PackIt!: Gamified Rectangle Packing

Thomas Garrison
Carnegie Mellon University, Pittsburgh, PA, USA

Marijn J. H. Heule
Carnegie Mellon University, Pittsburgh, PA, USA

Bernardo Subercaseaux
Carnegie Mellon University, Pittsburgh, PA, USA

Abstract

We present and analyze PackIt!, a turn-based game consisting of packing rectangles on an $n \times n$ grid. PackIt! can be easily played on paper, either as a competitive two-player game or in *solitaire* fashion. On the t-th turn, a rectangle of area t or $t+1$ must be placed in the grid. In the two-player format of PackIt! whichever player places a rectangle last wins, whereas the goal in the solitaire variant is to perfectly pack the $n \times n$ grid. We analyze necessary conditions for the existence of a perfect packing over $n \times n$, then present an automated reasoning approach that allows finding perfect games of PackIt! up to $n = 50$ which includes a novel SAT-encoding technique of independent interest, and conclude by proving an NP-hardness result.

2012 ACM Subject Classification Mathematics of computing → Combinatorics; Theory of computation → Discrete optimization

Keywords and phrases PackIt!, rectangle packing, SAT, NP-hardness

Digital Object Identifier 10.4230/LIPIcs.FUN.2024.14

Related Version *Full Version*: https://arxiv.org/abs/2403.12195

Supplementary Material
Software (Code associated to the paper): https://github.com/bsubercaseaux/packit
archived at swh:1:dir:4b7a9bb37e64301305bf01082703a27c23cae84f

Funding The authors are partially supported by the National Science Foundation (NSF) grant CCF-2108521.

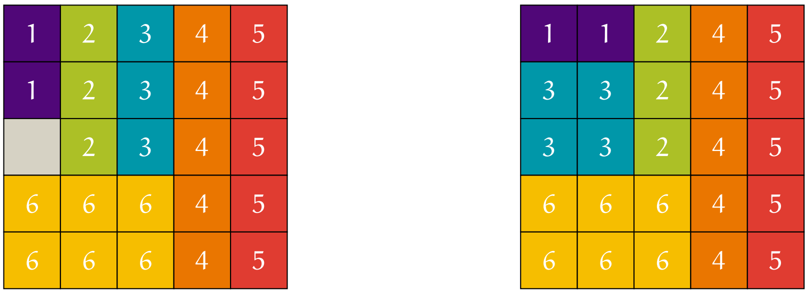
Acknowledgements We thank FUN2024 reviewers for their feedback and suggestions. We also thank Richard Green, for his comments and his blog post about our paper. The last author thanks Abigail Kamenenetsky for her help with a web implementation of the game.

1 Introduction

Pen-and-paper games have not only stimulated bored high school students for centuries, but also attracted the attention of mathematicians and computer scientists alike. From Tic-Tac-Toe to Conway's *Sprouts* [10], passing through *Dots and Boxes* [6], Sudoku, Hangman [1], and Nim [4], simple pen-and-paper games have had a long lasting impact in combinatorial game theory (e.g., the Sprague-Grundy theorem) and have offered landmark computational challenges (e.g., Sudokus require 17 clues to have a unique solution [13]). In this paper we introduce a new pen-and-paper game, PackIt!, and explore both mathematical and computational challenges concerning it.

1.1 Definition of PackIt!

The game proceeds by turns, and takes place over an $n \times n$ grid that we shall denote G. The main principle of PackIt! is very simple: on turn t (starting from 1), a rectangle r_t of area t or $t+1$ must be placed into G without intersecting any of the already placed rectangles.



(a) An *imperfect* game of PACKIT!. **(b)** A *perfect* game of PACKIT!.

Figure 1 Illustration of a couple of games of PACKIT!. Each rectangle a_t is labeled with t and depicted in a different color.

Formally, at the beginning of the game one defines the set of *used cells of the grid* as $U_0 := \varnothing$. On turn t, the corresponding player chooses $r_t = (h_t, v_t, x_t, y_t)$, with $h_t \cdot v_t \in \{t, t+1\}$, and $0 \leq x_t, y_t < n$. Define the cells used by this rectangle as the set

$$A_t := \{x_t, x_{t+1}, \ldots, x_{t+h_t-1}\} \times \{y_t, y_{t+1}, \ldots, y_{t+v_t-1}\},$$

so that the requirement for a valid turn is that $A_t \cap U_{t-1} = \varnothing$. After a valid turn, one sets $U_t := U_{t-1} \cup A_t$. Figure 1 illustrates some examples.

PackIt! as a game. PACKIT! can be played as a *solitaire* game, where the goal of the game is to complete a *perfect packing*, that is, to play so that after a valid sequence of turns it holds that $U_t = \{0, \ldots, n-1\} \times \{0, \ldots, n-1\}$. As depicted in Figure 1, we say the final board of such a game corresponds to a *perfect game of PACKIT!*. For two players, it suffices to alternate turns and when a player cannot play a valid turn, he or she is declared the *loser*. At this point, we suggest the reader to directly experiment with PACKIT!. A version of the game is available for *solitaire* mode at `https://packit.surge.sh`.

Organization. The main question about PACKIT! is:

for which values of n the $n \times n$ grid admits a perfect game of PACKIT!?

Section 2 presents arithmetic results that represent the initial steps toward answering this question. Then, Section 3 discusses the complexity of PACKIT!, showing that a particular version of the solitaire game is NP-hard. Finally, Section 4 is devoted to analyzing this question from a computational perspective. We present an initial backtracking implementation, which is then improved by a more complicated approach leveraging a novel SAT encoding.

2 Arithmetic Results

A perfect game of PACKIT! can be conceptually divided into two aspects:

- **(Rectangle selection)** As we denote by $|A_t|$ the area of the rectangle used in turn t, it must hold that in a perfect game of PACKIT! we have

  $$\sum_t |A_t| = n^2.$$

  Moreover, in order to fit every rectangle r_t of dimensions $h_t \times v_t$, it must hold that $\max(h_t, v_t) \leq n$. We will say that such a sequence of choices is *a valid rectangle sequence*.
- **(Packing Aspect)** Even if a sequence of area choices is valid, it can be the case that it is not possible to use such area choices in a perfect game of PACKIT!.

This section focuses on studying perfect games through the lens of the first aspect, as it is sometimes enough to determine the *tileability/untileability* of grids. Despite PackIt! being originally defined for a square grid, from now on we consider $m \times n$ grids as most of our ideas generalize nicely in that setting. Without loss of generality we will assume $n \geq m$ throughout the paper.

In order to state our results, we will need a couple of definitions. We denote by T_k the k-th *triangle number*, defined as $T_k = \sum_{i=1}^{k} i = \frac{k(k+1)}{2}$. Then, for any positive integer r, we denote by $\tau(r) = \arg\max_k \{T_k \mid T_k \leq r\}$.

An initial observation to understand whether an $m \times n$ grid admits a perfect packing is that the number of rectangles used in perfect PackIt! games depends entirely on the grid area $m \cdot n$, and not on its precise width or height

▶ **Lemma 1.** *For an $m \times n$ grid there is a unique number $K(m, n)$ such that if the $m \times n$ grid admits a perfect PackIt! game, then such a packing must use exactly $K(m, n)$ rectangles. In particular, $K(m, n) = \tau(m \cdot n)$.*

Proof. Assume, expecting a contradiction that for some $m \times n$ grid there are two sequences $A := (|A_1|, \dots, |A_{K_1}|)$ and $A' := (|A'_1|, \dots, |A'_{K_2}|)$, with $K_1 \neq K_2$, that can be used for perfect packings. Now, note that we must have

$$\sum_{t=1}^{K_1} |A_t| = m \cdot n = \sum_{t=1}^{K_2} |A'_t|. \tag{1}$$

By the game rules, we have that

$$\sum_{t=1}^{K_1} |A_t| \geq \sum_{t=1}^{K_1} t = T_{K_1}, \qquad \text{and} \qquad \sum_{t=1}^{K_1} |A_t| \leq \sum_{t=1}^{K_1} (t+1) = T_{K_1+1} - 1.$$

Using the same analysis for A', and Equation (1), we get

$$\max(T_{K_1}, T_{K_2}) \leq m \cdot n \leq \min(T_{K_1+1}, T_{K_2+1}) - 1.$$

As $K_1 \neq K_2$, let us assume without loss of generality that $K_1 > K_2$. Using that T is an increasing sequence, we have

$$T_{K_1} \leq m \cdot n \leq T_{K_2+1} - 1. \tag{2}$$

Now, as K_1 is an integer, $K_1 > K_2$ implies $K_1 \geq K_2 + 1$, from where Equation (2) becomes $T_{K_1} \leq m \cdot n \leq T_{K_1} - 1$, a clear contradiction. To obtain the second part of the lemma, note that when $K(m, n) := K_1 = K_2$ we get

$$T_{K(m,n)} \leq m \cdot n \leq T_{K(m,n)+1} - 1,$$

from where it follows by the definition of τ that $K(m, n) = \tau(m \cdot n)$. ◀

We can now define the notion of *gap*, which intuitively represents the number of turns t in which a rectangle of area $t + 1$ must be chosen. Let us say that any turn t at which a rectangle of area $t + 1$ is chosen is an *expansion turn*.

