On the Minimum Number of Pieces for Two-Dimensional Anti-Slide Using T-Tetrominoes

Kento KIMURA\textsuperscript{(a)}, Student Member, Kazuyuki AMANO\textsuperscript{(b)}, and Tetsuya ARAKI\textsuperscript{(c)}, Members

SUMMARY

Given a box of some specified size and a number of pieces of some specified shape, the anti-slide problem considers how to pack the pieces such that none of the pieces in the box can slide in any direction. The object is to find such a sparsest packing. In this paper, we consider the problem for the case of a two-dimensional square box using T-tetromino pieces. We show that, for a square box of side length $n$, the number of pieces in a sparsest packing is exactly $\lfloor \frac{2n}{3} \rfloor$ when $n \equiv 0 \pmod{3}$, and is between $\frac{2n}{3} - 1$ and $n - 1$ when $n \equiv 0 \pmod{3}$.

\textit{key words:} anti-slide packing, integer programming, puzzle

1. Introduction

Mathematical puzzles such as pencil puzzles provided us with interesting problems in various fields. In terms of complexity, many puzzles described in the literature have been shown to be NP-complete, which seems to be a common essence of puzzles [1]–[3]. Major complexity results for puzzles are surveyed by Hearn and Demaine [4]. In this paper, we investigate the essence of a solution of a puzzle. Next, we introduce the puzzle named “anti-slide.”

Given a $4 \times 4 \times 4$ box and a number of $2 \times 2 \times 1$ pieces, consider the problem of how to pack the pieces into the box such that none of the pieces can slide in any direction. We assume that there is no friction between the pieces, and a piece can never go out of the box. The object is to use a minimum number of pieces starting from 16 pieces. An optimal solution of this puzzle uses 12 pieces, as shown in Fig. 1. This puzzle, named “anti-slide,” was created by William Strijbos in 1994 [5].

In this paper, we consider anti-slide packing with aligned pieces and orthogonal movement. An anti-slide packing is called a stable packing. We define “stable” in Sect. 2. The problem is whether, given a three-dimensional cubic box and the number of pieces of some specified shape, there is a stable packing with a minimum number of pieces.

Amano, Nakano, and Yamazaki [6] gave an integer linear programming (IP) formulation of the problem for finding a sparsest stable packing and obtained upper and lower bounds on the density of such a packing of $2 \times 2 \times 1$ pieces for a three-dimensional box. It is also natural to consider such packings for various shapes of the pieces. In our previous work [7], we analyzed stable packings for several shapes of polycubes. We showed that the minimum number of pieces in a stable packing with the specified shape in Group A of Fig. 2 for an $n \times n \times n$ box is linear in $n$. However, the problem to determine the asymptotic behaviour of the minimum number of pieces in a stable packing for a three-dimensional box was left as an open problem. Note that a linear upper bound on the number of pieces for a two-dimensional square box gives a quadratic upper bound for a three-dimensional box: stack $n$ stable packings for an $n \times n$ box. In a related work, Takenaga, Xi, and Inada [8] considered a two-dimensional case using pentomino pieces. Namely, they...
enumerated anti-slide packings of pentomino pieces for a two-dimensional box of small size using ordered binary decision diagrams.

1.1 Contributions

In this paper, we focus on the problem of packing T-tetrominoes (Fig. 3) into a two-dimensional square box.

The L-tetromino is the simplest piece among the pieces shown in Fig. 2. However, it seems that a sparsest packing for an \( n \times n \) box is obtained by stacking \( n-1 \) L-tetrominoes diagonally, which is not so interesting. The T-tetromino is the second simplest. We will see in the following that the results suggest that the structure of a sparsest packing would be quite different depending on whether the side length of a box is divisible by three or not (see Sect. 2).

First, we conducted computer searches based on the IP formulation developed in [6] to find sparsest stable packings of T-tetromino pieces for an \( n \times n \) box for \( n \leq 15 \). Interestingly, the results suggest that the structure of a sparsest packing would be quite different depending on whether the side length of a box is divisible by three or not (see Sect. 2).

Let us see this more precisely. Let \( F(n) \) be the minimum number of T-tetromino pieces of a stable packing for an \( n \times n \) box. Our experimental results suggest that \( F(n) \sim 2n/3 \) for \( n \equiv 0 \mod 3 \), and \( F(n) \sim n \) for \( n \equiv 0 \mod 3 \).

