Odd Paths, Cycles and T-joins: Connections and Algorithms

Ildikó Schlotter[1,2] and András Sebő[3]

[1]HUN-REN Centre for Economic and Regional Studies, Hungary; schlotter.ildiko@krtk.hun-ren.hu
[2]Budapest University of Technology and Economics, Hungary
[3]CNRS, Univ. Grenoble Alpes, France; andras.sebo@grenoble-inp.fr

Abstract

Minimizing the weight of an edge set satisfying parity constraints is a challenging branch of combinatorial optimization as witnessed by the binary hypergraph chapter of Alexander Schrijver's book [*Combinatorial Optimization*, Springer-Verlag, 2003, Chapter 80]. This area contains relevant graph theory problems including open cases of the Max Cut problem and some multiflow problems. We clarify the interconnections between some of these problems and establish three levels of difficulties. On the one hand, we prove that the Shortest Odd Path problem in undirected graphs without cycles of negative total weight and several related problems are *NP*-hard, settling a long-standing open question asked by Lovász (Open Problem 27 in Schrijver's book [*Combinatorial Optimization*, Springer-Verlag, 2003]). On the other hand, we provide an algorithm for the closely related and well-studied Minimum-weight Odd T-Join problem for nonnegative weights: our algorithm runs in FPT time parameterized by c, where c is the number of connected components in some efficiently computed minimum-weight T-join. If negative weights are also allowed, then finding a minimum-weight odd $\{s,t\}$-join is equivalent to the Minimum-weight Odd T-Join problem for arbitrary weights, whose complexity is still only conjectured to be polynomially solvable. The analogous problems for digraphs are also considered.

1 Introduction

The Minimum-weight Odd T-Join problem (MOTJ) asks for an odd-cardinality T-join of minimum weight for a given subset T of vertices in an undirected, edge-weighted graph. The MOTJ problem is the graph special case of optimization in binary hypergraphs. This area contains relevant problems on graphs, including open cases of the Max Cut problem, or some multiflow problems. Stimulating minimax conjectures of Seymour's have been solved in special cases (see, e.g., Guenin and Abdi [1, 2, 16]), but optimization algorithms were not well known, not even for cases where the minimax theorems conjectured by Seymour hold.

In this paper we study a handful of interconnected optimization problems in undirected, edge-weighted graphs that involve parity constraints on certain edge sets. Such problems are considered in a general framework under the term "binary hypergraphs" [23, Chapter 80], subject to a huge number of deep results and conjectures since the 1970s until now (see, e.g., [1,2,12,15,16,25]; those published before 2002 are summarized in Schrijver's book [23]). A first round of problems like T-joins or odd cycles for nonnegative edge weights have been studied in the last century [10, 14]. Then time has come for adding more binary constraints [12, 15, 16], bringing in new results and new challenges.

In this work we consider the main algorithmic challenges. Among other variants, we study in undirected graphs the problems of finding a minimum-weight odd T-join for arbitrary weights, and for conservative weights the shortest odd cycle or the shortest odd path between two given vertices. The edge-weights are said to be *conservative* if there are no cycles with negative total weight. Our results are the following:

(a) The variant of Shortest Odd Cycle (SOC) where the solution has to contain a given vertex of the graph is *NP*-hard, *implying NP-hardness for the* Shortest Odd Path *(SOP) problem as well.* The complexity of the latter has been an open question by Lovász (see Open Problem 27 in Schrijver's book [23]) for the past twenty years.

(b) The MOTJ problem for nonnegative weights *can be solved in* $2^{|T|/2}O(n^4)$ *time* on an n-vertex graph. Our method is based on structural properties of shortest cycles in graphs with conservative weights, yielding an algorithm for SOC that is efficient when negative-weight edges span a bounded number of components.

(c) The SOC problem for conservative weights is polynomially equivalent to MOTJ, and although we do solve certain special cases in polynomial time, *it remains open in general.*

We prove in addition that finding two openly disjoint paths between two vertices with minimum total weight in an undirected graph with conservative weights in *NP*-hard; this problem, quite surprisingly, has also been open. A major benefit of our results is finding connections among a so far chaotic set of influential problems and sorting them into *NP-hard, polynomial-time solvable, and open* cases (cf. (a), (b), and (c), respectively). We will also see that some of the analogous problems for digraphs are easily reducible to tractable problems, while some others are equivalent with the problems we prove to be *NP*-hard (see Sections 4 and 5); some new related open problems also arise (see Section 5.3).

The SOP problem contains the following classical problem SP:

SHORTEST PATH IN UNDIRECTED CONSERVATIVE GRAPHS (SP)
Input: An undirected graph $G = (V, E)$ with weights $w : E \to \mathbb{Z}$, $s, t \in V$, and $k \in \mathbb{Z}$.

Question: Is G conservative with w, and if so, is there a path between s and t of weight at most k?

Indeed, to solve SP using SOP, simply add a new vertex t', and add both an edge and a path of two edges from t to t', each with weight 0, and then find a shortest odd (s, t')-path. As is well known, SP can be solved in polynomial time, but this cannot be done via shortest path algorithms for conservative digraphs:

To solve SP using techniques developed for digraphs, we would have to include each edge in both directions, and negative edges would lead to negative cycles consisting of two edges. Moreover, the algorithms for directed graphs are all based on the fact that subpaths of shortest paths are shortest and the triangle inequality holds, which is not true in the undirected case. In fact, it is well known from the very first results of matching theory that SP is both conceptually and algorithmically equivalent to finding a minimum-weight perfect matching in a graph (see some details and references in the first paragraph of Section 5.1).

Requiring odd cardinality from the paths to be optimized on will lead, as mentioned, to *NP*-completeness. In fact, our *NP*-hardness proof for SOP shows also the *NP*-hardness of the Shortest Odd Cycle through a Point (SOCp) problem that asks for a shortest odd cycle containing a given vertex in an undirected graph with conservative edge weights.

