

Higher Hardness Results for the Reconfiguration of Odd Matchings

Joseph Dorfer  

Graz University of Technology, Austria

Abstract

We study the reconfiguration of odd matchings of combinatorial graphs. Odd matchings are matchings that cover all but one vertex of a graph. A reconfiguration step, or flip, is an operation that matches the isolated vertex and, consequently, isolates another vertex. The flip graph of odd matchings is a graph that has all odd matchings of a graph as vertices and an edge between two vertices if their corresponding matchings can be transformed into one another via a single flip.

We show that computing the diameter of the flip graph of odd matchings is Π_2^p -hard. This complements a recent result by Wulf [FOCS25] that it is Π_2^p -hard to compute the diameter of the flip graph of perfect matchings where a flip swaps matching edges along a single cycle of unbounded size.

Further, we show that computing the radius of the flip graph of odd matchings is Σ_3^p -hard. The respective decision problems for the diameter and the radius are also complete in the respective level of the polynomial hierarchy. This shows that computing the radius of the flip graph of odd matchings is provably harder than computing its diameter, unless the polynomial hierarchy collapses.

Finally, we reduce SET COVER to the problem of finding shortest flip sequences. As a consequence, we show APX-hardness and that the problem cannot be approximated by a sublogarithmic factor. By doing so, we answer a question asked by Aichholzer, Brenner, Dorfer, Hoang, Perz, Rieck, and Verciani [GD25].

2012 ACM Subject Classification Mathematics of computing → Combinatorics; Mathematics of computing → Combinatoric problems; Mathematics of computing → Discrete mathematics

Keywords and phrases Graph Reconfiguration Problems, Flip Graphs, Polynomial Hierarchy, APX-hardness

Digital Object Identifier 10.4230/LIPIcs.STACS.2026.33

Related Version *Full Version*: <https://arxiv.org/abs/2602.09573>

Funding Austrian Science Foundation (FWF) 10.55776/DOC183.

Acknowledgements First, I would like to thank Lasse Wulf for introducing me to the relation between the polynomial hierarchy and diameter computation for flip graphs. Further, I would like to thank Oswin Aichholzer, Sofia Brenner, Hung Hoang, Daniel Perz, Christian Rieck, and Francesco Verciani for our previous joint work on odd matchings. Finally, I want to thank the reviewers for many helpful comments that helped improving the argumentation throughout the paper.

1 Introduction

Reconfiguration describes the process of changing one structure into another. It is often performed by small, reversible steps, so called *flips*. Reconfiguration has many applications in the areas of optimization [15] or enumeration [6, 18]. We refer to the following surveys for the discussion of additional applications of reconfiguration [19, 23]. Very recently, reconfiguration has also provided substantial insight into the complexity of computing the diameter of polytopes [20, 21], spiking in the result that computing the combinatorial diameter of a polytope is Π_2^p -hard [25]. Remarkably, all the recent results on the diameter of polytopes study the reconfiguration of perfect matchings.



© Joseph Dorfer;

licensed under Creative Commons License CC-BY 4.0

43rd International Symposium on Theoretical Aspects of Computer Science (STACS 2026).

Editors: Meena Mahajan, Florin Manea, Annabelle McIver, and Nguyễn Kim Thăng

Article No. 33; pp. 33:1–33:16

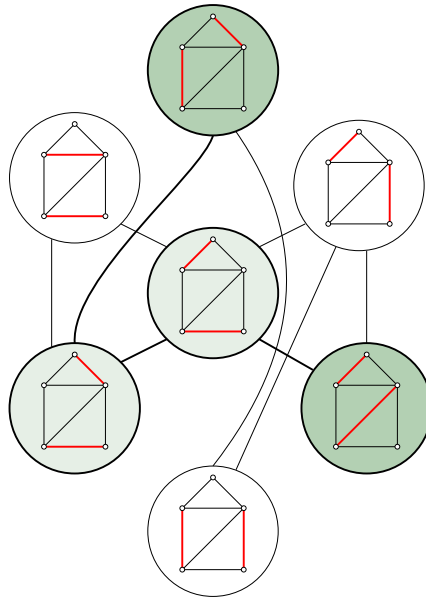


Leibniz International Proceedings in Informatics

LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



We study the related problem of the reconfiguration of *odd matchings* of graphs. An odd matching of a graph G is a matching consisting of edges of G such that all vertices of G are matched except for a single *isolated vertex*. A *flip* between two odd matchings is an operation that matches the isolated vertex v of the first matching to another vertex w . Subsequently, the vertex that previously shared an edge with w becomes the new isolated vertex. The *flip graph* of odd matchings of a graph G is a graph that has as vertex set all odd matchings of G and has edges between matchings whenever they can be transformed into one another via a single flip. A *flip sequence* between an initial matching M_{in} and a target matching M_{tar} is a sequence of matchings $M_{in} = M_0, M_1, \dots, M_{k-1}, M_k = M_{tar}$ such that consecutive matchings only differ by a single flip. In terms of the flip graph, a flip sequence is a path between two matchings. The index k denotes the *length* of a flip sequence. The *flip distance* between two odd matchings M_{in} and M_{tar} , denoted by $d(M_{in}, M_{tar})$, is the minimum over all k such that there exists a flip sequence of length k between M_{in} and M_{tar} . This can be interpreted as the length of a shortest path in the flip graph. The *diameter* of the flip graph is defined as $\max_{M_1, M_2} d(M_1, M_2)$ and describes the maximal flip distance between any pair of odd matchings in the flip graph. Similarly, the *radius* is defined as $\min_{M_1} \max_{M_2} d(M_1, M_2)$ and describes the minimum maximal distance of a matching M_2 to a *center* M_1 . Clearly, the diameter of a flip graph is bounded from below by its radius and bounded from above by twice the radius. For an illustration of many of the concepts, we refer to Figure 1.



■ **Figure 1** Illustration of a flip graph of odd matchings with a flip sequence of length 3 highlighted.

1.1 Related Work

Odd Matchings. The reconfiguration of odd matchings has been introduced in [2] for the setting of geometric odd matchings, that is, crossing-free odd matchings with straight line segments as edges between points in general position in the plane. The authors show that in this setting the flip graph is always connected with a diameter of $O(n^2)$ where n is the number of points. The study of combinatorial odd matchings has been initiated in [1]. The

authors provide a complete, polynomial time checkable characterization when the flip graph of odd matchings on a graph G is connected and show that any connected component of the flip graph has a diameter that is linear in the size of G . In the same paper, it is shown that the problem of deciding whether there exists a flip sequence of a certain length between two given odd matchings is NP-complete for both the geometric and the combinatorial setting. The reconfiguration of odd matchings also appears in form of the sliding block game gourds [13, 17] where the underlying graph is a triangular grid graph and the edges have colored or labeled end points. In every flip, the added edge inherits the colors or labels of the removed edge.

