Charles University in Prague
Faculty of Mathematics and Physics

MASTER THESIS

David Slabý

Competitive filling of a plane region

Department of Applied Mathematics

Supervisor of the master thesis: doc. RNDr. Pavel Valtr, Dr.
Study programme: Computer Science
Specialization: Discrete Models and Algorithms

Prague 2012

I am thankful to my supervisor doc. RNDr. Pavel Valtr, Dr. for his invaluable advices and comments and to students of Seminar on Combinatorial Problems in 2011 for introduction to the problem and cooperation during research, especially to Martin Böhm who also revised the whole thesis.

I declare that I carried out this master thesis independently, and only with the cited sources, literature and other professional sources.

I understand that my work relates to the rights and obligations under the Act No. 121/2000 Coll., the Copyright Act, as amended, in particular the fact that the Charles University in Prague has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 paragraph 1 of the Copyright Act.

In Prague, April 11, 2012 David Slabý

Název práce: Soutěživé vyplňování rovinné oblasti

Autor: David Slabý

Katedra: Katedra aplikované matematiky

Vedoucí diplomové práce: doc. RNDr. Pavel Valtr Dr., Katedra aplikované matematiky

Abstrakt: Dva hráči se střídají v umisťování jednotkových čtverečků na obdélníkovou hrací plochu, bez otáčení, jinak mohou být umístěny libovolně. Čtverečky se nesmí překrývat a hra končí, když už se nedá umístit další.

Výsledkem hry je počet tahů. Konstruktor se snaží tento výsledek maximalizovat a destruktor minimalizovat. Cílem této práce je co nejpřesněji určit výsledek hry za předpokladu, že oba hráči použijí optimální strategii.

Zde dokážeme nové odhady výsledku hry. Tato práce rozšiřuje výsledky popsané v článku Competitive rectangle filling, jehož autorem je Tamás Hubai.

Dále se zabýváme jinými tvary hracích ploch a pokládaných tvarů.

Klíčová slova: geometrická hra, čtverec, obdélník, konstruktor, destruktor

Title: Competitive filling of a plane region

Author: David Slabý

Department: Department of Applied Mathematics

Supervisor: doc. RNDr. Pavel Valtr Dr., Department of Applied Mathematics

Abstract: Two players take alternating turns filling a rectangular board with unit squares without rotation, but may be otherwise arbitrary. Squares may not overlap and the game ends when there is no space for the next one.

The result of the game is the number of turns. The constructor aims to maximize this quantity while the destructor wants to minimize it. We would like to get close to this value, provided that both players use their optimal strategy.

We prove some new lower and upper bound for the game. This thesis extends results given by Tamás Hubai in his paper Competitive rectangle filling.

Furthermore, we have a look at other board shapes and shapes to fill with.

Keywords: geometric game, square, rectangle, constructor, destructor

Contents

1. Introduction

1.1 Packing

Problems concerning packing (fitting as many objects into a container as possible) are widely studied in many variants and dimensions (see [2], [3]). First known contributions are applications of Lagrange's research [4] by Gauss [5]. Some great contributions were made by Minkowski (e.g. [6]) as he thoroughly studied properties of planar bodies. One of the most comprehensive monographs on the topic is *Covering and Packing* by Rogers [7], concerning mainly packing (and covering) in three or more dimensions. Study of packing of unit squares into a large rectangle also yielded a lot of interesting results: Erdős had shown in [8] that, for a large container, it is possible to have less than linear amount of unused area by rotating some squares, which is a rather surprising result. Some special cases are computed exactly in [9].

If we disallow any rotation, the solution is straightforward, but only as long as we do not invite a player who has an exactly opposite goal to our game.

1.2 The game

Throughout this work we analyze the following game, first presented by Tamás Hubai in [1]: we have a large rectangular board and two players, the *constructor* and the *destructor*, who alternate in taking turns. In each turn, a player must place a unit square anywhere onto the board, with its edges parallel to the board edges. These moves must not overlap. Once there is no place to put a unit square, the game ends. Constructor's goal is to maximize the number of moves, while the destructor minimizes it. Our aim is to achieve some good bounds on the number of moves if both players use their optimal strategy. We call this number the *value* of the game.

There is a number of two-player games, like Avoider-Enforcer games, where one player (the enforcer) tries to form a structure (say, a clique of given size in a graph) and the other player (the avoider) tries to avoid it. One of the most comprehensive books concerning combinatorial games is the famous monograph *Winning Ways for your Mathematical Plays* [11]. However, our game is a special one — players merely compete in minimizing or maximizing a value and it is not even defined who the winner is; actually, we would first need to know the value of the game to define a fair threshold for winning. But since there are two players with opposite goals, it seems correct to call it a game.

In [1], a few more variants of the game are discussed. The game evolved from its one-dimensional variant, where players pick non-overlapping unit line segments of a large line segment, one of them maximizing and the other minimizing total number of moves. A discrete variant, called Game of Kayles [12], allows players to pick one stone or two neighboring stones in a row of stones until there are none left. The interesting outcome is that both games converge to $\frac{3}{4}M$, M being the length of line segment or number of stones.

The analogous discrete variant of our game, called Two-dimensional Kayles, has the following rules: we have a board consisting of unit square tiles and each turn, a player may occupy one free tile, two neighboring free tiles or four free tiles forming a 2×2 square. The game ends once all tiles are occupied. The constructor wants to maximize the number of turns and the destructor wants the opposite. A neat proof is given showing that for a board with height and width even the number of moves is $\frac{9}{16}M$, M being the number of tiles.

1.3 Presented results

In Chapter 2 a new lower bound on the value of the game is presented, improving the previous bound presented in [1] considerably from $\frac{15}{32}M$ to $\frac{33}{64}M$, M being the maximal number of moves. The proof reduces the game to a discrete variant and using several observations and a linear program a new value is acquired. A new upper bound is shown in Chapter 3, reducing the bound given by Tamás Hubai from $\frac{3}{5}M$ to conjectured $\frac{9}{16}M$. The original proof contained an incorrect assumption, but the extended proof presented here avoids the mistake. The improvement is made by a thorough analysis of constructor's moves that jeopardize destructor's approach the most.

Chapter 4 generalizes these results by showing that if playing on a polygonal board inflated enough, the value of the game on such board has asymptotically the same bounds. Chapter 5 shows some basic approach in case of filling with polygons satisfying a specific, yet common property.

1.4 Playing on a small board

As a preface to the analysis of the game on an arbitrarily large board, let us try playing on one of size 4×4. We divide it into 16 unit squares — tiles. By trying several reasonable beginning moves, we conclude that we will not achieve better result than 9. Be aware that this chapter is a mere motivation and the following case analysis is not exhaustive.

First, we try beginning by a move in the center of our board. Now the destructor may fill the rest of the board in four more moves, but the constructor is able to disable two of them, prolonging the game to nine moves, no matter who made the first move. See Figure 1.1 — dark square refers to the beginning move, medium squares to the following moves and light ones are the rest — those only pick remaining empty spaces.

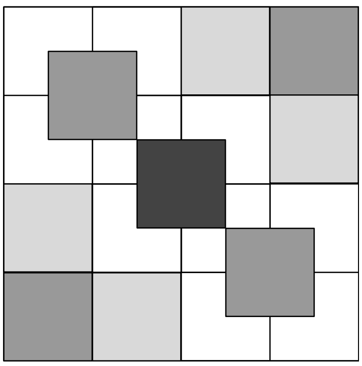

Figure 1.1: Beginning with a move to the center.

Another option is to solve each quadrant separately. The destructor would cover one in one move, while constructor's choice would be to cover a single tile out of four, e.g. the one closest to the center. When all quadrants are played into, there are six free tiles, which will be covered in five moves. See Figure 1.2.

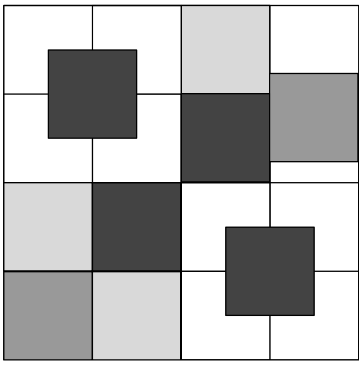

Figure 1.2: Beginning with a move to a quadrant.

The last presented option is a hybrid one — let us try a move on the boundary of two quadrants. The destructor covers four tiles, two from one quadrant and two from another, the constructor may want to take two, each from one quadrant. Now there are two moves that can cover three tiles at once, one will be used by the destructor, one disabled by the constructor, leaving us again with six free tiles coverable in five moves. See Figure 1.3.

