Winner Determination Algorithms for Colored Arc Kayles

Kanae Yoshiwatari[1,a)] Hironori Kiya[2,b)] Tesshu Hanaka[2,c)] Hirotaka Ono[1,d)]

Abstract: Cram, Domineering, and Arc Kayles are well-studied combinatorial games. They are interpreted as edge-selecting-type games on graphs, and the selected edges during a game form a matching. In this paper, we define a generalized game called Colored Arc Kayles, which includes these games. Colored Arc Kayles is played on a graph whose edges are colored in black, white, or gray, and black (resp., white) edges can be selected only by the black (resp., white) player, although gray edges can be selected by both black and white players. BW-Arc Kayles and Arc Kayles are restrictions of Colored Arc Kayles, where We first observe that the winner determination for Colored Arc Kayles can be done in $O^*(2^n)$ time by a simple algorithm, where n is the order of a graph. We then focus on the vertex cover number, which is linearly related to the number of turns, and show that Colored Arc Kayles, BW-Arc Kayles, and Arc Kayles are solved in time $O^*(1.4143^{\tau^2+3.17\tau})$, $O^*(1.3161^{\tau^2+4\tau})$, and $O^*(1.1893^{\tau^2+6.34\tau})$, respectively, where τ is the vertex cover number. Furthermore, we present an $O^*((n/\nu+1)^\nu)$-time algorithm for Arc Kayles, where ν is neighborhood diversity. We finally show that Arc Kayles on trees can be solved in $O^*(2^{n/2})(=O(1.4143^n))$ time, which improves $O^*(3^{n/3})(=O(1.4423^n))$ by a direct adjustment of the analysis of Bodlaender et al.'s $O^*(3^{n/3})$-time algorithm for Node Kayles.

Keywords: Combinatorial game theory, graph algorithm, Kayles, Node Kayles, Arc Kayles

1. Introduction

1.1 Background and Motivation

Cram, Domineering, and Arc Kayles are well-studied two-player mathematical games and interpreted as combinatorial games on graphs. Domineering (also called Stop-Gate) was introduced by Göran Andersson around 1973 under the name of Crosscram [6], [8]. Domineering is usually played on a checkerboard. The two players are denoted by Vertical and Horizontal. Vertical (resp., Horizontal) player is only allowed to place its dominoes vertically (resp., horizontally) on the board. Note that placed dominoes are not allowed to overlap. If no place is left to place a domino, the player in the turn loses the game. Domineering is a partisan game, where players use different pieces. The impartial version of the game is Cram, where two players can place dominoes both vertically and horizontally.

An analogous game played on an undirected graph G is Arc Kayles. In Arc Kayles, the action of a player in a turn is to select an edge of G, and then the selected edge and its neighboring edges are removed from G. If no edge remains in the resulting graph, the player in the turn loses the game. Figure 1 is a play example of Arc Kayles. In this example, the first player selects edge e_1, and then the second player selects edge e_2. By the first player selecting edge e_3, no edge is left; the second player loses. Note that the edges selected throughout a play form a maximal matching on the graph.

1 Nagoya University
2 Kyushu University
a) yoshiwatari.kanae.w1@s.mail.nagoya-u.ac.jp
b) h-kiya@econ.kyushu-u.ac.jp
c) hanaka@inf.kyushu-u.ac.jp
d) ono@nagoya-u.jp

Similarly, we can define BW-Arc Kayles, which is played on an undirected graph with black and white edges. The rule is the same as the ordinary Arc Kayles except that the black (resp., white) player can select only black (resp., white) edges. Note that Cram and Domineering are respectively interpreted as Arc Kayles and BW-Arc Kayles on a two-dimensional grid graph, which is the graph Cartesian product of two path graphs.

To focus on the common nature of such games with matching structures, we newly define Colored Arc Kayles. Colored Arc Kayles is played on a graph whose edges are colored in black, white, or gray, and black (resp., white) edges can be selected only by the black (resp., white) player, though grey edges can be selected by both black and white players. BW-Arc Kayles and ordinary Arc Kayles are special cases of Colored Arc Kayles. In this paper, we investigate Colored Arc Kayles from the algorithmic point of view.

