Fabiano N., Hayward R. (2020) New Hex Patterns for Fill and Prune. In: Cazenave T., van den Herik J., Saffidine A., Wu IC. (eds) Advances in Computer Games. ACG 2019. Lecture Notes in Computer Science, vol 12516. Springer, Cham. https://doi.org/10.1007/978-3-030-65883-0_7

New Hex Patterns for Fill and Prune

Nicolas Fabiano[1] and Ryan Hayward[2]

[1] Département d'Informatique, ENS Ulm
[2] Department of Computing Science, University of Alberta

Abstract. For a position in the game of Hex, a fill pattern is a subposition with one or more empty cells that can be filled without changing the position's minimax value. A cell is prunable if it can be ignored when searching for a winning move. We introduce two new kinds of Hex fill – mutual and near-dead – and some resulting fill patterns; we show four new permanently-inferior fill patterns; and we present three new prune results, based on strong-reversing, reversing, and game-history respectively. Experiments show these results slightly reducing solving time on 8×8 openings.

1 Introduction

Black and White alternate turns; on each turn, a player places one stone of their color on any unoccupied cell. The winner is the player who forms a chain of their stones connecting their two opposing board sides. See Fig. 1.

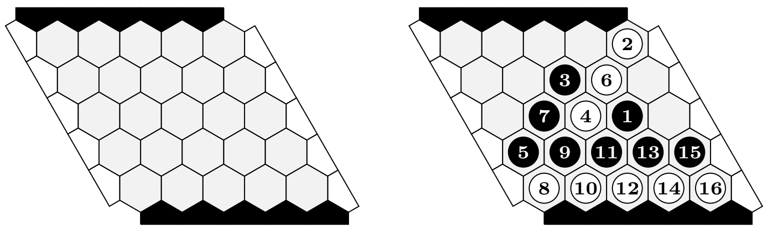

Fig. 1. An empty 5×5 Hex board and a game won by White.

Hex never ends in a draw [3] and it is never disadvantageous to move [14], so by strategy-stealing the first player has a winning strategy on an empty board [2]. But, for general sizes, no explicit first-player-wins strategy is known, nor is the location of a winning opening move. For boards up to 9×9 (resp. the 10×10 board) all (4) winning opening moves are known [12, 6].

Typical Hex solvers use tree search and these optimizations [1] : a) search promising moves first (which saves time when the move wins), e.g. with a neural network [4], b) prune provably redundant moves (which limits the growth of the tree size), c) detect wins early, e.g. by computing virtual connections [13, 8], d) use a transposition table to recognize positions already solved.

In this paper we present new fill and prune results. This is of interest from a combinatorial game theory (CGT) point of view and can speed up solving and playing. Fill replaces a position (whole-board pattern) with an equivalent one with fewer empty cells, which can help all optimizations, especially (b).

§2 gives definitions and results, §3 recalls known fill results, §4 presents new fill results, §5 presents new prune results, and §6 reports on experiments.

2 General analysis

$s(X)$ denotes a stone (or move) of player X in cell s. E.g., position C changes to $C + s(W)$ after White plays in s. $\overline{X}$ denotes the opponent of player X. The two players are usually B and W. The following definition – which we will apply to positions that differ by only a few stones – relies on the relation $\leq$ from CGT. We will only use it with positions that differ by just a few stones.

Definition 1 (A pre-order relation). *For two positions C, C', we say that $C \leq_X C'$ when for any player Y, if player X has a winning strategy in $C_{Y \text{ to play}}$, then she also has one in $C'_{Y \text{ to play}}$.*

For two patterns P, P', we say that $P \leq_X P'$ when, for every position C that contains P, if we call C' the position where it is replaced by P', then $C \leq_X C'$.

Note 2. Hex is a perfect-information deterministic no-draw 2-player game, so $\leq_W$ is exactly $\geq_B$, so $=_W$ is the same as $=_B$, which we write as $=$.

As with Def. 1, the following definitions and properties can be extended from positions to patterns: a pattern P has property A if, for every position C that contains P, C has property A.

Definition 3 (Dead cells and stones). *An empty cell s is* live *for player X when it is in a minimal (for inclusion) set of empty cells that, if X-colored, would join X's two sides. A cell is live for one player whenever it is live for the other [5]. A cell that is live for neither player is* dead*. A stone is* dead *when, if removed, the associated cell is dead.*

Theorem 4. *[5] If the empty cell s is dead, then $C = C + s(B) = C + s(W)$.*

In practice, dead cells can be detected either by seeing the board as a graph or by using patterns as those in Fig. 2.

