Discrete Applied Mathematics 162 (2014) 32–41

Contents lists available at ScienceDirect

Discrete Applied Mathematics

journal homepage: www.elsevier.com/locate/dam

More on the one-dimensional sliding-coin puzzle

Ting-Yu Lin, Shi-Chun Tsai *, Wen-Nung Tsai, Jong-Chuang Tsay

Department of Computer Science, National Chiao Tung University, Hsinchu 30050, Taiwan

A R T I C L E I N F O

Article history:
Received 23 May 2013
Accepted 14 August 2013
Available online 10 September 2013

Keywords:
Combinatorial games
Puzzles
Algorithms

A B S T R A C T

Consider a line of n nickels and n pennies with all nickels arranged to the left of all pennies, where $n \geq 3$. The puzzle asks the player to rearrange the coins such that nickels and pennies alternate in the line. In each move, the player is allowed to slide k adjacent coins to new positions without rotating. We first prove that for any integer $k \geq 2$ it takes at least n moves to achieve the goal. A well-known optimal solution for the case $k = 2$ matches the lower bound. We also give optimal solutions for the case $k = 3$.

1. Introduction

1.1. Preliminary: an old puzzle for $k = 2$

The sliding-coin puzzle is a solitaire game, which has numerous variants. The puzzle is defined with an initial configuration and a final configuration, and is associated with a moving rule. Consider the configuration on the left of Fig. 1, with three nickels to the left of three pennies. In this paper, we use a white stone and a black stone to denote a nickel and a penny, respectively. The player is allowed to slide a pair of adjacent coins to unoccupied positions in each move. The goal is to reach one of the configurations on the right of Fig. 1, such that black stones and white stones alternate in the line. The final configuration may not completely stay within the initial positions along the line, i.e. it may shift to the left or right as a whole.

The above puzzle can be solved in three moves, as shown in Fig. 2. We write $(i \rightarrow j)$ to denote a move such that two adjacent coins at positions i and $i + 1$ are moved to positions j and $j + 1$, respectively. Note that the player cannot rotate the pair of coins while sliding.

Since all the coins are arranged in the line, we call this puzzle the *one-dimensional sliding-coin puzzle*. In fact, this is an old puzzle. It has been illustrated as "Sheep and goats" [7], and as "empty and filled wine glasses" [4]. There is a very nice solution for $k = 2$ and general n shown in Kordemsky's famous puzzle book[6]. However, there was no clue about the optimality. We show that the solution is optimal.

Actually, there are many sliding-coin puzzles whose coin configurations are arranged on a plane. For two-dimensional sliding-coin puzzles, we refer to [3,5,2,1,8].

Unlike two-dimensional sliding-coin puzzles, the one-dimensional puzzle can be easily extended by adding the same number of nickels and pennies to the configuration or allowing the player to slide more coins in each move. These generalizations form a family of puzzles, and thus we can investigate the complexity of achieving an optimal solution. In this paper, we show a lower bound and some properties of optimal solutions, and explicitly solve the puzzle family optimally for

* Corresponding author. Tel.: +886 3 5131551.
E-mail addresses: linty@csie.nctu.edu.tw (T.-Y. Lin), sctsai@cs.nctu.edu.tw, sctsai@csie.nctu.edu.tw (S.-C. Tsai), tsaiwn@csie.nctu.edu.tw (W.-N. Tsai), jctsay@csie.nctu.edu.tw (J.-C. Tsay).

0166-218X/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.dam.2013.08.013

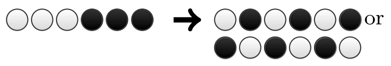


Fig. 1. Rearrange the configuration by sliding a pair of adjacent coins in each move.

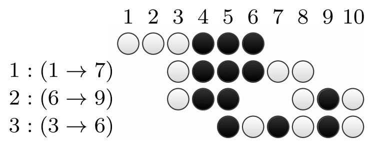


Fig. 2. A three-move solution to the puzzle for $n = 3$.

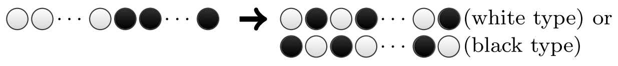


Fig. 3. Rearrange the left configuration into the right configuration by sliding k adjacent stones in each move.

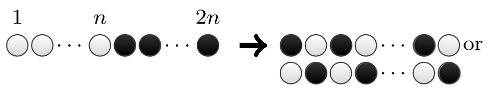


Fig. 4. The initial configuration has one pair of adjacent positions with different colors. The final configuration has $2n - 1$ pairs of adjacent positions with different colors.

$k = 3$. To the best of our knowledge, the one-dimensional sliding-coin puzzle for $k = 3$ has not been studied before. With similar ideas, we can solve optimally the puzzles for $k = 4$ and 5. For larger k, we leave the solution as an open question.

1.2. Definition and notions

For convenience, we give the following definitions.

Definition 1 (*$(i \to j)$-Move*)**.** An $(i \to j)$-move slides k adjacent stones, without rotating, from source positions $i, i + 1, \ldots, i + k - 1$ to destination positions $j, j + 1, \ldots, j + k - 1$, respectively.

The initial configuration is as shown on the left of the arrow in Fig. 3, with n white stones to the left of n black stones for $n \geq 3$. The final configuration is called *white type* if the first stone is white; otherwise, it is *black type*. We want to rearrange the initial configuration into one of the final configurations via a sequence of valid moves.