1.2 Related work

1.2.1 Cram and Domineering

Cram and Domineering are well studied in the field of combinatorial game theory. In [8], Gardner gives winning strategies for some simple cases. For Cram on $a \times b$ board, the second player can always win if both a and b are even, and the first player can always win if one of a and b is even and the other is odd. This can be easily shown by the so-called Tweedledum and Tweedledee strategy. For specific sizes of boards, computational studies have been conducted [17]. In [16], Cram's endgame databases for all board sizes with at most 30 squares are constructed. As far as the authors know, the complexity to determine the winner for Cram on general boards still remains open.

Finding the winning strategies of Domineering for specific

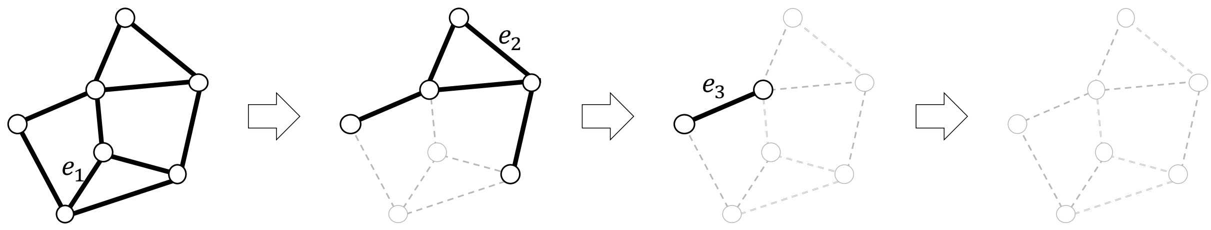

Fig. 1 A play example of Arc Kayles

sizes of boards by using computer programs is well studied. For example, the cases of 8×8 and 10×10 are solved in 2000 [3] and 2002 [4], respectively. The first player wins in both cases. Currently, the status of boards up to 11×11 is known [15]. In [18], endgame databases for all single-component positions up to 15 squares for Domineering are constructed. The complexity of Domineering on general boards also remains open. Lachmann, Moore, and Rapaport show that the winner and a winning strategy Domineering on $m \times n$ board can be computed in polynomial time for $m \in \{1, 2, 3, 4, 5, 7, 9, 11\}$ and all n [11].

1.2.2 Kayles, Node Kayles, and Arc Kayles

Kayles is a simple impartial game, introduced by Henry Dudeney in 1908 [7]. The name "Kayles" derives from French word "quilles", meaning "bowling". The rule of Kayles is as follows. Given bowling pins equally spaced in a line, players take turns to knock out either one pin or two adjacent pins, until all the pins are gone. As graph generalizations, Node Kayles and Arc Kayles are introduced by Schaefer [14]. Node Kayles is the vertex version of Arc Kayles. Namely, the action of a player is to select a vertex instead of an edge, and then the selected vertex and its neighboring vertices are removed. Note that both generalizations can describe the original Kayles; Kayles is represented as Node Kayles on sequentially linked triangles or as Arc Kayles on a caterpillar graph.

Node Kayles is known to be PSPACE-complete [14], whereas the winner determination is solvable in polynomial time on graphs of bounded asteroidal numbers such as cocomparability graphs and cographs by using Sprague-Grundy theory [1]. For general graphs, Bodlaender et al. propose an $O(1.6031^n)$-time algorithm [2]. Furthermore, they show that the winner of Node Kayles can be determined in time $O(1.4423^n)$ on trees. In [10], Kobayashi sophisticates the analysis of the algorithm in [2] from the perspective of the parameterized complexity and shows that it can be solved in time $O^*(1.6031^\mu)$, where μ is the modular width of an input graph[*1]. He also gives an $O^*(3^\tau)$-time algorithm, where τ is the vertex cover number, and a linear kernel when parameterized by neighborhood diversity.

Different from Node Kayles, the complexity of Arc Kayles has remained open for more than 30 years. Even for subclasses of trees, not much is known. For example, Huggans and Stevens study Arc-Kayles on subdivided stars with three paths [9]. To our best knowledge, no exponential-time algorithm for Arc Kayles is presented except for an $O^*(4^{\tau^2})$-time algorithm proposed in [13].