Definition 5 (Fill pattern). *An empty cell s is X-*fillable *when $C = C + s(X)$.*

E.g., any dead cell is W-fillable and B-fillable. In practice, X-fillable cells are usually X-filled and then considered as stones.

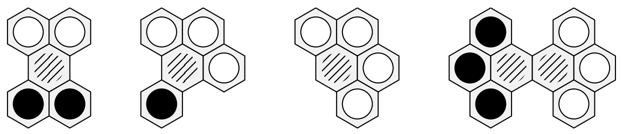

Fig. 2. Each dashed cell is dead. Any whitespace cell can be empty or colored.

Definition 6 (Inferior cells). *An empty cell s is X-*inferior *to another empty cell t when $C + s(X) \leq_X C + t(X)$.*

Definition 7 (Reversing a move). *A move $r(X)$ X-*reverses $s(\overline{X})$ *when $C + r(X) + s(\overline{X}) \geq_X C$.*

This CGT notion is useful: after $s(\overline{X})$, X should reply with $r(X)$. (Warning: if $s(\overline{X})$ is a blunder, X might have a better reply than $r(X)$.) In some (but not all) reverse patterns, $s(\overline{X})$ can be pruned, e.g. with vulnerable patterns [5]. In § 5.1 we give a condition sufficient to prune some reversible moves.

Definition 8 (X-iterativity). *A cell (resp. empty cell) s is X-*iteratively dead *(*Y-filled*) when we can iteratively X-fill some cells that leave s dead (Y-filled).*

Theorem 9. *If $s(\overline{X})$ is X-iteratively dead, then $C = C - s(\overline{X}) = C - s(\overline{X}) + s(X)$.*

Proof. Let T be the set of cells that is X-filled to kill s. Then, $C = C + T(X) = C - s(\overline{X}) + s(X) + T(X) \geq_X C - s(\overline{X}) + s(X) \geq_X C - s(\overline{X}) \geq_X C$. □

Theorem 10. *If s is X-iteratively X-filled (and so empty) then $C = C + s(X)$.*

Proof. Let T be the set of cells that is X-filled to fill s. Then, $C = C + T(X) = C + s(X) + T(X) \geq_X C + s(X) \geq_X C$. □

Some following results assume $C = C + s(X)$ as an hypothesis, so we can use iterative death/fill as sufficient conditions. This is also useful for pruning, in two ways:

- $C + s(X) = C + s(X) + t(X)$ implies that t is inferior to s for X, so provided that we explore s, or a move superior to s, we can prune t;
- $C + r(\overline{X}) + s(X) = C + r(\overline{X})$ implies that $s(X)$ is $\overline{X}$-reversible by $r(\overline{X})$.

3 Previous fill results

Dead and iteratively dead cells can be filled. Other fill results are known [7, 10]:

Definition 11 (X-captured cells). *A set of empty cells S is X-*captured *when X has a second-player strategy in S that guarantees every cell in S to be X-colored or dead.*

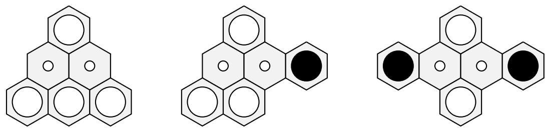

Fig. 3. Each white-dotted cell is W-captured.

Theorem 12. *If S is X-captured, then any subset of S can be X-filled.*

Proof. $C + S(X) \geq_X C$, and the strategy gives $C + S(X) \leq_X C$. □

In practice, captured cells are detected using patterns such as those in Fig. 3.

Definition 13 (X-permanently inferior cells). *Assume that there is an empty cell s and a set of empty cells T (with $s \notin T$) such that $\forall t \in T, P + t(\overline{X}) + s(X) = P + s(X) = P + t(X) + s(X)$. Then any cell $t \in T$ such that $P + t(X) + s(\overline{X}) = P + s(\overline{X})$ is said X-permanently inferior.*

Theorem 14. *If t is X-permanently inferior, then t can be X-filled.*

Proof. Let C be such a position and let $C' = C + t(X)$. In C (and in C'), each cell of T is X-inferior to s and $\overline{X}$-reversible by s, so we may assume that one player will play in s while T is still empty. But after this first move, regardless of who makes it, t can be filled. □

Before, only the three patterns shown in Fig. 4 were known.