▶ **Definition 2.** *For any $m \times n$ grid, we define its* gap, *$\gamma(m, n)$, as*

$$\gamma(m, n) = m \cdot n - T_{\tau(m \cdot n)}.$$

▶ **Lemma 3.** *For any sequence of turns that results in a perfect packing of an $m \times n$ grid, the number of expansion turns is exactly $\gamma(m, n)$.*

Proof. By Lemma 1, there must be exactly $K(m, n) = \tau(m \cdot n)$ turns in such a sequence. If for every turn $t \in \{1, \ldots, \tau(m \cdot n)\}$, a rectangle of area t were to be chosen, then the total area used would be exactly

$$\sum_{t=1}^{\tau(m \cdot n)} t = T_{\tau(m \cdot n)}.$$

Given that the total area used must be $m \cdot n$, we conclude there must be exactly $m \cdot n - T_{\tau(m \cdot n)}$ expansion turns. ◀

The next ingredient to analyze whether an $m \times n$ grid admits a perfect packing has to do with prime numbers, as if the area of a rectangle is a prime number p, then the only possibles rectangles are $p \times 1$ or $1 \times p$, which can limit our ability to pack it. We define the set $P(m, n)$ as

$$P(m, n) = \{p \mid n < p \leq K(m, n) \text{ and } p \text{ is prime}\}.$$

As the next results show, the comparison between the *gap* of a grid and the size of its corresponding P set plays a crucial role in understanding whether or not it allows a perfect packing. In particular, Theorem 4 shows how small gaps can forbid perfect packings, whereas Theorem 5 shows how large gaps can also be problematic.

▶ **Theorem 4** (Small gap). *For any $m \times n$ grid, if $\gamma(m, n) < |P(m, n)|$, then the grid does not allow a perfect game of* PackIt!*.*

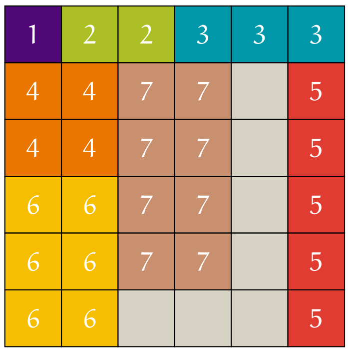


Figure 2 Illustration of the impossibility result for $n = 6$ resulting from Theorem 4. Even though turns 1 through 6 use the minimal possible area, the choice of area 8 on turn 7 is enough to make turn 9 possible, as only 8 empty cells remain (which is invariant under the concrete choice of packing).

Before a formal proof, let us present some intuition. Theorem 4 considers a gap that is *"too small"*, as the following example shows. Consider $m = n = 6$. One can easily check that, $\tau(6 \cdot 6) = 8$[1], and therefore the gap results in

$$\gamma(m, n) = m \cdot n - T_{\tau(m \cdot n)} = 6 \cdot 6 - \frac{8 \cdot 9}{2} = 0.$$

[1] A general formula for $\tau(r)$ is not too hard to derive. In particular, $\tau(r) = \left\lfloor \frac{\sqrt{8r+1}}{2} - \frac{1}{2} \right\rfloor$.

Then, $K(m,n) = \tau(m \cdot n) = 8$, and thus $P(m,n) = \{7\}$. As $K(m,n) = 8$, any perfect packing of the 6×6 grid will consist of 8 rectangles. We claim that in turn 7, the area chosen must be 7, or in other words, that choosing a rectangle of area 8 in turn 7 would forbid a perfect packing. Too see this, consider expecting a contradiction that a rectangle of area 8 is chosen on turn 7, and notice that then on the first 8 turns the smallest sum of areas we can achieve would be

$$1 + 2 + 3 + 4 + 5 + 6 + 8 + 8 = 37 > 36,$$

a contradiction. On the other hand, given 7 is a prime number, the only rectangles of area 7 are a 1×7 or a 7×1 rectangle, neither of which can be packed into a 6×6 grid. As either area choice for turn 7 leads to a contradiction, we conclude it is not possible to have a perfect game of PackIt! over the 6×6 grid. This example is illustrated in Figure 2, and is generalized in the next proof.

Proof of Theorem 4. Let $p \in P(m,n)$. At turn p, one must choose between area p or area $p+1$. If area p is chosen, then the rectangle must be either $1 \times p$ or $p \times 1$, due to the primality of p. However, by the definition of the set $P(m,n)$ we have $p > n \geq m$, and thus neither the $1 \times p$ nor the $p \times 1$ rectangle can be packed into the $m \times n$ grid. Assume, expecting a contradiction, that $\gamma(m,n) < |P(m,n)|$ and there exists a sequence of turns leading to a perfect packing for the $m \times n$ grid. As a result of the previous argument, every turn $p \in P(m,n)$ must be an expansion turn. As the number of expansion turns is equal to $\gamma(m,n)$ by Lemma 3, we have $\gamma(m,n) \geq |P(m,n)|$, which directly contradicts the assumption. ◀

▶ **Theorem 5** (Large gap). *For any $m \times n$ grid, let $\mathbf{1}_{K_p}$ be the indicator variable corresponding to whether $K(m,n)+1$ is a prime number or not. Then, the condition*

$$\gamma(m,n) > K(m,n) - |P(m,n)| - \mathbf{1}_{K_p}$$

implies the $m \times n$ grid does not allow a perfect game of PackIt!.

Before the proof, let us present some intuition for Theorem 5. Consider $m = n = 18$ (this example is illustrated in Figure 3). As a result, $\tau(18 \cdot 18) = 24$, and therefore the gap is

$$\gamma(m,n) = m \cdot n - T_{\tau(m \cdot n)} = 18 \cdot 18 - \frac{24 \cdot 25}{2} = 24.$$

We also have $K(m,n) = \tau(m \cdot n) = 24$, implying that any perfect packing of the 18×18 grid will consist of $K(m,n) = 24$ rectangles. We claim that on turn 18, both choices of area, 18 and 19, lead to contradictions. Let us see what happens if area 18 is chosen on turn 18. In this case, even if area $t+1$ is chosen on every turn $t \neq 18$, the maximum sum of the areas we can achieve is

$$2 + 3 + \ldots + 17 + 18 + \mathbf{18} + 20 + \ldots + 25 = 323 < 324,$$

implying the 324 cells of the 18×18 grid cannot be covered. On the other hand, if area 19 is chosen on turn 18, we run into a different issue: as 19 is a prime number it only allows for the rectangles 1×19 or 19×1, neither of which be can be packed into the 18×18 grid. As both cases lead to an impossibility, we conclude it is not possible to have a perfect game of PackIt! over the 18×18 grid. The proof for Theorem 5 generalizes this example.