Perfect Matchings. For the reconfiguration of perfect matchings removing and adding a single edge will not yield another perfect matching. Instead a flip in a matching M works as follows: Pick a cycle that alternates between edges that lie in M and edges that do not. The flip then removes all edges of M along the cycle and adds the edges of the cycle that were not in M . We can either allow cycles of arbitrary length, or bound the length of the cycle. When bounding the length of the cycle to only allow cycles of length four such that a flip removes two edges from M and adds two new edges it is shown to be PSPACE-complete to decide whether a given matching can be flipped into another matching in the combinatorial setting [8]. In the geometric setting it is a long-standing open question whether any perfect matching on any point set can be transformed into any other matching on the same point set when only allowing flips along 4-cycles. Up to now, there is not even a published proof that a perfect matching on any point set permits a valid flip of such a form. It further has been shown that deciding whether a flip sequence of a certain length exists is NP-hard in the geometric setting [7].

If, however, we allow flips along alternating cycles of unbounded size, the flip graph is connected in the geometric setting [14]. In the combinatorial setting it has been shown that deciding whether the flip distance between two perfect matchings is at most k is NP-complete, even for $k = 2$ [3, 16]. There also has been particular interest in the complexity of computing the diameter of the flip graph of perfect matchings in the combinatorial setting [20, 21, 25] spiking in the result that computing the diameter is Π_2^P -complete [25]. The research was motivated by its implications for the complexity of computing the diameter of polytopes [9].

Computing Central Structures in Flip Graphs. This year's CG:SHOP challenge deals with the search of central structures in flip graphs of triangulations under parallel flip operations [4]. The problem as it is phrased asks for a central structure in the flip graph for a small (compared to the size of the flip graph) finite set of triangulations that are part of the input. The problem is contained in NP and can be interpreted as an attempt to approximate the search of the center and the radius of the whole flip graph.

1.2 Our Contributions

In this paper, we provide three novel higher complexity results on the reconfiguration of combinatorial odd matchings.

First, we complement the main result in [25] by showing that for a given graph G and a parameter $k \in \mathbb{N}$ the problem of deciding whether the diameter of the flip graph of odd matchings of G is at most k is Π_2^P -complete. We do so by reducing directly from the Π_2^P -complete problem $\forall\exists$ -SAT.

► **Theorem 1.** *Given a graph G and a parameter $k \in \mathbb{N}$. Deciding whether the diameter of the flip graph of odd matchings of G is at most k is Π_2^P -complete.*

As a second result, we study the related problem of calculating the radius of the flip graph. We show that deciding whether for a given graph G the radius of the flip graph of odd matchings of G is at most some value k is Σ_3^P -complete. We do so by reducing directly from the Σ_3^P -complete problem $\exists\forall\exists$ -SAT.

► **Theorem 2.** *Given a graph G and a parameter $k \in \mathbb{N}$. Deciding whether the radius of the flip graph of odd matchings of G is at most k is Σ_3^P -complete.*

By showing Σ_3^P -completeness of the problem, we provide a naturally occurring problem that is complete in this complexity class. As discussed in [12], problems in this complexity class are not too well studied and the list of problems that are complete for that class are not too long.

Even though the concepts of diameter and radius seem very similar, we conclude that it is provably harder to compute the radius than to compute the diameter, unless the polynomial hierarchy collapses.

The authors of [1] show that it is NP-hard to compute shortest flip sequences between odd matchings and motivate the question about the existence of approximation algorithms. As a final result, we settle the question about approximability in the combinatorial setting by reducing SET COVER to the problem of finding shortest flip sequences. We will describe the global idea for the reduction and modify it to fit the definition of a PTAS-reduction.

► **Theorem 3.** *Given a graph G and two odd matchings M_{in} and M_{tar} of G . Computing the flip distance between M_{in} and M_{tar} is APX-hard. Further, it is NP-hard to approximate the flip distance by a factor better than $\Theta(\log(n))$.*

Full technical details and proofs for statements marked with (★) will appear in a full version of the paper.

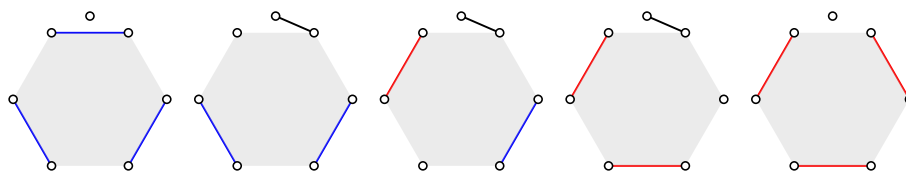
2 Preliminaries

2.1 Union of Odd Matchings

Let M_1 and M_2 be two odd matchings on the same graph G , then their union $M \cup M'$ admits the following connected components:

- one *alternating path* of even length (possibly zero) that connects the isolated vertices of M_1 and M_2 and alternates between edges of M_2 and M_1 ,
- *cycles* of even length alternating between M_1 and M_2 , and
- edges that lie in $M_1 \cap M_2$, called *happy edges*.

The above partition will be helpful when arguing lower bounds on the length of flip sequences. We will say that we *charge flips towards a component* if a flip sequence needs to perform that number of flips on this connected component. Some easy observations are: (1) If the alternating path contains k edges of M_1 and M_2 each, then we charge at least k flips towards the path; (2) If an alternating cycle contains k edges of M_1 and M_2 each, we charge at least $k + 1$ flips towards the cycle, k to flip all the edges of M_2 in and one additional flip to place the isolated vertex on the cycle, we will call these steps *switching a cycle* and (3) we either charge no flips at all towards a happy edge or at least two since if we remove the happy edge, we need to add it back in.



■ **Figure 2** Switching a cycle with three matching edges in four flips.

2.2 The Polynomial Hierarchy

The polynomial hierarchy was introduced by Stockmeyer [22] and provides a way to compare the complexity of problems beyond NP-hardness. Complexity classes are defined recursively. The lowest level is $\Sigma_0^P = \Pi_0^P = P$. Then, Σ_k^P for $k \geq 1$ is defined as the class of all problems that can be decided in non-deterministic polynomial time with the help of an oracle for the class Σ_{k-1}^P . Further, $\Pi_k^P = co\text{-}\Sigma_k^P$. In particular $\Sigma_1^P = \text{NP}$ and $\Pi_1^P = co\text{-}\text{NP}$.