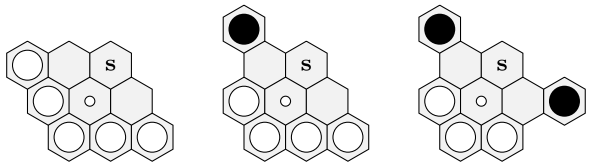

Fig. 4. Each white-dotted cell is W-permanently inferior.

4 New fill results

4.1 Near-death patterns

Definition 15 (Near-death pattern). *Pattern P with non-intersecting empty cell sets b and w and remaining empty cell set x is* near-death *when these hold:*

- *if b is B-colored and w is W-colored, then x is dead;*
- *if w is W-colored, then B has a second-player strategy in $b \cup x$ that leaves all cells of b black or dead;*

– *if b is B-colored, then W has a second-player strategy in $w \cup x$ that leaves all cells of w white or dead.*

We calls the cells of x nearly dead because, assuming optimal play, each player can save her side of the pattern and what remains will be dead:

Theorem 16. *For a near-death pattern P, $P = P + b(B) = P + w(W) = P + b(B) + w(W)$.*

Proof. First, we have $P + w(W) \geq_W P + b(B) + w(W) \geq_W P + b(B)$ and $P + w(W) \geq_W P \geq_W P + b(B)$, so it is sufficient to prove $P + b(B) \geq_W P + b(B) + w(W) \geq_W P + w(W)$, which by symmetry reduces to $P + b(B) \geq_W P + b(B) + w(W)$. Now following the white second-player strategy gives $P + b(B) \geq_W P + w(W) + b(B) + x(B) = P + w(W) + b(B)$. □

Note 17. In the hypothesis, one might be tempted to replace $b \cup x$ by $w \cup b \cup x$ (and to remove "if w is W-colored"). This is sufficient for $P = P + b(B) + w(W)$ but not $P = P + w(W)$, so monocolor-iterativity cannot be used for pruning.

Note 18. When $b = \emptyset$ the theorem implies that W-captured cells can be W-filled, which is not new. But some capture patterns found with $b = \emptyset$ are new.

Here is how to produce patterns from this theorem: choose a core death pattern; remove some stones; add a gadget (pattern of stones and empty cells) so that, for each cell that had a stone, that cell can be killed by playing in an empty cell; check whether the theorem applies. See Fig. 5, 6 and 7.

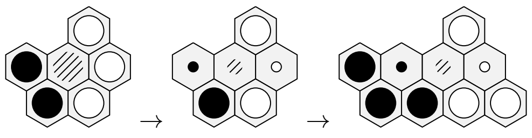

Fig. 5. Start with a pattern where the dashed cell is dead. Remove one stone for each player. On the final pattern, where two gadgets have been added, apply the theorem with b containing the black-dotted cell, w containing the white-dotted cell and x containing the dashed cell.

This method gives more than 100 new fill patterns of radius 3 (each pattern has up to $1 + 6 + 12 + 18 = 37$ cells).

4.2 Mutual-fill patterns

Theorem 19. *Let P be a pattern with empty cells S, and two disjoint subsets T, U of S. Denote $P' = P + T(B) + U(W)$ and $S' = S \backslash (T \cup U)$. Assume this:*

$$\exists b \in S, P + b(B) = P + S(B); \quad \exists w \in S, P + w(W) = P + S(W);$$
$$\exists b' \in S', P' + b'(B) = P + S(B); \quad \exists w' \in S', P' + w'(W) = P + S(W).$$

Then $P = P'$.

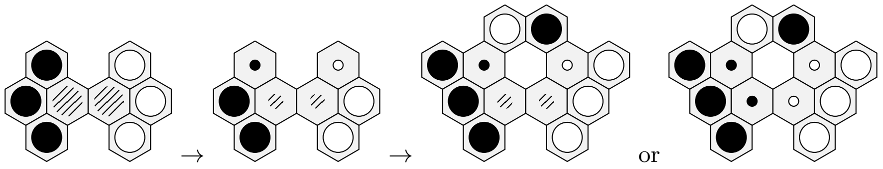

Fig. 6. A larger near-death pattern, constructed similarly. Here the theorem still applies if we move the left cell from x to b and the right one from x to w, giving a stronger result. This phenomenon often occurs when x has at least 2 cells.