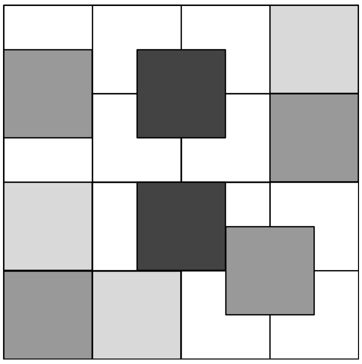

Figure 1.3: Beginning with a move on a boundary of two quadrants.

Although this mere guessing is far from a proof, it corresponds to the hypothesis that value of the general game is $\frac{9}{16}M$, M being total number of tiles. This suggests it may be reasonable to use these 4×4 squares to plot strategies for both constructor and destructor.

2. Strategy for the constructor

Dividing the board into 4×4 squares and analyzing them separately seems to be a promising approach. Obviously, this cannot be done in a straightforward fashion, for any player may make a move on the boundary of two such squares, affecting them both.

In this chapter we improve the original approach: we reduce the game to a discrete one in the same way and then take a closer look at possible moves in several phases of the game.

Let us start with several definitions.

Definition. *A tile is a unit square whose vertices' coordinates are integers.*

Definition. *A tile is* even *if its right bottom vertex coordinates are both even. If they are both odd, the tile is* odd*. Otherwise, if the parity differs, the tile is* general*.*

Definition. *A block is a 2×2 square whose left top vertex coordinates are even integers.*

Definition. *A field is a 4×4 square whose left top vertex coordinates are integers divisible by four.*

A block consists of four tiles. A field consists of four blocks. Any two tiles (blocks, fields) share no surface, at most an edge. In cases where we need them to be disjoint, suppose their right and bottom edges are not part of them.

Definition. *A move is a unit square placed on the board by a player; it must not overlap any other move (but may share a part of an edge).*

Definition. *A move belongs to a block (field), if move's left top vertex is in it.*

Definition. *An* active *tile is not covered by a move. Conversely, a* passive *tile is at least partially covered by a move. A passive tile is* obstructed*, if it is fully covered by a single move. To* inactivate *a point means to change it from active to passive.*

Observation. *It is always possible to obstruct any active tile by a move.*

2.1 Constructor's approach

Suppose we have a board $m \times n$, an integer $M = m \cdot n$ is total number of tiles and both m and n are divisible by 4. The strategy will divide the gameplay into four phases.

In the *first phase*, the constructor will obstruct even tiles in the following manner: whenever the destructor makes a move, it belongs to a block of a field. If the even tile in the block located diagonally (with respect to the field) is active, the constructor obstructs it (see Figure 2.2). If it is passive already, he obstructs any active even tile. As the outcome of this approach we have at least two obstructed even tiles in each field and their blocks are neighboring.

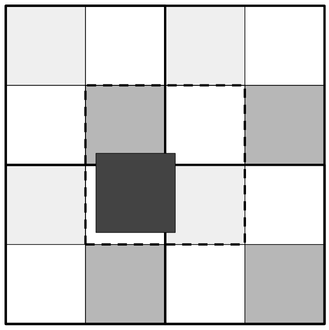

Figure 2.1: A field, blocks, tiles and a move. Grey tiles are even, light grey are odd, white are general, the dark move belongs to the left top block and makes tiles surrounded by the dashed border passive.

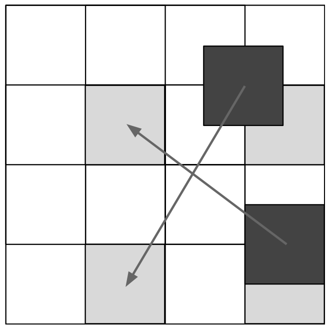

Figure 2.2: Constructor's responses to destructor's moves in first phase.

Definition. *D-move is a move that does not inactivate an even tile and inactivates two general tiles.*

Until all even tiles were passive, the destructor may have inactivated four tiles per turn. In the *second phase*, the constructor will focus on disabling D-moves — these disable two general moves. Any move inactivating three tiles at once in this phase is a D-move. The amount of these moves will be discussed in following section. In this phase, the constructor will disallow those D-moves by obstructing suitable general tiles.

Once all D-moves are wiped out, at most two tiles might be inactivated at once, one of them being an odd one. In this *third phase*, the constructor blocks these odd tiles. The last, *fourth phase* consists of mere deactivating of the rest of general tiles, one by one.

Once all the tiles are passive, it may be possible that there is still a space for a move, however, since by a strategy for the constructor we are trying to achieve a lower bound on the value of the game, we may suppose there are no more permitted moves.

2.2 Counting D-moves

Bounding the amount of D-moves is crucial to achieving a nice result. We find out there may be four times less than expected in the original proof.

Lemma 1. *There are at most two D-moves available per field.*

Proof. Since every D-move must partially cover four tiles, it needs a passive even tile that is not obstructed. If it is not obstructed, it must have a passive neighboring (general) tile. Therefore, for each such even tile there is at most one diagonal move, and there are at most two such even tiles per field. ☐

Definition. *V-move is a move that inactivates an even tile along with two general tiles.*

Claim 1. *If the constructor made two V-moves to a field, there is at most one D-move available in it.*

Proof. The two even tiles inactivated by destructor's V-moves belong to neighboring blocks. If the general tile between them is passive, one of these even tiles has three neighboring general tiles passive and cannot have a D-move. If the general tile between them is active, both diagonal moves need it and only one can use it (see Figure 2.3). ☐

Since every non-obstructed passive tile has a passive neighbor, at the beginning of second phase for each diagonal move (or overlapping tuple of diagonal moves) we may pinpoint a general tile contained in it.

Definition. *D-tile is a general tile contained in a D-move. If two D-moves of a field overlap, the tile is in their intersection.*

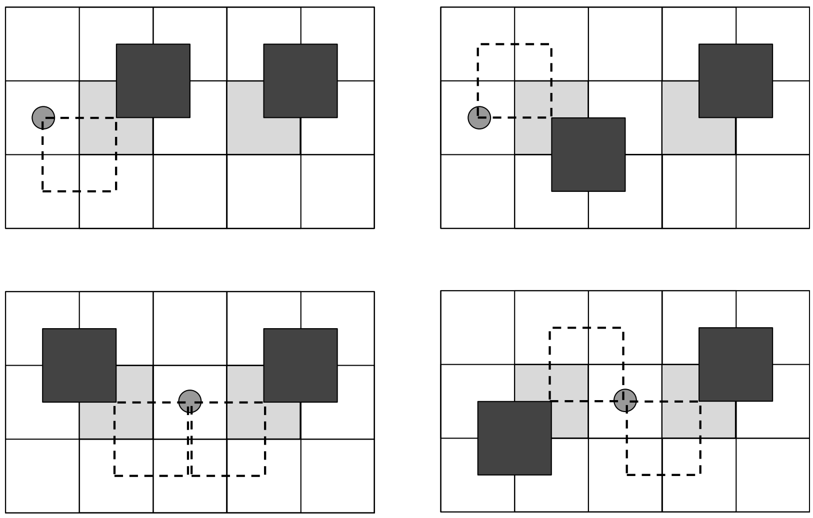

Figure 2.3: Four possible configurations of two neighboring V-moves with D-moves denoted by dashed squares and D-tile denoted by a bullet.

2.3 Competitive game of diminishing values

Instead of analyzing each of fields and their interactions, we define several values, and, using these, describe possible destructor's moves and moves allowed to the constructor.

- V is an upper bound on the number of fields without a V-move belonging to it.

- E is the number of active even tiles.
- O is the number of active odd tiles.
- G is the number of active general tiles.
- D is an upper bound on the number of D-tiles.

Using these values, we will remake the original game in the following manner: we describe moves by listing tiles that are inactivated by them; this corresponds to values E, O, G, D. If it is the first destructor's V-move in a field, also value V is used.

The constructor and the destructor choose an allowed set of values identifiers and decrease these values accordingly. The move is forbidden if a value would drop below zero. The game ends when E, O and G are zero (at that point no more moves are permitted anyway), the total number of these moves is the value of the game.

We recognize the following moves, denoted by their diminished values:

- $[VEOGG]$ — first V-move to a field, 4-move.
- $[DEOGG]$ — second V-move to a field, 4-move. If such a move is made, one of two D-tiles of the field is sacrificed.
- $[EOG]$ — unlike the previous move, both D-tiles of a field may be maintained if a move overlaps the previous 4-move in one general tile.
- $[EO]$ — an unlikely, yet allowed move inactivating non-general tiles.
- $[VEGG]$ — first V-move to a field, 3-move.
- $[DEGG]$ — second V-move to a field, 3-move. A D-tile is inactivated.
- $[DOGG]$ — using D-tile, a 3-move may be made.
- $[DGG]$ — another use of a D-tile is a purely diagonal move.
- $[EG]$ — 2-move with even tile.
- $[OG]$ — typical 2-move.
- $[G]$ — typical 1-move.
- $[E]$ — blocking an even tile.
- $[O]$ — blocking an odd tile.
- $[DG]$ — blocking a D-tile.