1.3 Our contribution

In this paper, we address winner determination algorithms for Colored Arc Kayles. We first propose an $O^*(2^n)$-time algorithm for Colored Arc Kayles. Note that this is generally faster than applying the Node Kayles algorithm to the line graph of an instance of Arc Kayles; it takes time $O(1.6031^m)$, where m is the number of the original edges. We then focus on algorithms based on graph parameters. We present an $O^*(1.4143^{\tau^2+3.17\tau})$-time algorithm for Colored Arc Kayles, where τ is the vertex cover number. The algorithm runs in time $O^*(1.3161^{\tau^2+4\tau})$ and $O^*(1.1893^{\tau^2+6.34\tau})$ for BW-Arc Kayles, and Arc Kayles, respectively. This is faster than the previously known time complexity $O^*(4^{\tau^2})$ in [13].

On the other hand, we give a bad instance for the proposed algorithm, which implies the running time analysis is asymptotically tight. Furthermore, we show that the winner of Arc Kayles can be determined in time $O^*((n/\nu+1)^\nu)$, where ν is the neighborhood diversity of an input graph. This analysis is also asymptotically tight, because there is an instance having $(n/\nu - o(1))^{\nu(1-o(1))}$. We finally show that the winner determination of Arc Kayles on trees can be solved in $O^*(2^{n/2}) = O(1.4143^n)$ time, which improves $O^*(3^{n/3})(= O(1.4423^n))$ by a direct adjustment of the analysis of Bodlaender et al.'s $O^*(3^{n/3})$-time algorithm for Node Kayles.

In this technical report, all proofs of theorems and lemmas are omitted. For more details, see [19].

2. Preliminaries

2.1 Notations and terminology

Let $G = (V, E)$ be an undirected graph. We denote $n = |V|$ and $m = |E|$, respectively. For an edge $e = \{u, v\} \in E$, we define $\Gamma(e) = \{e' \mid e \cap e' \neq \emptyset\}$. For a graph $G = (V, E)$ and a vertex subset $V' \subseteq V$, we denote by $G[V']$ the subgraph induced by V'. For simplicity, we denote $G - v$ instead of $G[V \setminus \{v\}]$. For an edge subset E', we also denote by $G - E'$ the subgraph obtained from G by removing all edges in E' from G. A vertex set S is called a *vertex cover* if $e \cap S \neq \emptyset$ for every edge $e \in E$. We denote by τ the size of a minimum vertex cover of G. Two vertices $u, v \in V$ are called *twins* if $N(u) \setminus \{v\} = N(v) \setminus \{u\}$.

Definition 1. *The* neighborhood diversity $\nu(G)$ *of* $G = (V, E)$ *is defined as the minimum number* w *such that* V *can be partitioned into* w *vertex sets of twins.*

In the following, we simply write ν instead of $\nu(G)$ if no confusion arises. We can compute the neighborhood diversity of G and the corresponding partition in polynomial time [12]. For any graph G, $\nu \leq 2^\tau + \tau$ holds.

2.2 Colored Arc Kayles

Colored Arc Kayles is played on a graph $G = (V, E_G \cup E_B \cup E_W)$, where E_G, E_B, E_W are mutually disjoint. The subscripts G, B, and

*1 The $O^*(\cdot)$ notation suppresses polynomial factors in the input size.

W of E_G, E_B, E_W respectively, stand for gray, black, and white. For every edge $e \in E_G \cup E_B \cup E_W$, let $c(e)$ be the color of e, that is, $c(e) =$ G if $e \in E_G$, B if $e \in E_B$, and W if $e \in E_W$. If $\{u, v\} \notin E_G \cup E_B \cup E_W$, we set $c(\{u, v\}) = \emptyset$ for convenience. As explained below, the first (black or B) player can choose only gray or black edges, and the second (white or W) player can choose only gray or white edges.

