroidal board. While we weren't able to run simulations on these boards due to time constraints, we do suspect that Algorithm 8.4.1 would also perform better in Torus Hex than its predecessor: If the algorithm is carried out on S — in the same way this thesis' simulations were carried out on S_n — forming a winning path becomes a matter of connecting a path to a copy of that path on a higher or lower board. As many of the changes made to Algorithm 7.1.1 were aimed at encouraging Red to connect, or plan to connect, to such other paths, Algorithm 8.4.1 will likely have an advantage in this regard.

The goal of this thesis was improving Algorithm 7.1.1 — hopefully to the point of optimality. Algorithm 7.1.1, in turn, was made by adjusting Algorithm 4.2.2 to be more effective on a $5 \times n$ board. While we were not able to present an optimal $5 \times n$ strategy in this thesis, we do seem to have come a step closer with Algorithm 8.4.1 — and were even able to provide an optimal strategy for a more general type of board with Algorithm 5.1.1. When looking at this process on the whole — including all the earlier strategies that led to this thesis — we have been getting closer to an optimal strategy each time, by learning from our losses, adjusting our strategy and trying again.

I think this makes one of Piet Hein's poems [7] a rather fitting way to end this thesis:

The road to wisdom? - Well, it's plain
and simple to express:

Err
and err
and err again
but less
and less
and less.

References

[1] Steve Alpern and Anatole Beck. “Hex Games and Twist Maps on the Annulus”. In: *The American Mathematical Monthly* 98.9 (1991), pp. 803–811. DOI: 10.2307/2324266.

[2] Broderick Arneson, Ryan B. Hayward, and Philip Henderson. “Solving Hex: Beyond Humans”. In: *Computers and Games* (2011). ISSN: 0302-9743. URL: https://webdocs.cs.ualberta.ca/~hayward/papers/solve.beyond.pdf.

[3] Nicolas Fabiano and Ryan Hayward. “New Hex Patterns for Fill and Prune”. In: *Advances in Computer Games* 12516 (2020), pp. 79–90. DOI: 10.1007/978-3-030-65883-0_7. URL: http://webdocs.cs.ualberta.ca/~hayward/papers/new-hex-fillin.pdf.

[4] David Gale. “The Game of Hex and the Brouwer Fixed-Point Theorem”. In: *The American Mathematical Monthly* 86.10 (1979), pp. 818–827. DOI: 10.1016/j.disc.2014.05.002.

[5] Martin Gardner. “The Scientific American Book of Mathematical Puzzles & Diversions”. In: *The Game of Hex* (1959), pp. 73–83.

[6] Ryan B. Hayward and Jack Van Rijswijck. “Hex and Combinatorics”. In: *Discrete Mathematics* 306.19 (2006), pp. 2515–2528. ISSN: 0012-365X. DOI: 10.1016/j.disc.2006.01.029.

[7] Piet Hein. *Grooks*. The M.I.T. Press, 1996.

[8] Samuel Clowes Huneke, Ryan Hayward, and Bjarne Toft. “A winning strategy for 3×n Cylindrical Hex”. In: *Discrete Mathematics* 331 (2014), pp. 93–97. ISSN: 0012-365X. DOI: 10.1016/j.disc.2014.05.002.

[9] Thomas Maarup. “Hex. Everything You Always Wanted to Know About Hex But Were Afraid to Ask”. [*This thesis was accessed via a so-called snapshot of the author’s own website — available using the Internet Archive’s Wayback Machine — as it was hard to find a reliable alternative. The snapshot was dated 09/02/2006 — the most recent snapshot in which thesis was available — and the url to this snapshot is provided here.*] University of Southern Denmark, 2005. URL: https://web.archive.org/web/20060209211555/http://maarup.net:80/thomas/hex/hex3.pdf.

[10] John Nash. “Some games and machines for playing them”. In: *RAND* (1952). URL: https://www.rand.org/pubs/documents/D1164.html.

[11] Art B. Owen. "Monte Carlo theory, methods and examples". In: *Simple Monte Carlo* (2013). URL: https://artowen.su.domains/mc/.

[12] Claude E. Shannon. "Computers and Automata". In: *Proceedings of the IRE* 41.10 (1953), pp. 1234–1241. DOI: 10.1109/JRPROC.1953.274273.

[13] Amber Van den Broek. "Hex Circles; Winning Strategies for variants of Hex using Pure Monte-Carlo". Leiden University, 2023.

[14] Marit Van Hees. "Hexenkringen: Onderzoek naar winnende strategieën voor Cilindrisch Hex". Leiden University, 2022.