Since $[G]$, $[E]$ and $[O]$ are possible moves, the game always ends.

Claim 2. *The value of this game is less or equal to the original game.*

Proof. We suppose that the only moves allowed to the constructor are the last four, since we may say for sure that any active tile can be obstructed. However, constructor's strategy uses exactly these. On the other hand, we allow the destructor to make even some obviously impossible moves (like beginning the game with a diagonal move). Since all the possible sequences of destructor's moves are available, the approach is correct. ☐

2.4 Computing the best destructor's response

Let us define four game phases according to constructor's moves. Let F (with corresponding indices) denote the amount of moves made by the constructor: F_E for even tiles, F_D for D-tiles, F_O for odd tiles and F_G for general tiles. Every phase has $2 \cdot F_{type}$ moves and thus the value of the game is $2(F_E + F_D + F_O + F_G)$.

By setting several constraints, we achieve a lower bound on the value of the game using a linear program. Each X_{type} denotes the amount of moves of given type made by the destructor. B_{type} are the initial values of V, D, E, O, G.

The exact input of our linear program follows.

$$
\begin{aligned}
\text{minimize } & 2(F_E + F_D + F_O + F_G) \\
\text{subject to } & F_* \geq 0 \\
& X_* \geq 0 \\
& \sum_{\tau : E \in \tau} X_\tau \leq F_E \\
& \sum_{\tau : E \in \tau \vee D \in \tau} X_\tau \leq F_E + F_D \\
& \sum_{\tau : E \in \tau \vee D \in \tau \vee O \in \tau} X_\tau \leq F_E + F_D + F_O \\
& \sum_{\tau} X_\tau \leq F_E + F_D + F_O + F_G \\
& \text{– bounds on destructor's moves in phases} \\
& F_E = B_E - \sum_{\tau : E \in \tau} X_\tau \\
& F_D = B_D - \sum_{\tau : D \in \tau} X_\tau \\
& F_O = B_O - \sum_{\tau : O \in \tau} X_\tau \\
& F_G = B_G - \sum_{\tau : G \in \tau} X_\tau - \sum_{\tau : GG \in \tau} X_\tau - F_D \\
& \text{– length of each phase} \\
& \sum_{\tau : V \in \tau} X_\tau \leq B_V \\
& \text{– limitation of first V-move in each field} \\
\text{with } & B_V = \frac{M}{16} \\
& B_D = \frac{M}{8} \\
& B_E = \frac{M}{4} \\
& B_O = \frac{M}{4} \\
& B_G = \frac{M}{2}
\end{aligned}
$$

Theorem 1. *The value of the game is at least* $\frac{33}{64}M$*.*

Proof. The above program yields, using `lp_solve` to compute the value, following results (zero values are omitted):

- $F_E = \frac{1}{8}M$
- $F_D = \frac{1}{32}M$
- $F_O = \frac{3}{64}M$
- $F_G = \frac{7}{128}M$
- $X_{VEOGG} = \frac{1}{16}M$
- $X_{DEOGG} = \frac{1}{16}M$
- $X_{DOGG} = \frac{1}{32}M$
- $X_{OG} = \frac{3}{64}M$
- $X_G = \frac{7}{128}M$

Value of objective function: $\frac{33}{64}M$.
As we have shown before, this value is a lower bound on the value of the game.
For exact input of `lp_solve` see Appendix A. ☐

According to these results, the best destructor's response to the current constructor's strategy is to make as many 4-moves as possible, then continues with 3-moves using D-tiles (there is at most one D-tile in a field) and then makes 2-moves while possible and finishes up with 1-moves.

3. Strategy for the destructor

The destructor's strategy outlined in this chapter is largely inspired by the strategy of the previous paper, but it had to be remastered, since the argument of converting the game to the discrete variant with coordinates having specific rational parts cannot be used in the presented manner. For instance, the required covering of four points with a move may not be possible — since the conversion proceeds by taking algebraic mean of previous coordinates, we cannot enforce the specific position to cover the quadruple of points. However, with caution, we may adapt the proof to the original, continuous version of the game. It will yield that the value of the game is at most $\frac{3}{5}M$.

Let us have a grid of points corresponding to tiles: take center of each tile and then shrink the grid with aspect ratio $1-\varepsilon$ with $\varepsilon < \frac{1}{max\{m,n\}}$ towards the game board center. We still have each point inside its original tile, but the distance between adjacent points is less than 1. Again, we begin with a few definitions we will need throughout the chapter.

Definition. *By point we will always mean a point of the* $1-\varepsilon$ *grid.*

Observation. *Every move covers at least one point.*

Definition. *We say a point is active if it is possible to make a move covering this one and no other point. A point is passive if no move that covers it may be made.*

If a point is neither active nor passive, it must be a dependent one:

Definition. *A dependent point is a non-passive point that can not be covered separately, i.e. every move covering it covers another active or dependent point. A dependent point* A *is dependent on a point* B *if every move covering* A *covers* B *as well. If* B *is dependent on* A *at the same time, we say these points are mutually dependent. If* B *or* C *is covered when* A *gets covered,* A *is loosely dependent on* B *and* C*. See Figure 3.1.*

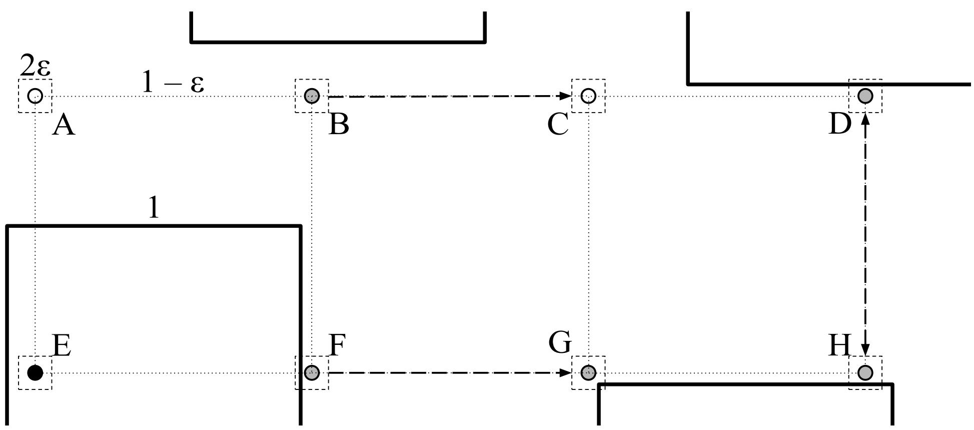

Figure 3.1: Points of grid, active (white), dependent (grey) and passive (black). Point yard (dashed square). Note that dependent B has its yard empty, G is loosely dependent on C and F, D and H are mutually dependent.

Observation. *If a point A is dependent on B and B becomes passive, A becomes passive as well.*

Definition. *A point's* yard *is its ε-neighborhood in uniform metrics (a square with sides of length 2ε and the point in the center).*

Observation. *If a point is active, its yard is empty (no move covers any part of it).*

Note that a dependent point may have an empty yard as well (see Figure 3.1, point B).

Observation. *The game ends when all points are passive.*

3.1 Clusters and their types

Definition. *A* cluster *is a set of four points whose tiles form a block.*

Definition. *A* move into a cluster *is a move that covers a point of the cluster.*

Observation. *Each cluster is a set of vertices of a square with side of length $1 - \varepsilon$.*

Let us define five types of clusters. Analysis of these types and possible transitions between them is the core of the proof.

Definition. *Cluster of type 4 has all its points coverable with a single move. Clusters of type i for $i = 0, 1, 2, 3$ are clusters not of type 4 with at most i possible further moves into it.*

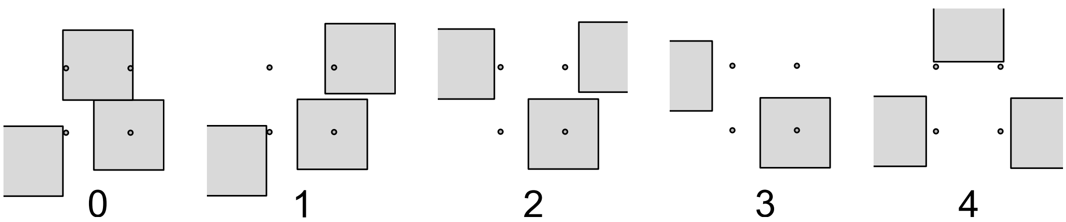

Figure 3.2: Types of clusters. Observe there is no trivial relation between number of active or dependent points and type of cluster. Also see that type 4 does not imply four more possible moves to the cluster.