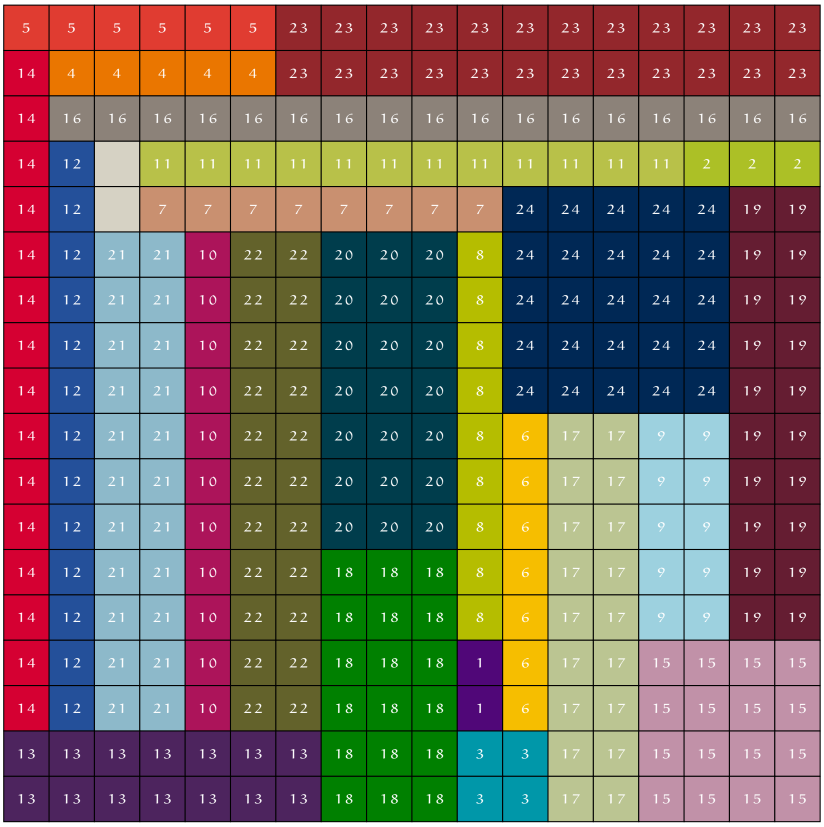


Figure 3 Illustration of the impossibility result for $n = 18$ (Theorem 5). Even though almost each rectangle t has area $t + 1$, except for $t \in \{18, 22\}$ (where $t + 1 > n$ is prime), the total area covered by turn 24 is only $322 = 18^2 - 2$, and naturally it is not possible to fill in the two remaining cells in turn 25.

Proof of Theorem 5. Let $p \in P(m, n)$. As $p \le K(m, n)$ by definition of $P(m, n)$, turn $p - 1$ is necessarily part of any perfect packing. At turn $p - 1$, one must choose between area $p - 1$ or area p. If area p is chosen, then the rectangle must be either $1 \times p$ or $p \times 1$, due to the primality of p. However, by the definition of the set $P(m, n)$ we have $p > n \ge m$, and thus neither the $1 \times p$ nor the $p \times 1$ rectangle can be packed into the $m \times n$ grid. We conclude that for each $p \in P(m, n)$, the turn $p - 1$ is not an expansion turn.

If $K(m, n) + 1$ is prime, then the rectangle turn $K(m, n)$ cannot be an expansion turn. By definition, $K(m, n) + 1 \notin P(m, n)$, so the number of turns that are not expansion turns is at least $|P(m, n)| + \mathbf{1}_{K_p}$. By Lemma 1, the number of expansion turns is exactly $\gamma(m, n)$, which together with the previous fact implies that the total number of turns is at least

$$|P(m, n)| + \mathbf{1}_{K_p} + \gamma(m, n). \tag{3}$$

Suppose, expecting a contradiction that

$$\gamma(m, n) > K(m, n) - |P(m, n)| - \mathbf{1}_{K_p}, \tag{4}$$

and yet there exists a sequence of turns leading to a perfect packing for the $m \times n$ grid. By combining Equation (3) and Equation (4), the total number of turns is at least

$$\begin{aligned}|P(m,n)| + \mathbf{1}_{K_p} + \gamma(m,n) &> |P(m,n)| + \mathbf{1}_{K_p} + \big(K(m,n) - |P(m,n)| - \mathbf{1}_{K_p}\big)\\ &= K(m,n),\end{aligned}$$

which is a contradiction, given the total number of turns must be exactly $K(m,n)$ according to Lemma 1. ◀

Combining Theorem 4 and Theorem 5, we obtain a range of values for the gap of an $m \times n$ grid in which perfect packings are *a priori* possible. So far, we have not found any examples of $m \times n$ grids whose gap belongs in this range and yet no perfect packings exist. Therefore, we pose the following conjecture

▶ **Conjecture 6.** *Let $m \leq n$ be positive integers. Then, if*

$$|P(m,n)| \;\leq\; \gamma(m,n) \;\leq\; K(m,n) - |P(m,n)| - \mathbf{1}_{K_p},$$

it is possible to complete a perfect game of PACKIT! *for the $m \times n$ grid.*

Interestingly, Theorem 4 is enough to construct infinite families of $n \times n$ grids that do not admit perfect packings.

▶ **Theorem 7.** *There are infinitely many positive integers n such that the $n \times n$ grid does not admit a perfect game of* PACKIT!.

Proof. By Theorem 4, if suffices to show that there are infinitely many values of n such that $\gamma(n,n) = 1$ and $|P(n,n)| > 1$. First, consider the following claim.

▷ Claim 8. For every $n \geq 100$, we have $K(n,n) \geq 1.4n$.

Proof of Claim 8. Let $\ell = \lfloor 1.4n \rfloor$. It suffices to argue that $T_\ell \leq n^2$. As $\ell > n \geq 100$, we have $\ell < \frac{1}{100}\ell^2$, which we can use as follows.

$$T_\ell = \frac{\ell^2 + \ell}{2} \leq \frac{\frac{101\ell^2}{100}}{2} = 101\ell^2/200,$$

and conclude by noting that

$$101\ell^2/200 \leq \frac{101}{200} \cdot \left(\frac{140}{100}n\right)^2 = \frac{1\,979\,600}{2\,000\,000}n^2 \leq n^2. \qquad ◁$$

Now, Schoenfeld proved in [15] that for every $n > 3 \cdot 10^6$, there is always a prime number between n and $\left(1 + \frac{1}{16957}\right)n$, which applied twice gives us that there are always (at least) two prime numbers between n and $\left(1 + \frac{1}{16957}\right)^2 n \leq 1.4n$. Therefore, for $n > 3 \cdot 10^6$ we always have $|P(n,n)| > 1$. It remains to prove that $\gamma(n,n) = 1$ infinitely often. We do this by using the theory of generalized Pell's equation. Indeed, the condition $\gamma(n,n) = 1$ can be written, by using notation $K := K(n,n)$, as

$$n^2 - \frac{K(K+1)}{2} = 1, \tag{5}$$

which after multiplying both sides by 8 and rearranging is equivalent to

$$8n^2 - (2K+1)^2 = 7.$$

Introducing the variable $t \coloneqq (2K+1)$ we consider the following equations.

$$t^2 - 8n^2 = -7, \tag{6}$$

$$\left(t^{(h)}\right)^2 - 8\left(n^{(h)}\right)^2 = 1. \tag{7}$$

While Equation (7) presents an *"homogeneous"* Pell equation, for which it is well known that infinitely many solutions exist over the positive integers (cf. the problem of square triangular numbers [2]), Equation (6) corresponds to a *"non-homogeneous"* equation, less frequently studied. Similarly to the theory of ordinary differential equations, we can obtain a set of solutions to the non-homogeneous equation by combining one *initial solution* for it with a set of solutions to its homogeneous counterpart. Indeed, assume the existence of a solution (n_0, t_0) to Equation (6) over the positive integers, and $\left(n_i^{(h)}, t_i^{(h)}\right)$ a sequence of solutions to Equation (7) over the positive integers, whose existence is standard (see e.g., [2]).

▷ Claim 9. The sequence (n_i, t_i), defined as

$$(n_i, t_i) \coloneqq \left(t_0 t_i^{(h)} + 8n_0 n_i^{(h)},\ t_0 n_i^{(h)} + n_0 t_i^{(h)}\right), \tag{8}$$

is an infinite family of solutions of Equation (6) over the positive integers.