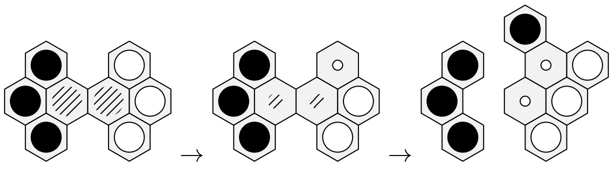

Fig. 7. A new W-capture pattern, constructed similarly

To prove the theorem, it suffices to use that in both P and P' whoever plays first in the pattern gets all of S. We omit the details.

Figure 8 shows the four mutual-fill patterns we found. The leftmost pattern applies on any empty board with at least two rows and columns.

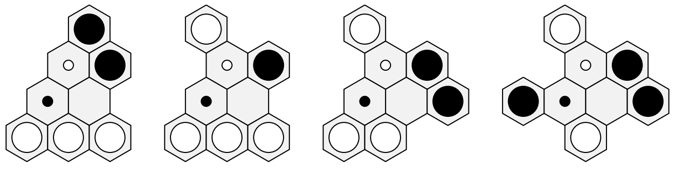

Fig. 8. In each pattern the dotted cells can be mutually filled.

Note 20. Warning: mutual-fill can change a move from losing to winning. E.g., apply the left-most pattern in Fig.8 to the $2{\times}2$ board: now each corner cell wins, whereas on the original empty board it is losing.

Note 21. This theorem fills cells for both players at the same time, so it cannot be used to derive strong-reversible or inferior cells via monocolor-iterativity.

4.3 New X-permanently inferior patterns

Along with the permanently inferior patterns of Fig. 4, we found the three new ones shown in Fig. 9. The leftmost occurs often on a border.

We also found a fourth more complicated pattern, shown in Fig. 10, which can occur near an acute corner of the board. We omit the proofs that these

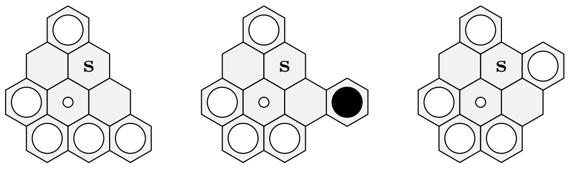

Fig. 9. Three new W-permanently inferior patterns.

patterns are permanently inferior. Combining the left patterns of Figs. 10 and 8 yields the right pattern of Fig. 10.

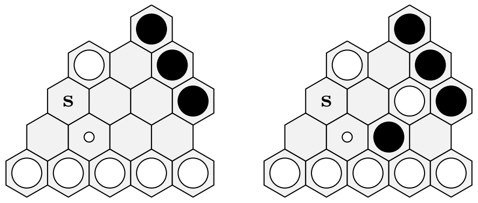

Fig. 10. Another new W-permanently inferior pattern, and a corollary.

5 New prune results

Previous prune results were based on mustplay (prune moves that let the opponent a simple way of making a winning connection) and monocolor-iterativity. This second approach is strengthened by the large number of new fill patterns. Other results were based on star decomposition [9] – some of which now apply via mutual-fill or near-death patterns – and on strategy-stealing arguments [1].

Prune results often have the form "if s wins, then t wins". Thus, we must take care to avoid logical implication cycles when pruning moves. Warning: some of our results here are based on game history, which requires extra consideration when implemented with a transposition table.

5.1 New theorem using reversibility

Here we strengthen a result from [9].

Definition 22 (X-strong-reversing). *A move $r(X)$ X*-strong-reverses *$s(\overline{X})$ when $C + r(X) + s(\overline{X}) = C + r(X)$.*

Definition 23 (Carrier of a X-strong-reverse pattern). *For a X-strong-reverse pattern P where r X-reverses a single cell s, we denote by F the set of the empty cells except r and s. This is the* carrier *of the pattern.*

Definition 24 (Blocking). *A X-strong-reverse pattern P_1* blocks *another one P_2 when r_1 is in F_2.*

In particular, some patterns (usually called "vulnerable") have an empty carrier, so they are never blocked.