To make sure each cluster matches exactly one type, we need to prove the following:

Lemma 2. *If a cluster is not of type 4, at most three moves can be made into it.*

Proof. Suppose we had a cluster that cannot be covered by a single move. No point is covered, but a yard of a point of the cluster is not empty and the point is (loosely) dependent on another point(s) of the cluster. It means at most three more moves can be made. $\square$

Observation. *At the beginning, all clusters are of type 4. Throughout the game the type of a cluster may only decrease.*

We recognize three types of moves:

Definition. *A* 4*-move touches all four yards of points in a cluster. A* 2*-move touches exactly two yards of points in a cluster. A* 1*-move touches only one yard of a point in a cluster.*

Lemma 3. *A* 4*-move may change type of eight neighboring clusters, but not if these are of type* 4*.*

Proof. To change type of a cluster, a move has to be made in a distance (by uniform metrics) less than 1 to any cluster's point, therefore only the eight neighboring clusters may be affected. To change type of a type 4 cluster, the move has to touch a yard of any of its points but a 4-move can not touch a yard of any other cluster. See Figure 3.3. ☐

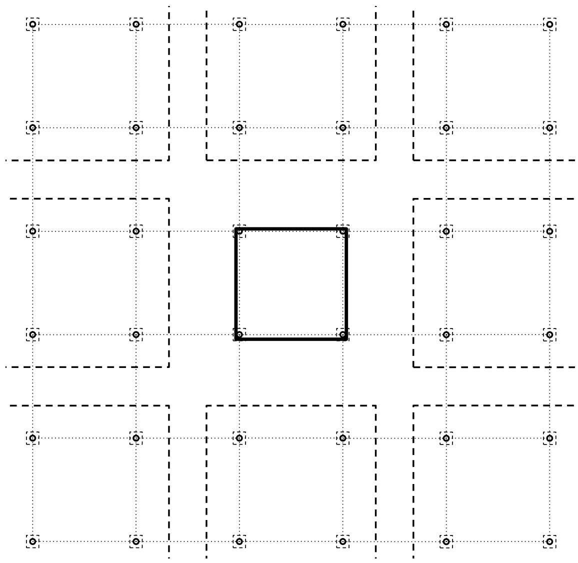

Figure 3.3: 4-move with surrounding clusters that cannot be affected if of type 4.

Lemma 4. *A* 2*-move may change type of one neighboring type* 4 *cluster. It may change type of up to five neighboring clusters if these are of type* 4*.*

Proof. Only five neighboring clusters have their points in the distance less than 1 to a 2-move. A 2-move can touch yards of at most one other cluster. See Figure 3.4. ☐

Lemma 5. *A* 1*-move may change type of up to three neighboring clusters.*

Proof. Only three neighboring clusters have their points in the distance less than 1 to a 1-move. A 1-move can touch yards of these three neighboring clusters. See Figure 3.4. ☐

We are now able to claim the following:

Claim 3. *A move may affect type of at most three more clusters of type* 4 *and at most eight more clusters of other type.*

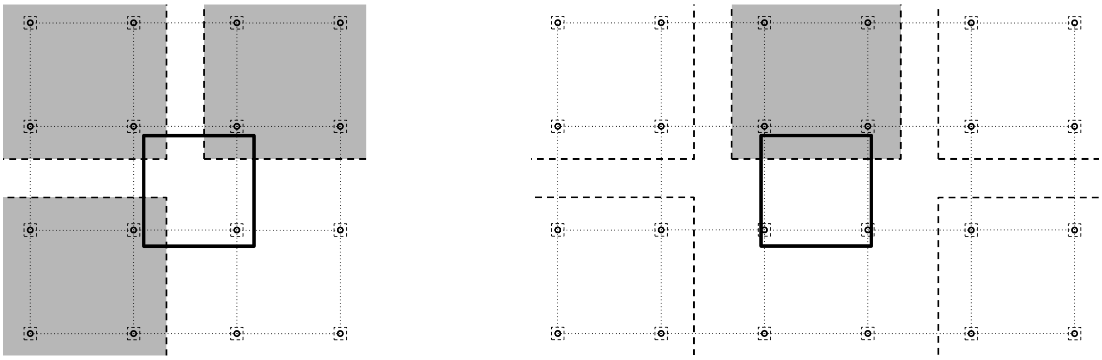

Figure 3.4: 1-move and 2-move, with clusters that can by seriously affected (grey) and those that cannot be affected if of type 4 (white).

Proof. Since every move is a 1-move, a 2-move or a 4-move, the claim follows directly from lemmas. ☐

Observation. *A* 4*-move turning a cluster of type* 4 *to type* 0 *does not change type of any other type* 4 *cluster.*

Claim 4. *It is always possible to turn cluster of type* 3 *to type* 1 *or type* 0*.*

Proof. First, suppose no point is passive. Since the cluster is not of type 4, one of yards is not free, making its point (loosely) dependent on another point(s) of the cluster. Again, since we cannot cover all four points with a single move, another yard is not free. It must be the one positioned diagonally, since otherwise it would either not block the covering 4-move or it would yield two passive points. But this means we have two dependent points, not mutually, and we cannot place three more moves into such configuration.

Thus there must be a passive point. Since no other point may be passive, the one is covered by a move. No other point can be dependent on another one in the cluster, therefore the neighboring points have free ε-space on the outer side. Existence of a move covering the corner point guarantees that at least one of neighboring points can be covered together with it, yielding possible transition to type 1 (or 0). See Figure 3.5. ☐

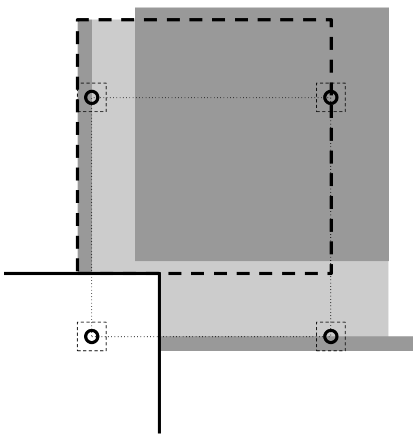

Figure 3.5: Type 3 cluster with grey free space and a dashed move to type 1.

We will not set up any restriction on changing of cluster types by the destructor — we will suppose he might lower type of several clusters anyhow, with respect to Claim 3. It will be crucial to know the constructor cannot change the type of more than four type 4 clusters at once.

Now we define a potential in the very same way as it is described in the original paper, proving the desired bound on the value of the game.

Theorem 2. *The value of the game is at most $\frac{3}{5}M$.*

Proof. Let C_{type} be the number of clusters for each type $0 \dots 4$.

We define the following potential: $\phi = 0C_0 + 1C_1 + 2C_2 + \frac{5}{2}C_3 + \frac{12}{5}C_4$.

We want to prove that the destructor may enforce reducing the potential by at least 2 each doubleturn.

Observe that the potential grows only in the case where a cluster of type 4 is turned into type 3.

While $C_4 > 0$, the destructor can turn a type 4 cluster into type 0, diminishing the potential by at least $\frac{12}{5}$. The constructor may respond by turning up to four type 4 clusters into type 3, increasing the potential by $\frac{2}{5}$, so the total diminisher is at most 2.

If $C_4 = 0$ and $C_3 > 0$, the destructor turns type 3 into type 1, reducing potential by at least $\frac{3}{2}$ and since the constructor must make a move, he must at least turn a cluster of type 3 into 2, taking $\frac{1}{2}$ from the potential. If C_3 is zero as well, they both can decrease the potential by 1 by affecting clusters of type 2 and 1.

As $\phi = 0$ if and only if $C_{1\dots4} = 0$, i.e. when the game ends, we know that number of moves until end of the game is at most equal to the potential. Since at the beginning of the game $\phi \leq \frac{12}{5}\frac{M}{4} = \frac{3}{5}M$, we have an upper bound on the number of moves. □

In the other half of the chapter we improve the strategy to lower the upper bound to $\frac{9}{16}M$. During analysis of the potential we found out that while there are type 4 clusters, the constructor changes as many of them as possible into type 3, i.e. covers four points at once. It is rather suspicious that it should be his best strategy. We will see that having a closer look at the surroundings of these moves will prove the suspicion correct.

3.2 Cradles

For our further investigations we focus on those constructor's moves that considerably increase the potential — those moves that turn several type 4 clusters into type 3.

Definition. *A cradle is a set of points consisting of two neighboring clusters that has yards of two of its points, each from different cluster, at least partially covered with a single move —* the central move.

Observe that a player may be able to turn two neighboring clusters into a cradle or to turn four clusters into two cradles. We will find out that the potential of a cradle can be set to a significantly lower value than a union of two type 3 clusters.

Without loss of generality, we imagine the cradle positioned horizontally, with covered points in the upper part.

Definition. *A side move is a move into a cradle covering two leftmost or two rightmost points, sharing a part of the edge with the central move.*

For cradle and side moves see Figure 3.6.