In this paper, we determine the value of \( F(n) \) exactly for the case of \( n \not\equiv 0 \mod 3 \) by proving matching upper and lower bounds of \( F(n) = \lfloor 2n/3 \rfloor \). For the case of \( n \equiv 0 \mod 3 \), we have succeeded only in proving the bound of \( 2n/3 - 1 \leq F(n) \leq n - 1 \), although we believe that the upper bound is tight for \( n \geq 9 \). Note that when \( n = 6 \), the smallest number of pieces in a stable packing is 4 (Fig. 3).

The organization of this paper is as follows. In Sect. 2, we define the terms and show our experimental results. In Sect. 3, we show the upper bounds on \( F(n) \) by giving an explicit construction of stable packings of T-tetromino pieces for an \( n \times n \) box. In Sect. 4, we show the lower bounds on \( F(n) \). Finally, in Sect. 5, we discuss the reasons why \( F(n) \) behaves differently depending on whether \( n \) is divisible by three or not.

2. Definitions and Experimental Results

For a positive integer \( \alpha \), \( \{1, 2, \ldots, \alpha\} \) denotes a set \{1, 2, \ldots, \alpha\}.

We describe integral and orthogonal. Integral means that we assume all the coordinates of corner points of pieces are integers. By using the integral, we view the \( n \times n \) box as a two-dimensional array of cells of unit size. Each cell is identified by \( (i, j) \in [n] \times [n] \). Orthogonal means that we assume that each piece is axis-aligned and each piece slides to the orthogonal direction, i.e., parallel to \( x \)- or \( y \)-axes.

For an \( n \times n \) box and a number of T-tetromino pieces, we say that a situation is a packing if each piece goes in the box and is restricted by the integral and the orthogonal conditions. We say that a packing is a stable packing if none of pieces in a packing can slide in any direction.

A T-tetromino is a polyomino shaped like the letter T consisting of four unit squares. We say that a piece is placed at \( (i, j) \) if the central square of the piece is adjusted to \( (i, j) \). Also, we say that a piece is placed with direction \( d \in \mathcal{R} \), where \( \mathcal{R} = \{U, R, D, L\} \), each of which is shown in Fig. 4. For example, if a T-tetromino piece is placed at \( (2, 2) \) with direction \( U \), then the cells \((2, 2), (3, 2), (1, 2), \) and \((2, 3)\) are occupied by this piece.

By using the IP formulation in Amano et al. [6], the problem can be represented as an IP problem. We solved this problem by using an IP solver [9] for \( n \leq 15 \). The results are shown in Table 1, and examples of packings that attain \( F(n) \) for \( n = 10, 11, \ldots, 15 \) are shown in Fig. 5.

### Table 1 Values of \( F(n) \) for \( n \leq 15 \).

| \( n \) | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| \( F(n) \) | 2 | 3 | 4 | 4 | 5 | 8 | 6 | 7 | 11 | 8 | 9 | 14 |

3. Upper Bound

From Fig. 5, we can observe that a sparsest packing for an \( n \times n \) box looks like a diagonal line for \( n \not\equiv 0 \mod 3 \) and a V-shape for \( n \equiv 0 \mod 3 \). We believe that this holds also for higher values of \( n \).

An easy calculation shows that the number of pieces used in these packings is \( \sim \frac{1}{2}n \) for the former, and \( \sim n \) for the latter. In this section, we show the upper bounds on \( F(n) \) matching this by giving an explicit construction of stable packings.

**Theorem 1.** For every \( n \geq 4 \) such that \( n \equiv 1 \mod 3 \), we have

\[
F(n) \leq \frac{2(n-1)}{3},
\]

and for every \( n \geq 5 \) such that \( n \equiv 2 \mod 3 \), we have

\[
F(n) \leq \frac{2n-1}{3}.
\]

**Proof.** First we consider the case \( n \equiv 1 \mod 3 \).
The proof is by induction. The base case, \( F(4) = 2 \), is obvious. Place two T-tetromino pieces as in the center part of Fig. 5(a). The induction step is also obvious by comparing Fig. 5(a) and (d). Precisely, we can obtain a specified stable packing for the \((n + 3) \times (n + 3)\) box by adding two pieces to a stable packing for the \(n \times n\) box; add one piece with \(R\)-direction to the top right corner and the other piece with \(U\)-direction to the bottom left corner. This completes the proof for the case \( n \equiv 1 \mod 3 \).