Definition 2 (*(n, k)-Game*)**.** Given integers $k \geq 2$ and $n \geq 3$, we call the corresponding sliding coin puzzle the (n, k)-game.

For example, the puzzle shown in Fig. 1 is a $(3, 2)$-game. We say that a stone becomes *stationary* in a solution if it is not slid anymore in the rest of the moves, i.e. it has reached the correct position as in the final configuration. For example, in Fig. 2, the stone at position 5 and the stone at position 8 in row 1 are both stationary. We say a stone is stationary in a solution if it is stationary from the very beginning.

We denote a solution as *optimal* if there is no other solution using fewer moves. It is common for a puzzle to have more than one optimal solution. A *k-block* indicates k adjacent consecutive positions in a configuration.

The rest of the paper is organized as follows. Section 2 studies the lower bound and some general properties of optimal solutions. For completeness, in Section 3, we show a recursive algorithm for the $(n, 2)$-game which generates optimal solutions. Section 4 gives two algorithms for the $(n, 3)$-game, one for odd n and the other for even n, since the optimal solutions of the two cases are very different. Section 5 concludes with some open problems.

2. Lower bound and properties

We study some properties about optimal solutions which are independent of k. These will help in searching for the solutions. We first show the minimum number of moves required to solve a puzzle by counting adjacent positions with white and black stones, i.e. ○● or ●○.

Theorem 2.1. *For integers $n \geq 3$ and $k \geq 2$, it takes at least n moves to solve the (n, k)-game.*

Proof. We prove the theorem by contradiction. Suppose that the game can be solved in $n-1$ moves. Observe that, in Fig. 4, in the initial configuration there is only one pair of positions with different types of stone, i.e. the nth stone being white and the

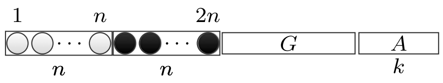


Fig. 5. The gap, indicated by G, in the configuration.

$(n+1)$th stone being black. The first move contributes at most one pair of adjacent positions with different colors, and each of the following moves creates at most two more such pairs. Thus, in $n-1$ moves, there are at most $1+1+2(n-2) = 2n-2$ pairs of adjacent positions with different colors created. However there are $2n-1$ such pairs in the final configuration, which is a contradiction. □

From Theorem 2.1, we know that n moves are the minimum to solve the puzzle. In the following, we study some necessary conditions for an n-move solution. In other words, if any of the conditions does not hold, the solution must have at least $n+1$ moves. We first investigate the patterns of adjacent positions created by moves in an optimal solution.

Theorem 2.2. *Let "pair" denote two adjacent positions possessing two stones with different colors. If a solution of the (n, k)-game has exactly n moves, then one of the following must hold.*

1. *The first move creates no pair, and each of the remaining $n-1$ moves creates two pairs.*
2. *There are two moves, including the first move, each creating one pair, and each of the remaining $n-2$ moves creates two pairs.*

Proof. We know that the initial configuration has one pair and that the final configuration has $2n-1$ pairs. If any move destroys a pair, it must take more than n moves to solve the puzzle. Therefore if the solution has exactly n moves, it has no move which splits any pair. According to the number of pairs created by the first move, there are two possible cases.

Case 1. Suppose that the first move creates no pair. To achieve $2n-1$ pairs in n moves, each of the remaining $n-1$ moves must create $((2n-1)-1-0)/(n-1) = 2$ pairs.

Case 2. Suppose that the first move creates one pair. Then $(2n-1)-1-1 = 2n-3$ pairs must be created in $n-1$ moves. Exactly one other move creates one pair. Each of the remaining $n-2$ move must create two pairs to achieve the goal.

This complete the proof. □

Next, we investigate the possible k-blocks outside of the initial configuration used in an n-move solution.

Theorem 2.3. *Let M be the $2n$ positions occupied by the $2n$ stones in the initial configuration. For an n-move solution to the (n, k)-game, there is no gap between M and any k-block used by any move during the rearrangement.*

Proof. We prove the theorem by contradiction. Let A be a k-block used on the right of M, and let there be a gap G between M and A, where G consists of at least one position, as shown in Fig. 5.

Without loss of generality, assume that block A is used in the first move. Otherwise, there would exist moves that create no pair and one pair, respectively, and this would not lead to an optimal solution by Theorem 2.2. Thus it is clear that the first move creates no pair, and there are two possible cases.

Case 1. Suppose that at least two different k-blocks (including A) are used during the rearrangement. When the second k-block (other than A) is first used, the corresponding move creates at most one pair, since there is a gap between it and M, by assumption. It is impossible for there to be an optimal solution, since it violates condition 1 of Theorem 2.2.

Case 2. Suppose that A is the only k-block outside of M used during the rearrangement. Note that the first move must use a k-block outside of M. Then, in the final configuration, the $2n$ alternate stones must stay within M. Note that A cannot be the target destination more than once, otherwise there would be more than one move creating no pair. Let C be the k stones in order slid in the first move. C must have stones in alternating order in the initial configuration, otherwise C cannot be fitted into the final configuration or it needs to use one more k-block.