We refer to [24] for a definition of the polynomial hierarchy that is easier to work with in our setting. A *language* is a set $L \subseteq \{0, 1\}^*$. A language L is contained in Π_2^P if there exists some polynomial-time computable function V such that for all $w \in \{0, 1\}^*$ for suitable $m_1, m_2 \leq poly(|w|)$:

$$w \in L \Leftrightarrow \forall y_1 \in \{0, 1\}^{m_1} \exists y_2 \in \{0, 1\}^{m_2} : V(w, y_1, y_2) = 1$$

Similarly, a language L is contained in Σ_3^P if there exists some polynomial-time function V such that for all $w \in \{0, 1\}^*$ for some suitable $m_1, m_2, m_3 \leq poly(|w|)$:

$$w \in L \Leftrightarrow \exists y_3 \in \{0, 1\}^* \forall y_2 \in \{0, 1\}^{m_2} \exists y_1 \in \{0, 1\}^{m_3} : V(w, y_1, y_2, y_3) = 1$$

In [24] problems are provided that are complete for the respective stages of the polynomial hierarchy. The problem $\forall\exists\text{-SAT}$ given by all Boolean formulas ϕ on variables x_1, \dots, x_{m_1} and y_1, \dots, y_{m_2} such that for all assignments of x_1, \dots, x_{m_1} there exists an assignment of y_1, \dots, y_{m_2} such that $\phi(x_1, \dots, x_{m_1}, y_1, \dots, y_{m_2}) = 1$ is Π_2^P -complete. Further, the problem $\exists\forall\exists\text{-SAT}$ given by all Boolean formulas ψ on variables $x_1, \dots, x_{m_1}, y_1, \dots, y_{m_2}, z_1, \dots, z_{m_3}$ such that there exists an assignment of x_1, \dots, x_{m_1} such that for all assignments of y_1, \dots, y_{m_2} there exists an assignment of z_1, \dots, z_{m_3} such that $\psi(x_1, \dots, x_{m_1}, y_1, \dots, y_{m_2}, z_1, \dots, z_{m_3}) = 1$ is Σ_3^P -complete. We assume without loss of generality that all Boolean formulas are given in conjunctive normal form (CNF).

2.3 APX-hardness

For an extensive introduction to the concepts see [10]. Let APX be the set of all problems in NP that allow for a constant factor approximation. A problem is *APX-hard* if there is a *PTAS-reduction* from every problem in APX to said problem. A PTAS-reduction from problem A to B is a set of three functions f, g, α , that are polynomial-time computable for a fixed ε such that:

- the function f maps an instance of A to an instance of B .
- the function g takes an instance x of A and an approximate solution of $f(x)$ in B and computes an approximation of x
- the function α maps error parameters of problems in A to corresponding parameters of problems in B
- if the solution y to $f(x)$ is a $1 + \alpha(\varepsilon)$ -approximation to the optimal solution, then $g(x, y, \varepsilon)$ is a $1 + \varepsilon$ solution to x .

In particular, if there exists no polynomial-time approximation scheme for A and B PTAS-reduces to A , then there is also no polynomial-time approximation scheme for B and if A cannot be approximated within some factor $1 + \varepsilon$ then B cannot be approximated within $1 + \alpha(\varepsilon)$.

2.4 Set Cover

Consider the integers from 1 to n and a collection of sets $S = \{s_1, \dots, s_t\}$ such that $s_i \subseteq \{1, \dots, n\}$ and $\bigcup_{i=1}^t s_i = \{1, \dots, n\}$. The SET COVER problem asks for a given integer k whether there exists a subset $S' \subseteq S$ such that $|S'| = k$ and $\bigcup_{s_i \in S'} s_i = \{1, \dots, n\}$. SET COVER is known to be APX-hard, since for example SET COVER problems are a superset of the vertex cover problem [5]. Further, the size of a smallest set S^* that covers all integers in $\{1, \dots, n\}$ can in general not be approximated by a sublogarithmic factor [11], unless $P=NP$.

3 Computing the Diameter is Π_2^P -complete

We reduce directly from $\forall\exists$ -SAT to computing the diameter of the flip graph. We will introduce gadgets for clauses and for each type of variable.

As a high level idea, the goal of a flip sequence will be to switch the cycles in all clause gadgets by switching the cycles in all the variable gadgets in a way that places isolated vertices next to clause gadgets. The alternating cycle of an \exists -gadget can then be switched in two ways depending on which of the two edges from a central vertex v to the gadget is used to enter the gadget. Each direction for the switch will correspond to an assignment of the variable based on what clauses the isolated vertex is placed next to. In a \forall -gadget, there is only one way to enter the gadget from v , so the way to switch the cycle is fixed. If the cycle is not crossing in the drawing, the isolated vertex will be placed next to gadgets of clauses that contain the positive literal, and, if the cycle is crossing, next to clauses containing the negative literal.

We now introduce all gadgets of our reduction in more detail. The reduction will be built around one, aforementioned, central vertex v .

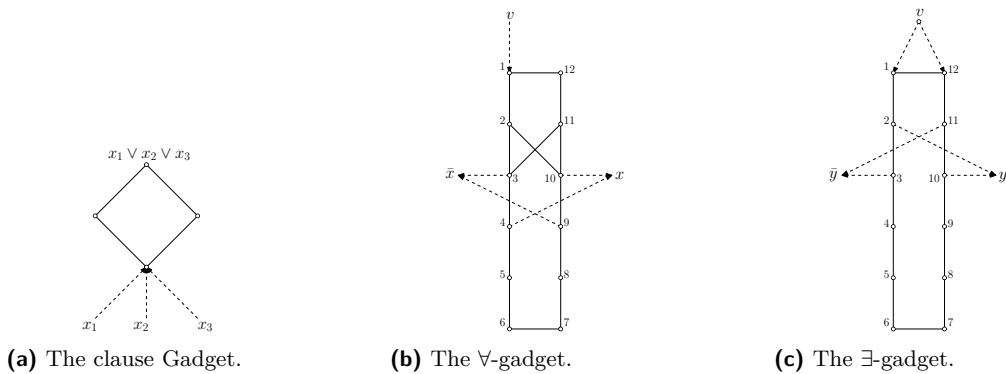
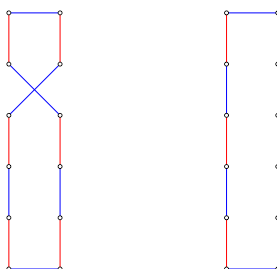


Figure 3 Gadgets of the reduction.

Clause gadget (Figure 3a): The clause gadget is a 4-cycle that has one vertex which is incident to the later introduced variable gadgets. The idea is to force a perfect matching of the cycle in the initial matching and the other perfect matching in the target matching such that all alternating cycles in variable gadgets need to be switched at some point.

\forall -gadget (Figure 3b): For a given variable x , the variable gadget consists of a cycle of length twelve, with vertices labeled 1 to 12 along the cycle, as well as two additional diagonals from vertex 2 to vertex 10 and vertex 3 to vertex 11 in the cycle that form the crossing as seen in Figure 3b. The vertex with label 1 has an edge that is connected to a central vertex v . Two vertices, labeled 3 and 9, have edges to all clause gadgets that correspond to clauses that contain \bar{x} . Two vertices, labeled 4 and 10, are incident to edges to all clause gadgets that correspond to clauses that contain x . If $M_{in} \cup M_{tar}$ contains an alternating cycle on a \forall -gadget, then this cycle can either contain the introduced crossing or not. To see a crossed and uncrossed cycle, see Figure 4. The two options will encode the truth value of the corresponding variable.



■ **Figure 4** Left: a crossed alternating cycle in a \forall -gadget, Right: an uncrossed alternating cycle.