Appendix A

Moves Made in Relevant Games

A.1 Games Lost by Algorithm 8.4.1

The lists below show the moves made during each of the games lost during the simulations of Algorithm 8.4.1. They are sorted into groups based on the same criteria as were used when analysing the results of the various algorithms discussed in this thesis in Chapter 9, i.e. based on board size and starting player.

A.1.1 5×9 Board, Blue Starting Player

Several positions that occur late in the first game listed here are shown in Figure 10.2.

Red - rule	Blue
–	$(2,7)$
$(4,7)$ - 14	$(4,6)$
$(5,6)$ - 13	$(5,5)$
$(1,5)$ - 12	$(3,9)$
$(2,8)$ - 7	$(2,3)$
$(3,3)$ - 14	$(2,4)$
$(1,4)$ - 7	$(1,3)$
$(4,1)$ - 12	$(3,4)$
$(5,3)$ - 6	$(5,4)$
$(4,4)$ - 2	$(3,5)$
$(4,5)$ - 2	$(3,6)$

Red - rule	Blue
–	$(2,2)$
$(1,2)$ - 12	$(1,7)$
$(3,7)$ - 14	$(1,3)$
$(5,3)$ - 12	$(4,5)$
$(3,5)$ - 7	$(4,4)$
$(5,4)$ - 7	$(5,5)$
$(2,7)$ - 12	$(3,4)$
$(1,5)$ - 6	$(1,4)$
$(2,4)$ - 2	$(3,3)$
$(2,3)$ - 2	$(3,2)$

A.1.2 5×12 Board, Red Starting Player

Red - rule	Blue
$(3,6)$ - 0	$(4,3)$
$(1,3)$ - 14	$(2,11)$
$(1,11)$ - 12	$(5,4)$
$(4,4)$ - 12	$(5,3)$
$(2,1)$ - 12	$(1,4)$
$(3,3)$ - 6	$(3,4)$
$(2,4)$ - 2	$(1,5)$
$(2,5)$ - 2	$(2,6)$
$(3,5)$ - 8	$(4,2)$
$(2,2)$ - 10	$(2,3)$
$(3,2)$ - 2	$(2,8)$
$(3,8)$ - 14	$(3,7)$
$(2,7)$ - 12	$(1,8)$
$(2,9)$ - 12	$(4,7)$
$(5,8)$ - 6	$(1,7)$
$(5,7)$ - 2	$(1,6)$
$(5,6)$ - 2	$(4,6)$
$(5,5)$ - 2	$(4,5)$

A.1.3 5×12 Board, Blue Starting Player

Red - rule	Blue
–	$(2,6)$
$(1,6)$ - 14	$(2,2)$
$(3,2)$ - 12	$(1,4)$
$(2,4)$ - 7	$(1,1)$
$(3,1)$ - 15	$(1,7)$
$(5,7)$ - 12	$(1,5)$
$(2,5)$ - 9	$(3,3)$
$(2,3)$ - 9	$(4,9)$
$(3,9)$ - 14	$(4,8)$
$(5,8)$ - 7	$(5,9)$
$(2,11)$ - 12	$(3,8)$
$(1,9)$ - 6	$(1,8)$
$(2,8)$ - 2	$(3,7)$
$(2,7)$ - 2	$(3,6)$

Red - rule	Blue
–	$(4,1)$
$(1,2)$ - 11	$(5,7)$
$(1,7)$ - 14	$(4,4)$
$(5,4)$ - 12	$(4,5)$
$(2,5)$ - 12	$(2,10)$
$(4,10)$ - 14	$(2,6)$
$(1,6)$ - 9	$(5,8)$
$(4,8)$ - 7	$(1,8)$
$(3,10)$ - 12	$(4,7)$
$(2,8)$ - 6	$(2,7)$
$(3,7)$ - 2	$(4,6)$
$(3,6)$ - 2	$(3,5)$

Red - rule	Blue
–	$(2,2)$
$(1,2)$ - 12	$(4,10)$
$(1,10)$ - 14	$(2,11)$
$(5,12)$ - 11	$(1,11)$
$(5,11)$ - 6	$(2,9)$
$(1,9)$ - 12	$(1,3)$
$(5,3)$ - 12	$(4,5)$
$(3,5)$ - 14	$(4,4)$
$(5,4)$ - 7	$(5,5)$
$(2,7)$ - 12	$(3,4)$
$(1,5)$ - 6	$(1,4)$
$(2,4)$ - 2	$(3,3)$
$(2,3)$ - 2	$(3,2)$

A.2 Game in which Algorithm 8.4.1 Uses Rule 18

The list below shows the moves made during the only game during the simulations of Algorithm 8.4.1 in which rule 18 is used. The game was played on a 5×12 board, with Blue as the starting player and was won by Red.