Proof of Claim 9. By assumption, (n_0, t_0) is a solution of Equation (6), and $\left(n_i^{(h)}, t_i^{(h)}\right)$ is a solution of Equation (7). Thus, we have

$$\begin{aligned}
-7 &= (t_0^2 - 8n_0^2)\left(\left(t_i^{(h)}\right)^2 - 8\left(n_i^{(h)}\right)^2\right) \\
&= (t_0 + \sqrt{8}n_0)(t_0 - \sqrt{8}n_0)\left(t_i^{(h)} + \sqrt{8}n_i^{(h)}\right)\left(t_i^{(h)} - \sqrt{8}n_i^{(h)}\right) \\
&= \left[(t_0 + \sqrt{8}n_0)\left(t_i^{(h)} + \sqrt{8}n_i^{(h)}\right)\right] \cdot \left[(t_0 - \sqrt{8}n_0)\left(t_i^{(h)} - \sqrt{8}n_i^{(h)}\right)\right] \\
&= \left[\left(t_0 t_i^{(h)} + 8n_0 n_i^{(h)}\right) + \sqrt{8}\left(t_0 n_i^{(h)} + n_0 t_i^{(h)}\right)\right] \\
&\quad \cdot \left[\left(t_0 t_i^{(h)} + 8n_0 n_i^{(h)}\right) - \sqrt{8}\left(t_0 n_i^{(h)} + n_0 t_i^{(h)}\right)\right] \\
&= \left(t_0 t_i^{(h)} + 8n_0 n_i^{(h)}\right)^2 - 8\left(t_0 n_i^{(h)} + n_0 t_i^{(h)}\right)^2 \\
&= n_i^2 - 8t_i^2.
\end{aligned}$$

◀

As we can provide an initial solution $(n_0, t_0) := (11, 31)$ to Equation (6), we conclude by Claim 9 that it has infinitely many solutions over the positive integers. We now finish the proof by the following claim.

▷ Claim 10. Every solution (n_i, t_i) to Equation (6) over the positive integers with $n_i > 3 \cdot 10^6$ corresponds to a value of n such that the $n \times n$ grid does not admit a perfect game of PackIt!.

Proof of Claim 10. Let (n_i, t_i) be a solution to Equation (6) and let us argue that the $n_i \times n_i$ does not admit a perfect game of PackIt!. First, consider that t_i must be odd, as $t_i^2 = 1 + 8n_i^2$, by Equation (6). Therefore $(t_i - 1)/2$ is a positive integer. We now a argue that $(t_i - 1)/2$ indeed matches the definition of $K(n_i, n_i)$. Let us denote $(t_i - 1)/2$ by K', and we will argue that indeed $K' = K(n_i, n_i)$. To see, this, consider that as Equation (6) has the same set of solutions as Equation (5), it must be the case that

$$n_i^2 - \frac{K'(K'+1)}{2} = 1,$$

implying that $T_{K'} = n_i^2 - 1 \leq n_i^2$. Moreover, we have that

$$T_{K'+1} = T_{K'} + (K'+1) = n_i^2 + K' > n_i^2,$$

thereby confirming that $K' = \tau(n_i^2) = K(n_i, n_i)$. Taking $n := n_i$, we have by construction that $\gamma(n,n) = 1$, and as $n > 3 \cdot 10^6$ we have $|P(n,n)| > 1$. Therefore the condition of Theorem 4 applies to n, implying the $n \times n$ grid does not admit a perfect packing. This concludes the proof of the entire theorem. ◁

◀

Let us define notation $\gamma^{-1}(c)$ to denote the set $\{n \in \mathbb{N}^{>0} \mid \gamma(n,n) = c\}$. The previous proof showed that there are infinitely many values of $n \in \gamma^{-1}(1)$ that do not admit perfect packings. We now show a much stronger statement.

▶ **Theorem 11.** *For every value $c \geq 0$, only a finite number of values $n \in \gamma^{-1}(c)$ allow for a perfect packing of the $n \times n$ grid.*

Proof. By Theorem 4, it suffices to show that for every value $c \geq 0$, there are only finitely many values of n such that

$$|P(n,n)| = \{n < p \leq K(n,n) \mid p \text{ is prime}\} \leq c$$

We will do so by using the following improvement on Bertrand's postulate due to Dusart.

▶ **Proposition 12** ([8]). *For every value of $n > 3275$, there exists a prime number p such that*

$$n < p \leq n\left(1 + \frac{1}{2\ln^2 n}\right).$$

In particular, if we apply Proposition 12 exactly $c+1$ times, we obtain that

$$\left|\left\{n < p \leq n\left(1 + \frac{1}{2\ln^2 n}\right)^{c+1} \;\middle|\; p \text{ is prime}\right\}\right| \geq c+1, \quad \text{for every } n > 3275.$$

Now, let us see that for every sufficiently large n it holds that

$$n\left(1 + \frac{1}{2\ln^2 n}\right)^{c+1} \leq K(n,n),$$

which will be enough to conclude. Indeed, recall that by Claim 8 we have that $K(n,n) \geq 1.4n$ for $n \geq 100$, and hence it only remains for us to show that for sufficiently large n we have

$$\left(1 + \frac{1}{2\ln^2 n}\right)^{c+1} \leq 1.4,$$

which must be true since the LHS is monotonically decreasing in n and its limit when n goes to infinity is 1. ◀

▶ **Theorem 13.** *For every even $n \geq 2$, the $2 \times \frac{n^2}{2}$ grid always admits a perfect game of* PackIt!.

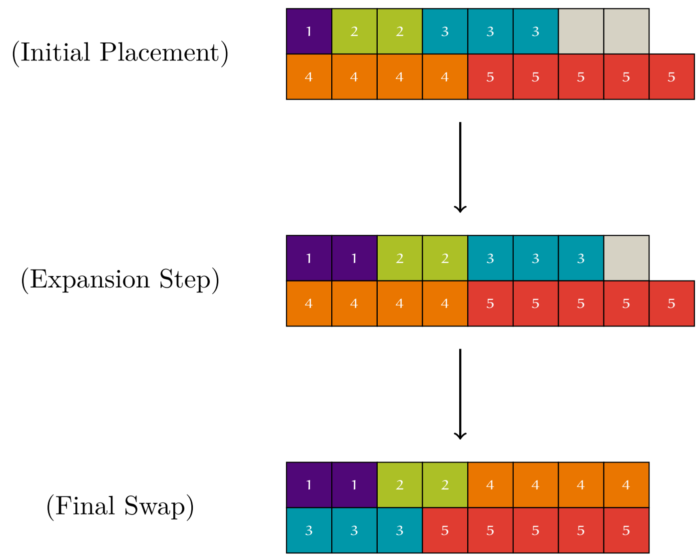


Figure 4 Illustration of Case 1 for the proof of Theorem 13, for $n = 4$. In this case $\gamma_n = 1$.

Proof. The proof is constructive. Let $K := K\left(2, \frac{n^2}{2}\right)$. As a first step, we place the first $n-1$ rectangles (i.e., $1 \times t$ for $t \in \{1, \dots, n-1\}$) in the first row, one after another, thus covering the first $\frac{n(n-1)}{2} < \frac{n^2}{2}$ cells of the first row. Some of these rectangles will be *expanded* later on in order to fill up the first row, meaning that the rectangle $1 \times t$, used in turn t will be replaced by a rectangle $1 \times (t+1)$. The remaining $K - n - 1$ rectangles, for $t \geq n$, will be placed on the second row. We might have to move some rectangles from the first row to the second row or vice-versa. The proof proceeds by cases over $\gamma\left(2, \frac{n^2}{2}\right)$, which we will abbreviate by γ_n to alleviate notation.