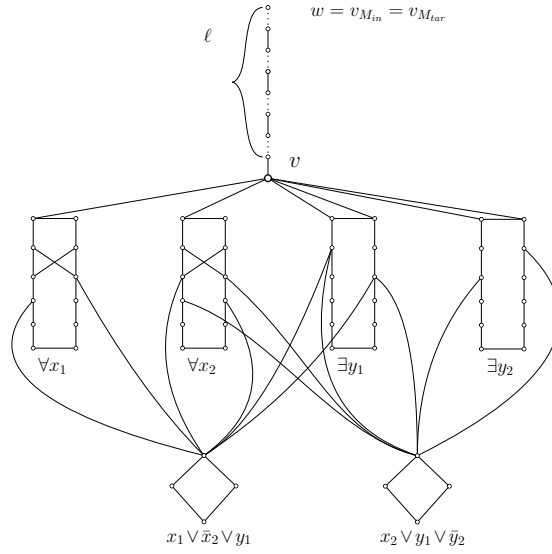
\exists -gadget (Figure 3c): The \exists -gadget corresponding to a variable y consists of a cycle with twelve edges and vertices labeled 1 to 12 along the cycle. Two adjacent vertices, labeled 1 and 12 are incident to an edge to the vertex v . Two vertices, labeled 2 and 10 are incident to edges to all gadgets of clauses that contain y . Similarly, two vertices, labeled 3 and 11 are connected to gadgets of clauses that contain \bar{y} . If the union $M_{in} \cup M_{tar}$ contains an alternating cycle on an \exists -gadget, the cycle can be switched in two ways depending on which of the two edges incident to v is used to switch the cycle. These two choices will encode the two truth values of the corresponding variable.

Forcing the position of the isolated vertex: If we take s to be the number of vertices in all clause gadgets, \forall -gadgets and \exists -gadgets combined, we obtain by [1, Theorem 10] that any connected component of the flip graph of odd matchings of the constructed graph has diameter at most $c \cdot (s + 1)$ for a constant c . We attach to v a path P of length $\ell = 2c \cdot (s + 1)$ (See Figure 5). If the isolated vertex of at least one of the matchings is placed at the vertex w of P that is farthest away from v , then flipping edges along the path already takes $\frac{\ell}{2}$ flips, which is at least as much as it takes to reconfigure two odd matchings if both have their isolated vertex placed on some clause gadget, \forall -gadget, or \exists -gadget.

The reduction: Now, let ϕ be a Boolean formula on variables x_1, \dots, x_{m_1} and y_1, \dots, y_{m_2} . We construct a graph G as follows: Start from a single vertex v . For every x_i introduce a \forall -gadget and connect it to v as described above. For every y_i introduce an \exists -gadget and connect it to v as described. Further, introduce a clause gadget for every clause C in ϕ and connect each clause gadget to all the variable gadgets that correspond to variables in the clause. The connection happens in one of two ways described above depending on whether $x \in C$ or $\bar{x} \in C$ (resp. y or z). Then add the path of length ℓ at v to obtain a final graph G' .

First, we need to make sure that the diameter is well defined in this setting.

► **Proposition 4** (\star). *The flip graph of odd matchings of G' is connected.*



■ **Figure 5** The instance graph for the quantified formula $\forall x_1, x_2 \exists y_1, y_2 : (x_1 \vee \bar{x}_2 \vee y_1) \wedge (x_2 \vee y_1 \vee \bar{y}_2)$.

We show Proposition 4 by showing that G' fulfills the characterization of graphs for which the flip graph is connected from [1, Theorem 3].

From now on, we will call a pair of matchings M_{in} and M_{tar} such that $d(M_{in}, M_{tar})$ equals the diameter of the flip graph a *maximizing pair*. We will make structural observations on how a maximizing pair looks like. Also, let $v_{M_{in}}$ and $v_{M_{tar}}$ denote the isolated vertices of M_{in} and M_{tar} .

► **Lemma 5** (\star). *Let M_{in} and M_{tar} be a maximizing pair. Then:*

- (1) $v_{M_{in}} = v_{M_{tar}}$ is the vertex w of P that is farthest away from v .
- (2) $M_{in} \cup M_{tar}$ contains an alternating cycle on every clause gadget.
- (3) $M_{in} \cup M_{tar}$ contains an alternating cycle on every \exists -gadget.
- (4) $M_{in} \cup M_{tar}$ contains an alternating cycle on every \forall -gadget.

► **Theorem 1.** *Given a graph G and a parameter $k \in \mathbb{N}$. Deciding whether the diameter of the flip graph of odd matchings of G is at most k is Π_2^P -complete.*

Proof. Containment in Π_2^P follows from the definition of Π_2^P and the fact that the underlying flip distance problem that has to be solved for every pair of matchings is contained in NP.

Let ϕ on variables x_1, \dots, x_{m_1} and y_1, \dots, y_{m_2} and clauses C_1, \dots, C_K be a $\forall\exists$ -SAT instance and G' be the graph as constructed above.

▷ **Claim 6.** If ϕ is a YES-instance of $\forall\exists$ -SAT, then the diameter of the flip graph of odd matchings of G' is at most $\ell + 7(m_1 + m_2) + 3K$.

Proof. Assume ϕ is a YES-instance. In order to upper bound the diameter of the flip graph of odd matchings of G' it is sufficient to consider pairs of matchings that fulfill the conditions of Lemma 5. Let M_{in} and M_{tar} be a pair with these properties. We consider $M_{in} \cup M_{tar}$ on every \forall -gadget for some variable x_i . If they form an uncrossed cycle, then we consider the positive assignment of x_i , otherwise the negative assignment. Since ϕ is a YES-instance of $\forall\exists$ -SAT, for the given assignment of the x_i there exists a satisfying assignment of the y_i such that $\phi(x_1, \dots, x_{m_1}, y_1, \dots, y_{m_2}) = 1$. We can construct a flip sequence from M_{in} to M_{tar} based on the assignment of y_1, \dots, y_{m_2} .

The flip sequence starts by moving the isolated vertex from w to v . Then the flip sequence switches all cycles that belong to \forall -gadgets and if the isolated vertex is placed next to an unswitched clause gadget, the flip sequence switches this cycle as well. Afterwards, the flip sequence switches all \exists -gadgets. If y_i is assigned a positive (resp. negative) value, then the flip sequence traverses the gadget such that the isolated vertex is placed next to clause gadgets containing y_i (resp. \bar{y}_i). Since $\phi(x_1, \dots, x_{m_1}, y_1, \dots, y_{m_2}) = 1$, every clause gadget will eventually be switched. The flip sequence ends by moving the isolated vertex back from v to w . There are ℓ flips for moving the isolated vertex between w and v , seven flips per variable gadget and three flips per clause gadget, which gives us the aspired length of the flip sequence. \triangleleft

\triangleright **Claim 7.** ϕ is a YES-instance of $\forall\exists$ -SAT whenever the diameter of the flip graph of odd matchings of G' is at most $\ell + 7(m_1 + m_2) + 3K$.