The proof for the case \( n \equiv 2 \mod 3 \) is analogous, referring to Fig. 5(b) and (e).

**Theorem 2.** For every \( n \geq 6 \) such that \( n \equiv 0 \mod 3 \), we have \( F(n) \leq n - 1 \).

**Proof.** The proof is similar to the proof of Theorem 1.

We use the packing of Fig. 5(c) as the base for the case \( n \equiv 0 \mod 6 \), and the packing of Fig. 5(f) as the base for the case \( n \equiv 3 \mod 6 \).

Then, we can obtain a stable packing for the \((n + 6) \times (n + 6)\) box from a packing for the \(n \times n\) box by adding six pieces: insert three pieces with \(L\)-direction in the middle of the left slope and insert three pieces with \(R\)-direction in the middle of the right slope. Similarly, we can obtain a stable packing for the \(6 \times 6\) box consisting of five pieces from a packing for the \(12 \times 12\) (Fig. 5(c)) box by removing six pieces: remove three pieces with \(L\)-direction on the left slope and three pieces with \(R\)-direction on the right slope. A packing for the \(9 \times 9\) box consisting of eight pieces can be obtained from the packing for the \(15 \times 15\) box (Fig. 5(f)) in a similar way. This suffices to prove the theorem.

### 4. Lower Bound

In this section, we give the lower bound on \( F(n) \), proving that the upper bound shown in Theorem 1 is exactly tight.

**Theorem 3.** For every \( n \geq 4 \), we have \( F(n) \geq \frac{7}{3}n - 1 \).

Since \( F(n) \) takes only an integer, the upper bounds in Theorem 1 and the lower bound in Theorem 3 are shown to be identical.

**Corollary 4.** For every \( n \geq 4 \) such that \( n \equiv 0 \mod 3 \), we have \( F(n) = \left\lfloor \frac{2n}{3} \right\rfloor \).

The rest of this section is devoted to the proof of Theorem 3. Roughly speaking, we will show that, given any stable packing, we can always pick a set of pieces of size \( \geq 2n/3 - 1 \) from the packing by applying an appropriate procedure.

#### 4.1 Preparation

Consider a stable packing \( P_n \) for an \( n \times n \) box. Let \( W_l \) denote the left vertical boundary of the box. Similarly, let \( W_r, W_d, \) and \( W_u \) denote the right vertical, the bottom horizontal, and the top horizontal boundary of the box.

We introduce an algorithm that defines a “backbone” set \( \mathcal{P} \) of pieces for a given packing \( P_n \). See Algorithm 1. In what follows, we verify that Algorithm 1 surely halts and outputs a set \( \mathcal{P} \) for any stable packing. Note that every piece
Algorithm 1 Select a set of pieces from any stable packing

1: procedure Select($P_n$)
2: Pick an arbitrary piece in $P_n$ that touches $W_i$.
3: Designate the picked piece as $p_1$.
4: $m := 1$
5: while $p_m$ does not touch $W_i$ do
6: Pick an arbitrary piece that touches $p_m$ from the right.
7: $m := m + 1$
8: Designate the picked piece as $p_m$.
9: end while
10: $P_{m+} := \{p_1, \ldots, p_m\}$.
11: Pick an arbitrary piece that touches $W_d$.
12: Designate this picked piece as $p'$.
13: if $p' \notin P_{m+}$ then
14: $m := m + 1$
15: $p_m := p'$
16: end if
17: $p'' := p'$
18: while $p''$ does not touch $W_d$ do
19: Pick an arbitrary piece that touches $p''$ from above.
20: Designate this picked piece as $p'$.
21: if $p' \notin P_{m+}$ then
22: $m := m + 1$
23: $p_m := p'$
24: end if
25: $p'' := p'$
26: end while
27: $P := \{p_1, \ldots, p_m\}$
28: output $P$ and $P_{l-r}$.
29: end procedure