Two players alternatively choose an edge of G. Player B can choose an edge in $E_G \cup E_B$ and player W can choose an edge in $E_G \cup E_W$. That is, there are three types of edges; E_B is the set of edges that only the first player can choose, E_W is the set of edges that only the second player can choose, and E_G is the set of edges that both the first and second players can choose. Once an edge e is selected, the edge and its neighboring edges (i.e., $\Gamma(e)$) are removed from the graph, and the next player chooses an edge of $G - \Gamma(e)$. The player that can take no edge loses the game. Since (Colored) Arc Kayles is a two-person zero-sum perfect information game and ties are impossible, one of the players always has a winning strategy. We call the player having a winning strategy the *definite winner*, or simply *winner*.

The problem that we consider in this paper is defined as follows:

Input: $G = (V, E_G \cup E_B \cup E_W)$, active player in $\{B, W\}$.

Question: Suppose that players B and W play Colored Arc Kayles on G from the active player's turn. Which player is the winner?

Remark that if $E_B = E_W = \emptyset$, Colored Arc Kayles is equivalent to Arc Kayles and if $E_G = \emptyset$, it is equivalent to BW-Arc Kayles.

To simply represent the definite winner of Colored Arc Kayles, we introduce two Boolean functions f_B and f_W. The $f_B(G)$ is defined such that $f_B(G) = 1$ if and only if the winner of Colored Arc Kayles on G from player B's turn is player B. Similarly, $f_W(G)$ is the function such that $f_W(G) = 1$ if and only if the winner of Colored Arc Kayles on G from player W's turn is the player W. If two graphs G and G' satisfy that $f_B(G) = f_B(G')$ and $f_W(G) = f_W(G')$, we say that *G and G' have the same game value on Colored Arc Kayles*.

3. Basic Algorithm

In this section, we show that the winner of *Colored* Arc Kayles on G can be determined in time $O^*(2^n)$. We first observe that the following lemma holds by the definition of the game.

Lemma 1. *Suppose that Colored Arc Kayles is played on $G = (V, E_G \cup E_W \cup E_B)$. Then, player* B *(resp.,* W*) wins on G with player* B*'s (resp.,* W*'s) turn if and only if there is an edge $\{u, v\} \in E_G \cup E_B$ (resp., $\{u, v\} \in E_G \cup E_W$) such that player* W *(resp.,* B*) loses on $G - u - v$ with player* B*'s (resp.,* W*'s) turn.*

This lemma is interpreted by the following two recursive formulas:

$$f_B(G) = \bigvee_{\{u,v\} \in E_G \cup E_B} \neg \left(f_W(G - u - v)\right), \qquad (1)$$

$$f_W(G) = \bigvee_{\{u,v\} \in E_G \cup E_W} \neg \left(f_B(G - u - v)\right). \qquad (2)$$

By these formulas, we can determine the winner of G with either first or second player's turn by computing $f_B(G)$ and $f_W(G)$ for all induced subgraphs of G. Since the number of all induced subgraphs of G is 2^n, it can be done in time $O^*(2^n)$ by a standard dynamic programming algorithm.

Theorem 1. *The winner of Colored Arc Kayles can be determined in time $O^*(2^n)$.*

4. Parameterization by vertex cover

In this section, we propose winner determination algorithms for Colored Arc Kayles parameterized by the vertex cover number. As mentioned in Introduction, the selected edges in a play of Colored Arc Kayles form a matching. This implies that the number of turns is bounded above by the maximum matching size of G and thus by the vertex cover number. Furthermore, the vertex cover number of the input graph is bounded by twice of the number r of turns of Arc Kayles. Intuitively, we may consider that a game taking longer turns is harder to analyze than games taking shorter turns. In that sense, the parameterization by the vertex cover number is quite natural.

In this section, we propose an $O^*(1.4143^{\tau^2+3.17\tau})$-time algorithm for Colored Arc Kayles, where τ is the vertex cover number of the input graph. It utilizes similar recursive relations shown in the previous section, but we avoid to enumerate all possible positions by utilizing equivalence classification.

Before explaining the equivalence classification, we give a simple observation based on isomorphism. The isomorphism on edge-colored graphs is defined as follows.

Definition 2. *Let $G = (V, E)$ and $G' = (V', E')$ be edge-colored graphs where $E = \bigcup_{i=1}^{r} E_i$ and $E' = \bigcup_{i=1}^{r} E'_i$. Then G and G' are called* isomorphic *if for any pair of $u, v \in V$ there is a bijection $f : V \to V'$ such that $\{u, v\} \in E_i$ if and only if $\{f(u), f(v)\} \in E'_i$.*

The following proposition is obvious.