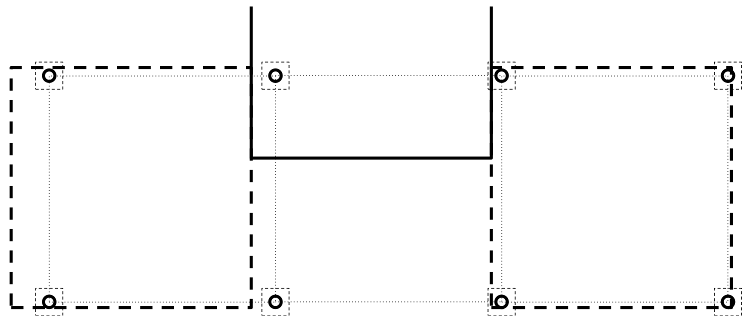

Figure 3.6: A cradle with central move and outlined side moves (dashed).

Both players will be interested in these moves. We describe a configuration of a cradle with three measures (M, S, P):

- M denotes the maximal number of further moves to this cradle,
- S denotes the number of permitted side moves,
- P denotes the number of side moves placed.

3.3 Potential of cradles

We need the following set of observations:

Observation. $0 \leq S + P \leq 2$.

Observation. $0 \leq M \leq 6$.

Lemma 6. $P \geq 1 \Rightarrow M \leq 3$.

Proof. If a side move is made, it is not possible to make more than three more moves to this cradle, since at most two moves fit into the bottom part. □

Lemma 7. $P = 2 \Rightarrow M \leq 1$.

Proof. If both side moves are made, only one more move into the cradle can be done. □

Observation. *Every side move must decrease M by at least one.*

Claim 5. $M \leq 4 + S$.

Proof. Suppose the opposite; if $S = 1$, there is no passive point nor a point dependent on another point of this cradle. There is still a possibility of side bottom point having a move next to itself in a distance at most 3ε, but even though it is not dependent, at most three moves fit into the bottom part. If $S = 0$, since no move disables both side moves at once, we may solve them separately, except for the case both are disabled by the special move — in this case we calculate that there is only space for two moves in the bottom line. See Figure 3.7. ☐

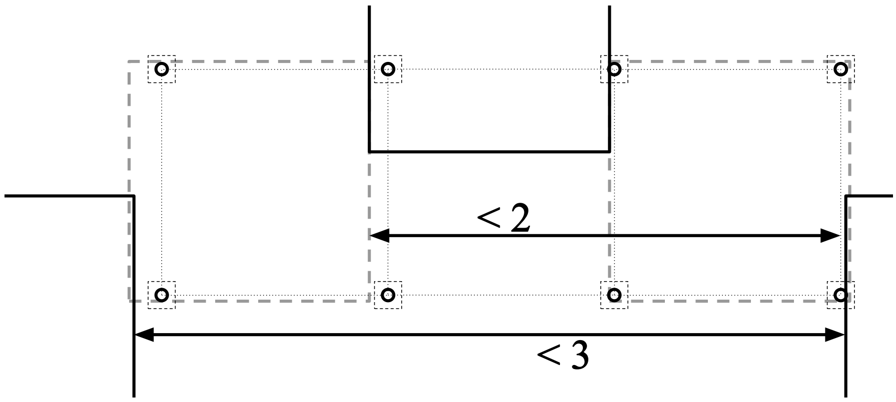

Figure 3.7: Special moves blocking side moves reducing number of possible moves without making a point dependent or passive.

Lemma 8. $P = 1 \wedge S = 0 \Rightarrow M \leq 2$.

Proof. Two points are covered by the side turn and one point is made dependent or passive. Since the other side move is not possible, either there is one more passive or dependent point or at most one further move fits into the bottom part. ☐

In Table 3.1 we define a potential for each possible (M, S, P) triple. We also calculate the *temperature* of each configuration — the difference between the highest and the lowest potential that can be reached from given position by a move into the cradle.

M \ S,P	0,2	0,1	1,1	0,0	1,0	2,0
0	0/–	0/–	–	0/–	–	–
1	1/0	1/0	1/0	1/0	1/0	–
2	–	2/0	2/0	2/0	2/0	2/0
3	–	–	2.5/1	3/0	3/0	3/0
4	–	–	–	4/0	4/0	4/0
5	–	–	–	–	4.5/2	4.5/1.75
6	–	–	–	–	–	4.5/2

Table 3.1: Potential/temperature of a cradle with specific M, S, P.

As every move into the cradle decreases M, we may always set potential equal to M with zero temperature. Explanation of values in cells differing from it follows.

Empty cells are a result of the rules $M \geq S$, $M \leq 4 + S$, $P \geq 1 \Rightarrow M \leq 3$, $P = 2 \Rightarrow M \leq 1$ and $P = 1 \wedge S = 0 \Rightarrow M \leq 2$.

Making the second side move in the $(3, 1, 1)$ position leads to $(1, 0, 2)$ or $(0, 0, 2)$ and any other move decreases the potential at least to 2, thus we may set the potential to $\frac{5}{2}$ with temperature 1. This configuration is actually equivalent to a type 3 cluster.

The value of the cell $(4, 0, 0)$ has to be $4/0$, and since we did not exclude the option of diminishing S without reducing M simultaneously, we keep the other two values in the row at the same values to avoid an unexpected increase of the potential.

The $(5, 1, 0)$ configuration may be turned to $(2, 0, 1)$ or to one with lower M, and on the other hand, we cannot get to a bigger potential than 4. Thus the potential must be at least 4, but we increase it to $\frac{9}{2}$ to simplify our approach. This does not change its temperature, which is 2. Similarly, $(5, 2, 0)$ may become $(3, 1, 1)$, so its potential is at least $\frac{17}{4}$, but we increase it to $\frac{9}{2}$, maintaining the temperature $\frac{7}{4}$.

The last configuration $(6, 2, 0)$ may lead to $(3, 1, 1)$ or to one with even lower potential, while the smallest decrease is to $(5, 2, 0)$ or $(5, 1, 0)$ configurations, yielding value of the cell $\frac{9}{2}/2$.

We see that the potential never exceeds $\frac{9}{2}$. Furthermore, thanks to artificial increases of the potential for $M = 5$, the temperature only decreases during the game. Therefore, if the destructor always plays into the warmest cradle, the potential is decreased by at least 2 each doubleturn except for a few of cases when the constructor picks a last move of currently highest temperature. But since amount of different temperatures is small natural number, it will not seriously affect the result.

3.4 Resulting strategy

We need to ensure that central move never creates two non-neighboring type 3 clusters. This could happen if the other two clusters surrounding the move already belong to a cradle. See Figure 3.8.

Claim 6. *It is possible to reorder constructor's moves in such a way that each move that affects some clusters of type* 4 *turns them into at most one type* 3 *cluster and up to two cradles.*

Proof. We will reorder constructor's moves once there are no more type 4 clusters. First, we sort all moves that can possibly create a cradle by their y and x coordinates, respectively. Now, in their new order, we assign clusters that would form their cradles and are not yet assigned to any other move to them. Now, each of these moves has between zero and four assigned clusters, but never only two non-neighboring. Therefore assigned clusters of a move may be converted to at most two cradles and at most one cluster of type other than 4. See Figure 3.9. ☐

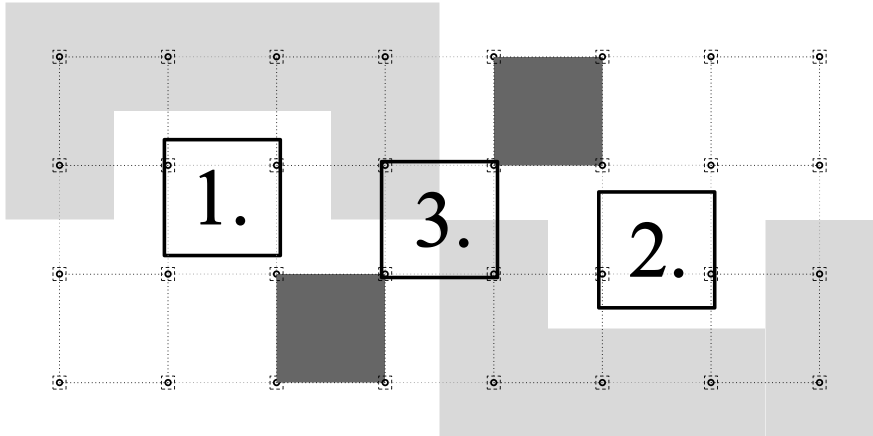

Figure 3.8: An order of central moves yielding two non-neighboring type 3 clusters in a single move.