In a stable packing must touch another piece or a boundary of the box in order to avoid sliding. For $m’ = 1, 2, \ldots$, let $x(m’)$ denote the maximum value of the $x$-coordinate of a cell occupied by $p_{m’}$ designated in line 8 (or in line 3 for $m’ = 1$). Because every piece with the shape of T-tetromino is touched from right, $x(m’)$ has the following properties: first, $x(m’)$ is a monotonically non-decreasing function; second, if $x(m’ + 1) - x(m’)=0$, then $x(m’ + 2) - x(m’ + 1) \geq 1$. These two properties imply that $x(m) = n$ for some $m$ with $m \leq 2n$. This guarantees that the first while loop in Algorithm 1 will eventually be terminated. $P_{l-r}$ (in line 10) is a sequence of picked pieces in line between $W_i$ and $W_r$. A similar argument also shows the termination of the second while loop in the algorithm. This suffices to show that Algorithm 1 halts for any stable packing. We should note that Algorithm 1 surely outputs a set $P$ since every piece in a stable packing must touch another piece or a boundary of the box in order to avoid sliding.

An example of a stable packing and the corresponding output of Algorithm 1 is illustrated in Fig. 6. In this example, the algorithm outputs the set $\mathcal{P}$ of nine pieces labeled 1 to 9 and the set $P_{l-r}$ is consisting of five pieces labeled 1 to 5 in Fig. 6.

The following definition is the key to our analysis.

Definition 5. For a set of pieces $\mathcal{P}$ placed in an $n \times n$ box and for $i \in [n]$, we say that the $i$-th column ($i$-th row, respectively) is marked if a cell with the $x$-coordinate $i$ ($y$-coordinate $i$, respectively) is occupied by some piece in $\mathcal{P}$.

It is obvious that $P_n$ with a stable packing marks all columns and rows. Suppose that $\mathcal{P} = \{p_1, \ldots, p_m\}$ is an output of Algorithm 1. It is clear that $\mathcal{P}$ marks all columns and rows. For each $k \leq m$, let $M(k)$ denote the total number of rows and columns that are marked by $\{p_1, \ldots, p_k\}$.

For example, the piece labeled 1 in Fig. 6(b) marks the fifth, sixth, and seventh rows and the first and second columns, and thus $M(1) = 5$.

We define $Q_4$, $Q_3$, and $Q_{\leq 2}$ as follows:

$Q_4 := \{ p_k \mid k \in [\mathcal{P}] \setminus \{1\}, M(k) - M(k - 1) = 4 \}$,
$Q_3 := \{ p_k \mid k \in [\mathcal{P}] \setminus \{1\}, M(k) - M(k - 1) = 3 \}$,
$Q_{\leq 2} := \{ p_k \mid k \in [\mathcal{P}] \setminus \{1\}, M(k) - M(k - 1) \leq 2 \}$.

By the definition, it is obvious that $M(m) = 2n$ and that $\mathcal{P} = Q_4 \cup Q_3 \cup Q_{\leq 2} \cup \{ p_1 \}$.

4.2 Analysis of $P_{l-r}$ of $\mathcal{P}$

Given a set $P_{l-r}$ of pieces and $k \in [[P_{l-r}]]$, let $G_{\text{max},y}(k)$ and $G_{\text{min},y}(k)$ be the maximum and minimum values of the $y$-coordinate marked by the first $k$-th pieces of $P_{l-r}$. Similarly, let $G_{\text{max},x}(k)$ be the maximum value of the $x$-coordinate marked by the first $k$-th pieces of $P_{l-r}$.

For $k \in [[P_{l-r}]]$, let $H_{\text{max},y}(k)$ and $H_{\text{min},y}(k)$ be the maximum and minimum values of the $y$-coordinate of a cell occupied by the $k$-th piece in $P_{l-r}$. Similarly, let $H_{\text{max},x}(k)$ be the minimum value of the $x$-coordinate of a cell occupied by the $k$-th piece in $P_{l-r}$.