For $k = 2$, C must be the pair at positions n and $n+1$, i.e. ○●. However, after moving C to A, all of the pairs in M are of the same color. The second move creates at most one pair, violating condition 1 of Theorem 2.2. For $k \geq 3$, there are no stones in alternating order in the initial configuration. Again, this is impossible. Therefore, the claim is true. □

Theorem 2.4. *An n-move solution to the (n, k)-game uses one of the following k-block patterns during the rearrangement (see Fig. 6), where L_i and R_i denote k-blocks outside of M and there is no gap between adjacent k-blocks.*

1. *R_1.*
2. *L_1.*
3. *R_1 and L_1.*
4. *R_1 and R_2.*
5. *L_1 and L_2.*

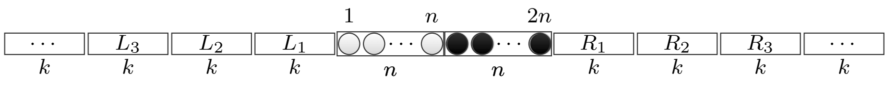


Fig. 6. The k-blocks in the initial configuration.

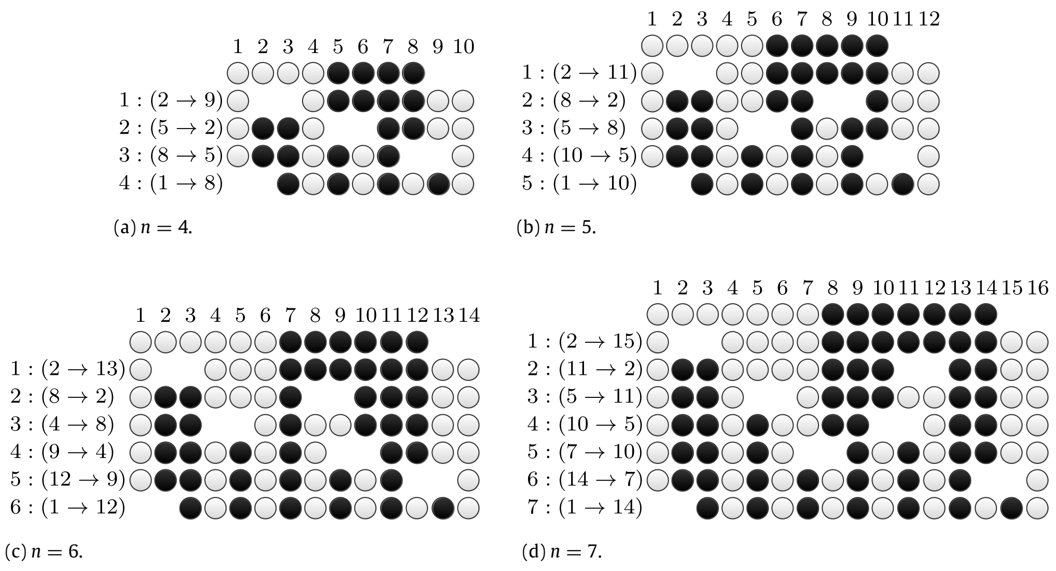


Fig. 7. The optimal solutions for the $(n, 2)$-game with $n = 4, 5, 6, 7$.

Proof. From Theorem 2.3, we know that there is no gap between M and the k-blocks used during the rearrangement. Therefore, if the solution uses at most two k-blocks, it can use only one of the five k-block set listed above.

We claim that the optimal solution does not use more than two different k-blocks. By contradiction, without loss of generality, assume that an optimal solution uses three different k-blocks, say R_1, R_2, and L_1. No matter which order the three k-blocks are first used in the solution, neither of the conditions in Theorem 2.2 is satisfied, since there are three moves that contribute at most one pair each. □

The above theorem is very useful for characterizing the possible optimal solution.

3. Optimal solution for the $(n, 2)$-game

We start from the family with $k = 2$, although this family of games has been solved [6]. To motivate the idea, we give the optimal solutions for small n in Section 3.1. Then we develop a recursive algorithm which generates optimal solutions in Section 3.2. The basic idea of the algorithm is reduce the size of the puzzle by 4, and then solve the subproblem recursively.

3.1. Base cases

The solution for $n = 3$ is as shown in Section 1.1, which is a special case. The solutions for $n = 4$, 5, 6, and 7 are shown as follows. Each solution has exactly n moves.

Theorem 3.1. *For $k = 2$, the following sequences are the optimal solutions for each corresponding n.*

- *For $n = 3$:* $(1 \to 7)$, $(6 \to 9)$, $(3 \to 6)$.
- *For $n = 4$:* $(2 \to 9)$, $(5 \to 2)$, $(8 \to 5)$, $(1 \to 8)$.
- *For $n = 5$:* $(2 \to 11)$, $(8 \to 2)$, $(5 \to 8)$, $(10 \to 5)$, $(1 \to 10)$.
- *For $n = 6$:* $(2 \to 13)$, $(8 \to 2)$, $(4 \to 8)$, $(9 \to 4)$, $(12 \to 9)$, $(1 \to 12)$.
- *For $n = 7$:* $(2 \to 15)$, $(11 \to 2)$, $(5 \to 11)$, $(10 \to 5)$, $(7 \to 10)$, $(14 \to 7)$, $(1 \to 14)$.

Proof. It is easy to check the correctness and the optimality of the solutions in Figs. 2 and 7. □

The optimal solutions for $n = 4$, 5, 6, and 7 will be used as the base cases in the algorithm. Note that the final configurations are black type and are shifted to the right by two positions. This property is useful to develop the algorithm.