(Case 1: $\gamma_n \leq \frac{n}{2}$). As introduced earlier, the first step is to place the first $n-1$ rectangles in the row one and the rest in row two. For the moment we do not care if row two is too long or row one too short; we will deal with that in a moment. Next, expand the first γ_n rectangles of row one. Originally, row one was $\frac{n^2}{2} - \frac{n(n-1)}{2} = \frac{n}{2}$ cells too short, and after the expansion of the first γ_n rectangles it is $\frac{n}{2} - \gamma_n$ cells too short. By Lemma 3, the γ_n expansions in row one, guarantee that the total area of rectangles in row one and two adds up to exactly n^2. As a result, row two must be exactly $\frac{n}{2} - \gamma_n$ cells too large. If γ_n were to be exactly $\frac{n}{2}$, we would be done immediately. Otherwise, we will swap a rectangle from row one with a rectangle from row two. Indeed, note that $r_{\frac{n}{2}+\gamma_n}$, the $1 \times \frac{n}{2} + \gamma_n$ rectangle, is still on row one, and it was not expanded. Therefore, we can swap $r_{\frac{n}{2}+\gamma_n}$ (from row one) with r_n (from row two). As a result, row one has grown by $n - \left(\frac{n}{2} + \gamma_n\right) = \frac{n}{2} - \gamma_n$ cells, and row two has shrunk by the same amount. Therefore both rows have reached their desired length. This case is illustrated in Figure 4.

(Case 2: $\frac{n}{2} < \gamma_n < n - 1$). As before, placing the first $n-1$ rectangles in row one makes the first row $\frac{n}{2}$ cells too short. Then, if we place rectangles $r_n, \dots, r_K$ in row two, given that in total γ_n expansions are required to achieve the total desired area (Lemma 3), it must be the case that row two is $\gamma_n - \frac{n}{2}$ cells too short. Naively, we would simply expand $\frac{n}{2}$ rectangles in the first row, and $\gamma_n - \frac{n}{2}$ in the second row. However, the second row might contain fewer than $\gamma_n - \frac{n}{2}$ rectangles. To address this, we will transfer a rectangle from row one to row two, and perform more expansions on row one, which concentrates most of the rectangles. Let us identify which rectangle will be moved from row one to row two. Let us define

$$i = \gamma_n - \frac{n}{2}.$$

Transfer r_i from row one to row two, and expand the first $\gamma_n < n-1$ of the rectangles in row one. Since γ_n expansions have been made, the total area is exactly $\frac{n^2}{2}$, and thus it only remains to argue that the top row has exactly $\frac{n^2}{2}$ cells covered. This is indeed the case as

$$\frac{n(n-1)}{2} + \gamma_n - i = \frac{n(n-1)}{2} + \gamma_n - \left(\gamma_n - \frac{n}{2}\right) = \frac{n^2}{2}.$$

(Case 3: $\gamma_n \geq n-1$). Place the first $n-1$ rectangles in row one and the rest in row two. Expanding all $n-1$ rectangles in the first row, and then expand $\gamma_n - (n-1)$ rectangles in the second row. Let $i = \frac{n}{2} - 2$ (if $n \in \{2, 4\}$, the result can be checked manually, and therefore we assume $i \geq 1$ is a valid index for a rectangle). Move rectangle r_i from row one to row two. As in the previous case, it only remains to argue that the number of cells in the top row is exactly $\frac{n^2}{2}$. This is indeed the case as the number is determined by

$$\frac{n(n-1)}{2} + (n-1) - (i+1) = \frac{n(n-1)}{2} + (n-1) - \left(\frac{n}{2} - 1\right) = \frac{n^2}{2}.$$

Having covered all cases, we conclude the entire proof. ◀

3 Complexity Results

In turn t of a game of PackIt!, the turn in which each of the already placed rectangles was packed into the grid is irrelevant, and therefore a *partially filled grid* G of dimensions $n \times n$ can be represented as an $n \times n$ matrix over $\{0, 1\}$. We will assume this representation uses $O(n^2)$ bits. Consider now the following problem:

PROBLEM: :	SolitairePackIt!
INPUT :	A partially filled grid G, and a turn number t given in binary.
OUTPUT :	Whether it is possible to complete a perfect packing for G starting from turn t.

We will analyze the complexity of SolitairePackIt! next, but before that, let us remark that the definition of the problem does not require the partial filling of G to be achievable in $t-1$ turns. We leave the complexity of SolitairePackIt! with the additional restriction that G must be achievable in $t-1$ turns as an open problem. That being said, we can present our main complexity result.

▶ **Theorem 14.** *SolitairePackIt! is* NP*-complete.*

Proof. Let $n \times n$ be the the dimensions of G. Membership in NP is easy to see: the certificate is a description of the turns $t, ..., t+m$, where $m = K(n, n) \leq n^2$, and it suffices to check that at each turn $t+i$, a rectangle of the appropriate area was placed without overlapping with any of the previously placed rectangles. For hardness, we reduce from a variant of the well-known 3 partition problem, proven to be NP-hard by Hulett, Will and Woeginger [12]. The overall reduction is inspired by the analysis of Tetris by Breukelaar et al. [5]. Consider the Restricted-3-Partition problem defined as follows.

PROBLEM: :	Restricted-3-Partition
INPUT :	A set of integers, $\mathcal{A} = \{\alpha_1, \ldots, \alpha_n\}$, with n a multiple of 3, such that if we define $T := \frac{\sum_{i=1}^{n} \alpha_i}{(n/3)}$, then $T/4 < \alpha_i < T/2$ for every $i \in [n]$.
OUTPUT :	Whether it is possible to partition $\mathcal{A}$ into $n/3$ sets of 3 elements, all of them having sum exactly T.

Consider now the 4-Restricted-3-Partition, defined exactly as above but with the additional restriction that all numbers α_i are multiples of 4. This additional restriction preserves NP-hardness as every 3-partition P defined as

$$\{4\alpha_1, \ldots, 4\alpha_n\} \overset{P}{\longmapsto} (\{4\alpha_i, 4\alpha_j, 4\alpha_k\}, \ldots, \{4\alpha_x, 4\alpha_y, 4\alpha_z\})$$

is in a one-to-one correspondence with a 3-partition P' defined as

$$\{\alpha_1, \ldots, \alpha_n\} \overset{P'}{\longmapsto} (\{\alpha_i, \alpha_j, \alpha_j\}, \ldots, \{\alpha_x, \alpha_y, \alpha_z\}).$$

We can therefore reduce directly from 4-Restricted-3-Partition. Let $\mathcal{A}$ be an input instance of 4-Restricted-3-Partition. We now show how to construct an associated instance of SolitairePackIt!. First, we will present the required *gadgets*, which are illustrated in Figure 5.

E-gadgets. An E-gadget consists of a $T \times 3$ grid, in which the first and third column are completely filled, whereas the middle column is completely empty (hence E(mpty)-gadget). An illustration is presented in Figure 5b.

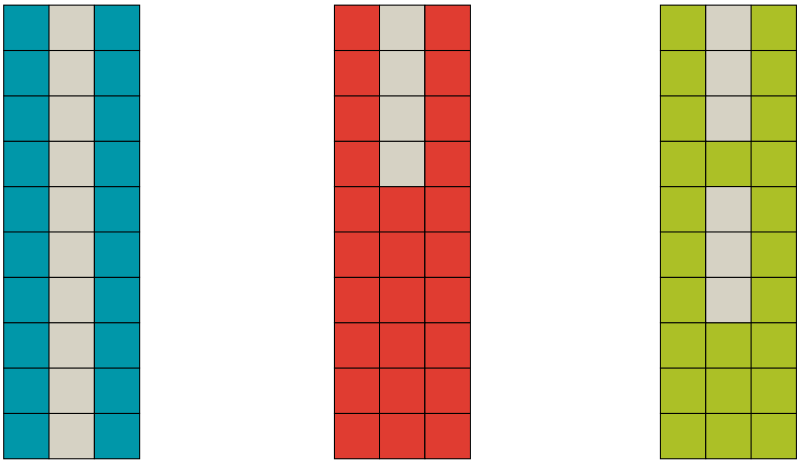
S-gadgets. Given an integer $\alpha \geq 1$, an $S(\alpha)$-gadget consists of a $T \times 3$ grid, in which the first and third column are completely filled, whereas only the bottom $T - \alpha$ rows of the middle column are filled. In other words, $S(\alpha)$-gadgets have a *single* "hole" of $\alpha \times 1$, hence their name. An illustration is presented in Figure 5b.