However, the Shortest Odd Cycle (SOC) problem of finding a shortest odd cycle in undirected conservative graphs seems significantly easier. The SOC problem has been studied from multiple viewpoints and under various names; one of these is MOTJ. Although SOC is known to be in *RP* when edge weights are uniformly 1 (due to Geelen and Kapadia [12]; see more about this in Section 3), its polynomial-time solvability remains open! The study of SOC and MOTJ has lead to deep structural results:

Seymour [23, 25] conjectured minimax theorems for the problem of finding a shortest odd T-join if certain minors are excluded; one of these, the *flowing conjecture*, postulates the existence of a fractional dual solution for a minimum transversal of odd T-joins, while the *cycling conjecture* bets on the existence of an integer dual solution for nonnegative weights. We need not and will not enter into these linear programming aspects in this note, but it is interesting to mention that these conjectures have been solved in the $|T| \leq 2$ special case by Guenin and Abdi [1, 2, 16], without regard for algorithms. On the other hand, a randomized polynomial algorithm has been given for SOC by Geelen and Kapadia [12], making polynomial-time solvability plausible. In Section 5.2 we discuss these connections.

Main contribution. On the positive side, in Section 3 we propose a fixed-parameter tractable (FPT) algorithm for the Shortest Odd Cycle problem for conservative weights, parameterized by the number of connected components spanned by all negative edges (Theorem 3.7). As a consequence, MOTJ with nonnegative weights can be solved by an algorithm that first computes a minimum-weight T-join F and runs in time $2^c O(n^4)$, where c is the number of connected components in F, and n the number of vertices in the graph.

As a further corollary, Cook, Espinoza, and Goycoolea's FPT result [8] for MOTJ with parameter $|T|$ and nonnegative weights follows, with a different proof and slightly different time complexity (Corollary 3.8).

The main surprise—causing at the same time some disappointment—is *the NP-completeness of SOCp (Theorem 4.3), in contrast with SOC, which remains open.* As an immediate corollary, surprisingly, Lovász's problem SOP is also *NP*-complete (Corollary 4.4). In Corollary 4.5 we further obtain *NP*-completeness of the Disjoint Shortest Paths (DISP) problem where, given two vertices s and t in an undirected, conservative graph, the task is to find two openly disjoint paths between s and t with minimum total weight.

Finally, we present certain connections including equivalences between the studied but still open problems (Theorem 5.4).

Organization. In Section 2 we introduce the most important notation, terminology and some basic facts. In Section 3 we make an inventory of the positive results concerning MOTJ. Besides mentioning some existing results and recalling the main conjecture about MOTJ, we present simple structural results for the Shortest Odd Cycle problem in conservative graphs, leading first to a polynomial algorithm for MOTJ with nonnegative weights in the case when we can find a minimum-weight T-join that is connected (Section 3.1), and then to efficient algorithms for MOTJ and for SOC (Section 3.2). Our *NP*-hardness results for the problems SOCp, SOP, and DISP are presented in Section 4.

The results of the paper reveal new possibilities for considering special cases that may deserve more focus. We put forward their relations and some open problems concerning them, summarized in Section 5, together with some conclusions.

2 Preliminaries

We start with basic notation for graph-theoretic concepts and for properties of edge-weight functions. We then proceed by giving the precise definitions of the problems already mentioned in Section 1 and stating some well-known facts about them that will be useful later on.

Notation for graphs. Given an undirected graph $G = (V, E)$, for some $F \subseteq E$ and $v \in V$ let $d_F(v)$ denote the *degree* of v in F, i.e., the number of edges in F incident to v. Let $V(F)$ denote the set of vertices that are incident to some edge in F. Let $G[F]$ denote the subgraph of G *spanned* by F, that is, the graph $(V(F), F)$.

A *cycle* in an undirected graph $G = (V, E)$ is a nonempty set C of edges such that $G[C]$ is connected, and $d_C(v) = 2$ for each vertex $v \in V(C)$. In a digraph $G = (V, E)$, a (directed) *cycle* additionally satisfies that all in- and out-degrees in $G[C]$ are equal to 1. For two distinct vertices s and t in a graph, an (s, t)*-path* has the same definition, except that the two *endpoints*, s and t, have degree 1 in the undirected case; in the directed case, s has in-degree 0 and out-degree 1, while t has out-degree 0 and in-degree 1. A cycle C with $s \in V(C)$ is also considered to be an (s, t)-path with $s = t$. For two sets of vertices $S, T \subseteq V$ of a graph, an (S, T)*-path* is an (s, t)-path for some $s \in S$ and $t \in T$. If P is a path and $a, b \in V(P)$, then the subpath of P between $a, b \in V(P)$ is denoted by $P(a, b)$.

Note that we have defined cycles and (s,t)-paths as edge sets. With a slight abuse of terminology, a *path* in G may also be a subgraph spanned by an (s,t)-path for distinct vertices s and t, and we also consider a single vertex in G to be a *trivial* path.

Two paths are said to be *vertex-disjoint* (or *edge-disjoint*) if they do not have a common vertex (or edge, respectively), and they are said to be *openly disjoint* if they can only share vertices that are endpoints on both paths.

A *T-join* in an undirected graph $G = (V, E)$ for some $T \subseteq V$ is a subset of edges $J \subseteq E$, such that $d_J(v)$ is odd if $v \in T$ and even if $v \in V \setminus T$. An $\emptyset$-join is the disjoint union of cycles; inclusionwise minimal, nonempty $\emptyset$-joins are exactly the cycles. A T-join with $|T| = 2$, that is, with $T = \{s, t\} \subseteq V$, is the disjoint union of an (s,t)-path and some cycles, so the inclusionwise minimal ones are (s,t)-paths.

A cycle, a path, a T-join, and, generally, any edge set is *odd (even)* if it contains an odd (respectively, even) number of edges.

Weight functions. We denote by $\mathbb{R}, \mathbb{R}_+, \mathbb{Z}, \mathbb{Z}_+$, and $\mathbb{N}$ the set of real, nonnegative real, integer, nonnegative integer, and positive integer numbers, respectively. For a function $f : D \to \mathbb{R}$ and some $D' \subseteq D$, let $f(D') := \sum_{d \in D'} f(d)$ as usual. In an optimization problem over a set of feasible edge sets (e.g., over paths between two vertices or over all cycles), a *w-minimum* solution is one that has minimum weight according to a given edge-weight function w; if w is clear from the context, we might also say that such a solution is *shortest*.