3.2. Algorithm

The following recursive algorithm generates optimal solutions for $k = 2$.

```
1  function SlidingCoinK2(start, n);
2  begin
       // start is the starting position of the configuration.
       // n is the coin number of each type.
3      if $n < 3$ then
4          print "It is infeasible!";
5      else if $n = 3, 4, 5, 6$ or 7 then
6          foreach $(p \to q)$ in the corresponding solution in Theorem 3.1 do
7              print "$(p + start - 1 \to q + start - 1)$ ";
8          end
9      else
10         print "$(start + 1 \to 2n + start)$ ";
11         print "$(2n + start - 4 \to start + 1)$ ";
12         SlidingCoinK2(start+4, n-4);
13         print "$(2n + start - 1 \to start + 4)$ ";
14         print "$(start \to 2n + start - 1)$ ";
15     end
16 end
```

Algorithm 3.1: SlidingCoinK2 generates optimal solutions for the $(n, 2)$-game.

Theorem 3.2. SlidingCoinK2$(1, n)$ *generates optimal solutions correctly for* $n \geq 3$. *Moreover, each solution shifts the black type final configuration to the right by two positions, except for* $n = 3$.

Proof. We prove the theorem by induction on n. By Theorem 3.1, it is clear that the algorithm generates optimal solutions correctly for the special case $n = 3$ and the base cases $n = 4, 5, 6$, and 7.

Suppose that the algorithm generates an optimal solution for the $(n, 2)$-game. We prove that it generates an optimal solution as well for the $(n+4, 2)$-game. For convenience, assume that the initial configuration for the $(n+4, 2)$-game starts at position i, as shown in row (1) of Fig. 8. After performing the two moves generated by Steps 10 and 11 of the algorithm, the configuration becomes row (2) and (3), respectively.

Now the shaded sub-configuration in row (3) is simply the $(n, 2)$-game, and its initial configuration starts at position $i + 4$. By the induction hypothesis, the recursive call in Step 12 of the algorithm generates an n-move solution for the corresponding $(n, 2)$-game. Since the n-move solution shifts the black type final configuration to the right by two positions, the shaded sub-configuration can be rearranged as shown in row (4).

Finally, by performing the two moves generated by Steps 13 and 14 of the algorithm, the black type final configuration is reached, as shown in row (6). It takes $n + 4$ moves to solve the puzzle of size $n + 4$, and this is the optimal solution. □

4. Optimal solution for the $(n, 3)$-game

In this section, we explore the $(n, 3)$-game. Our approach is similar to the one for the $(n, 2)$-game in Section 3. We first show the optimal solutions for some small n in Section 4.1. However, the inductive steps of the solutions for odd n and even n are different. We develop two different algorithms in Section 4.2.

4.1. Special cases and base cases

The solution for $n = 3$ will be used as a base case in the algorithm for odd n. As for even n, $n = 4, 6$, and 8 are special cases, while $n = 10, 12$, and 14 are the base cases for even n. Note that the initial configuration for even n starts at position 4.

Theorem 4.1. *For* $k = 3$, *the following sequences are the optimal solutions for each corresponding* n.

- *For* $n = 3$: $(2 \to 7), (6 \to 2), (1 \to 6)$.
- *For* $n = 4$: $(6 \to 12), (10 \to 6), (5 \to 10), (12 \to 5)$.
- *For* $n = 6$: $(8 \to 16), (5 \to 8), (14 \to 5), (10 \to 14), (6 \to 10), (9 \to 6), (16 \to 9)$.
- *For* $n = 8$: $(10 \to 20), (6 \to 10), (14 \to 6), (18 \to 14), (12 \to 18), (7 \to 12), (10 \to 7), (5 \to 10), (20 \to 5)$.
- *For* $n = 10$: $(19 \to 1), (10 \to 19), (20 \to 24), (15 \to 10), (3 \to 20), (22 \to 15), (8 \to 3), (1 \to 8), (9 \to 22), (4 \to 9)$.

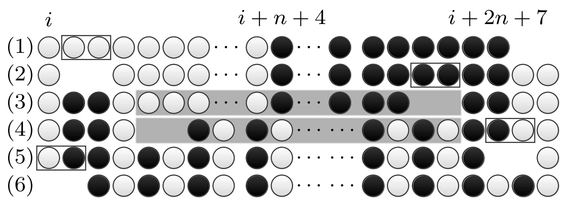


Fig. 8. The inductive step in the proof of Theorem 3.2. The problem size is $n+4$ and the initial configuration starts at position i. Each row becomes the next row after moving the coins in the frame.

- *For $n = 12$:* $(23 \to 1), (12 \to 23), (24 \to 28), (19 \to 12), (3 \to 24), (26 \to 19), (14 \to 26), (8 \to 14), (15 \to 3), (1 \to 15), (13 \to 8), (4 \to 13)$.
- *For $n = 14$:* $(27 \to 1), (10 \to 27), (28 \to 32), (21 \to 10), (14 \to 21), (3 \to 28), (8 \to 3), (19 \to 8), (23 \to 19), (1 \to 23), (21 \to 14), (30 \to 21), (9 \to 30), (4 \to 9)$.