D-gadgets. Given an integer $\alpha \geq 1$, a $D(\alpha)$-gadget consists of a $T \times 3$ grid, in which the first and third column are completely filled, and the middle column is filled only at row $\alpha + 1$ and rows $\{2\alpha + 2, 2\alpha + 3, \ldots, T\}$. In other words, $D(\alpha)$-gadgets have two "holes" of $\alpha \times 1$, i.e., a *double* hole, hence their name. An illustration is presented in Figure 5c.

With these gadgets, we can now construct a $(T + n) \times (T + n)$ grid as follows. First, horizontally concatenate exactly $n/3$ identical E-gadgets. Next, concatenate an $S(1)$-gadget to the right of the current construction. Then, for every odd value m such that $3 \leq m < \max(\mathcal{A})$, concatenate a $D(m)$-gadget to the right of the current construction if $m - 1 \notin \mathcal{A}$, and instead an $S(m + 1)$-gadget to the right of the current construction otherwise.

Afterwards, if the resulting grid has length $T \times T'$, we complete a $T \times (T + n)$ grid by concatenating a $T \times (T + n - T')$ completely filled grid to the right of the current construction. This is well-defined, meaning that $T' < T + n$, as we show next. First, consider that, as each gadget uses exactly 3 columns, we have

$$\begin{aligned} T' &= 3 \cdot n/3 + 3 \cdot \left|\{3 \leq m < \max(\mathcal{A}) \mid m \text{ is odd}\}\right| \\ &\leq n + 3 \left\lceil \frac{\max(\mathcal{A}) - 3}{2} \right\rceil < n + 3 \frac{\max(\mathcal{A})}{2}. \end{aligned}$$



(a) An E-gadget. **(b)** An $S(4)$-gadget. **(c)** A $D(3)$-gadget.

Figure 5 Illustration of the gadgets for $T = 10$.

Next, consider that

$$T = \left(\sum_{\alpha \in \mathcal{A}} \alpha \right) / (n/3) \leq 3 \max(\mathcal{A}).$$

Then, as $\max(\mathcal{A}) \leq T/2$ by the definition of 4-Restricted-3-Partition, we have

$$T' \leq n + 3 \cdot \frac{\max(\mathcal{A})}{2} \leq n + 3 \cdot \frac{T}{4} < T + n.$$

Finally, to go from the resulting $T \times (T + n)$ grid to a $(T + n) \times (T + n)$ grid it suffices to concatenate a completely filled $n \times (T + n)$ grid at the bottom of the previous grid. This construction is illustrated in Figure 6. We are now ready to prove the correctness of our reduction. Let $G_{\mathcal{A}}$ be the $(T + n) \times (T + n)$ grid constructed by the above process.

▶ **Lemma 15.** *The instance $(G_{\mathcal{A}}, 1)$ is a Yes-instance for SolitairePackIt! if and only if $\mathcal{A}$ is a Yes-instance for 4-Restricted-3-Partition.*

Proof. ($\Longleftarrow$) Let us start with the backward direction since it is simpler. Assume there is a solution to the partition problem with sets $S_1, \dots, S_{n/3}$, where each set has exactly 3 elements and its sum is exactly T. Then, we can complete a perfect packing of G as follows. On each turn $1 \leq t \leq \max(\mathcal{A})$:

Case I) If $t \in \mathcal{A}$, then let i be the index such that $t = \alpha_i$, and j be the index of the set S_j such that $\alpha_i \in S_j$. Then, on this turn we can place a rectangle of dimensions $t \times 1$ into the j-th E-gadget of $G_{\mathcal{A}}$.

Case II) If $t = 4k$ for some positive integer k but $t \notin \mathcal{A}$, then by construction there is a $D(t+1)$-gadget, which can be filled by placing a $(t + 1) \times 1$ rectangle on this turn, and a $(t + 1) \times 1$ rectangle on the next turn.

Case III) If $t = 4k + 1$ and $t - 1 \in \mathcal{A}$, then by construction there is an $S(t+1)$-gadget, which can be filled by placing a $(t + 1) \times 1$ rectangle on this turn.

Case IV) If $t = 4k + 1$ for some integer k, and $t - 1 \notin \mathcal{A}$, then this turn has been covered in Case **II**).

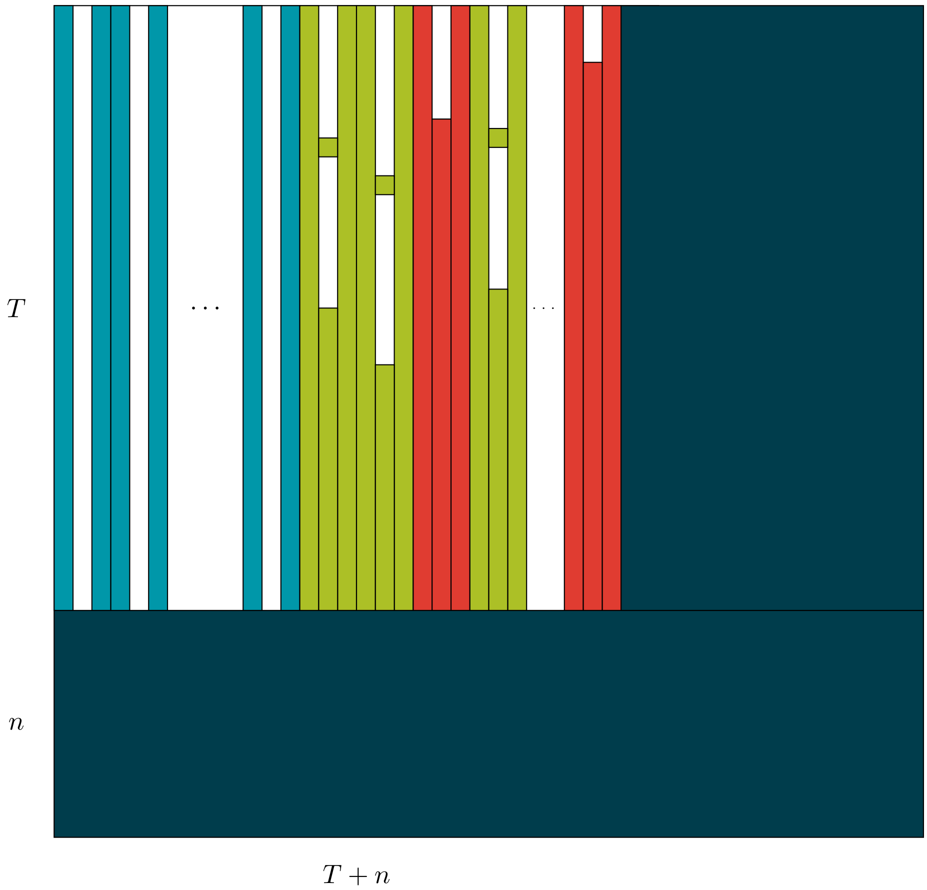


Figure 6 Illustration of the construction of $G_{\mathcal{A}}$ for Theorem 14. Note that n could be larger than T, and thus this figure is not necessarily in scale.

Case V) If $t = 4k+2$ for some integer k, then by by construction there is a $D(t+1)$-gadget, which can be filled by placing a $(t+1) \times 1$ rectangle on this turn, and a $(t+1) \times 1$ rectangle on the next turn.

Case VI) If $t = 4k+3$ then this turn has been covered in Case **V**).

As a result of the turns of Case **I**), every E-gadget will be completely filled since by definition, if $\alpha_i, \alpha_k, \alpha_\ell \in S_j$, then $\alpha_i + \alpha_k + \alpha_\ell = T$. As there are exactly $n/3$ identical E-gadgets in $G_{\mathcal{A}}$, they will all be filled. Note as well that the gadgets used in every case are different. In particular, the only S-gadgets in the construction are for $t+1 = 4k+2$ with $t-1 \in \mathcal{A}$, which are all used by Case **III**). Similarly, all $D(m)$-gadgets for $m = 4k+1$ for some integer k are used by Case **II**), whereas all $D(m)$-gadgets for $m = 4k+3$ are used by Case **V**). Given all gadgets are perfectly filled up, we have a perfect packing of $G_{\mathcal{A}}$.