Proof. Assume the diameter of the flip graph of odd matchings is at most $\ell + 7(m_1 + m_2) + 3K$. For a given assignment of x_1, \dots, x_{m_1} , we construct a pair of matchings M_{in} and M_{tar} . If x_i appears in its positive form, $M_{in} \cup M_{tar}$ has an uncrossed cycle in the \forall -gadget that corresponds to x_i , otherwise $M_{in} \cup M_{tar}$ has a crossed cycle. Additionally, $M_{in} \cup M_{tar}$ has an alternating cycle on all \exists -gadgets and clause gadgets, and $v_{M_{in}} = v_{M_{tar}} = w$.

Since there exists a flip sequence from M_{in} to M_{tar} of length at most $\ell + 7(m_1 + m_2) + 3K$, exactly seven flips are performed in every variable gadget and exactly three flips are performed in every clause gadget. Less flips would not suffice to switch the cycle in the respective gadget. With more flips we would exceed the length of the flip sequence. This means there is an isolated vertex placed next to every clause gadget at some point in a flip sequence that switches every variable gadget once. We construct an assignment of y_1, \dots, y_{m_2} from where the isolated vertex was placed during the traversal of the corresponding variable gadgets. This assignment satisfies $\phi(x_1, \dots, x_{m_1}, y_1, \dots, y_{m_2}) = 1$. Repeating this procedure for all assignments of x_1, \dots, x_{m_1} shows that ϕ is a YES-instance of $\forall\exists$ -SAT. \triangleleft

The theorem follows from a combination of the two claims. \blacktriangleleft

4 Computing the Radius is Σ_3^P -complete

We reduce from $\exists\forall\exists$ -SAT to computing the radius of the flip graph. We now introduce all gadgets for our reduction that will again be built around one central vertex v .

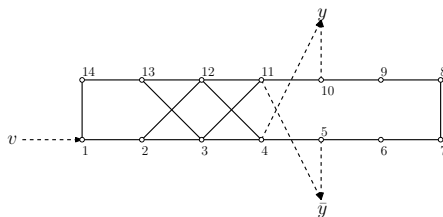


Figure 6 \forall -gadget for the reduction from $\exists\forall\exists$ -SAT to computing the radius.

Clause gadget (Figure 3a): The clause gadget is the same as in Section 3.

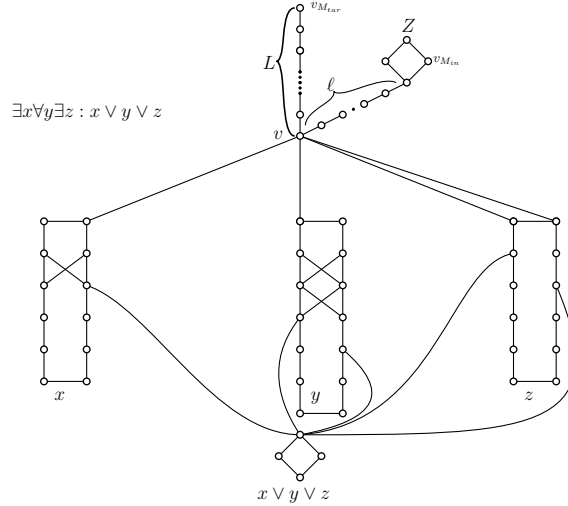
First \exists -gadget (Figure 3b): The first \exists -gadget is the same as the \forall -gadget of Section 3.

\forall -gadget (Figure 6): The \forall -gadget corresponding to a variable y consists of a cycle with 14 edges and vertices labeled 1 to 14 along the cycle, as well as four diagonals, from 2 to 12, 3 to 11, 4 to 12, and 3 to 13, of the cycle that form the two crossings in Figure 6.

33:10 Higher Hardness Results for the Reconfiguration of Odd Matchings

Two vertices, 4 and 10, have edges to clauses that contain the positive literal y and two vertices, 5 and 11, with edges to clauses containing the negative literal \bar{y} and vertex 1 has an edge from the vertex v .

Second \exists -gadget (Figure 3c): The second exists gadget coincides with the \exists -gadget of Section 3. Notation wise, the variables corresponding to those gadgets will now be called z instead of y .



■ **Figure 7** Instance graph for the quantified formula $\exists x \forall y \exists z : x \vee y \vee z$.

Forcing position of the isolated vertices: We conclude the construction by attaching two paths to v . Let s be the number of vertices in clause and variable gadgets. We set $\ell = 2c \cdot (s + 1)$ where c is chosen according to [1, Theorem 10] such that ℓ is larger than twice the diameter of the flip graph of odd matchings on v and the clause and variable gadgets. We add a path p of length ℓ and attach one end to v and to the other end we attach a four-cycle Z . Further, let $L = 2c \cdot (s + \ell + 1)$ and add a path P of length L and attach one end of P to v . No matter what a matching M_{in} in the center of the flip graph looks like, a matching M_{tar} can maximize the distance to M_{in} by having an isolated vertex at $v_{M_{tar}}$, the end vertex of P that is not attached to v and by differing from M_{in} on Z . By doing so, any flip sequence is forced to traverse both p and P . Vice versa, by having the isolated vertex of M_{in} placed on Z it can be guaranteed that a flip sequence traverses p only once instead of twice.

The reduction: Now, let ψ be a Boolean formula on variables x_1, \dots, x_{m_1} , y_1, \dots, y_{m_2} and z_1, \dots, z_{m_3} . We construct a graph G as follows: Start from a single vertex v . For every x_i introduce a first \exists -gadget and connect it to v as described above. Repeat for all y_i and \forall -gadgets and all z_i and the second \exists -gadget. Further, for every clause C in ψ introduce a clause gadget and connect the gadget to all the variable gadgets that correspond to variables in the clause. The connection happens in one of two described ways depending whether $x \in C$ or $\bar{x} \in C$ (resp. y or z). Then add the paths p and P as described to obtain a final graph G'' . Again, we need to make sure that the radius is well defined in this setting.

► **Proposition 8** (*). *The flip graph of odd matchings on G'' is connected.*

Again, we verify this by checking the characterization from [1, Theorem 3].

We will again give a characterization of pairs of matchings, which can determine the radius. We will argue that, without loss of generality, we do not have to consider all other pairs of matchings but only a set of candidates. As a high level argument, we make sure that in a candidate pair we cannot make a change to M_{tar} locally in a single gadget such that the flip distance to M_{in} increases and we cannot immediately make a change to M_{in} in a single gadget such that the flip distance decreases even if M_{tar} is changed locally to respond to the changes to M_{in} .

► **Lemma 9** (\star). *Without loss of generality, to determine the radius of the flip graph, it is sufficient to consider pairs of matchings M_{in} and M_{tar} that have the following properties:*

- (1) $v_{M_{tar}}$ is the vertex on P that is farthest away from v .
- (2) $v_{M_{in}}$ is placed on Z .
- (3) $M_{in} \cup M_{tar}$ contains an alternating cycle on every clause gadget.
- (4) $M_{in} \cup M_{tar}$ contains an alternating cycle on every variable gadget.