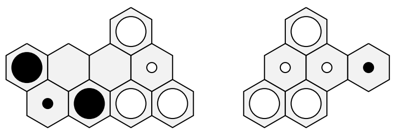

Fig. 11. (left) The black-dotted cell B-strong-reverses the white-dotted cell (carrier size 2). (right) The black-dotted cell B-strong-reverses each white-dotted cell (carrier size 1 each), and neither pattern blocks the other.

Theorem 25. *Let (P_n) be a sequence of (possibly overlapping) X-strong-reverse patterns of a position C. Assume that for all $k < n$, P_n does not block P_k, and that $\overline{X}$ has a winning move in C. Then $\overline{X}$ has a winning move that is not one of the s_n (or, if there is no other move, then skipping the next move wins).*

Proof. The special case where there is no other move is left as an exercise for the reader.

Argue by contradiction: assume that the theorem is false, and let $(P_n)_{1\leq n\leq N}$ be a counter-example with N minimum, i.e. the only winning moves (with $\overline{X}$ to play) are among the s_n (so $N \geq 1$)

By applying the result to $(P_n)_{1\leq n\leq N-1}$, we know that s_N wins, so $C + r_N(X) + s_N(\overline{X}) = C + r_N(X)$ (with $\overline{X}$ to play) wins for $\overline{X}$. In this new position, we can apply the result to (P_n), from which we remove the potential patterns for which $r_N = s_n$ (because $\overline{X}$ cannot play in s_n) or $r_N = r_n$ (but anyway in this case s_n is $\overline{X}$-fillable, so if s_n wins for $\overline{X}$ then any move or even skipping the next move wins) (this removes at least $n = N$), and we get that there is a winning move t that is not one of the s_n. This means that $C + r_N(X) + t(\overline{X})$ (with X to play) wins for $\overline{X}$, so t was a winning move for $\overline{X}$ in C, contradiction. □

5.2 New reverse patterns

Theorem 26. *Let P be a pattern with exactly 3 empty cells s, t, u such that, in $P + t(X) + u(X)$, s is dead, and $P + s(\overline{X}) + t(X) + u(\overline{X}) \geq_X P + s(\overline{X}) + t(\overline{X}) + u(X)$. Then $t(X)$ reverses $s(\overline{X})$.*

Proof. First, we prove that $P + s(\overline{X}) + t(X) \geq_X P + s(\overline{X}) + u(X)$. We denote $P_1 = P + s(\overline{X}) + u(X)$ and $P_2 = P + s(\overline{X}) + t(X)$. Let C_1 be a position containing P_1, and let C_2 be the position obtained from C_1 by replacing P_1 with P_2. Assume that C_1 wins for X with a strategy S. We will show that C_2 also wins for X.

To win in C_2, X follows S, pretending that there is P_1 instead of P_2 until one player moves in P_2. There are three cases:

- If neither player moves in P_2, then at the end of the game, after a X move, X would have a winning connection in C'_1 and thus in $C'_1 + t(\overline{X})$; and $C'_1 + t(\overline{X}) \leq_X C'_2 + u(\overline{X}) \leq_X C'_2$.
- If S tells X to play in t then X plays instead in u, and we get a position that S proves wins.
- If $\overline{X}$ plays in u, then S tells us that $C'_1 + t(\overline{X})$ wins for X, and given $C'_1 + t(\overline{X}) \leq_X C'_2 + u(\overline{X})$ we get a position at least as good as one that S proves wins.

Now we prove that $P \leq_X P_2$. Let C be a position containing P, and let C_2 be the position obtained from C by replacing P with P_2. Assume that C wins for X with a strategy S. We will show that C_2 also wins for X.

To win in C_2, X follows S, pretending that there is P instead of P_2, until one player moves in P_2. Again, there are three cases:

- If neither player moves in P_2, argue as in the same case above.
- If S tells X to play somewhere in P, then X plays in u instead. Then s is dead so we can consider s to be X-colored, so we get a position at least as good as one that S proves wins.
- If $\overline{X}$ plays in P_2, then this must be in u given this is the only empty cell. Then X pretends that $\overline{X}$ has played in s in P, and follows S accordingly. P_2 is now full, so $\overline{X}$ can not play there any more. The case where X also does not play there can be treated as previously. So we may assume that at some point S tells X to play in P, and we may assume that this is in t since $P + s(\overline{X}) + t(X) \geq_X P + s(\overline{X}) + u(X)$. But then S tells us that, after this move and $\overline{X}$'s reply in u, X wins. So the current position wins.