($\Longrightarrow$) For the forward direction, assume it is possible to perfectly pack the grid $G_{\mathcal{A}}$ starting from turn 1. Let $G^P_{\mathcal{A}}$ be any perfect packing completing $G_{\mathcal{A}}$. Note immediately that by construction, every rectangle placed in $G^P_{\mathcal{A}}$ from turn 1 onward must have dimension $t \times 1$ for some positive integer t. Intuitively, we will now prove that the choices made in the backward direction of the proof are forced.

▶ **Definition 16.** *For any turn $t \geq 1$, we say the rectangle placed in $G^P_{\mathcal{A}}$ on turn t is* proper *if either*

1. *$t = 1$, and the rectangle placed in $G^P_{\mathcal{A}}$ on this turn was a 1×1 rectangle placed in the only $S(1)$-gadget of $G_{\mathcal{A}}$.*
2. *$t > 1$ is odd, and $t - 1 \in \mathcal{A}$, and the rectangle placed in $G^P_{\mathcal{A}}$ on this turn was a $(t+1) \times 1$ placed in the only $S(t+1)$-gadget of $G_{\mathcal{A}}$.*
3. *$t > 1$ is odd and $t - 1 \notin \mathcal{A}$, and the rectangle placed in $G^P_{\mathcal{A}}$ on this turn was a $t \times 1$ placed in one of the two spaces of the only $D(t)$-gadget of $G_{\mathcal{A}}$.*
4. *$t \in \mathcal{A}$, and the rectangle placed in $G^P_{\mathcal{A}}$ on this turn was placed in one of the E-gadgets.*
5. *t is even but $t \notin \mathcal{A}$, and the rectangle placed in $G^P_{\mathcal{A}}$ on this turn was a $(t+1) \times 1$ placed in one of the two spaces of the only $D(t+1)$-gadget of $G_{\mathcal{A}}$.*

▷ Claim 17. Every turn $t \geq 1$ where a rectangle was placed in $G^P_{\mathcal{A}}$ must have been proper.

Proof of Claim 17. We prove the claim by induction on t. The base case is $t = 1$, for which a single $S(1)$-gadget exists in the construction, and given that the 1×1 empty space in this gadget must be filled in $G^P_{\mathcal{A}}$, the only turn on which it can be filled is turn 1. Therefore the base case works. For the inductive case, assume the claim holds up to t and let us show it holds for $t + 1$.

- If $t + 1$ is odd and $t \in \mathcal{A}$, then we claim the rectangle placed on turn $t + 1$ must have been a $(t + 2) \times 1$ rectangle in the only $S(t + 2)$-gadget of $G_{\mathcal{A}}$. Indeed, if this were not the case, said gadget could only have been filled by a $(t + 2) \times 1$ rectangle placed on turn $(t + 2)$, since all previous turns have been proper and thus not placed anything in the $S(t + 2)$-gadget. However, given there are two empty spaces of size $(t + 3)$ into the only $D(t + 3)$-gadget of $G_{\mathcal{A}}$ (which must exist since $t \in \mathcal{A} \implies t + 2 \notin \mathcal{A}$ as all elements of $\mathcal{A}$ are multiples of 4), and no previous turns could have placed anything into them as they are proper by inductive hypothesis, then we conclude that on turn $(t + 2)$ a rectangle of size $(t + 3)$ must have been placed into the only $D(t + 3)$-gadget of $G_{\mathcal{A}}$.
- If $t + 1$ is odd and $t \notin \mathcal{A}$, then given all the previous turns have been proper, it must be that the only $D(t + 1)$-gadget of $G_{\mathcal{A}}$ has only received a $(t + 1) \times 1$ rectangle placed on turn t, according to (5) in the definition of proper turn. Therefore, a single $(t + 1) \times 1$ empty space remains in the only $D(t + 1)$-gadget of $G_{\mathcal{A}}$, and it must be that is filled on this turn, as any posterior turns will have rectangles of area at least $t + 2$.
- If $t + 1$ is even but $t + 1 \notin \mathcal{A}$, then given all turns so far have been proper, there are two empty $(t + 2) \times 1$ spaces in the only $D(t + 2)$-gadget of $G_{\mathcal{A}}$, and given none can be filled after turn $t + 3$, and at most one can be filled in turn $t + 2$, we conclude that turn $t + 1$ must fill one.
- If $t + 1 \in \mathcal{A}$, and this turn were to be improper, then the rectangle placed on this turn must be placed either in an $S(t')$-gadget or in a $D(t')$-gadget.
  In either case we will reach a contradiction. Note first that $t' > t + 2$: in the construction of $G_{\mathcal{A}}$, as t is odd and $t - 1 \notin \mathcal{A}$, when $m = t$ a $D(t)$-gadget was created, and the next gadget created is a $S(t + 3)$-gadget when $m = t + 2$, since $m - 1 \in \mathcal{A}$. Next, note that the remaining empty space on the $S(t')$-gadget or the $D(t')$-gadget partially filled on turn $t + 1$ must be at least $t' - (t + 2) > 0$. If $t' - (t + 2) < t + 2$, then that remaining empty space can never be filled in posterior turns, where all rectangles have area at least $t + 2$, a contradiction. Otherwise, $t' - (t + 2) > t + 1$, meaning that $t' > 2t + 3$. Because an $S(t')$-gadget or a $D(t')$-gadget exists, we deduce from the construction that $t' \leq \max(\mathcal{A})$. This implies that

  $$\max(\mathcal{A}) > t' - 1 > 2t + 2 = 2(t + 1),$$

meaning that two elements of $\mathcal{A}$, namely

$$\alpha_i := \max(\mathcal{A}), \quad \alpha_j := t+1,$$

hold $\alpha_i > 2\alpha_j$. But by definition of 4-Restricted-3-Partition that would imply the following contradiction:

$$T/4 < \alpha_j < \alpha_i/2 < (T/2)/2 = T/4. \qquad \triangleleft$$

By Claim 17, we have that for every $\alpha_t \in \mathcal{A}$, a rectangle of area α_t has been placed inside an E-gadget. Given that $T/2 < \alpha_t < T/4$ for every t, there must be exactly 3 rectangles placed inside every E-gadget. Let $\alpha_i^{(1)}, \alpha_i^{(2)}, \alpha_i^{(3)}$ be the areas of the three rectangles placed inside the i-th E-gadget. As for every i, by hypothesis, the i-th E-gadget is perfectly filled and had t empty cells to be filled, we conclude that that $\alpha_i^{(1)} + \alpha_i^{(2)} + \alpha_i^{(3)} = T$, from where it follows that $\mathcal{A}$ is a Yes-instance to the 4-Restricted-3-Partition problem. This concludes the proof of Lemma 15. ◀

Given the reduction presented above can clearly be carried out in polynomial time, we conclude hardness from the correctness proved in Lemma 15, and consequently this finishes the entire proof of Theorem 14. ◀

4 Computing Perfect PackIt! games

Even though Theorem 14 does not directly imply that it is hard to find perfect packings for an $n \times n$ grid (or to decide whether such a packing exist), it arguably gives evidence for this being a hard combinatorial challenge.

In many combinatorial problems SAT-solving can dramatically outperform backtracking approaches. This also happens to be the case for computing perfect PackIt! games, where even after several optimizations, a backtracking approach only allowed us to find perfect packings up to $n = 20$. In contrast, by using a novel SAT encoding technique we were able to find perfect packings up to $n = 50$ in under 24 hours of computation. As in Section 2, we divide the problem into two stages: (i) finding a set of rectangles (h_t, v_t) such that

- Their total area is n^2, meaning that $\sum_t h_t \cdot v_t = n^2$.
- The t-th rectangle has area t or $t+1$, meaning that $h_t \cdot v_t \in \{t, t+1\}$ for every t.
- All rectangles fit into the $n \times n$ grid, meaning that $\max(h_t, v_t) \leq n$.

and (ii), packing the rectangles obtained in the previous stage without overlaps. Note that due to the area condition, if a valid rectangle selection is packed without overlapping, then they must cover the entire $n \times n$ grid.