Proposition 1. *If edge-colored graphs G and G' are isomorphic, G and G' have the same game value for Colored Arc Kayles.*

Let S be a vertex cover of $G = (V, E_G \cup E_W \cup E_B)$, that is, any $e = \{u, v\} \in E_G \cup E_W \cup E_B$ satisfies that $\{u, v\} \cap S \neq \emptyset$. Note that for $v \in V \setminus S$, $N(v) \subseteq S$ holds. We say that two vertices $v, v' \in V \setminus S$ are *equivalent with respect to S in G* if $N(v) = N(v')$ and $c(\{u, v\}) = c(\{u, v'\})$ holds for $\forall u \in N(v)$. If two vertices $v, v' \in V \setminus S$ are equivalent with respect to S in G, $G - u - v$ and $G - u - v'$ are isomorphic because the bijective function swapping only v and v' satisfies the isomorphic condition. Thus, we have the following lemma.

Lemma 2. *Suppose that two vertices $v, v' \in V \setminus S$ are equivalent with respect to S in G. Then, for any $u \in N(v)$, $G - u - v$ and $G - u - v'$ have the same game value.*

By the equivalence with respect to S, we can split $V \setminus S$ into equivalence classes. Note here that the number of equivalence classes is at most $4^{|S|}$, because for each $u \in S$ and $v \in V \setminus S$, edge $\{u, v\}$ does not exist, or it can be colored with one of three colors if exists; we can identify an equivalent class with $\boldsymbol{x} \in \{\emptyset, G, B, W\}^S$, a 4-ary vector with length $|S|$. For $S' \subseteq S$, let $\boldsymbol{x}[S']$ denotes the vector by dropping the components of $\boldsymbol{x}$ except the ones corresponding to S'. Also for $u \in S$, $\boldsymbol{x}[u]$ denotes the component corresponding to u in $\boldsymbol{x}$. Then, V is partitioned into $V_S^{(\boldsymbol{x})}$'s, where $V_S^{(\boldsymbol{x})} = \{v \in V \setminus S \mid \forall u \in S : c(\{v, u\}) = \boldsymbol{x}[u]\}$. We arbitrarily define the representative of non-empty $V_S^{(\boldsymbol{x})}$ (e.g., the vertex with the smallest ID), which is denoted by $\rho(V_S^{(\boldsymbol{x})})$. By using ρ, we also define the representative edge set by

$$E^R(S) = \bigcup_{\boldsymbol{x} \in \{\emptyset, \mathrm{G}, \mathrm{B}, \mathrm{W}\}^S} \{\{u, \rho(V_S^{(\boldsymbol{x})})\} \in E_{\mathrm{G}} \cup E_{\mathrm{B}} \cup E_{\mathrm{W}} \mid u \in S\}.$$

By Lemma 2, we can assume that both players choose an edge only in $E^R(S)$, which enables to modify the recursive equations (1) and (2) as follows: For a vertex cover S of G, we have

$$f_B(G) = \bigvee_{\{u,v\} \in (E_{\mathrm{G}} \cup E_{\mathrm{B}}) \cap (E^R(S) \cup S \times S)} \neg (f_W(G - u - v))), \quad (3)$$

$$f_W(G) = \bigvee_{\{u,v\} \in (E_{\mathrm{G}} \cup E_{\mathrm{W}}) \cap (E^R(S) \cup S \times S)} \neg (f_B(G - u - v))). \quad (4)$$

Note that this recursive formulas imply that the winner of Colored Arc Kayles can be determined in time $O^*((\tau^2 + \tau \cdot 4^\tau)^\tau) = O^*((4^{\tau + \log_4 \tau})^\tau) = O^*(4^{\tau^2 + \tau \log_4 \tau}) = O^*(5.6569^{\tau^2})$, because the recursions are called at most $|S|$ times and $\tau + \log_4 \tau \leq 1.25\tau$ for $\tau \geq 1$.