With Lemma 9 in mind, we describe the high level strategies for M_{in} , M_{tar} and a shortest flip sequence:

Since all clause gadgets contain alternating cycles, a shortest flip sequence will have to switch them all by entering them through a variable gadget. Adjacency relations of gadgets correspond to containment relations of variables in clauses.

For the first \exists -gadget, while switching an uncrossed cycle, the isolated vertex will be next to clause gadgets containing the positive literal, otherwise, if the cycle is crossed, the clauses with negative literals can be switched. Also note, that M_{in} can determine, whether the cycle is crossed or uncrossed. M_{in} may choose not to control the existence of the crossing if the value of a particular variable does not matter.

For the \forall -gadget, while switching the cycle, the isolated vertex is next to clauses with positive literals if the cycle is crossing twice or not at all, however if it crosses once, the isolated vertex is placed next to clauses with negative literals. While M_{in} can control to add one crossing to the cycle or not, the existence of a second crossing will then always be controlled by M_{tar} .

For the second \exists -gadget, there is only one option for an alternating cycle up to swapping M_{in} and M_{tar} . The direction of traversal is then chosen by the flip sequence depending on which of the two edges incident to v is chosen to enter the gadget. The direction of traversal then determines whether the isolated vertex will be placed next to clauses with positive or negative literals.

► **Theorem 2.** *Given a graph G and a parameter $k \in \mathbb{N}$. Deciding whether the radius of the flip graph of odd matchings of G is at most k is Σ_3^P -complete.*

Proof. Containment in Σ_3^P follows from the definition of Σ_3^P and the fact that the underlying flip distance problem that has to be solved for every matching M_{tar} is contained in NP.

Let ψ on variables x_1, \dots, x_{m_1} , y_1, \dots, y_{m_2} and z_1, \dots, z_{m_3} and clauses C_1, \dots, C_K be a $\exists\forall\exists$ -SAT instance and G'' be the graph as constructed above.

▷ **Claim 10.** *If ψ is a YES-instance of $\exists\forall\exists$ -SAT, then the radius of the flip graph of odd matchings of G'' is at most $\frac{L+\ell}{2} + 7(m_1 + m_3) + 8m_2 + 3K + 2$.*

Proof. We construct M_{in} based on the truth assignment of x_1, \dots, x_{m_1} that is part of a solution of ψ . If x_i has a positive truth value, then we match its corresponding variable gadget such that any completion of the matching to an alternating cycle is uncrossed. If x_i has a negative truth value, match the variable gadget such that the cycle has a crossing. Place the isolated vertex on Z and match the remaining vertices to form an arbitrary perfect matching. The existence of such perfect matchings is shown as a part of the proof of Proposition 8.

33:12 Higher Hardness Results for the Reconfiguration of Odd Matchings

Now for any M_{tar} that maximizes the distance to M_{in} , we look at the \forall -gadgets. By Lemma 9 we know that $M_{in} \cup M_{tar}$ forms an alternating cycle on the \forall -gadget. If the cycle in the gadget to some y_i is crossed zero times or twice, we translate this to a positive assignment of y_i , if it is crossed once, translate to a negative assignment of y_i .

Since ψ is a YES-instance of $\exists\forall\exists$ -SAT there exists an assignment of z_1, \dots, z_{m_3} such that combined with the initial assignment of x_1, \dots, x_{m_1} and the translated assignment of y_1, \dots, y_{m_2} it holds that $\psi(x_1, \dots, x_{m_1}, y_1, \dots, y_{m_2}, z_1, \dots, z_{m_3}) = 1$. We build our flip sequence between M_{in} and M_{tar} around the assignment of the z_i 's.

A flip sequence from M_{in} to M_{tar} looks as follows: Perform flips to make the matchings coincide on Z in up to two flips (since the isolated vertex is already placed on Z). Move the isolated vertex along p to v in $\frac{\ell}{2}$ flips. Switch all of the first \exists -gadgets using seven flips per gadget and if the isolated vertex is placed next to an unswitched clause gadget, switch it using three flips. Switch all \forall -gadgets using eight flips per gadget and if the isolated vertex is placed next to an unswitched clause gadget, switch it using three flips. For every \exists -gadget that corresponds to some z_i switch the \exists -gadget such that if z_i has a positive truth value, the isolated vertex is placed next to clauses that contain the positive literal z_i . However, if z_i has a negative truth value, the isolated vertex should be placed next to clauses that contain the negative literal \bar{z}_i . Every switch of a \exists -gadget takes again seven flips. In the end, move the isolated vertex along P using $\frac{L}{2}$ flips.

The total length of the flip sequence adds up to $\frac{L+\ell}{2} + 7(m_1 + m_3) + 8m_2 + 3K + 2$. \triangleleft

\triangleright **Claim 11.** ψ is a YES-instance of $\exists\forall\exists$ -SAT whenever the radius of the flip graph of odd matchings of G'' is at most $\frac{L+\ell}{2} + 7(m_1 + m_3) + 8m_2 + 3K + 2$.

Proof. Let M_{in} be the center of the flip graph of matchings of G'' . In particular, the flip distance from M_{in} to any other odd matching of G'' is at most the radius.

Consider the first set of variable gadgets corresponding to x_1, \dots, x_{m_1} and see, in which way they are matched. If the gadget corresponding to x_i is matched, such that any completion to an alternating cycle is uncrossed, assign a positive truth value to x_i , if any completion to an alternating cycle is crossed, assign a negative value. It can happen that the matching can be completed to both a crossed and uncrossed alternating cycle, in that case assign an arbitrary truth value.

Now, let there be an arbitrary truth assignment of y_1, \dots, y_{m_2} . We construct M_{tar} based on this assignment. In every \forall -gadget, we complete M_{in} to an alternating cycle that has zero or two crossings if the corresponding y_i has a positive assignment, and otherwise to have one crossing if y_i has a negative assignment. On all other gadgets, complete M_{in} arbitrarily to an alternating cycle. Let M_{tar} have a perfect matching on Z such that M_{tar} and M_{in} differ on Z . Let M_{tar} have its isolated vertex at the very end of P and complete M_{tar} with a perfect matching on the remainder of P and p .