Proof. For $n = 3, 4, 6, 8, 10, 12$, and 14, it is easy to check the correctness and the optimality of the solutions. The details for $n = 10, 12, 14$ are shown in Fig. 9. □

4.2. Algorithms

4.2.1. Algorithm for odd n

The following recursive algorithm generates optimal solutions for the $(n, 3)$-game with $n \geq 3$ as an odd number.

```
function SlidingCoinK3Odd($n, gap$);
begin
    // $n$ is the number of coins of each type.
    // $gap$ is the number of positions between the $(n+1)$th coin and the $(n+2)$th coin in
       the configuration of $n$ nickels and $n$ pennies.
    if $n < 3$ then
        print "It is infeasible!";
    else if $n = 3$ then
        print "$(2 \to n + gap + 4)$";
        print "$(n + gap + 3 \to 2)$";
        print "$(1 \to n + gap + 3)$";
    else
        print "$(n - 1 \to 2n + gap + 1)$";
        print "$(2n + gap - 1 \to n - 1)$";
        SlidingCoinK3Odd($n$-2, $gap$+2);
    end
end
```

Algorithm 4.1: SlidingCoinK3Odd generates optimal solutions for the $(n, 3)$-game with odd n.

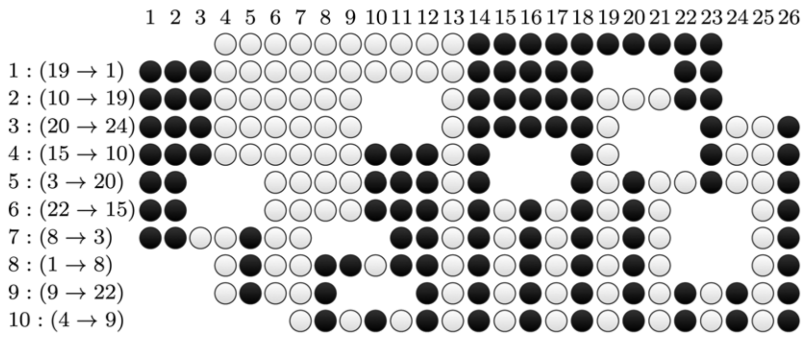
Theorem 4.2. *For odd $n \geq 3$,* SlidingCoinK3Odd$(n, 0)$ *generates the optimal solutions correctly. Moreover, each solution has the following three properties.*

1. *Each solution has the white type final configuration shifted to the right by three positions.*
2. *The two coins at positions n and $n+1$ are both stationary after the second move.*
3. *The $(n+2)$th coin (the second penny) is stationary.*

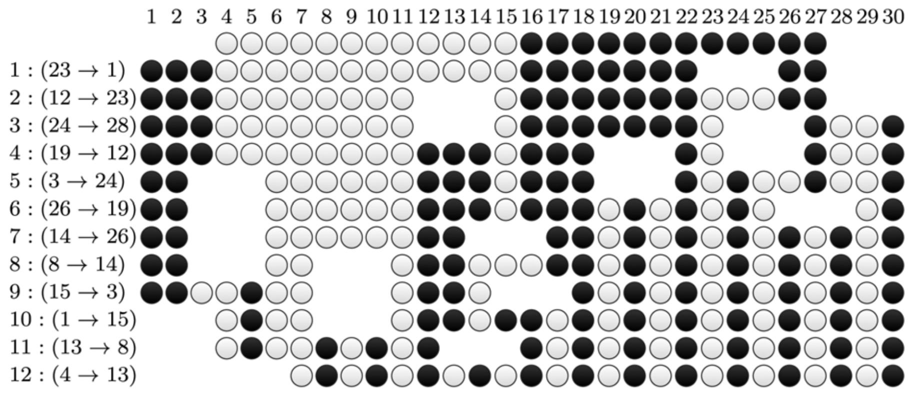
Before proving the correctness of the algorithm, let us observe two more optimal solutions generated by SlidingCoinK3Odd $(n, 0)$, for $n = 5$ and $n = 7$, as shown in Fig. 10. The two solutions and the one for $n = 3$ all satisfy the three properties in Theorem 4.2.

Suppose that there is a gap, with unoccupied positions or stationary coins, between the $(n+1)$th coin and the $(n+2)$th coin in the initial configuration. For example, assume that there is a gap between the first penny (at positions 6) and the second penny (at position 7) in the initial configuration of Fig. 10(a). Since the $(n+2)$th coin is stationary, we can still apply the solution to the new configuration by properly shifting all the coins after the $(n+2)$th coin to the right by the size of the gap. The gap in the configuration will not affect the moves in the solution. Thus the coins in the final configuration are in alternating order (but in two parts) as if the gap does not exist.

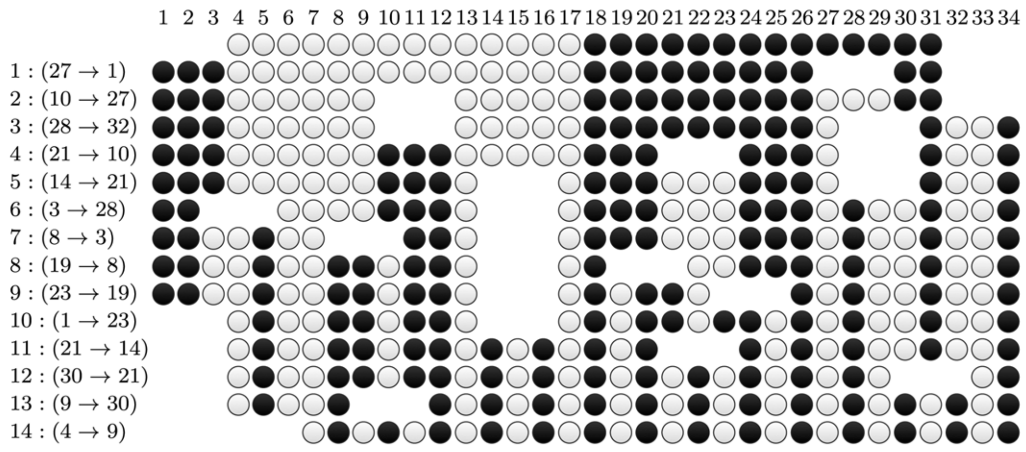
Proof of Theorem 4.2. We prove the theorem by induction on odd n. By Theorem 4.1, it is clear that the algorithm correctly generates the optimal solution for the base case $n = 3$ (with $gap = 0$), and that the solution satisfies the three properties.