An undirected graph $G = (V, E)$ is *conservative* with weights $w : E \to \mathbb{R}$, if $w(C) \geq 0$ for any cycle C of G.

For arbitrary $w : E \to \mathbb{R}$ and $F \subseteq E$, let $w[F] : E \to \mathbb{R}$ denote the function defined by

$$w[F](e) := \begin{cases} -w(e) & \text{if } e \in F, \\ w(e) & \text{if } e \in E \setminus F. \end{cases}$$

Denote the *symmetric difference* of two sets X and Y by $X \Delta Y := (X \setminus Y) \cup (Y \setminus X)$. Then clearly $w[F](X) = w(X \setminus F) - w(X \cap F) = w(X \Delta F) - w(F)$ for any $X \subseteq E$. In particular, *F is a w-minimum T-join for some vertex set T if and only if $w[F]$ is conservative*, we will refer to this as *Guan's lemma* (stated by Guan [19] for the "Chinese postman problem" for nonnegative weights). Indeed, for any cycle X, the set $X \Delta F$ is also a T-join; therefore $w[F](X) = w(X \Delta F) - w(F) \geq 0$ by the definition of F.

In order to avoid some ambiguities and irrelevant case analysis, we will *normalize* any given rational weight function, thereby simplifying some arguments and algorithms. Given a conservative, rational weight function we first multiply it by the smallest common denominator of its values, which does not change the optimal sets, increases the size of the input only polynomially, and can be carried out in polynomial time. An integer weight function $w : E \to \mathbb{Z}$ is further normalized by defining $w'(e) := 2|E| \cdot w(e) + 1$ for each edge $e \in E$. The normalized weight function w' will then satisfy $w'(X) < w'(Y)$ for each pair of edge sets X and Y with $w(X) < w(Y)$, and equal-weight edge sets of different cardinalities get different w'-weights. When searching for a minimum-weight edge set with a given property, normalization does not essentially change the problem, since at least one optimal edge set remains optimal. Note that if w is conservative, then w' will also be conservative; furthermore, *odd sets will have different weights than even ones.* Moreover, the only $\emptyset$-join of weight 0 in a conservative graph will be the empty set. We will say that a weight function is *normal* if it is obtained as a result of the above normalization method; in particular, a normal weight function does not assign zero weight to any edge. Since any weight function can be normalized, assuming normal weight functions does not restrict generality.

Problem definitions and classical results. Consider the following two problems, which differ only in that the second one confines the searched cycle to contain a given vertex:

SHORTEST ODD CYCLE IN CONSERVATIVE GRAPHS (SOC)
Input: An undirected graph $G = (V, E)$ conservative with $w : E \to \mathbb{Z}$ and $k \in \mathbb{Z}$.
Question: Is there an odd cycle C in G whose weight is at most k?

SHORTEST ODD CYCLE IN CONSERVATIVE GRAPHS THROUGH A POINT (SOCp)
Input: An undirected graph $G = (V, E)$ conservative with $w : E \to \mathbb{Z}$, $p \in V$, and $k \in \mathbb{Z}$.
Question: Is there an odd cycle C in G with $p \in V(C)$ whose weight is at most k?

The following problem MOTJ is closely related to SOC: on the one hand, MOTJ is a generalization of SOC (consider the case $T = \emptyset$, which yields exactly SOC); on the other hand, SOC is exactly the problem of finding the "improving step" for reaching an optimum in MOTJ; the two problems are therefore polynomially equivalent (cf. Theorem 5.4).

MINIMUM-WEIGHT ODD T-JOIN (MOTJ)
Input: An undirected graph $G = (V, E)$ with $w : E \to \mathbb{Z}$, $T \subseteq V$, and $k \in \mathbb{Z}$.
Question: Is there an odd T-join in G with total weight at most k?

The following problem can be thought of as an alternative form of SOCp. It is the following formulation whose complexity was questioned by Lovász [23, Open Problem 27, p. 517]:

SHORTEST ODD PATH IN CONSERVATIVE GRAPHS (SOP)
Input: An undirected graph $G = (V, E)$ conservative with $w : E \to \mathbb{Z}$, $s, t \in V$, and $k \in \mathbb{Z}$.
Question: Is there an odd (s, t)-path in G with total weight at most k?

We will use the SOCp formulation more in what follows; the straightforward relation of SOCp and SOP, in particular their polynomial equivalence, will be apparent from Theorem 4.3 and Corollary 4.4 thereafter.

Analogously to SOCp and SOP, we also define the problems SECp and SEC by replacing "odd" with "even" in the definitions.

The problems that will turn out to be *NP*-hard (SOCp and SOP) will actually already be *NP*-hard for conservative weight functions with values in $\{-1, 1\}$. We denote the problems restricted to such weight functions by putting ± 1 in a subscript; for example, $\mathrm{SOP}_{\pm 1}$ means SOP restricted to weight functions taking only values from $\{-1, 1\}$. A subscript $+$ means a restriction to nonnegative weights. The following theorem summarizes well-known results.

Theorem 2.1. *SP, SOP_+, SEP_+, $SOCp_+$, $SECp_+$, and SOC_+, SEC_+ are polynomially solvable.*

Proof. SP can be solved in polynomial time ([23, Section 29.2], see explanations in the beginning of Section 5.1), as mentioned in the introduction. SOP_+ and SEP_+ can be solved in polynomial time by the well-known "Waterloo folklore" algorithm related to Edmonds' classical work on matchings [14]; see also [23, Section 29.11e]. Then SOCp_+ on an instance $(G = (V, E), w, p, k)$ can be solved by solving SEP_+ on $(G' = (V, E \setminus \{pr\}), w', p, r, k')$ for each edge $pr \in E$ incident to p, where w' is the restriction of w to $E \setminus \{pr\}$ and $k' = k - w(pr)$. We can reduce SECp_+ to SOP_+ similarly. Finally, SOC_+ on an instance $(G = (V, E), w, k)$ can be solved by computing SOCp_+ on (G, w, p, k) for all $p \in V$, and SEC_+ can be reduced to SECp_+ similarly. The execution time of all these problems is polynomial in the input size. □

SOP_+ and SEP_+ can actually be solved in a simpler and much faster way—in $O(m \log n)$ time on a graph with n vertices and m edges—than finding a maximum-weight matching, due to Derigs [9]. We will see this in terms of SOC_+ and SEC_+ being much easier than matchings: they can be solved by using only Dijkstra's shortest path algorithm (Proposition 5.1).