In the following, we give a better estimation of the number of induced subgraphs appearing in the recursion. Once such subgraphs are listed up, we can apply a standard dynamic programming to decide the necessary function values, or we can compute f_B and f_W according to the recursive formulas with memorization, by which we can skip redundant recursive calls. In order to estimate the number of induced subgraphs appearing in the recursion, we focus on the fact that the position of a play in progress corresponds to the subgraph induced by a matching.

Lemma 3. *The number of nodes in recursion trees of equations (3) and (4) for Colored Arc Kayles is $O((r+1)^{|S|^2/4} 3^{|S|} |S|^2)$, where r is the used colors.*

The proof of Lemma 3 is omitted. The following theorem immediately holds by Lemma 3 and the fact that a minimum vertex cover of G can be found in time $O^*(1.2738^\tau)$, where τ is the vertex cover number of G [5].

Theorem 2. *The winners of Colored Arc Kayles, BW-Arc Kayles, and Arc Kayles can be determined in time $O^*(1.4143^{\tau^2+3.17\tau})$, $O^*(1.3161^{\tau^2+4\tau})$, and $O^*(1.1893^{\tau^2+6.34\tau})$, respectively, where τ is the vertex cover number of a graph.*

We have shown that the winner of Arc Kayles can be determined in time $O^*(1.1893^{\tau^2+6.34\tau})$. The following theorem shows that the analysis is asymptotically tight, which implies that for further improvement, we need additional techniques apart from ignoring vertex-cover-based isomorphic positions. We here give such an example in Figure 2.

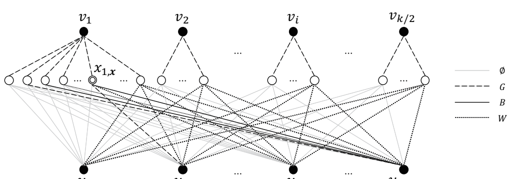

Fig. 2 The constructed graph $G = (U \cup V \cup X, E)$.

Theorem 3. *There is a graph for which the algorithm requires $2^{\tau^2/2}$ recursive calls for Colored Arc Kayles.*

The proof of Theorem 3 is omitted. By the similar construction, we can show the following theorem.

Theorem 4. *There is a graph for which the algorithm requires 1.3161^{τ^2} and 1.1893^{τ^2} recursive calls for BW-Arc Kayles and Arc Kayles, respectively.*

Remark 1. *Although Theorems 3 and 4 give lower bounds on the running time of the vertex cover-based algorithms, the proof implies a stronger result. In the proof of Theorem 3, we use ID's of the vertices in U. By connecting $2i$ pendant vertices to u_i, we can regard them as ID of u_i. Furthermore, we make U a clique by adding edges. These make the graphs not automorphic, which implies that the time complexity of an algorithm utilizing only isomorphism is at least the value shown in Theorems 3 or 4.*

5. Parameterization by neighborhood diversity

In this section, we deal with neighborhood diversity ν, which is a more general parameter than vertex cover number. We first give an $O^*((n/\nu)^{O(\nu)})$-time algorithm for Arc Kayles. This is an XP algorithm parameterized by neighborhood diversity. On the other hand, we show that there is a graph having at least $O^*((n/\nu)^{\Omega(\nu)})$ non-isomorphic induced subgraphs, which implies the analysis of the proposed algorithm is asymptotically tight.

By Proposition 1, if we list up all non-isomorphic induced subgraphs, the winner of Arc Kayles can be determined by using recursive formulas (1) and (2). Let $\mathcal{M} = \{M_1, M_2, \ldots, M_\nu\}$ be a partition such that $\bigcup_i M_i = V$ and vertices of M_i are twins each other. We call each M_i a *module*. It is easily seen that non-isomorphic induced subgraphs of G are identified by how many vertices are selected from which module.

Lemma 4. *The number of non-isomorphic induced subgraphs of a graph of neighborhood diversity ν is at most $(n/\nu + 1)^\nu$.*

Without loss of generality, we select an edge whose endpoints are the minimum indices of vertices in the corresponding module. By Proposition 1, the algorithm in Section 3 can be modified to run in time $O^*((n/\nu + 1)^\nu)$.