Now take a flip sequence from M_{in} to M_{tar} of length at most $L + \ell + 7(m_1 + m_3) + 8m_2 + 3K + 2$. By our observations on lower bounds it follows that the flip sequence has to spend exactly seven flips on every \exists -gadget, eight flips on every \forall gadget and three flips on every clause gadget. We reconstruct an assignment of z_1, \dots, z_{m_3} from how their corresponding gadgets are switched. Since, every clause gadget has been flipped in the flip sequence, every clause contains at least one literal which has the right truth value assigned by the assignment from our constructions. Therefore $\psi(x_1, \dots, x_{m_1}, y_1, \dots, y_{m_2}, z_1, \dots, z_{m_3}) = 1$. Since we can do this for any assignments of y_1, \dots, y_{m_2} , this shows that ψ is a YES-instance of $\exists\forall\exists$ -SAT. \triangleleft

The theorem then follows from the combination of the two claims. \blacktriangleleft

5 Computing the Flip Distance is APX-hard

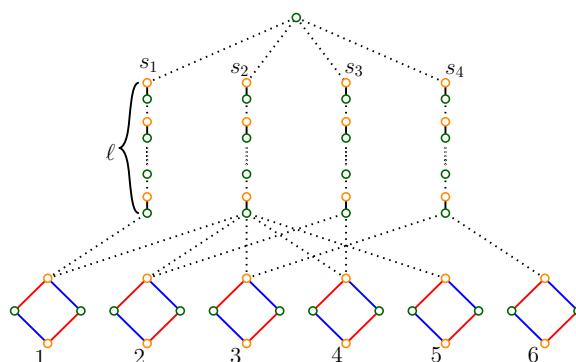


Figure 8 Instance for the hardness-proof. A SET COVER instance with $n = 6$, $s_1 = \{1\}$, $s_2 = \{1, 2, 3, 4, 5\}$, $s_3 = \{2, 4\}$, and $s_4 = \{3, 6\}$ is reduced. Edges in $M_{in} \setminus M_{tar}$ are colored red, edges from $M_{tar} \setminus M_{in}$ blue, and edges in $M_{in} \cap M_{tar}$ black.

The reduction from SET COVER to finding minimum flip sequences is illustrated in Figure 8. For every set in S from the SET COVER instance we add a path with ℓ vertices to the graph G''' , for some even ℓ that will be fixed later. We construct M_{in} and M_{tar} such that they form a perfect matching of the paths. For every integer from 1 to n (that denote the elements within the sets), we add a 4-cycle to G''' such that M_{in} and M_{tar} cover the cycle alternatingly. At last, we add one single vertex to G''' such that this vertex is the isolated vertex in both M_{in} and M_{tar} . For each path corresponding to a set s_i , we add an edge from one end of the path to the isolated vertex and edges from the other end to all gadgets that correspond to integers that are contained in s_i . The strategy for a flip sequence is to traverse a set gadget in order to be able to flip the integer gadgets for integers that are contained in the set. After that the flip sequence has to traverse the set gadget back to the initial isolated vertex. Since G''' is bipartite (see green/orange partition in Figure 8), there are no shortcuts that skip the traversal of set gadgets.

► **Proposition 12.** *There exists a set cover of size c if and only if $d(M_{in}, M_{tar}) \leq c \cdot \ell + 3 \cdot n$.*

Proof. Assume there exists a set cover of size c . To transform M_{in} to M_{tar} all cycles that correspond to an integer from 1 to n need to be switched. The only way to switch such a cycle is to flip along a path that corresponds to a set s_i that is incident to e to place the isolated vertex next to the integer gadget and then switch the cycle. At last, add all edges from the path that corresponds to s_i back to the isolated vertex. If we repeat this for all vertices s_i in a set cover, all cycles in integer gadgets will be switched in the end and we reached M_{tar} . Adding and removing the happy edges along a set gadget takes ℓ flips. Switching an integer gadget takes three flips. So the total length of the flip sequence is $c \cdot \ell + 3 \cdot n$.

For the opposite direction, we first need to see that we indeed have to take the detour back to the isolated vertex. Observe that G''' is a bipartite graph. In Figure 8 we give a 2-coloring of G''' with two colours, orange and green. The isolated vertex can then only be placed on green vertices. This prevents flip sequences from taking any shortcuts between gadgets, by placing the isolated vertex directly on a different path when leaving an integer gadget.

33:14 Higher Hardness Results for the Reconfiguration of Odd Matchings

Now assume that we have a flip sequence from M_{in} to M_{tar} of length at most $c \cdot \ell + 3 \cdot n$. All cycles in integer gadgets have been switched. In particular, every cycle had the isolated vertex placed next to it at some point of the flip sequence. We set S^* to be the set of all sets whose gadget contained the isolated vertex next to an integer gadget at some point. S^* is clearly a set cover. By our initial observation, in order to place an isolated vertex in a set gadget next to an integer gadget, we need to charge at least ℓ flips towards the set gadget. Therefore, we get that $|S^*| \leq \frac{c \cdot \ell + 3 \cdot n - 3 \cdot n}{\ell} = c$ ◀

► **Theorem 3.** *Given a graph G and two odd matchings M_{in} and M_{tar} of G . Computing the flip distance between M_{in} and M_{tar} is APX-hard. Further, it is NP-hard to approximate the flip distance by a factor better than $\Theta(\log(n))$.*

Proof. Assume, there exists an efficient algorithm A that approximates the flip distance between any two odd matchings on the same graph up to a factor of $(k - \varepsilon)$ for an arbitrary but constant k and $\varepsilon \in (0, k - 1)$.

For a given instance of the minimum set cover problem. Construct matchings M_{in} and M_{tar} as described and set $\ell = 6 \lceil \frac{n}{\varepsilon} \rceil$. Then, A computes a flip sequence between M_{in} and M_{tar} of length $d < (k - \varepsilon)d^*$, where $d^* = d(M_{in}, M_{tar})$

By Proposition 12 we obtain a set cover of size $c \leq \frac{d - 3 \cdot n}{\ell}$.

Since $d < (k - \varepsilon)d^*$, we can express c in terms of the size of the minimum vertex cover c^* :

$$\begin{aligned} c &\leq \frac{(k - \varepsilon)d^* - 3 \cdot n}{\ell} = \frac{(k - \varepsilon)(c^* \cdot \ell + 3 \cdot n) - 3 \cdot n}{\ell} \\ &= (k - \varepsilon)c^* + \frac{(k - 1 - \varepsilon)\varepsilon}{2} \leq (k - \frac{\varepsilon}{2})c^* \end{aligned}$$

Therefore, for any approximation of the flip distance, we can compute an approximation of the size of the minimum set cover. In particular, if we approximate the flip distance with $k = o(\log(n))$, we obtain the same approximation for a minimum set cover, which is not possible, unless $P=NP$. ◀

6 Open Questions

We have shown that computing the diameter of the flip graph of combinatorial odd matchings is Π_2^P -complete and computing the radius of a the flip graph is Σ_3^P -complete. By doing so, we provide two naturally occurring problems that fall into these complexity classes. Further, by reducing SET COVER to the problem of finding shortest flip sequences between odd matchings we show that the problem is APX-hard and does not admit any global constant factor approximation.

The following open questions arise from our results:

1. Since the reconfiguration of matchings is closely tied to complexity results in polytopes we motivate the question whether it is Σ_3^P -hard to compute the combinatorial radius/center of a polytope. We remark that the result for the diameter of the flip graph in [25] uses a so called *canonical structure*, that is, a structure that can be reached from any other structure in a reasonable number of flips. An upper bound on the diameter then follows from flipping from any initial structure to the canonical structure and then to the target structure. Therefore, by computing the radius, a center of the flip graph is implicitly also computed. This means that so far computing the radius is only shown to be Π_2^P -hard.