Problems concerning odd or even paths are not really different, since they can be reduced to one another by introducing a new vertex t' and an edge tt'. However, no such reduction is known between problems concerning odd and even cycles. In fact, even the existence of nonempty even cycles happens to be inherently more difficult, to the extent that its complexity is not yet completely settled in directed graphs; see Proposition 4.2 and Section 5.1.

We finish the list of helpful preliminaries with further notation and observations. Given a graph $G = (V, E)$ and a conservative weight function $w : E \to \mathbb{Z}$, we denote the set of edges with negative weight by $E^- = \{e \in E : w(e) < 0\}$, and let us write $E^+ = E \setminus E^-$. Observe that each connected component K of $G[E^-]$ is a tree, because w is conservative on G. For any two vertices u and v in K, let $K(u, v)$ denote the unique (u, v)-path in K.

Proposition 2.2. *Suppose $G = (V, E)$ is conservative with w, and P is a w-minimum (u, v)-path for some vertices $u, v \in V$. Then for each connected component K of $G[E^-]$, either P and K are vertex-disjoint, or their intersection is a path.*

Proof. For a contradiction, suppose that there is a connected component K of $G[E^-]$ whose intersection with P is nonempty, and not a path. Then there exist two distinct vertices a and b in $V(P) \cap V(K)$ so that $K(a, b)$, is edge-disjoint from P.

Using that w is conservative on G, we get $w(P(a, b) \cup K(a, b)) \geq 0$. Since every edge in K has negative weight, this implies $w(K(a, b)) < 0 < w(P(a, b))$. Then $w(P \setminus P(a, b) \cup K(a, b)) < w(P)$, contradicting the choice of P. □

3 Are MOTJ and SOC tractable?

In this section we collect evidence that MOTJ is tractable in its full generality, and present some new cases when this can be already proved.

The conjecture of polynomial-time solvability of MOTJ is primarily supported by Geelen and Kapadia's algorithm [12] establishing that MOTJ for uniformly 1 weights belongs to RP, saving this unweighted special case of the problem from being suspected as NP-hard (which would imply $NP = RP$) and suggesting the following conjecture, which is therefore open and very tempting already in the unweighted case.

Conjecture 3.1. *MOTJ and SOC can be solved in polynomial time.*

We stated the conjecture for the general, weighted case, since we anticipate that any algorithm for the unweighted cardinality case works without essential changes: it is usually easy to follow what a given algorithm would do after each edge is subdivided into as many parts as its weight, without actually performing the subdivision.

This conjecture is equivalent to a whole range of equivalent conjectures, since MOTJ can be reduced to several special cases, including the case when weights are nonnegative or when $|T| \leq 2$ (see Theorem 5.4 and some remarks thereafter). However, restricting $|T|$ and simultaneously assuming nonnegative weights seems to make the problem easier (Corollary 3.8), confirming Conjecture 3.1 under these assumptions. In Section 3.1 we present two approaches for solving MOTJ$_+$ for $|T| = 2$, and the second one is generalizable to the case when a minimum T-join with a bounded number of components is given. In Section 3.2 we investigate the general case that leads us to an FPT algorithm for the equivalent SOC where the parameter is the number of connected components spanned by all negative edges. As a corollary we also obtain the fixed-parameter tractability of MOTJ$_+$ with parameter $|T|$, already proved by Cook, Espinoza, and Goycoolea [8], with a slightly worse dependence on $|T|$ but better on n, the number or vertices in the input graph.

3.1 MOTJ$_+$ with a connected minimum-weight T-join

We start by proving that MOTJ$_+$ is polynomial-time solvable if $|T| \leq 2$, a case for which Seymour's conjectures mentioned in the introduction have also been proved. (Guenin [16] characterized for $|T| \leq 2$, in terms of the two small excluded minors of Seymour, when inclusionwise minimal odd T-joins are "ideal"; Abdi and Guenin [1,2] proved that in this special case, actually a stronger minimax theorem holds.)

A simple $O(n^3)$ algorithm is known in this case from Cook, Espinoza, and Goycoolea [8, Proposition 5.3] (see the paragraph after Corollary 3.8). Our goal here is to introduce the reader to certain structural properties of shortest odd cycles in conservative graphs that also allow us to solve MOTJ$_+$ when a minimum-weight T-join is connected. These properties will then carry us further to the case of several components.

We first state a clarifying observation on inclusionwise minimal odd T-joins for $|T| = 2$.

Lemma 3.2 ([2,16])**.** *Let $G = (V, E)$ be a graph, $s, t \in V$, and $F \subseteq E$. Then F is an inclusionwise minimal odd $\{s, t\}$-join if and only if it is an odd (s, t)-path or is of the form $P \cup C$ where P is an even (s, t)-path and C an odd cycle that is edge-disjoint from P and satisfies $|V(P) \cap V(C)| \leq 1$.*

Proof. Clearly, any odd (s, t)-path and any edge set $P \cup C$ as defined in the statement of the lemma is an inclusionwise minimal odd $\{s, t\}$-join. Conversely, any $\{s, t\}$-join F is the union of an (s, t)-path P and pairwise edge-disjoint cycles. So, if F is an inclusionwise minimal odd $\{s, t\}$-join, then it contains neither even cycles, nor more than one odd cycle. (An even cycle or two odd cycles could be deleted from F, contradicting minimality.)

If F contains no cycle, then $F = P$ where P is an odd (s, t)-path. Otherwise, $F = P \cup C$ where P and C *are edge-disjoint*, and C is an odd cycle; P is then an even path, since $|F|$ is odd. It remains to prove that $|V(P) \cap V(C)| \leq 1$.