□

To produce patterns from this theorem, we combine one left-gadget (to make sure that W-coloring t and u kills s) with one right-gadget (to make sure that B-coloring s and t kills u, which in practice is the only interesting case) from Fig. 12. This yields 18 patterns, two of which are shown in Fig. 13.

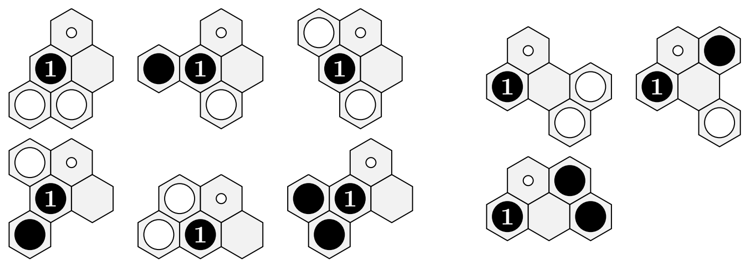

Fig. 12. The 1 (resp. dotted) (resp. empty) cell coresponds to s (t) (u). (left) The six left-gadgets. (right) The three right-gadgets.

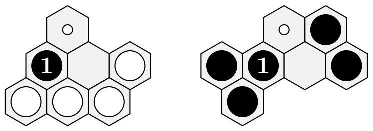

Fig. 13. In each pattern, the white-dotted cell W-reverses 1.

Note 27. Unlike strong-reverse moves, which are mostly pruned (see Thm 25), a reversible move be good, or even a unique winning move. Often, when $s(\overline{X})$ is really good, $\overline{X}$ should move in $u(\overline{X})$ immediately after $t(X)$. We have not formalized this observation (in particular the definition of "really good").

5.3 Self-reversibility

Observation 28. *Let s,t,u be empty cells such that* $C + s(X) + t(\overline{X}) + u(X) = C + t(\overline{X}) + u(X)$. *Then* $C + u(X) \geq_X C + s(X) + t(\overline{X}) + u(X)$.

In practice, this means that the move $u(X)$ after the moves $s(X), t(\overline{X})$ can be pruned (up to logical implication cycles), because X should have played in u instead of s. E.g., this can happen in the games in Fig. 14.

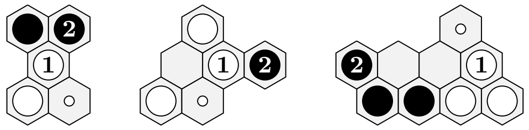

Fig. 14. In each game, after moves 1 and 2, White should not play in the white-dotted cell.

The conditions of this theorem are easily detected: for a possible move, can you prove (e.g. by monocolor-iterativity) that it would make your previous move useless? If yes, then you should move elsewhere.

6 Experiments

We ran some experiments using the Benzene software package (maintained by Chao Gao at `https://github.com/cgao3/benzene-vanilla-cmake`) that includes the player MoHex and a parallel depth-first proof number (DFPN) search Hex solver [11, 12].

We implemented near-death and new permanently inferior patterns. These are simply fill patterns, so this was straightforward. We did not implement mutual fill, due to technical issues mentioned in Note 20.

We also implemented applications of Theorem 25 (in greedy fashion) and the new reversible patterns. We did not implement self-reversibility: we suspect that the cost of checking for logical implication cycles would yield negligible improvement.

We compared different implementations by solving a fixed set of 1-move Black openings on the 8×8 board. The results are shown in Table 15.

	none	Nd	Pi	F	F+St	F+St+Rp	F+St+Rp'	F+Rp'
a1	103432	93774	96544	93416	93398	95287	88844	88838
b2	341130	356745	316543	320501	327763	384242	316390	316695
c3	195830	193217	190741	195419	195423	240345	202943	202936
Σ	640392	643736	603828	609336	616584	719874	608177	608469
d4[1]	155659	200313	372527	198621	198649	184876	186944	361603
Σ	796051	844049	976355	807957	815233	904750	795121	970072

Fig. 15. Number of nodes created, depending on features enabled. Openings a1 and b2 lose, while c3 and d4 win. Nd: near-death patterns and derived strong-reversible and inferior patterns; Pi: new permanently inferior patterns and derived patterns; F: Nd+Pi; St: strong-reversible theorem, replacing theorem from [9]; Rp: reversible patterns, reverser always picked; Rp': reversible patterns where, for OR (resp. AND) node n, reverser r is picked only when r's disproof (proof) number is less than or equal to the minimum proof (disproof) number among n's children.