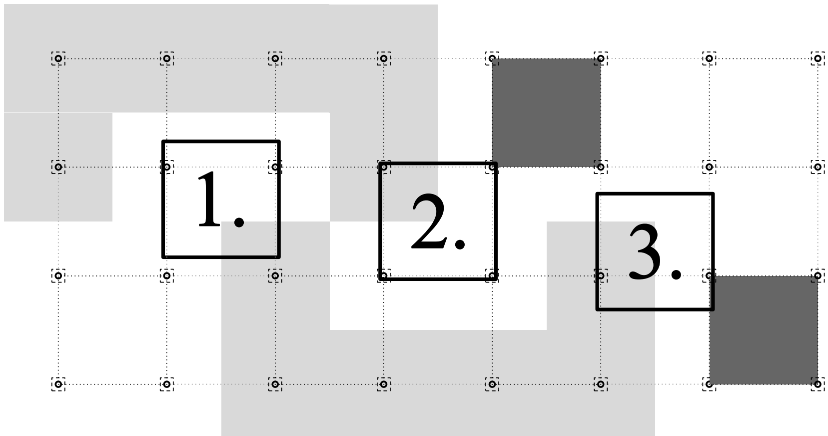

Figure 3.9: The same moves correctly reordered.

Claim 7. *The potential of cluster of type* 4 *can be set to* $\frac{9}{4}$.

Proof. The destructor can turn these clusters into type 0, the constructor has more options (but we have excluded the option of creating two non-neighboring type 3 clusters):

- turn one type 4 cluster into type 3 cluster: $2x \geq 2 + \frac{5}{2} \Rightarrow x \geq \frac{9}{4}$,
- turn two type 4 clusters into cradle: $3x \geq 2 + \frac{9}{2} \Rightarrow x \geq \frac{13}{6}$,
- turn three type 4 clusters into cradle and a type 3 cluster: $4x \geq 2 + \frac{5}{2} + \frac{9}{2} \Rightarrow x \geq \frac{9}{4}$ or
- turn four type 4 clusters into two cradles: $5x \geq 2 + 2 \cdot \frac{9}{2} \Rightarrow x \geq \frac{11}{5}$.

We can set potential of a type 4 cluster to the highest calculated value $\frac{9}{4}$. □

Theorem 3. *The value of the game is at most* $\frac{9}{16}M + k$, *k being a constant.*

Proof. At the beginning of the game the destructor picks type 4 cluster and turns it into type 0 — these moves have temperature $\frac{5}{2}$, higher than any move to a cradle. The constructor's best move is to turn it into a type 3 cluster. Once there are no type 4 clusters left, reassign the cradle clusters by reordering the moves. Now pick the warmest moves in cradles or regular clusters. Since the temperature never rises, the potential is decreased as required except for a finite number of cases — at most the number of possible different temperatures, yielding the small constant k. At the beginning, the potential is $\frac{9}{4}\frac{M}{4} = \frac{9}{16}M$, giving us an upper bound on number of moves. $\square$

This outcome suggests that the constructor should not need to make 2-moves and 4-moves to achieve the best result. He can merely pick single points in type 4 clusters while there are any and then compete with the destructor whether type 3 clusters will be covered with three or four moves — and it is remarkably similar to the strategy of the constructor presented in the previous chapter.

4. Playing on a polygonal board

So far we did suppose our game board is rectangular with sides of integer lengths. This short chapter shows that a gameplay on any polygonal board that is large enough has asymptotically the same bounds on the value of the game.

First, observe that if the board may be any polygon, the value of the game may be an arbitrarily close to both 0 and S, S being the surface of the polygon. Indeed, suppose we have a union of M unit squares in a row where those at even positions are slightly shifted. The value of the game played on such board is exactly M. On the other hand, if we shrink such a polygon by any small ε, we get a game that ends as soon as it begins, yielding its value 0 (see Figure 4.1).

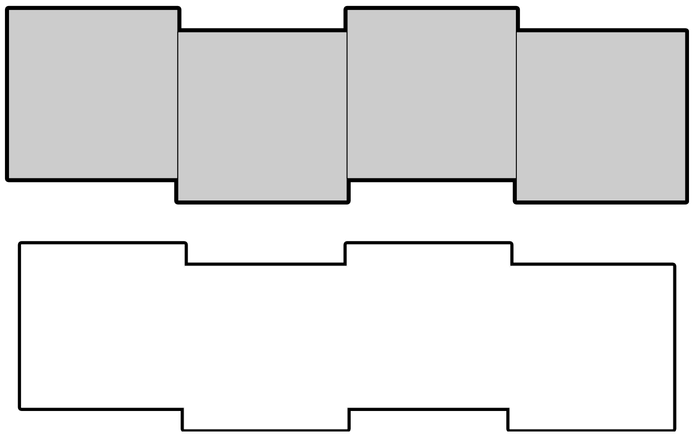

Figure 4.1: Polygonal boards with maximal and minimal values.

We see it is easy to construct polygonal boards to please both constructor and destructor. By simple alteration of the construction we may construct a board for any desired value of the game. However, any polygonal board inflated enough has the same asymptotic value as a rectangular one.

Let us have the polygonal board placed arbitrarily to a coordinates grid. We want to inflate it enough to make fields contained in it (not intersecting the border) cover significant part of the board. We will show that the remaining area left along the border (denoted by R) may be arbitrarily small part of the whole area. In strategies for both constructor and destructor we may ignore the moves that does not belong to a field, assuming the other player may use the remaining area to his liking.

Lemma 9. *Any point of R is sure not farther than 5 in uniform metrics from border, that is $5\sqrt{2}$ in Euclidean metrics.*

Proof. Any point farther than 4 in uniform metrics from any edge is definitely contained in an inner field. It would suffice for the strategy for the constructor where every move partially covering a tile inactivates it. In the strategy for the destructor, we need a move to cover a grid point placed somewhere in the tile of the field to be a move into the field. Thus, any point farther than 5 in uniform metrics can be only contained in a move to an inner field. ☐

Let $p(P)$ be the perimeter of P.

Claim 8. *The area inside a polygon P of points at most at Euclidean distance x from border is at most $p(P) \cdot x$.*

Proof. The area of rectangles of height x above each edge cover the computed area except for reflex angles. There is a sector of size $\frac{\alpha-\pi}{2}x^2$ at each reflex angle α (see Figure 4.2).

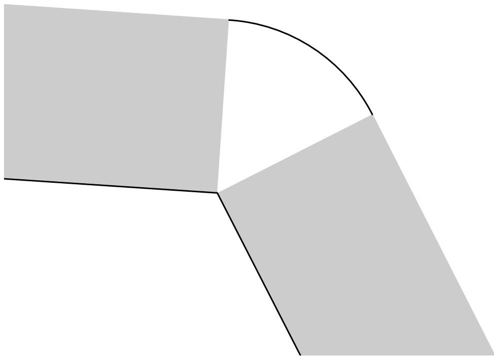

Figure 4.2: The area close to border not covered in rectangles above edges.

However, the area in the intersection of rectangles adjacent to convex angles and the area outside the polygon adjacent to acute angles adds up to compensate it. The intersection at an obtuse angle α contains a sector of area $\frac{\pi-\alpha}{2}x^2$. The intersection at an acute angle only contains a sector of area $\frac{\alpha}{2}x^2$, but the area outside the polygon contains another two sectors of area $\frac{\frac{\pi}{2}-\alpha}{2}x^2$. See Figure 4.3.

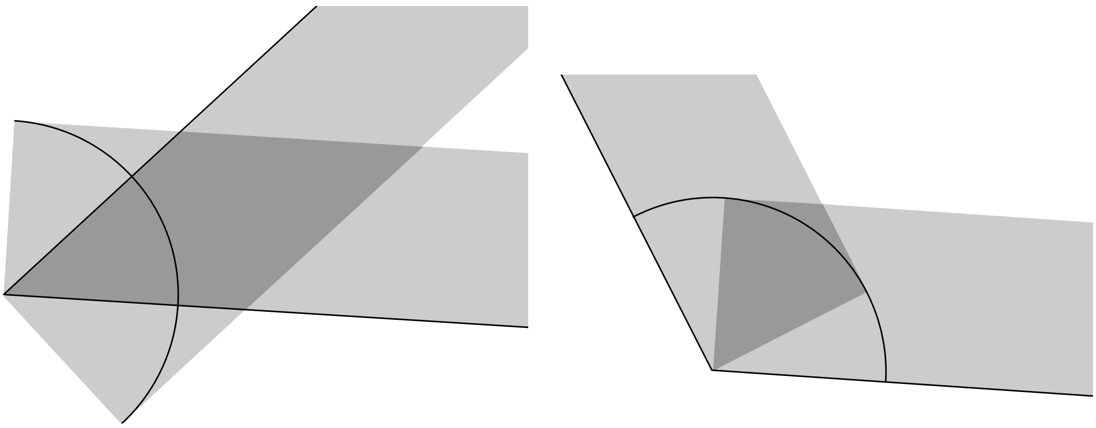

Figure 4.3: The extra area near convex angles.