Suppose for a contradiction that $|V(P) \cap V(C)| > 1$. Traversing P from s to t, let a and b be the first and last encountered vertex of C, respectively. Since $|V(P) \cap V(C)| \geq 2$ we have that both a, b exist, $a \neq b$, and therefore a and b divide C into two (a, b)-paths, C_1 and C_2. So $P \cup C$ contains three, pairwise edge-disjoint (a, b)-paths: $P(a, b)$, C_1, and C_2, two of which necessarily have the same parity. Deleting those two from F, we still get an odd $\{s, t\}$-join, contradicting the inclusionwise minimality of F. □

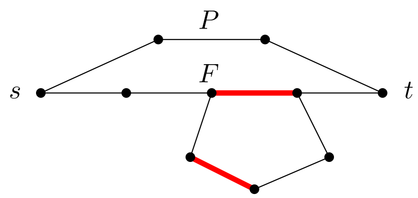
A simple algorithm for MOTJ$_+$ with $|T| = 2$. Lemma 3.2 easily yields a simple polynomial algorithm for finding a minimum-weight odd (or even) $\{s, t\}$-join F. Assuming that w is normal, F is clearly inclusionwise minimal and thus can be searched in the form given in Lemma 3.2:

Step 1. Compute in the input graph a minimum-weight odd (s, t)-path P_{odd}, a minimum-weight even (s, t)-path P_{even}, and a minimum-weight odd cycle C.

Step 2. Let F be the shorter one among P_{odd} and $P_{\text{even}} \Delta C$; if their weights are equal, choose arbitrarily.

The correctness of the above algorithm follows from Lemma 3.2. To see this, it suffices to observe that $P_{\text{even}} \Delta C$ is an odd $\{s, t\}$-join, and its weight does not exceed that of any $P' \cup C'$ where P' is an even path and C' an odd cycle edge-disjoint from P', since $w(P) \leq w(P')$ and $w(C) \leq w(C')$. Therefore, by Lemma 3.2 we know that F indeed has minimum weight among all inclusionwise minimal $\{s, t\}$-joins. Furthermore, according to Lemma 2.1, each of P_{even}, P_{odd}, and C can be computed in polynomial time.

We also introduce another approach for the case $|T| = 2$ that brings us closer to the extension of polynomial solvability of MOTJ$_+$ when a minimum T-join of a constant number of components can be constructed. Our next algorithm relies heavily on Proposition 3.3, illustrated by Figure 1. Recall that inclusionwise minimal odd $\emptyset$-joins are cycles, and recall also Lemma 3.2 and Guan's lemma. The following observation shows that MOTJ is equivalent to SOC.



$C = F \cup P$ with $w[F](C) = 1$;
C is a $w[F]$-minimal odd cycle.

Figure 1: Illustration of Proposition 3.3. Bold, red lines depict edges of weight -1, and all other edges have weight 1. The shortest (s,t)-path F is the middle horizontal line; it is even and of weight 2. According to Proposition 3.3, $P = F\Delta C$ is a w-shortest odd $\{s,t\}$-join.

Proposition 3.3. *Let $w : E \to \mathbb{R}$ be arbitrary, and let $T \subseteq V$ with $|T|$ even. Assume that F is a w-minimum T-join and $|F|$ is even. Then F_{odd} is a w-minimum odd T-join if and only if $F_{\text{odd}} = F\Delta C$ for some $w[F]$-minimum odd $\emptyset$-join C. In particular, if C is a $w[F]$-minimum odd cycle, then $F\Delta C$ is a w-minimum odd T-join.*

Proof. Let F be as in the condition. Then correspondence $F_{\text{odd}} \leftrightarrow F_{\text{odd}}\Delta F$ is a bijection between odd T-joins and odd $\emptyset$-joins. Note further that

$$w[F](F\Delta F_{\text{odd}}) = w(F_{\text{odd}} \setminus F) - w(F \setminus F_{\text{odd}}) = w(F_{\text{odd}}) - w(F).$$

Since $w(F)$ is a fixed value, we obtain that $F\Delta F_{\text{odd}}$ minimizes $w[F]$ over all odd $\emptyset$-joins exactly if F_{odd} minimizes w over all odd T-joins. It remains to observe that since $w[F]$ is conservative, a $w[F]$-minimum odd cycle is always a $w[F]$-minimum odd $\emptyset$-join. $\square$

By Proposition 3.3, finding a w-minimum odd T-join in an instance (G, w, T, k) of MOTJ can be reduced to finding a $w[J]$-minimum odd cycle in the same graph G where J is a w-minimum T-join. Next, we present Proposition 2.2 for shortest odd cycles.

Lemma 3.4. *Suppose the graph $G = (V,E)$ is conservative with $w : E \to \mathbb{R}$, and C is a w-minimum odd cycle. Then for each connected component K of $G[E^-]$, either C and K are vertex-disjoint, or their intersection is a path.*

Proof. The proof is essentially the same as that of Proposition 2.2. For a contradiction, suppose that there is a connected component K of $G[E^-]$ whose intersection with C is not a path. Then there exist two distinct vertices a and b in $V(C) \cap V(K)$ so that the unique (a,b)-path in K, denoted by $K(a,b)$, is edge-disjoint from C and, consequently, from both (a,b)-paths C_1 and C_2 into which a and b divides C; note that $|C_1|$ and $|C_2|$ have different parities.

Using that w is conservative on G, we get $w(C_i \cup K(a,b)) \geq 0$ for both $i = 1,2$. Recall also that $w(K(a,b)) < 0$ since every edge in K has negative w-weight. Hence, $w(C_i \cup K(a,b)) < w(C_i)$ and $w(C_i) > 0$ for $i = 1,2$. Therefore $C_i \cup K(a,b)$ for $i = 1,2$ are two cycles of weight less than $w(C)$, and one of them is odd, which contradicts the definition of C. $\square$

Corollary 3.5. *Let $G = (V, E)$ be a graph with a normal weight function $w : E \to \mathbb{R}_+$, let P be a w-minimum (s,t)-path for some vertices $s, t \in V$, and let C be a $w[P]$-minimum odd cycle. Then either C and P are vertex-disjoint, or their intersection is a path.*

Proof. Since w is nonnegative, a w-minimum (s,t)-path is also a w-minimum $\{s,t\}$-join. By Guan's lemma mentioned in the introduction, $w[P]$ is therefore conservative. Since w is nonnegative and normal, we know that $w(e) > 0$ for all $e \in E$. Therefore, P is a connected component in the subgraph of negative-weight edges according to $w[P]$, and the statement follows directly from Lemma 3.4. □

Another simple algorithm for MOTJ$_+$ with $|T| = 2$. We now present a second polynomial algorithm for finding a minimum-weight odd $\{s,t\}$-join for a nonnegative weight function w, using Proposition 3.3 and Corollary 3.5. Again, we assume that w is normal.