(a) $n = 10$.



(b) $n = 12$.

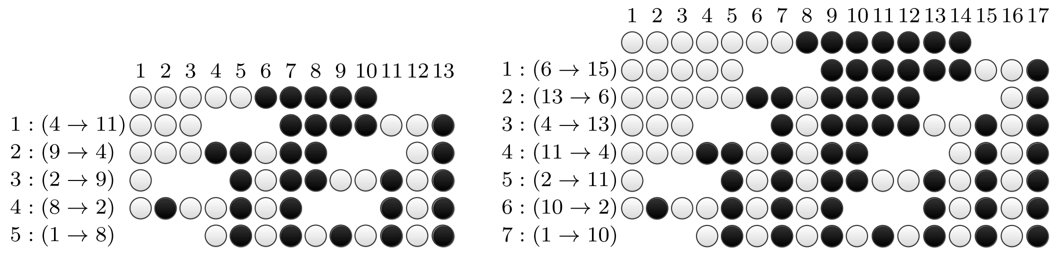


(c) $n = 14$.

Fig. 9. The optimal solutions for the $(n, 3)$-game with $n = 10, 12, 14$.

Suppose that the algorithm generates an optimal solution for the $(n-2, 3)$-game which satisfies the three properties. We prove that it generates an optimal solution correctly for the $(n, 3)$-game which satisfies the three properties as well. Assume that the initial configuration for the $(n, 3)$-game starts at position 1, as shown in row (1) of Fig. 11. After performing the two moves generated by Steps 10 and 11 of the algorithm, the configuration becomes row (2) and row (3), respectively.

Now the shaded sub-configuration in row (3) is simply the $(n-2, 3)$-game, separated by a gap of size 2 (at positions n and $n+1$). By the assumption and previous observation, the shaded sub-configuration can be solved in $n-2$ moves, even if there is a gap between the first penny and the second penny of the shaded sub-configuration. Thus, after the recursive call in Step 12, the shaded sub-configuration becomes the white type final sub-configuration and shifts to the right by three positions, as shown in row (4). The solution uses n moves, which is optimal. The two coins ●○ at positions n and $n+1$ and the two ○● at positions $2n+2$ and $2n+3$ integrate with the shaded sub-configuration. Thus the $2n$ coins are rearranged



(a) $n = 5$. (b) $n = 7$.

Fig. 10. Two optimal solutions generated by `SlidingCoinK3Odd`.

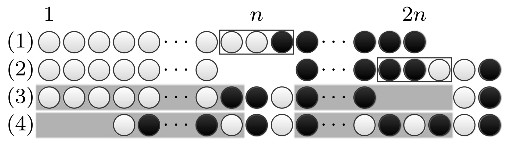


Fig. 11. The inductive step in the proof of Theorem 4.2. The problem size is n, and the initial configuration starts at position 1.

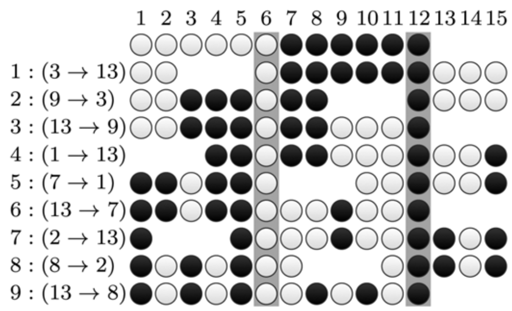


Fig. 12. A solution for an auxiliary puzzle which can be used to derive the solutions for even n. The coins in the shaded blocks are stationary.

correctly into the white type final configuration, shifted to the right by three positions. The second property is satisfied, since the two coins at positions n and $n + 1$ are not moved by the solution to the $(n - 2, 3)$-game. The third property is also satisfied, since the second penny is not moved by the first two moves, and it is also the second penny of the corresponding $(n - 2, 3)$-game. □

4.2.2. Algorithm for even n

We observe that the structures of solutions for odd n and even n are quite different. The puzzle family for even n is not as easy as that for odd n, since it cannot be solved by simple recursion. We need a different approach to tackle the $(n, 3)$-game for even n.

Consider an auxiliary puzzle of size 6, where the partial solution is shown in Fig. 12. Note that the final configuration is different from the original puzzle, which is black type from position 1 to position 6 and is white type in the second half. This solution is designed such that we can combine it with a known solution S which satisfies the conditions shown in Lemma 4.3. Note that in Fig. 12 the three patterns ○○○ , ○○● , and ●○● , in this order, appear at positions 13 to 15. This helps us to find a solution for a puzzle of larger size.