In addition, we define $X(k)$, $Y_{\text{max}}(k)$ and $Y_{\text{min}}(k)$ as follows:

$X(k) = G_{\text{max},x}(k) - G_{\text{max},x}(k - 1)$,
$Y_{\text{max}}(k) = G_{\text{max},y}(k) - G_{\text{min},y}(k - 1)$,
$Y_{\text{min}}(k) = G_{\text{min},y}(k - 1) - G_{\text{min},y}(k)$,

where $X(k)$, $Y_{\text{max}}(k)$, and $Y_{\text{min}}(k)$ are undefined when $k = 1$. According to the definitions of these functions and Algorithm 1, we have the following:

\begin{equation}
0 \leq X(k) \leq 3,
0 \leq Y_{\text{max}}(k) \leq 2,
0 \leq Y_{\text{min}}(k) \leq 2,
Y_{\text{max}}(k) \cdot Y_{\text{min}}(k) = 0.
\end{equation}
We call the cell \((G_{\text{max},s}(k), G_{\text{max},g}(k))\) the \(k\)-th top corner. Also, we call the cell \((G_{\text{max},b}(k), G_{\text{min},g}(k))\) the \(k\)-th bottom corner. We refer to a top corner or a bottom corner as a corner. We say that the \(k\)-th top corner (or similarly, \(k\)-th bottom corner) is filled if it is occupied by the \(k\)-th piece in \(P_{t-r}\).

We introduce two types of 0/1 variables:

\[
\{ s_{\text{max}}[k] \mid k \in |P_{t-r}| \},
\]

and

\[
\{ s_{\text{min}}[k] \mid k \in |P_{t-r}| \}.
\]

The variable \(s_{\text{max}}[k]\) takes the value 1 if and only if the \(k\)-th top corner is filled. Similarly, the variable \(s_{\text{min}}[k]\) takes the value 1 if and only if the \(k\)-th bottom corner is filled.

The following lemma says that every piece in \(Q_4\) must be picked in the first while loop in Algorithm 1.

**Lemma 6.** \(Q_4 \subset P_{t-r}\).

**Proof.** The proof is immediate by seeing that every piece picked in the second while loop (in lines 18–26) in Algorithm 1 can mark only rows since all columns have been marked during the first while loop (in lines 5–9).

The following lemma says that \(p_k \in Q_4\) fills neither the \(k\)-th top nor bottom corner. In addition, \(p_k \in Q_4\) touches \(p_{k-1}\) that fills the \((k-1)\)-th top or bottom corner.

**Lemma 7.** For every integer \(k\) such that \(2 \leq k \leq |P_{t-r}|\), if \(p_k \in Q_4\), then \(s_{\text{max}}[k] = 0, s_{\text{min}}[k] = 0, \) and either \(s_{\text{max}}[k-1] = 1\) or \(s_{\text{min}}[k-1] = 1\).

**Proof.** Suppose that \(p_k \in Q_4\). This means that

\[X(k) + Y_{\text{max}}(k) + Y_{\text{min}}(k) = 4.
\]

By Eq. (1), there are four possibilities for the values of \(X(k), Y_{\text{max}}(k),\) and \(Y_{\text{min}}(k)\).

\[
\begin{align*}
\text{Case I:} & \quad \begin{cases} X(k) = 3 \\
Y_{\text{max}}(k) = 1 \\
Y_{\text{min}}(k) = 0 \end{cases} \\
\text{Case II:} & \quad \begin{cases} X(k) = 3 \\
Y_{\text{max}}(k) = 0 \\
Y_{\text{min}}(k) = 1 \end{cases} \\
\text{Case III:} & \quad \begin{cases} X(k) = 2 \\
Y_{\text{max}}(k) = 2 \\
Y_{\text{min}}(k) = 0 \end{cases} \\
\text{Case IV:} & \quad \begin{cases} X(k) = 2 \\
Y_{\text{max}}(k) = 0 \\
Y_{\text{min}}(k) = 2 \end{cases}
\end{align*}
\]

**Case I.** The situation of this case is illustrated in Fig. 7. Since \(X(k) = 3\) and \(Y_{\text{max}}(k) = 1\), the piece \(p_k\) is with \(U\)-direction and the piece \(p_{k-1}\) fills the \((k-1)\)-th top corner. This implies that \(s_{\text{max}}[k] = 0\) and \(s_{\text{max}}[k-1] = 1\).

It remains to show that \(s_{\text{min}}[k] = 0\). To this purpose, it is sufficient to show that \(H_{\text{min},g}(k) - G_{\text{min},g}(k) \geq 1\). In this case, we have

\[H_{\text{min},g}(k) = G_{\text{max},g}(k-1).
\]

This can be verified as follows.

\[
H_{\text{min},g}(k) - G_{\text{min},g}(k) \\
\geq G_{\text{max},g}(k-1) - G_{\text{min},g}(k) \\
= G_{\text{max},g}(k-1) - G_{\text{min},g}(k-1) \geq 1.
\]

**Case II.** This case is “upside down” to Case I, and can be verified analogously.

**Cases III and IV.** In these cases, we can easily verify that the piece \(p_k\) is with \(R\)-direction. The rest of the proof is similar to the proof for Case I.