Our experiments show that the benefit of the new patterns is in most cases positive. Using St is less beneficial. These features did not help as consistently as we expected, perhaps in part because a small change in fill or prune can cause DFPN search to move to a different part of the tree.

Feature Rp' is always better than Rp, perhaps because the presence of a reverser is positively correlated with the presence of a move more easily proved to be winning. Feature Rp' appears globally positive.

Overall, our preferred setting is to enable all new features.

Note 29. Each new pattern requires more pattern matching and so slows node creation. There are few reversible patterns, so they slow runtime negligibly. But the new fill patterns yield thousands of inferior and strong-reversible patterns. Thus, the previous tests were run with only with patterns as large as in Fig. 5. E.g., we excluded the pattern in Fig. 6. The slowdown almost exactly compensates for the reduction in size of the resulting search tree, but – given that the slowdown is a constant factor while pruning can yield exponentially smaller trees – we expect that for larger boards using the new patterns will be beneficial.

[1] For opening d4, after White's strongest reply d5, Black has two main winning moves: g4 and c5. The latter has a smaller search tree, with ~ 65000 nodes instead of ~ 240000. This explains the differences for this opening, depending on the initial choices. One cannot fairly compare features that differ on this g4-c5 choice.

Acknowledgements

We thank Chao Gao for many helpful comments and suggestions.

References

1. Broderick Arneson, Ryan B. Hayward, and Philip Henderson. Solving Hex: Beyond Humans. In H.J̃aap van den Herik, Hiroyuki Iida, and Aske Plaat, editors, *Computers and Games 2010 Revised Selected Papers*, volume 6515 of *Lecture Notes in Computer Science*, pages 1–10. Springer, 2011.
2. Anatole Beck, Michael N. Bleicher, and Donald W. Crowe. *Excursions into Mathematics.* Worth, New York, 1969.
3. David Gale. Game of Hex and the Brouwer Fixed Point Theorem. *American Math Monthly*, 86(10):818–827, 1979.
4. Chao Gao, Martin Müller, and Ryan Hayward. Focused depth-first proof number search using convolutional neural networks for the game of hex. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 3668–3674, 2017.
5. Ryan Hayward and Jack van Rijswijck. Hex and Combinatorics. *Discrete Math*, 306(19-20):2515–2528, 2006.
6. Ryan B. Hayward and Bjarne Toft. *Hex, the full story.* CRC Press, London, 2019.
7. Philip Henderson. *Playing and Solving Hex.* PhD thesis, University of Alberta, Edmonton, Alberta, Canada, Fall 2010. `https://webdocs.cs.ualberta.ca/~hayward/theses/ph.pdf`.
8. Philip Henderson, Broderick Arneson, and Ryan B. Hayward. Hex, braids, the crossing rule, and xh-search. In *Advances in Computer Games, 12th International Conference, ACG 2009, Pamplona, Spain, May 11-13, 2009. Revised Papers*, pages 88–98, 2009.
9. Philip Henderson and Ryan B. Hayward. Captured-reversible moves and star-decomposition domination in Hex. *Integers*, 13(CG1):1–15, January 2013.
10. Philip Henderson and Ryan B. Hayward. *A Handicap Strategy for Hex*, pages 129–136. MSRI and Cambridge University Press, 2015.
11. Shih-Chieh Huang, Broderick Arneson, Ryan B. Hayward, Martin Müller, and Jakub Pawlewicz. Mohex 2.0: A pattern-based MCTS hex player. In *Computers and Games - 8th International Conference, CG 2013, Yokohama, Japan, August 13-15, 2013, Revised Selected Papers*, pages 60–71, 2013.
12. Jakub Pawlewicz and Ryan B. Hayward. Scalable Parallel DFPN Search. In *Computers and Games 2013 Revised Selected Papers*, volume 8427 of *Lecture Notes in Computer Science*, pages 138–150. Springer, 2014.
13. Jakub Pawlewicz, Ryan B. Hayward, Philip Henderson, and Broderick Arneson. Stronger Virtual Connections in Hex. *IEEE Transactions on Computation Intelligence and AI in Games*, 7(2):156–166, June 2015.
14. Y. Yamasaki. Theory of Division Games. *Pub. Res. Inst. Math. Sciences*, 14:337–358, 1978.