Now it only suffices to say that each angle either needs (in case of reflex ones) area of $\frac{(\alpha-\pi)}{2}x^2$ or provides (in case of convex ones) $\frac{(\pi-\alpha)}{2}x^2$. Since in every polygon the average angle is less than π, the provided area is bigger than the area needed. $\square$

Let $a(P)$ denote the area of polygon P.

Theorem 4. *For any polygon P and $\varepsilon > 0$ there is a positive real q such that value of the game on $P \cdot q$ (P inflated by factor q) is between $(\frac{33}{64} - \varepsilon)a(P \cdot q)$ and $(\frac{9}{16} + \varepsilon)a(P \cdot q)$.*

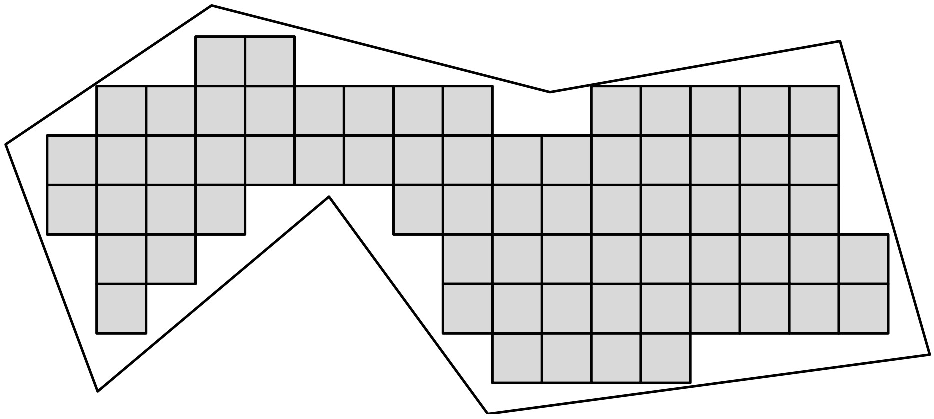

Figure 4.4: A polygonal board with inner fields.

Proof. The width of the surface R along polygon sides can be bounded by $5\sqrt{2}$, therefore $a(R) \leq 5\sqrt{2}p(P)$.

Since $p(P \cdot q) = q \cdot p(P)$ while $a(P \cdot q) = q^2 \cdot a(P)$, $\lim_{q \to \infty} \frac{p(P \cdot q)}{a(P \cdot q)} = 0$. Similarly, $\lim_{q \to \infty} \frac{a(R)}{a(P \cdot q)} = 0$.

Therefore we may simply use the fields not intersecting any edge of the board to achieve the expected result and the value of the whole game is asymptotically the same. $\square$

5. Filling with translates of a convex polygon

The case of filling the board with squares is a very special one. Not only that it is possible to tile the plane with a square, but since a square is also Cartesian product of two line segments, it is much easier to come up with an effective strategy for the destructor — any two copies of a square placed carefully can obstruct an area arbitrarily close to the size of the square. This is certainly not true for most other convex polygons.

5.1 Convex polygons

Since any affine transformation of the game does not affect the gameplay, we may play with any affine transformation of a square, i.e. any parallelogram. The chapter 4 shows that a change of shape of the board does not seriously affect the value of the game if the board is large enough.

But what about other convex polygons? We find out that given any convex polygon, we can find a centrally symmetric convex polygon that can be played with equivalently.

Claim 9. *For each convex polygon P there is a centrally symmetric polygon C such that the set $S(P)$ of translates of P has no overlapping ones if and only if $S(C)$ of translates of C has no overlapping ones.*

Proof. Place a translate of P anywhere onto the plane and choose a point x of P. Images of x of all translates of P touching the original one form a centrally symmetric polygon. This polygon is called the *difference body* of P. See Figure 5.1. When scaled by a factor of one half we get the polygon C: two translates of P touch if and only if translates of C at the same locations touch as well. See [6] for details and a more thorough proof. □

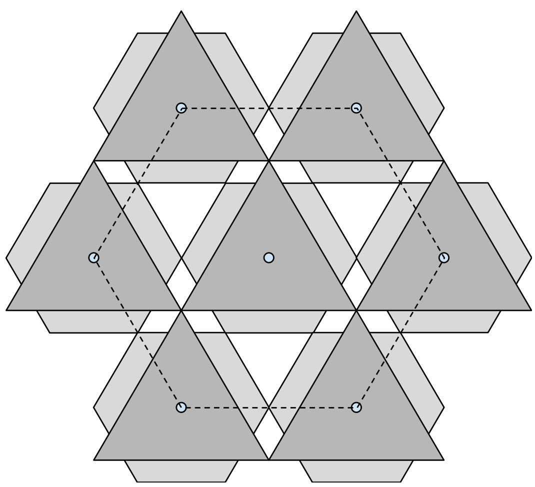

Figure 5.1: Difference body (dashed) of a triangle.

Therefore, we may focus on analyzing the case of filling with translates of a centrally symmetric polygon.

For each centrally symmetric polygon P we can find an inscribed parallelogram of size at least $\frac{2}{\pi}a(P)$. See [14] for details. We will use an affine transformation to turn the parallelogram into unit square.

Lemma 10. *The symmetric polygon is confined to the outer dual square of its maximal inscribed square.*

Proof. If any point of the polygon was outside the dual square, some vertex of the inscribed square can be moved there, stretching an altitude of a triangle of the square, violating maximality. See Figure 5.2. □

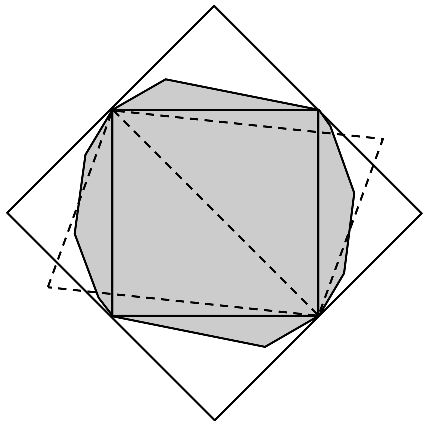

Figure 5.2: Maximal inscribed square of a polygon and the confining outer dual square. An example of a parallelogram violating maximality (dashed).

5.2 More general strategy for the destructor

The following strategy for the destructor works properly for a special, yet large set of centrally symmetric convex polygons: it must have a vertex outside its maximal inscribed square at each side, i.e. no edge of the square is an edge of the polygon. Let Ξ denote this set.

Now, let us make an orthogonal lattice of unit squares with the neighboring ones at distance 2.

Lemma 11. *If there are two neighboring squares covered with a translate of the convex polygon, a space of $2 - \frac{\pi}{2}$ is obstructed (it cannot be covered).*

Proof. If the space were confined between four translates of the polygon, the area would be $1 - (\frac{\pi}{2} - 1)$, since it is a unit square with the excess area of the polygon beyond the inscribed square. See Figure 5.3.

The convexity guarantees that enclosing the space between two more translates leaves the smallest area.

Here we need the special property of Ξ, otherwise those two more translates could cover arbitrarily large part of the space. □

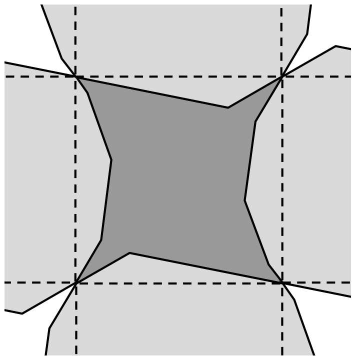

Figure 5.3: Space (dark) confined between four translates of a polygon (light).

Claim 10. *The destructor can cover $\frac{M}{12} - o(M)$ tuples of neighboring lattice squares during a game, M being the number of lattice squares on the board.*

Proof. Let us have a graph where vertices are non-covered squares and every two neighboring squares are connected with an edge. The graph has $2M - o(M)$ edges at the beginning.

No move can remove more than two vertices at once, and any two such vertices are neighboring (see Figure 5.4). It means not more than 7 edges can be taken each move. In odd moves the destructor picks any vertex of degree at least two, because the constructor cannot remove all its neighbors at once. In even moves he picks one of neighbors. This way he will remove at most 7 edges each two turns along with two vertices. The destructor cannot continue once there are less than $\frac{M}{2}$ edges since there might not be a vertex of degree two or more.