Theorem 5. *There is an $O^*((n/\nu + 1)^\nu)$-time algorithm for Arc Kayles.*

The idea can be extended to Colored Arc Kayles and BW-Arc Kayles. In $G = (V, E_{\mathrm{G}} \cup E_{\mathrm{B}} \cup E_{\mathrm{W}})$, two vertices $u, v \in V$ are called *colored twins* if $c(\{u, w\}) = c(\{v, w\})$ holds $\forall w \in V \setminus \{u, v\}$. We then define the notion of colored neighborhood diversity.

Definition 3. *The* colored neighborhood diversity *of $G = (V, E)$ is defined as minimum ν' such that V can be partitioned into ν' vertex sets of colored twins.*

In Colored Arc Kayles or BW-Arc Kayles, we can utilize a partition of V into modules each of which consists of colored twins. If we are given a partition of the vertices into colored modules, we can decide the winner of Colored Arc Kayles or BW-Arc Kayles like Theorem 5. Different from ordinary neighborhood diversity,

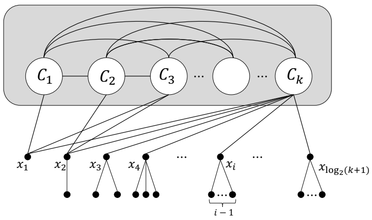

Fig. 3 The constructed graph G with neighborhood diversity $\nu = k + 2\log_2(k+1)$.

it might be hard to compute colored neighborhood diversity in polynomial time.

Theorem 6. *Given a graph $G = (V, E_{\mathrm{G}} \cup E_{\mathrm{B}} \cup E_{\mathrm{W}})$ with a partition of V into ν' modules of colored twins, we can compute the winner of Colored Arc Kayles on G in time $O^*((n/\nu'+1)^{\nu'})$.*

In the rest of this section, we give a bad instance for the proposed algorithm as shown in Figure 3, although the detailed proof is omitted. The result implies that the analysis of Theorem 5 is asymptotically tight.

Theorem 7. *There is a graph having at least $(n/\nu + 1 - o(1))^{\nu(1-o(1))}$ non-isomorphic positions of Arc Kayles.*

6. Arc Kayles for Trees

In [2], Bodlaender et al. show that the winner of Node Kayles on trees can be determined in time $O^*(3^{n/3}) = O(1.4423^n)$. It is easy to show by a similar argument that the winner of Arc Kayles can also be determined in time $O(1.4423^n)$. It is also mentioned that the analysis is sharp apart from a polynomial factor because there is a tree for which the algorithm takes $\Omega(3^{n/3})$ time. The example is also available for Arc Kayles; namely, as long as we use the same algorithm, the running time cannot be improved.

In this section, we present that the winners of Arc Kayles on trees can be determined in time $O^*(2^{n/2}) = O(1.4143^n)$, which is attained by considering a tree (so-called) unordered. Since a similar analysis can be applied to Node Kayles on trees, the winner of Node Kayles on trees can be determined in time $O^*(2^{n/2})$. We omit the proof for Node Kayles to avoid repetition.

Let us consider a tree $T = (V, E)$. By Sprague–Grundy theory, if all connected subtrees of T are enumerated, one can determine the winner of Arc Kayles. Furthermore, by Proposition 1, once a connected subtree T' is listed, we can ignore subtrees isomorphic to T'. Here we adopt isomorphism of rooted trees.

Definition 4. *Let $T = (V, E, r)$ and $T' = (V', E', r')$ be trees rooted at r and r', respectively. Then, T and T' are called* isomorphic *if for any pair of $u, v \in V$ there is a bijection $f : V \to V'$ such that $\{u, v\} \in E_i$ if and only if $\{f(u), f(v)\} \in E'_i$ and $f(r) = f(r')$.*

In the following, we estimate the number of non-isomorphic connected subgraphs of T based on isomorphism of rooted trees. For $T = (V, E)$ rooted at r, a connected subtree T' rooted at r is called an *AK-rooted subtree* of T, if there exists a matching $M \subseteq E$ such that $T[V \setminus M]$ consists of T' and isolated vertices. Note that M can be empty, AK-rooted subtree T' must contain root r of T, and the graph consisting of only vertex r can be an AK-rooted subtree.