2. Can similar results on the complexity of computing the diameter and radius also be shown in a geometric setting? While in the combinatorial setting, we can control, which edges are part of the input graph G and which are not, in the geometric setting all edges between any two points in the plane can theoretically occur in M_{in} , M_{tar} or any intermediate matching, the only degree of freedom for constructions is the placement of the points.

References

- 1 Oswin Aichholzer, Sofia Brenner, Joseph Dorfer, Hung P. Hoang, Daniel Perz, Christian Rieck, and Francesco Verciani. Flipping Odd Matchings in Geometric and Combinatorial Settings. In Vida Dujmović and Fabrizio Montecchiani, editors, *33rd International Symposium on Graph Drawing and Network Visualization (GD 2025)*, volume 357 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 12:1–12:18, Dagstuhl, Germany, 2025. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.GD.2025.12.
- 2 Oswin Aichholzer, Anna Brötzner, Daniel Perz, and Patrick Schnider. Flips in odd matchings. *Computational Geometry*, 129:102184, 2025. doi:10.1016/j.comgeo.2025.102184.
- 3 Oswin Aichholzer, Jean Cardinal, Tony Huynh, Kolja Knauer, Torsten Mütze, Raphael Steiner, and Birgit Vogtenhuber. Flip distances between graph orientations. *Algorithmica*, 83(1):116–143, 2021. doi:10.1007/s00453-020-00751-1.
- 4 Oswin Aichholzer, Joseph Dorfer, Sándor Fekete, Phillip Keldenich, Peter Kramer, Dominik Krupke, and Stefan Schirra. Cg:shop 2026: Central triangulation under parallel flip operations, 2025. URL: <https://cgshop.ibr.cs.tu-bs.de/competition/cg-shop-2026/#problem-description>.
- 5 Paola Alimonti and Viggo Kann. Some APX-completeness results for cubic graphs. *Theoretical Computer Science*, 237(1):123–134, 2000. doi:10.1016/S0304-3975(98)00158-3.
- 6 David Avis and Komei Fukuda. Reverse search for enumeration. *Discrete Applied Mathematics*, 65(1-3):21–46, 1996. doi:10.1016/0166-218X(95)00026-N.
- 7 Carla Binucci, Fabrizio Montecchiani, Daniel Perz, and Alessandra Tappini. Flipping matchings is hard, 2025. doi:10.48550/arXiv.2503.02842.
- 8 Marthe Bonamy, Nicolas Bousquet, Marc Heinrich, Takehiro Ito, Yusuke Kobayashi, Arnaud Mary, Moritz Mühlenthaler, and Kunihiko Wasa. The perfect matching reconfiguration problem. In *International Symposium on Mathematical Foundations of Computer Science (MFCS)*, pages 80:1–80:14, 2019. doi:10.4230/LIPIcs.MFCS.2019.80.
- 9 Vašek Chvátal. On certain polytopes associated with graphs. *Journal of Combinatorial Theory, Series B*, 18(2):138–154, 1975. doi:10.1016/0095-8956(75)90041-6.
- 10 Pierluigi Crescenzi. A short guide to approximation preserving reductions. In *Proceedings of the Twelfth Annual IEEE Conference on Computational Complexity, Ulm, Germany, June 24-27, 1997*. IEEE Comput. Soc, 1997. doi:10.1109/ccc.1997.612321.
- 11 Irit Dinur and David Steurer. Analytical approach to parallel repetition. In *Proceedings of the Forty-Sixth Annual ACM Symposium on Theory of Computing, STOC '14*, pages 624–633, New York, NY, USA, 2014. Association for Computing Machinery. doi:10.1145/2591796.2591884.
- 12 Christoph Grüne and Lasse Wulf. Completeness in the polynomial hierarchy for many natural problems in bilevel and robust optimization, 2024. arXiv:2311.10540.
- 13 Joep Hamersma, Marc J. van Kreveld, Yushi Uno, and Tom C. van der Zanden. Gourds: A sliding-block puzzle with turning. In *International Symposium on Algorithms and Computation (ISAAC)*, pages 33:1–33:16, 2020. doi:10.4230/LIPIcs.ISAAC.2020.33.
- 14 Michael E. Houle, Ferran Hurtado, Marc Noy, and Eduardo Rivera-Campo. Graphs of triangulations and perfect matchings. *Graphs and Combinatorics*, 21:325–331, 2005. doi:10.1007/s00373-005-0615-2.

33:16 Higher Hardness Results for the Reconfiguration of Odd Matchings

- 15 Takehiro Ito, Erik D. Demaine, Nicholas J. A. Harvey, Christos H. Papadimitriou, Martha Sideri, Ryuhei Uehara, and Yushi Uno. On the complexity of reconfiguration problems. *Theoretical Computer Science*, 412(12-14):1054–1065, 2011. doi:10.1016/J.TCS.2010.12.005.
- 16 Takehiro Ito, Naonori Kakimura, Naoyuki Kamiyama, Yusuke Kobayashi, and Yoshio Okamoto. Shortest reconfiguration of perfect matchings via alternating cycles. *SIAM Journal on Discrete Mathematics*, 36(2):1102–1123, 2022. doi:10.1137/20M1364370.
- 17 Naonori Kakimura and Yuta Mishima. Reconfiguration of labeled matchings in triangular grid graphs. In *International Symposium on Algorithms and Computation (ISAAC)*, pages 43:1–43:16, 2024. doi:10.4230/LIPIcs.ISAAC.2024.43.
- 18 Torsten Mütze. Combinatorial gray codes – an updated survey. *Electronic Journal of Combinatorics*, 2024. Dynamic survey. doi:10.37236/11023.
- 19 Naomi Nishimura. Introduction to reconfiguration. *Algorithms*, 11(4):52, 2018. doi:10.3390/A11040052.
- 20 Christian Nöbel and Raphael Steiner. Complexity of polytope diameters via perfect matchings. In *Proceedings of the 2025 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2234–2251. SIAM, 2025. doi:10.1137/1.9781611978322.74.
- 21 Laura Sanità. The diameter of the fractional matching polytope and its hardness implications. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 910–921. IEEE, 2018. doi:10.1109/FOCS.2018.00090.
- 22 Larry J. Stockmeyer. The polynomial-time hierarchy. *Theoretical Computer Science*, 3(1):1–22, 1976. doi:10.1016/0304-3975(76)90061-X.
- 23 Jan van den Heuvel. The complexity of change. In *Surveys in Combinatorics 2013*, volume 409 of *London Mathematical Society Lecture Note Series*, pages 127–160. 2013. doi:10.1017/CB09781139506748.005.
- 24 Celia Wrathall. Complete sets and the polynomial-time hierarchy. *Theoretical Computer Science*, 3(1):23–33, 1976. doi:10.1016/0304-3975(76)90062-1.
- 25 Lasse Wulf. Computing the polytope diameter is even harder than NP-hard (already for perfect matchings), 2025. doi:10.48550/arXiv.2502.16398.