Actually, the approach here can be used as a general framework. With a similar idea, we solved the $(n, 4)$-game and the $(n, 5)$-game. Consider the solution in Fig. 12, and see how to use it to construct a solution for a larger puzzle. The detail of the construction is shown in the proof of Lemma 4.3, which leads to an algorithm for even n.

Lemma 4.3. *Consider a solution S for the* $(n, 3)$*-game satisfying the following two conditions.*

1. *There exists a nickel and a penny such that both coins are stationary.*
2. *The three patterns* ○○○ , ○○● , *and* ●○● , *in this order, are slid in three moves of* S.

Then S can be extended to be a solution S' *for the* $(n + 6m, 3)$*-game, where the integer* $m \geq 1$.

Proof. Let S be a solution for the $(n, 3)$-game satisfying the above conditions, and let C be the initial configuration. Suppose that a and b are the positions of the stationary nickel and penny in C. We give a construction which extends S to a solution S' for the $(n + 6, 3)$-game. We prove that S' also satisfies the above two conditions.

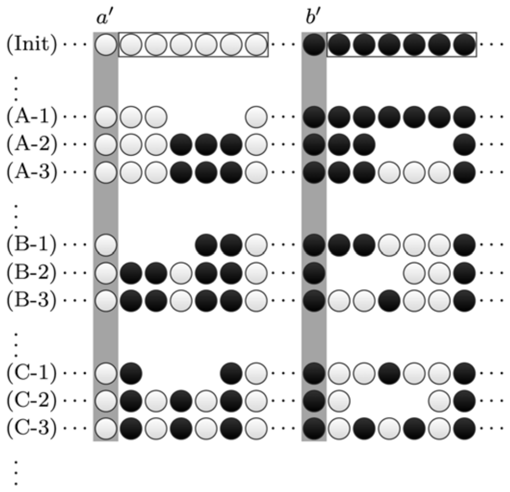


Fig. 13. The modifications of the three moves in the proof of Lemma 4.3. The coins in the shaded blocks at a' and b' are stationary. The newly added coins are in the frame of the first row.

First, we add six nickels and six pennies to the right of a and b in C, respectively. Let C' be the modified initial configuration. Let a' and b' be the positions in C' corresponding to the coins at positions a and b in C, respectively. The added coins are in the frame as shown in row (Init) of Fig. 13.

Initially, let S' be the partial solution inherited from S such that the moves consisting of the positions after a' and b' are properly shifted to the right accordingly as in C'. For the moment, S' does not move any of the newly added 12 coins. By condition 1, the coins at positions a' and b' are both stationary in C' when performing the moves of S'. For now, after performing the moves of S', the original coins can be slid to reach the final configuration without touching the newly added 12 coins. Next, we need to further modify the moves of S' such that the added coins are in alternating order and integrated into the final configuration.

Let $(x_1 \to y_1)$ be the move in S sliding ○○○, which is guaranteed by condition 2. Let $(x'_1 \to y'_1)$ be the corresponding move in S'. We replace $(x'_1 \to y'_1)$ with the following three moves, as shown in Fig. 13.

(A-1): $(a' + 3 \to y'_1)$.
(A-2): $(b' + 3 \to a' + 3)$.
(A-3): $(x'_1 \to b' + 3)$.

The net effect of the three moves on the original coins is the same as $(x'_1 \to y'_1)$, i.e. ○○○ is slid to y'_1, and x'_1 becomes unoccupied. The sub-configuration of the added coins is shown in Fig. 13.

Similarly, let $(x'_2 \to y'_2)$ be the move in S' sliding ○○●. Replace $(x'_2 \to y'_2)$ with the following.

(B-1): $(a' + 1 \to y'_2)$.
(B-2): $(b' + 1 \to a' + 1)$.
(B-3): $(x'_2 \to b' + 1)$.

Finally, let $(x'_3 \to y'_3)$ be the move in S' sliding ●○●. Replace $(x'_3 \to y'_3)$ with the following.

(C-1): $(a' + 2 \to y'_3)$.
(C-2): $(b' + 2 \to a' + 2)$.
(C-3): $(x'_3 \to b' + 2)$.

After move (C-3), the added nickels at positions $a' + 1, \dots, a' + 6$ and pennies at positions $b' + 1, \dots, b' + 6$ are in alternating order, as shown in Fig. 13. Since S is a correct solution, after the last move of S', the coins at $a' + 7$ and $b' + 7$ are a penny and a nickel, respectively. The added coins fit into the final configuration of S'. Hence, S' is a correct solution as well, and has exactly six more moves than S does.

Observe Fig. 13: the nickel at position $a' + 6$ and the penny at position $b' + 6$ are both stationary while performing S'. The three patterns ○○○, ○○●, and ●○●, in this order, are slid by the three move (A-1), (B-1), and (C-1) in S'. Therefore, S' satisfies the above two conditions. The construction can be applied repeatedly. Hence S can be extended to optimal solutions for the $(n + 6m, 3)$-game, where $m \geq 1$. $\square$

See the solution for the (16, 3)-game in Fig. 14 as an example. It is generated by the construction shown in the proof of Lemma 4.3, which uses the solution for the (10, 3)-game in Fig. 9(a) and the solution to the auxiliary puzzle in Fig. 12.