Red - rule	Blue
–	$(3,9)$
$(1,9)$ - 14	$(4,10)$
$(5,11)$ - 11	$(1,8)$
$(2,8)$ - 12	$(4,12)$
$(5,12)$ - 9	$(5,10)$
$(1,10)$ - 6	$(3,7)$
$(2,7)$ - 12	$(2,6)$
$(3,6)$ - 12	$(4,3)$
$(1,3)$ - 14	$(5,4)$
$(2,1)$ - 12	$(1,4)$
$(3,3)$ - 6	$(3,4)$
$(2,4)$ - 2	$(2,5)$
$(1,5)$ - 2	$(3,2)$
$(2,3)$ - 10	$(5,6)$
$(4,5)$ - 18	$(5,5)$
$(1,6)$ - 2	$(1,7)$
$(5,7)$ - 2	$(5,8)$
$(3,8)$ - 4	$(3,5)$
$(2,2)$ - 3	$(4,7)$
$(4,8)$ - 1	–

Appendix B

Cylindrical Hex Boards with Known Strategies

Table B.1 shows which $m \times n$ boards with $m, n \leq 10$ have a known optimal strategy.

	$m=1$	$m=2$	$m=3$	$m=4$	$m=5$	$m=6$	$m=7$	$m=8$	$m=9$	$m=10$
$n=1$	Any	Any, 4.2.1, **5.1.1**	Any, 4.2.2, **5.1.1**	Any, 4.2.1, **5.1.1**	Any, **5.1.1**, **8.4.1**	Any, 4.2.1, **5.1.1**	Any, **5.1.1**	Any, 4.2.1, **5.1.1**	Any, **5.1.1**	Any, 4.2.1, **5.1.1**
$n=2$	Any	4.2.1	4.2.2, **5.1.1**	4.2.1, **5.1.1**	**5.1.1**, **8.4.1**	4.2.1, **5.1.1**	**5.1.1**	4.2.1, **5.1.1**	**5.1.1**	4.2.1, **5.1.1**
$n=3$	Any	4.2.1	4.2.2	4.2.1, **5.1.1**	**5.1.1**, **8.4.1**	4.2.1, **5.1.1**	**5.1.1**	4.2.1	**5.1.1**	4.2.1
$n=4$	Any	4.2.1	4.2.2	4.2.1	**5.1.1**, **(8.4.1)***	4.2.1, **5.1.1**	**5.1.1**	4.2.1, **5.1.1**	**5.1.1**	4.2.1, **5.1.1**
$n=5$	Any	4.2.1	4.2.2	4.2.1		4.2.1, **5.1.1**	**5.1.1**	4.2.1, **5.1.1**	**5.1.1**	4.2.1, **5.1.1**
$n=6$	Any	4.2.1	4.2.2	4.2.1		4.2.1	**5.1.1**	4.2.1, **5.1.1**	**5.1.1**	4.2.1, **5.1.1**
$n=7$	Any	4.2.1	4.2.2	4.2.1		4.2.1		4.2.1, **5.1.1**	**5.1.1**	4.2.1, **5.1.1**
$n=8$	Any	4.2.1	4.2.2	4.2.1		4.2.1		4.2.1	**5.1.1**	4.2.1, **5.1.1**
$n=9$	Any	4.2.1	4.2.2	4.2.1		4.2.1		4.2.1		4.2.1, **5.1.1**
$n=10$	Any	4.2.1	4.2.2	4.2.1		4.2.1		4.2.1		4.2.1

Table B.1: An overview of which Cylindrical Hex boards currently have a known optimal strategy. If a known strategy exists, the corresponding algorithm is listed according to the numbering used in this thesis.
All new strategies introduced in this thesis are marked in bold. The strategy marked by (...)* was only shown to be optimal when Red is the starting player.

When $m, n = 1$, only one move can be made. This move is a winning move for either player, so the starting player is always the winner, regardless of colour.

When $m = 1$ and $n > 1$, any move on the board is still a winning move for Blue. As such, Blue is guaranteed to win, regardless of whether they begin or where make their move.

For $m = 2$ and $m = 3$, optimal strategies are known that work regardless of the value of n — namely 4.2.1 and 4.2.2 respectively.

For $m \geq 4$, a distinct pattern begins to appear, which also extends to the larger boards not shown in this table: Every other column — that is, every column where $m = 2k$ — is entirely coloured red due to the existence of Algorithm 4.2.1, and every cell in the 'top-right' of the table — that is, every cell where $m > n$ — is coloured red due to the existence of Algorithm 5.1.1.