Step 1. Compute a w-minimum (s,t)-path J. If J is odd, return it and stop.

Step 2. Let C be the cycle of smaller $w[J]$-value between the following two cycles:

(a) a w-shortest odd cycle in $E \setminus J$;

(b) a cycle $J(u,v) \cup P$ with the smallest $w[J]$-weight where $u, v \in V(J)$, $u \neq v$, and P is a w-minimum (u,v)-path in $E \setminus J$ of parity different from $J(u,v)$.

Step 3. Return $J \Delta C$.

Computing a w-minimum $\{s,t\}$-path in Step 1 can be done in polynomial time by Theorem 2.1. (In fact, we only need SP$_+$ here, solved, e.g., by Dijkstra's algorithm.) Step 2(a) is the SOC$_+$ problem which is solvable in polynomial time by Theorem 2.1, as are the SEP$_+$ and SOP$_+$ problems of Step 2(b). Hence, the above algorithm runs in polynomial time.

Since w is nonnegative, the w-minimum (s,t)-path J of Step 1 is a w-minimum $\{s,t\}$-join, so if it is odd, then the algorithm returns the correct result. It remains to prove that *the odd cycle C computed in Step 2 is a $w[J]$-minimum odd cycle*, since then the correctness of the algorithm follows immediately from Proposition 3.3.

Let C' be a $w[J]$-minimum odd cycle. Recall that w is normal, and hence no edge has weight 0 according to w. Therefore, the edges with negative $w[J]$-weight are exactly the edges of J, and so by Corollary 3.5, either C' is edge-disjoint from J (implying that C' is a w-shortest cycle in $E \setminus J$), or C' is formed by the union of a nontrivial subpath of J and a path in $E \setminus J$ of different parity between the same vertices. Hence, we obtain an odd cycle with $w[J]$-weight at most $w[J](C')$ in either Step 2(a) or Step 2(b).

It is not hard to see that the same approach works for solving MOTJ$_+$ for the more general case when a connected w-minimum T-join is given. In the next section, we generalize this algorithm to work without any assumption on the number of components. The running time of the equivalent SOC, however, is fixed-parameter tractable if the parameter is the number of connected components of negative edges.

3.2 FPT algorithms for SOC and MOTJ$_+$

In this section we show that MOTJ$_+$ is polynomial-time solvable when we are given a w-minimum T-join with a fixed number of connected components. By Proposition 3.3, this is reduced to finding a shortest odd cycle with respect to a conservative weight function with a fixed number of *negative components*, defined as the connected components of $G[E^-]$, the subgraph spanned by all negative-weight edges. The main result of this section is Theorem 3.7, establishing that SOC is fixed-parameter tractable with parameter c, the number of negative components of the input graph.

Finding a shortest odd cycle when there is only one negative component is based on Lemma 3.4 via Corollary 3.5. Even though the assertion in Lemma 3.4 holds for an arbitrary number of components, a naive, "brute force" approach based on this lemma provides only an algorithm with $c^c O(n^{2(c+1)})$ running time where $n = |V|$, which is polynomial if c is fixed, but does not confirm fixed-parameter tractability. *The structural observations of Lemma 3.6 below make it possible to achieve a running time of* $2^c O(n^4)$.

To state Lemma 3.6 which, together with the crucial property expressed in Lemma 3.4, will form the basis of our FPT algorithm with parameter c, we need some additional notation. We will use the notation $[k] := \{1, \dots, k\}$ for $k \in \mathbb{N}$. Let $K_1, \dots, K_c$ be the edge sets of the negative components in G. We define the graph G_I as $G_I = (V, E_I)$ where $E_I = E^+ \cup \bigcup_{i \in I} K_i$ for any index set $I \subseteq [c]$; in particular, $G_\emptyset = (V, E^+)$ and $G_{[c]} = G$. For $F \subseteq E$ let $I(F) := \{i \in [c] : F \cap K_i \neq \emptyset\}$ denote the set of indices of negative components having an edge in F. Whenever we consider any subgraph of G with weight function w, we will implicitly use the restriction of w to this subgraph; this will not cause any confusion.

Lemma 3.6. *Suppose that $G = (V, E)$ is conservative with a normal weight function w, and $C \subseteq E$ is a w-minimum odd cycle partitioned into paths $P_1, \dots, P_m \subseteq E$ ($m \in \mathbb{N}, m \geq 2$) so that the sets $I(P_i) \subseteq [c]$, $i \in [m]$, are pairwise disjoint. Then for any family of pairwise disjoint sets I_i, $i \in [m]$, with $I(P_i) \subseteq I_i \subseteq [c]$, the following statements (a) and (b) hold for $G_i := G_{I_i}$:*

- *(a) Each P_i for $i \in [m]$ but at most one is shortest among paths in G_i between the endpoints of P_i, where the exception is also shortest in G_i among paths of the same parity as P_i.*
- *(b) Suppose that the path P'_i for each $i \in [m]$ is shortest among paths in G_i of the same parity as P_i between the endpoints of P_i. Then the paths P'_i, $i \in [m]$, are pairwise openly disjoint.*

Furthermore, there exists a partition of C into $P_1, \dots, P_m$ and index sets $I_1, \dots, I_m$ satisfying the above conditions with $m \leq 3$ and such that either $P_1 = \{e\}$ for some $e \in E_+$ or $P_1 = C \cap K_j$ for some $j \in [c]$, and if $m = 3$, then $I_1 = I(P_1)$, $I_2 = I(P_2)$, $I_3 = [c] \setminus (I_1 \cup I_2)$, and P_i is shortest in G_i for $i \in [3]$.