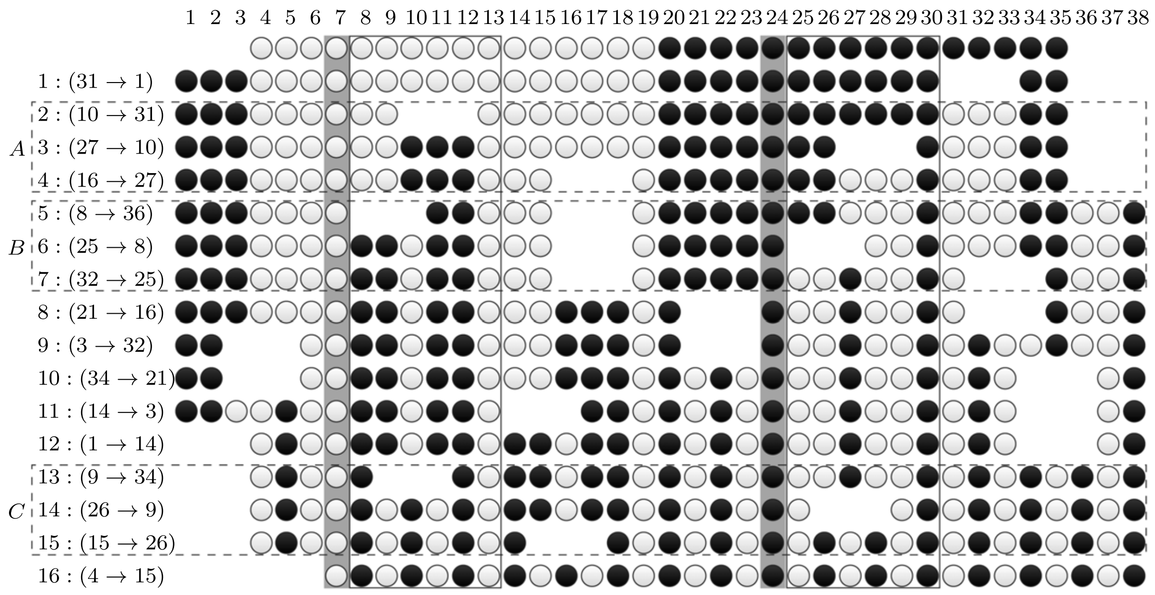


Fig. 14. An optimal solution for the $(16, 3)$-game, which is extended from the solutions for the $(10, 3)$-game in Fig. 9(a) by using the construction in Lemma 4.3. The added coins are in the solid frame. The moves in A, B and C (in the dashed frame) are extended from moves 2, 3, and 9 in Fig. 9(a), respectively.

In Fig. 9(a), the coins at positions 7 and 18 are stationary and the three moves 2, 3, and 9 move the three patterns ○○○ , ○○● , and ●○● , respectively. The moves in A, B, and C in Fig. 14 are extended from moves 2, 3, and 9 in Fig. 9(a).

Theorem 4.4. *For any even integer $n \geq 10$, there exists an optimal solution for the $(n, 3)$-game.*

Proof. For $n = 10$, 12, and 14, it is clear that each has an optimal solution by Theorem 4.1. It is easy to check that each solution satisfies the conditions in Lemma 4.3 from Fig. 9(a)–(c). The construction in the proof of Lemma 4.3 implies that, if S has n moves, then S' has $n + 6m$ moves, $m \geq 1$. Hence the solutions for $n = 10$, 12, and 14 can be used as the base cases, and can be extended to be the solutions for all even $n \geq 16$. □

Based on the above, we can generate the optimal moves for any even number $n \geq 10$. With a similar idea, we can show that the $(n, 4)$-game and the $(n, 5)$-game can be played in n moves.

5. Conclusion and remarks

We prove that n moves is the minimum number of moves required to solve the sliding puzzle, and show how to generate optimal solutions for the (n, k) sliding-coin puzzle families with $k = 2$ and 3. Actually, we have extended the idea for $k = 3$ to solve the cases $k = 4$ and 5 optimally. We leave the following as open questions.

1. Can the (n, k) sliding-coin puzzle be solved in n moves for any k and sufficiently large n?
2. Can an arbitrary initial configuration of equal number of nickels and pennies be rearranged to alternate coins? Can it be done in n moves?

References

[1] M. Abellanas, S. Bereg, F. Hurtado, A.G. Olaverri, D. Rappaport, J. Tejel, Moving coins, Computational Geometry 34 (1) (2006) 35–48.
[2] E.D. Demaine, M.L. Demaine, Puzzles, art, and magic with algorithms, Theory of Computing Systems 39 (3) (2006) 473–481.
[3] E.D. Demaine, M.L. Demaine, H.A. Verrill, Coin-moving puzzles, in: More Games of No Chance, Cambridge University Press, 2002, pp. 405–431.
[4] M. Gardner, Mathematical Puzzles of Sam Loyd, Dover Publications, 1959.
[5] E. Hordern, Sliding Piece Puzzles, Oxford University Press, 1987.
[6] B.A. Kordemsky, The Moscow Puzzles: 359 Mathematical Recreations, Dover Publications, 1992.
[7] G. Mott-Smith, Mathematical Puzzle, for Beginners and Enthusiasts, Dover Publications, 1954.
[8] Sliding coin puzzles and games, http://www.alethis.net/amusements/coin/coin.html.