Appendix C

The Solution to Piet Hein's First Polygon Problem

In Figure 1.1 of Chapter 1, we presented the very first Polygon problem published by Piet Hein in Poletiken on December 26th, 1942. The position shown in this problem was recreated here, in Figure C.1, using the same colour stones as those used in the original problem — black and white instead of red and blue. White can guarantee a win by making a single move. The question is simple: What move should White make?

The solution to this problem is shown in Figure C.2.

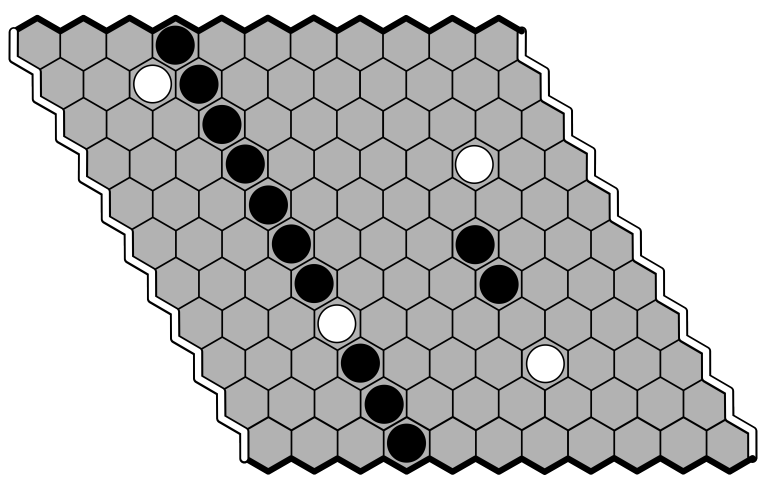


Figure C.1: A recreation of Piet Hein's first Polygon problem. From this position, White is able to guarantee a win in just one move. How?

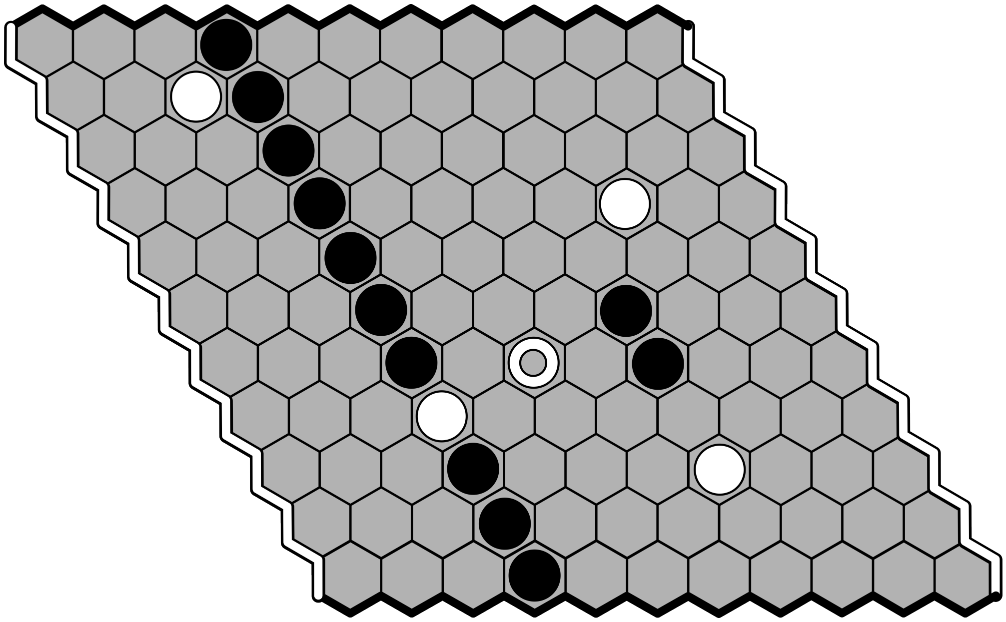


Figure C.2: The solution to Piet Hein's first Polygon problem: Play on $(6, 7)$!

One can check that, from the position in Figure C.1, White is always able to connect the white cell in $(4, 8)$ to the left side of the board. Likewise, White can always connect $(9, 4)$ and $(8, 9)$ to the right side of the board. In order to guarantee a win, White should thus make sure that they can always connect $(4, 8)$ to either $(9, 4)$ or $(8, 9)$.

By playing on $(6, 7)$, like in Figure C.2, White creates a bridge to $(4, 8)$. Regardless of Black's next move, White will be able to indirectly connect to at least one of $(9, 4)$ or $(8, 9)$. This thus guarantees a win for White if they continue to play optimally.