Each tuple of doubleturns at most 21 edges and four vertices are removed and one neighboring tuple is covered. Let K denote the number of tuples of doubleturns until there might not be enough edges left:

$$M - 6K = 2(2M - o(M) - 21K) \Rightarrow K = \frac{M}{12} - o(M)$$

It remains to note that constructor's moves taking less vertices (and therefore less edges) are less profitable for him. $\square$

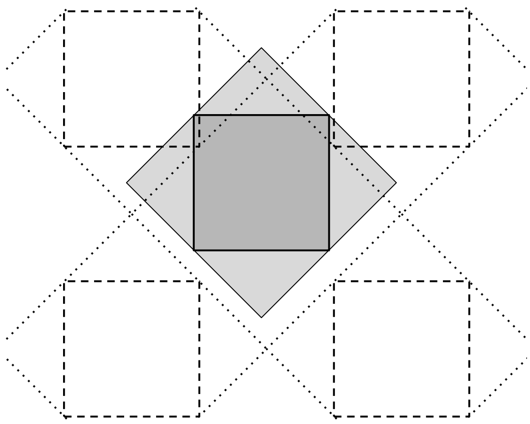

Figure 5.4: Part of the lattice with an inscribed and outer square of a move covering at most two squares.

Theorem 5. *At least* $\frac{1-\frac{\pi}{4}}{24}S - o(S)$ *may remain uncovered on a rectangular board with even width and height of area* S*.*

Proof. We may put unit lattice squares with centers at odd coordinates and then use the strategy for the destructor. Every lattice square corresponds to area of 4 and neighboring tuples yield $2 - \frac{\pi}{2}$ per twelve squares. $\square$

Although the result is probably still far from the value of the game, the approach shows that for every polygon of Ξ it is possible to keep a constant fraction of the board uncovered.

6. Conclusion

6.1 New bounds

The author of the original paper had conjectured that the value of the game (for n and m divisible by 4) is $\frac{9}{16}nm$. It also provided a lower bound $\frac{15}{32}nm$ and upper bound $\frac{3}{5}nm$. Proofs presented here considerably improved these bounds — lower bound to $\frac{33}{64}nm$ and the upper bound to the conjectured $\frac{9}{16}nm$.

We may guess why the lower bound still eludes the desirable value. The main idea of our proof rests in minimizing the possible number of 3-moves after no 4-moves (i.e. D-moves) are possible. Notice that if we managed to extinguish all the D-moves it would result in the value of $\frac{9}{16}nm$. However, there is no assurance it is possible to do so. If it is not, it may still be possible to reach the value, since there may be permitted moves after all the tiles are inactivated due to the simplification of the active/passive tiles approach.

We have actually presented a new discrete game, a hybrid of Two-dimensional Kayles and our game: we have a rectangular board of tiles and a player either changes one active tile into an obstructed one, two neighboring non-obstructed into passive ones (i.e. non-active, non-obstructed) or four non-obstructed into passive ones. We have shown that the game has lower value than the original one, but the following question remains unanswered:

Question 1. *Does the hybrid game always have the same value as our game?*

We also must not rule out the possibility that the conjecture is false and that the value of the game is actually lower. Notice that both Two-dimensional Kayles game and playing on a 4×4 board, that yield the conjectured value, forbid or may avoid the D-moves. Since these games are the main reason to think $\frac{9}{16}nm$ is the right value and D-moves are the main drawback in proving it, it may seem this value is not the correct answer.

However, this conjecture presented in [1] still holds:

Conjecture 1. *The value of the game converges to $\frac{9}{16}M$, M being the area of the board.*

6.2 Generalized games

We have shown in Chapter 4 a board yielding any value can be easily constructed. On the other hand, inflating a board normalizes it by the means that by increasing the inflation factor the value of the game converges to a value between our computed bounds, and it is natural to expect it converges to the value of the game on a rectangular board.

Chapter 5 shows a basic destructor's strategy for tiling with more general polygons, ensuring that it is possible to keep some area unoccupied for all polygons of the special set Ξ. There is still a lot of research to do in this area, such as:

Question 2. *What is the value of the game when filling with circles / symmetric hexagons / other shapes?*

Question 3. *Which (convex) polygon yields the lowest value of the game? Which yields the highest?*

We may also join questions of both generalizing chapters to express the following:

Conjecture 2. *For any polygonal board and any convex filling polygon the following holds: with increasing inflation factor of the board the value of the game converges to the value of the game on a large rectangular board with the same filling polygon.*

Bibliography

[1] Hubai, T. *Competitive rectangle filling.* Manuscript (2009), 24 pages.

[2] Beck, A.; Bleicher, M. N. *Packing convex sets into a similar set.* Acta. Math. Acad. Sci. Hungar. 22 (1972), 283-303.

[3] Torquato, S.; Jiao, Y. *Dense packings of the Platonic and Archimedean solids.* Nature, Volume 460, Issue 7257, pp. 876-879 (2009).

[4] Lagrange, J. L. *Recherches d'arithmétique.* Nouveaux mémoires de l'Académie royale des sciences et belles-lettres de Berlin (1773), 265-312.

[5] Gauss, C. F. *Untersuchungen über die Eigenschaften der positiven ternären quadratischen Formen von Ludwig August Seeber.* Göttingische gelehrte Anzeigen (1831).

[6] Minkowski, H. *Dichteste gitterförmige Lagerung kongruenter Körper.* Nachr. K. Ges. Wiss. Göttingen, 1904, 311-355, in Gesammelte Abhandlungen. Vol. II, Teubner, Berlin, 1911, pp. 3-42.

[7] Rogers, C. A. *Packing and Covering.* New York: Cambridge University Press (2008), 120 pages.

[8] Erdős, P.; Graham, R. L. *On Packing Squares with Equal Squares.* J. Combin. Theory Ser. A 19 (1975), 119-123.

[9] Friedman, E. *Packing unit squares in squares: A survey and new results.* Electronic Journal of Combinatorics, Dynamic Surveys #DS7 (2000).

[10] Friedman, E. *Erich's packing center.* `http://www2.stetson.edu/~efriedma/packing.html`

[11] Berlekamp, E. R.; Conway, J. H.; Guy, R. K. *Winning ways, for your mathematical plays.* New York: Academic Press (1982), Volume 1-4.

[12] Sibert, W. L.; Conway, J. H. *Mathematical Kayles.* International Journal of Game Theory, 20/3 (1992), 237-246.

[13] Mount, D. M; Silverman, R. *Packing and Covering the Plane with Translates of a Convex Polygon.* Journal of Algorithms Volume 11, Issue 4 (1990), 564-580.

[14] Jin, K. *Finding the Maximum Area Parallelogram in a Convex Polygon.* 23rd Canadian Conference on Computational Geometry (CCCG'11). `http://2011.cccg.ca/`

A. Linear program

The following data is input for the program `lp_solve`. This linear program was used to calculate the value of objective function in 2.4. Values are scaled by a factor of $\frac{128}{M}$ to avoid variables and non-integers in result, so the value of objective function 66 yielded by this input corresponds to the value $\frac{33}{64}M$.

Variables identifiers F_*, X_* are the same as in 2.4, constants B_* are replaced by their values.

```
min: 2 F_E + 2 F_D + 2 F_O + 2 F_G;

F_E >= 0;
F_D >= 0;
F_O >= 0;
F_G >= 0;

X_VEOGG >= 0;
X_DEOGG >= 0;
X_EOG >= 0;
X_EO >= 0;
X_VEGG >= 0;
X_DEGG >= 0;
X_DOGG >= 0;
X_DGG >= 0;
X_EG >= 0;
X_OG >= 0;
X_E >= 0;
X_O >= 0;
X_G >= 0;

F_E = 32 - X_VEOGG - X_DEOGG - X_EOG - X_EO - X_DEGG - X_VEGG
    - X_EG - X_E;
F_D = 16 - X_DEOGG - X_DEGG - X_DOGG - X_DGG;
F_O = 32 - X_VEOGG - X_DEOGG - X_EOG - X_EO - X_DOGG - X_OG
    - X_O;
F_G = 64 - 2 X_VEOGG - 2 X_DEOGG - X_EOG - 2 X_DEGG - 2 X_VEGG
    - X_EG - 2 X_DOGG - 2 X_DGG - X_OG - X_G - F_D;

X_E + X_EG + X_DEGG + X_VEGG + X_EOG + X_EO + X_DEOGG
    + X_VEOGG <= F_E;
X_DGG + X_DOGG + X_E + X_EG + X_DEGG + X_VEGG
    + X_EOG +X_EO + X_DEOGG + X_VEOGG <= F_E + F_D;
X_O + X_OG + X_DGG + X_DOGG + X_E + X_EG + X_DEGG + X_VEGG
    + X_EOG + X_EO + X_DEOGG + X_VEOGG <= F_E + F_D + F_O;
X_G + X_O + X_OG + X_DGG + X_DOGG + X_E + X_EG + X_DEGG + X_VEGG
    + X_EOG + X_EO + X_DEOGG + X_VEOGG <= F_E + F_D + F_O + F_G;

X_VEOGG + X_VEGG <= 8;
```