Lemma 5. *Any tree rooted at r has $O^*(2^{n/2})(= O(1.4143^n))$ non-isomorphic AK-rooted subtrees rooted at r.*

We omit the proof here.

Theorem 8. *The winner of Arc Kayles on a tree with n vertices can be determined in time $O^*(2^{n/2}) = O(1.4143^n)$.*

Acknowledgments

This work is partially supported by JSPS KAKENHI JP17H01698, JP17K19960, JP19K21537, JP20H05967, JP21H05852, JP21K17707, JP21K19765, JP21K21283, JP22H00513.

References

[1] Bodlaender, H. L. and Kratsch, D.: Kayles and Nimbers, *Journal of Algorithms*, Vol. 43, No. 1, pp. 106–119 (2002).
[2] Bodlaender, H. L., Kratsch, D. and Timmer, S. T.: Exact algorithms for Kayles, *Theoretical Computer Science*, Vol. 562, pp. 165–176 (2015).
[3] Breuker, D., Uiterwijk, J. and van den Herik, H.: Solving 8 × 8 Domineering, *Theoretical Computer Science*, Vol. 230, No. 1, pp. 195–206 (2000).
[4] Bullock, N.: Domineering: Solving large combinatorial search spaces, *ICGA Journal*, Vol. 25, No. 2, pp. 67–84 (2002).
[5] Chen, J., Kanj, I. and Xia, G.: Improved upper bounds for vertex cover, *Theoretical Computer Science*, Vol. 411, No. 40-42, pp. 3736–3756 (2010).
[6] Conway, J. H.: *On numbers and games*, CRC Press (2000).
[7] Dudeney, H.: *The Canterbury Puzzles*, Courier Corporation (2002).
[8] Gardner, M.: Mathematical Games: Cram, crosscram and quadraphage: new games having elusive winning strategies, *Scientific American*, Vol. 230, No. 2, pp. 106–108 (1974).
[9] Huggan, M. A. and Stevens, B.: Polynomial Time Graph Families for Arc Kayles., *Integers*, Vol. 16, p. A86 (2016).
[10] Kobayashi, Y.: On Structural Parameterizations of Node Kayles, *Proceedings of the 21st Japanese Conference on Discrete and Computational Geometry and Graphs (JCDCGGG 2018)*, Lecture Notes in Computer Science, Vol. 13034, Springer, pp. 96–105 (2018).
[11] Lachmann, M., Moore, C. and Rapaport, I.: Who Wins Domineering on Rectangular Boards?, *arXiv preprint math/0006066* (2000).
[12] Lampis, M.: Algorithmic Meta-theorems for Restrictions of Treewidth, *Algorithmica*, Vol. 64, No. 1, pp. 19–37 (2012).
[13] Lampis, M. and Mitsou, V.: The Computational Complexity of the Game of Set and Its Theoretical Applications, *Proceedings of the 11th Latin American Symposium on Theoretical Informatics (LATIN 2014)*, Lecture Notes in Computer Science, Vol. 8392, Springer, pp. 24–34 (2014).
[14] Schaefer, T. J.: On the Complexity of Some Two-person Perfect-information Games, *Journal of Computer and System Sciences*, Vol. 16, No. 2, pp. 185–225 (1978).
[15] Uiterwijk, J. W.: 11×11 Domineering Is Solved: The First Player Wins, *International Conference on Computers and Games*, Springer, pp. 129–136 (2016).
[16] Uiterwijk, J. W.: Construction and investigation of Cram endgame databases, *ICGA Journal*, Vol. 40, No. 4, pp. 425–437 (2018).
[17] Uiterwijk, J. W.: Solving Cram Using Combinatorial Game Theory, *Advances in Computer Games*, Springer, pp. 91–105 (2019).
[18] Uiterwijk, J. W. and Barton, M.: New results for Domineering from combinatorial game theory endgame databases, *Theoretical Computer Science*, Vol. 592, pp. 72–86 (2015).
[19] Yoshiwatari, K., Kiya, H., Hanaka, T. and Ono, H.: Winner Determination Algorithms for Graph Games with Matching Structures, *Combinatorial Algorithms* (Bazgan, C. and Fernau, H., eds.), Cham, Springer International Publishing, pp. 509–522 (2022).