For stage (i), we use a pseudo-polynomial dynamic programming approach, similar to the one used for the standard subset sum problem. For stage (ii) we use a sophisticated SAT encoding that uses only $O(n^3)$ many clauses as opposed to the naive $O(n^4)$ encoding. Due to space constraints, both the dynamic programming formulation and the SAT encoding is presented in the extended arXiv version of this paper, at `https://arxiv.org/abs/2403.12195`.

4.1 Computational Results

All experiments have been run on a personal computer with the following specifications:

- MacBook Pro M1, 2020, running Sonoma 14.3
- 16GB of RAM
- 8 cores (but all experiments were run in a single thread).

In terms of software, we experimented with different SAT-solvers, and obtained the best results using the award-winning solver Kissat [3]. We tested every value of n between 5 and 50 and such that neither Theorem 4 nor Theorem 5 applies, and for every value we were able to find a perfect game of PackIt! in under 24 hours. For each such value, we used the dynamic programming approach to generate a valid selection of rectangles, and simply used the first one obtained. Given the number of valid selections of rectangles is likely exponential in n, it could be that some valid selections are significantly easier to pack than others. The fact that we obtained perfect packings simply using the first valid rectangle selection obtained via dynamic programming confirms the robustness of the SAT approach.

Detailed results are presented in Table 1. As it is common for families of satisfiable formulas, the runtime is not strictly monotone with n, even though the size of the encoding is (both the number of variables and clauses).

Table 1 Computational results for $n \in \{5, \ldots, 50\}$. Perfect packings for $n \in \{1, \ldots, 4\}$ are trivial.

n	#vars	#clauses	SAT runtime
5	141	424	0.0s
(Theorem 4 applies) 6	-	-	-
7	297	1101	0.0s
8	375	1482	0.0s
9	510	2228	0.02s
10	611	2797	0.02s
11	780	3921	0.02s
12	904	4732	0.03s
13	1037	5673	0.19s
14	1254	7375	0.16s
15	1410	8584	0.04s
16	1661	10838	0.56s
17	1840	12397	0.20s
(Theorem 5 applies) 18	-	-	-
19	2327	17184	0.20s
20	2538	19339	2.47s
21	2871	23037	2.08s
22	3105	25582	2.04s
(Theorem 4 applies) 23	-	-	-
24	3729	33117	4.43s
25	3995	36396	2.80s
26	4410	41980	2.69s
27	4699	45737	23.21s
28	5148	52283	8.45s
29	5460	56636	17.24s
(Theorem 5 applies) 30	-	-	-
31	6278	69109	34.26s
32	6622	74340	48.17s
33	7153	83288	36.37s
34	7520	89207	107.23s
(Theorem 4 applies) 35	-	-	-
36	8475	105934	747.46s
37	8874	112997	194.33s
38	9487	124629	502.20s
39	9909	132324	442.62s
40	10556	145392	129.71s
41	11001	153969	6117.58s
42	11455	162890	2088.45s
43	12150	177744	923.03s
44	12627	187501	579.50s
45	13356	203857	3185.11s
46	13856	214540	2188.39s
(Theorem 5 applies) 47	-	-	-
48	15142	244107	48102.44s
49	15674	256188	23337.97s
50	16485	276182	15925.77s

5 Concluding Remarks

We have analyzed several aspects of PackIt!:

1. Every $2 \times \frac{n^2}{2}$ grid admits a perfect PackIt! game.
2. For every $n \leq 50$ such that neither Theorem 4 nor Theorem 5 applies, the $n \times n$ grid admits a perfect PackIt! game. In other words, Conjecture 6 is true for all values of $n \leq 50$.

We hope that both our mathematical and computational techniques can be applicable to similar packing problems. The *"Mondrian Art Puzzle"* [9, 14] asks for perfect packings of $n \times n$ grids but where all rectangles must use the same area. Recently, the *MIT CompGeom Group* has studied perfect packings for rectangular grids with square pieces [11]. Then, in terms of concrete PackIt! questions, we pose the following challenges:

1. Prove or refute Conjecture 6.
2. Is there always a perfect packing of the $m \times n$ grid when $\gamma(m, n) = K(m, n)/2$? In this case, exactly half of the turns are expansion turns. In particular, this might be easier to show assuming m and n are even.
3. What is the complexity of PackIt! as a 2-player game? It is well known that complexity tends to increase in 2-player formulations (see e.g., [7]), so could PackIt! be complete for the class PSPACE?

In terms of our web implementation of PackIt!, future work includes the design of an online multiplayer mode, and AIs that could be faced as opponents.

References

1 Jérémy Barbay and Bernardo Subercaseaux. The Computational Complexity of Evil Hangman. In Martin Farach-Colton, Giuseppe Prencipe, and Ryuhei Uehara, editors, *10th International Conference on Fun with Algorithms (FUN 2021)*, volume 157 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 23:1–23:12, Dagstuhl, Germany, 2020. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.FUN.2021.23.

2 Edward J Barbeau. *Pell's Equation*. Problem Books in Mathematics. Springer, New York, NY, 2003 edition, January 2003.

3 Armin Biere, Katalin Fazekas, Mathias Fleury, and Maximillian Heisinger. CaDiCaL, Kissat, Paracooba, Plingeling and Treengeling Entering the SAT Competition 2020. In Tomas Balyo, Nils Froleyks, Marijn Heule, Markus Iser, Matti Järvisalo, and Martin Suda, editors, *Proc. of SAT Competition 2020 – Solver and Benchmark Descriptions*, volume B-2020-1 of *Department of Computer Science Report Series B*, pages 51–53. University of Helsinki, 2020.

4 Charles L. Bouton. Nim, A Game with a Complete Mathematical Theory. *Annals of Mathematics*, 3(1/4):35–39, 1901. doi:10.2307/1967631.

5 Ron Breukelaar, Erik D. Demaine, Susan Hohenberger, Hendrik Jan Hoogeboom, Walter A. Kosters, and Dadvid Liben-Nowell. Tetris is Hard, Even to Approximate. *International Journal of Computational Geometry & Applications*, 14:41–68, April 2004. doi:10.1142/S0218195904001354.

6 Kevin Buchin, Mart Hagedoorn, Irina Kostitsyna, and Max van Mulken. Dots & boxes is PSPACE-complete, 2021. arXiv:2105.02837.

7 Erik. D Demaine, William Gasarch, and Mohammad Hajiaghayi. Computational Intractability: A Guide to Algorithmic Lower Bounds. https://hardness.mit.edu/.

8 Pierre Dusart. *Autour de la fonction qui compte le nombre de nombres premiers*. PhD thesis, Université de Limoges, 1998. Thèse de doctorat dirigée par Robin, Guy Mathématiques appliquées. Théorie des nombres Limoges 1998. URL: http://www.theses.fr/1998LIMO0007.

9 Natalia García-Colín, Dimitri Leemans, Mia Müßig, and Érika Roldán. There is no perfect mondrian partition for squares of side lengths less than 1001. *arXiv preprint arXiv:2311.02385*, 2023.

10 Martin Gardner. Mathematical games. *Scientific American*, 223(4):120–123, October 1970. `doi:10.1038/scientificamerican1070-120`.

11 MIT CompGeom Group, Zachary Abel, Hugo A. Akitaya, Erik D. Demaine, Adam C. Hesterberg, and Jayson Lynch. When can you tile an integer rectangle with integer squares?, 2023. `arXiv:2308.15317`.

12 Heather Hulett, Todd G. Will, and Gerhard J. Woeginger. Multigraph realizations of degree sequences: Maximization is easy, minimization is hard. *Operations Research Letters*, 36(5):594–596, September 2008. `doi:10.1016/j.orl.2008.05.004`.

13 Gary McGuire, Bastian Tugemann, and Gilles Civario. There is no 16-clue sudoku: Solving the sudoku minimum number of clues problem, 2013. `arXiv:1201.0749`.

14 Cooper O'Kuhn. The mondrian puzzle: A connection to number theory, 2018. `arXiv:1810.04585`.

15 Lowell Schoenfeld. Sharper Bounds for the Chebyshev Functions $\theta(x)$ and $\psi(x)$. II. *Mathematics of Computation*, 30(134):337–360, 1976. `doi:10.2307/2005976`.