We call such a partition into two or three paths a *compact* partition. In a compact partition the only path P_i that may not be shortest in G_i between its endpoints is the path P_1, and only if $m = 2$ and $P_1 = \{e\}$ for some $e \in E^+$, when an even path shorter than $w(e)$ may exist in $G[E^+]$.

Finally, note that the condition on the disjointness of index sets is a formalization of the requirement that each path of the partition should have "its own negative components" that are not used by any of the other paths. We know from Lemma 3.4 that the intersections with these components are paths.

Proof. To see (a), suppose first for a contradiction that there is a path P'_i in G_i of the same parity as P_i and between the same endpoints with weight $w(P'_i) < w(P_i)$. Then $C' = (C \setminus P_i)\Delta P'_i$ is an odd $\emptyset$-join. Since $P'_i \subseteq E_{I_i}$, our assumption on the disjointness of I_i from any I_j where $j \neq i$ implies that $C \setminus P_i$ and P'_i may share only edges in E^+. As a consequence, C' has weight at most $w(C) - w(P_i) + w(P'_i) < w(C)$, which contradicts the definition of C.

To finish the proof of (a), suppose that there exist two distinct indices i and j in $[m]$ so that P_i and P_j are not shortest paths between their endpoints in G_i and in G_j, respectively. Then let $\hat{P}_i$ and $\hat{P}_j$ be shortest paths between the endpoints of P_i and P_j, respectively, in G_i and G_j. We conclude that the parity of P_i and P_j differs from the parity of $\hat{P}_i$ and $\hat{P}_j$, respectively, since the former two are shortest for their parity but not shortest, while the latter two are shortest. Using again our assumptions on I_i and I_j, we obtain that $\hat{C} = (C \setminus (P_i \cup P_j))\Delta\hat{P}_i\Delta\hat{P}_j$ is an odd $\emptyset$-join with weight at most $w(C) - w(P_i) - w(P_j) + w(\hat{P}_i) + w(\hat{P}_j) < w(C)$, a contradiction.

In order to prove (b), note first that the paths P'_i, $i \in [m]$, must be *pairwise edge-disjoint* since, using similar arguments as before, we know that any two of them can only share edges of E^+, and if they do have a common edge, then there exists a smaller cycle; more formally: let $i, j \in [m]$ with $i \neq j$, and define $C' := (C \setminus (P_i \cup P_j))\Delta P'_i\Delta P'_j$, which is an odd $\emptyset$-join with $w(C') \leq w(C)$, so by the minimality of C, the equality holds here. Since $I(P_i) \cap I(P_j) = \emptyset$, a common edge of P'_i and P'_j would have positive weight (since w is normal), which would imply $w(C') < w(C)$. Thus, P'_i and P'_j are edge-disjoint.

Suppose now for a contradiction that P'_i and P'_j $(i, j \in [m], i \neq j)$ are not openly disjoint, so there exists some $x \in V(P'_i) \cap V(P'_j)$ that is an inner vertex of at least one of P'_i and P'_j. Then C', defined as above, contains a cycle C'' as a nonempty proper subset, so C'' and $C' \setminus C''$ partition C' into two nonempty $\emptyset$-joins, exactly one of which is odd; denote it by Q. Since there are no cycles of weight 0 in G by the normality of w, we get that $w(Q) < w(C') = w(C)$, a contradiction.

Finally, in order to prove the last sentence of the lemma, choose first $P_2 \subseteq C$ so that it satisfies $I(P_2) \cap I(C \setminus P_2) = \emptyset$ and is a shortest path in $G_{I(P_2)}$ between two distinct vertices u and v on C. To see that such a path exists, consider any partition of C into paths Q_1 and Q_2 with $I(Q_1) \cap I(Q_2) = \emptyset$; then by (a), at least one Q_i, $i \in [2]$, is shortest in $G_{I(Q_i)}$. Moreover, choose P_2 so that it is inclusionwise maximal among all paths satisfying these requirements.

Choose $P_1 \subseteq C \setminus P_2$ so that it is consecutive with P_2 on C and consists either of one positive edge or a negative path that continues until the next positive edge on C. In the latter case, by Lemma 3.4 this path contains the entire intersection of C with the component of $G[E^-]$

containing P_1, and therefore $I(P_1) \cap I(C \setminus P_1) = \emptyset$, in particular $I(P_1) \cap I(P_2) = \emptyset$.

If $P_1 \cup P_2 = C$, then we are done; otherwise, let $P_3 := C \setminus (P_1 \cup P_2)$. Clearly, $I(P_3)$ is disjoint from $I(P_1)$ and $I(P_2)$, because $I(P_i) \cap I(C \setminus P_i) = \emptyset$ for $i = 1, 2$. Moreover, defining I_1, I_2, and I_3 as in the statement of the lemma, P_3 is also a shortest path in G_{I_3}; otherwise, the partition of C into two paths $\{P_1 \cup P_2, P_3\}$ would contradict (a), since by the maximal choice of P_2 we know that $P_1 \cup P_2$ is also not a shortest path in $G_{I_1 \cup I_2}$. Similarly, P_1 is also a shortest path in G_{I_1}; otherwise, the partition of C into two paths $\{P_1, P_2 \cup P_3\}$ would contradict (a). $\square$

We are now ready to present the main result of this section.

FPT-algorithm for SOC with parameter c:

Step 0. Normalize w, and initialize $\mathcal{Q} = \emptyset$.

Step 1. For all $I \subseteq [c]$, compute a shortest (x, y)-path $P(x, y, I)$ in G_I for all $x, y \in V$.

Step 2. For all $u, v \in V$ with $u \neq v$:

- **(a)** if $uv \in E^+$, then let $R \leftarrow \{uv\}$, and perform (c).
- **(b)** if $u, v \in V(K_j)$ for some $j \in [c]$, then let $R \leftarrow K_j(u, v)$, and perform (c).
- **(c)** For all $x \in V$, $I_u \subseteq [c] \setminus I(R)$ and $I_v = [c] \setminus (I_u \cup I(R))$:
  if $Q = R \cup P(u, x, I_u) \cup P(x, v, I_v)$ is an odd cycle, then add Q to $\mathcal{Q}$.