The next lemma says that if \(p_k \in Q_3\) and the \((k-1)\)-th top and bottom corners are not filled, then the \(k\)-th top and bottom corners are not filled as well.

**Lemma 8.** For every integer \(k\) such that \(2 \leq k \leq |P_{t-r}|\), if \(p_k \in Q_3\), \(s_{\text{max}}[k-1] = 0,\) and \(s_{\text{min}}[k-1] = 0,\) then \(s_{\text{max}}[k] = 0\) and \(s_{\text{min}}[k] = 0.\)

**Proof.** Suppose that \(p_k \in Q_3,\) which means that

\[X(k) + Y_{\text{max}}(k) + Y_{\text{min}}(k) = 3.
\]

Suppose also that \(s_{\text{max}}[k-1] = 0\) and \(s_{\text{min}}[k-1] = 0\). Then, by Eq. (1), the values of \(X(k), Y_{\text{max}}(k),\) and \(Y_{\text{min}}(k)\) are one of the following five cases.

\[
\begin{align*}
\text{Case I:} & \quad \begin{cases} X(k) = 3 \\
Y_{\text{max}}(k) = 0 \\
Y_{\text{min}}(k) = 0 \end{cases} \\
\text{Case II:} & \quad \begin{cases} X(k) = 2 \\
Y_{\text{max}}(k) = 1 \\
Y_{\text{min}}(k) = 1 \end{cases} \\
\text{Case III:} & \quad \begin{cases} X(k) = 2 \\
Y_{\text{max}}(k) = 0 \\
Y_{\text{min}}(k) = 2 \end{cases} \\
\text{Case IV:} & \quad \begin{cases} X(k) = 1 \\
Y_{\text{max}}(k) = 0 \\
Y_{\text{min}}(k) = 2 \end{cases} \\
\text{Case V:} & \quad \begin{cases} X(k) = 1 \\
Y_{\text{max}}(k) = 0 \\
Y_{\text{min}}(k) = 2 \end{cases}
\end{align*}
\]

**Case I.** Since \(X(k) = 3,\) the direction of the piece \(p_k\) is \(U\) or \(D.\) If \(p_k\) is \(U\)-direction, then \(s_{\text{max}}[k] = 0\) is obviously true. Then, we have

\[H_{\text{min},g}(k) - G_{\text{min},g}(k) \geq 1.
\]

This implies that \(s_{\text{min}}[k] = 0.\)

If \(p_k\) is \(D\)-direction, then this is “upside down” to the
4.3 Proof of the Main Theorem

Lemma 10 says that each piece in an output \( \mathcal{P} \) of Algorithm 1 marks at most (about) three rows or columns on average. By combining this with the fact that the total number of marked rows or columns is \( 2n \), we obtain a desired lower bound of (roughly) \( 2n/3 \).

Let us see the formal proof of the main theorem.

**Proof of Theorem 3.** Fix an integer \( n \geq 4 \) and a stable packing \( P_n \) for an \( n \times n \) box. We obtain \( \mathcal{P} \) by applying Algorithm 1 for \( P_n \), and then define \( Q_4 \), \( Q_3 \), and \( Q_{\leq 2} \).

By the definition, we have \( |\mathcal{P}| = |Q_4| + |Q_3| + |Q_{\leq 2}| + 1. \) We can bound \( M(|\mathcal{P}|) \) from above by

\[
M(|\mathcal{P}|) \leq 4|Q_4| + 3|Q_3| + 2|Q_{\leq 2}| + 5 \cdot 1
= 3(|Q_4| + |Q_3| + |Q_{\leq 2}| + 1)
+ (|Q_4| - |Q_{\leq 2}|) + 2
= 3|\mathcal{P}| + (2 + |Q_4| - |Q_{\leq 2}|).
\]

By recalling that \( M(|\mathcal{P}|) = 2n \), this is equivalent to

\[
\frac{2}{3} n - \frac{2}{3} + \frac{|Q_4| - |Q_{\leq 2}|}{3} \leq |\mathcal{P}|.
\]

By applying Lemma 10, we have

\[
\frac{2}{3} n - 1 \leq |\mathcal{P}| \leq F(n),
\]

which completes the proof of the theorem. \( \square \)

5. Concluding Remarks

In this paper, we investigated the anti-slide packings for a square box of side length \( n \) using T-tetrominoes andproved that the number of pieces in a sparsest packing solution \( F(n) = \lfloor 2n/3 \rfloor \) when \( n \neq 0 \) (mod 3) and \( 2n/3 - 1 \leq F(n) \leq n-1 \) when \( n \equiv 0 \) (mod 3). There is still a substantial gap between the upper and lower bounds for the case that the side length is divisible by 3. We believe that the upper bound is in fact tight, although we have not succeeded in proving this.

One may wonder whether a diagonal packing would be possible even for the case \( n \equiv 0 \) (mod 3). However, a simple observation of the packings shown in Fig. 5 reveals that this is not the case. We can observe that, for every \( k = 1, 2, \ldots \), the sum of the largest \( x \)-coordinate and the largest \( y \)-coordinate of a cell occupied by the \( k \)-th piece, counting from the bottom left corner, in a diagonal packing is not divisible by 3. We conjecture that an extra \( n/3 \) pieces are needed to prevent all of these pieces from sliding. To show this formally would be an interesting future work.

Acknowledgements

This research was partially supported by JSPS Grant Numbers 18K11152 and 18H04090. We would like to thank Editage (www.editage.com) for English language editing.
References

[1] E.D. Demaine, Y. Okamoto, R. Uehara, and Y. Uno, “Computational complexity and an integer programming model of Shakashaka,” IEICE Trans. Fundamentals, vol.E97-A, no.6, pp.1213–1219, June 2014.

[2] Z. Abel, E.D. Demaine, M.L. Demaine, S. Eisenstat, J. Lynch, and T.B. Schardl, “Finding a Hamiltonian path in a cube with specified turns is hard,” J. Inf. Process., vol.21, no.3, pp.368–377, 2013.

[3] K. Haraguchi and H. Ono, “BlockSum is NP-complete,” IEICE Trans. Inf. & Syst., vol.E96-D, no.3, pp.481–488, March 2013.

[4] R.A. Hearn and E.D. Demaine, Games, Puzzles and Computation, A.K. Peters Ltd., 2009.

[5] “Anti-Slide,” http://www.johnrausch.com/PuzzleWorld/puz/anti_slide.htm, accessed April 8, 2020.

[6] K. Amano, S. Nakano, and K. Yamazaki, “Anti-slide,” J. Inf. Process., vol.23, no.3, pp.252–257, 2015.

[7] K. Kimura, K. Amano, and T. Araki, “The analysis of anti-slide (in Japanese),” IEICE Technical Report, vol.119, no.340, COMP2019-43, pp.101–107, 2019.

[8] Y. Takenaga, X. Yang, and A. Inada, “Anti-slide placements of pentominoes,” The Japan Conf. on Discrete and Comput. Geometry, Graphs, and Games, pp.121–122, 2019.

[9] “Gurobi Optimizer,” https://www.gurobi.com/, accessed April 8, 2020.

Kento Kimura received his M.E. degree in electronics and informatics, mathematics, and physics from Gunma University in 2019. Presently, he is a Ph.D. student in the Department of Electronics and Informatics, Mathematics, and Physics, Gunma University, Kiryu, Japan. His research interests include discrete mathematics.

Kazuyuki Amano received his M.E. and D.E. degrees in information sciences from Tohoku University in 1993 and 1996, respectively. Presently, he is a professor in the Department of Computer Science, Gunma University, Kiryu, Japan. His research interests include the theory of computation and combinatorics. He enjoys writing computer programs to solve puzzles and problems of discrete mathematics.

Tetsuya Araki received his B.E., M.E., and Ph.D. degrees in electrical and electronic engineering from Kobe University, Kobe, Japan, in 2008, 2010, and 2013, respectively. He was a researcher at the National Institute of Informatics from 2013 to 2017 and a research assistant professor at Tokyo Metropolitan University from 2017 to 2019. He is currently an assistant professor in the Department of Computer Science, Gunma University, Kiryu, Japan. His research interests are graph algorithms, social networks, and data mining.