Step 3. If $\mathcal{Q} \neq \emptyset$, then **return** $Q \in \mathcal{Q}$ with the minimum weight; otherwise, **return** "There is no odd cycle in G."

Running time. Step 0 can be performed in linear time, as explained in Section 2. Step 1 computes shortest paths for all pairs of vertices in 2^c different graphs with conservative weights. By Theorem 2.1, the SP problem can be solved in polynomial time; the book by Korte and Vygen describes an $O(n^4)$ time algorithm [18, Theorem 12.14] for this problem. Step 2 has $O(n^2)$ iterations, and inside each of these (c) in turn checks for at most $2^c n$ edge sets whether it forms an odd cycle. As this takes $O(n)$ time for each set, Step 2 takes altogether $2^c O(n^4)$ time, so this is the total time used by the FPT-algorithm.

Theorem 3.7. *If G is nonbipartite, the above algorithm returns a w-minimum odd cycle, and its running time is $2^c O(n^4)$.*

Proof. We have already proved the assertion on the complexity, so let us prove the correctness of our algorithm.

If G is bipartite, $R \cup P(u, x, I_u) \cup P(x, v, I_v)$ of Step 2 is even for all possible choices, since it is a closed walk in a bipartite graph. So, $\mathcal{Q}$ remains empty, and the algorithm returns a correct answer. Otherwise, let C be a shortest odd cycle; we show that the algorithm puts into $\mathcal{Q}$ an odd cycle of the same weight as C and thus returns a correct solution.

By the final assertion of Lemma 3.6, C admits a compact partition $P_1, \dots, P_m$, where either P_1 consists of an edge $uv \in E^+$, corresponding to a choice in Step 2(a) of our algorithm, or $P_1 = C \cap K_j$ for some $j \in [c]$, that is, P_1 is a negative path between two distinct vertices u and v in K_j, corresponding to a choice in Step 2(b). Hence, at least once in Step 2 the path P_1 gets chosen as R. Recall that in our compact partition, *any other path P_i ($2 \leq i \leq m \leq 3$) in the partition is shortest in G_i* where $G_i = G_{I(P_i)}$.

Now if $m = 2$, then P_2 is a shortest (u, v)-path in G_2. Consider the choice of Step 2(c) for $P(u, x, I_u)$ with $x := v$ and $I_u := I(P_2)$. Since with these choices $P(u, x, I_u) = P(u, v, I_2)$ is also a shortest (u, v)-path in G_2, we get $w(P(u, x, I_u)) = w(P_2)$. Since w is normal, we also know that $P(u, x, I_u)$ has the same parity as P_2. Since now $P(x, v, I_v) = P(v, v, I_v)$ is a trivial path independently of the choice of I_v, we get that $w(P(x, v, I_v)) = 0$. By claim (b) of Lemma 3.6, we also know that P_1 and $P(u, x, I_u)$ are openly disjoint, and thus Q is a cycle. Moreover, the weight of $Q = P_1 \cup P(u, x, I_u) \cup P(v, x, I_v)$ is $w(P_1) + w(P_2) = w(C)$. Hence, Q has the same weight and then by normality also the same parity as C, and is therefore a w-minimum odd cycle contained in $\mathcal{Q}$, as claimed.

If $m = 3$, then the shortest path P_2 in G_2 and the shortest path P_3 in G_3 have a common endpoint; denote it by x. Again, setting $I_u := I(P_2)$ we get that $I_v = [c] \setminus (I(P_1) \cup I(P_2)) = I_3$ also holds by our definitions. Moreover, $P(u, x, I_u)$ and $P(x, v, I_v)$ have the same weight and, by the normality of w, the same parity as P_2 and P_3, respectively. Applying claim (b) of Lemma 3.6 to the paths P_1, $P(u, x, I_u)$, and $P(x, v, I_v)$, we get that they are mutually openly disjoint. Hence, we can conclude again that $Q = P_1 \cup P(u, x, I_u) \cup P(x, v, I_v)$ is a cycle and has the same weight and parity as C. $\square$

It is easy to see that the $w \geq 0$ special case of the FPT-algorithm consists only of n shortest path computations and does not rely on matchings. (Indeed, then the enumeration of the components of E^- disappears, and one execution of Dijkstra's shortest path algorithm computes a shortest path from a given vertex to any other.) This is not surprising, since it is well known that an odd walk can be determined by n shortest path computations in an auxiliary graph, for both undirected and directed graphs (see Proposition 5.1). The same method is not suitable for determining shortest even cycles, since the proof of Lemma 3.6 relies on symmetric differences and $\emptyset$-joins, and heavily uses the fact that a shortest odd $\emptyset$-join contains a shortest odd cycle, while a shortest even $\emptyset$-join is the empty edge set having weight 0. In undirected graphs shortest even cycles for nonnegative weights can, of course, be determined by solving SOP_+ problems (solvable in polynomial time according to Theorem 2.1) for the endpoints of edges. However, for directed graphs the problem is more difficult (see in Section 5.1, under the paragraph "Digraphs").

Theorem 3.7 has the immediate consequence for MOTJ_+ that after computing a w-minimum T-join F, *a w-minimum odd T-join can be computed in $2^c O(n^4)$ time where c denotes the number of connected components of F.*

Indeed, computing F takes $O(n^3)$ time for any T; see [23, Section 29.2]. If F is odd, we are done. If not, by Proposition 3.3 the minimum odd T-join problem is equivalent to determining a

$w[F]$-shortest odd cycle C in the graph G, and the set of negative edges of $w[F]$ has at most c negative components.

Since any inclusionwise minimal T-join consists of at most $|T|/2$ connected components, we also obtain the following.

Corollary 3.8. *Given an instance (G, w, T) of MOTJ where w is nonnegative, a w-minimum odd T-join (if it exists) can be computed in $2^{|T|/2}O(n^4)$ time.*

As already mentioned, the fact that MOTJ_+ can be solved in FPT time parameterized by $|T|$ has already been proved by Cook, Espinoza, and Goycoolea [8, Proposition 5.3]. Their algorithm runs in time $O(2^{|T|} + |T|^2n^2 + n^3)$, so its dependence on n is better than in Corollary 3.8, but its dependence on $|T|$ is slightly worse.

4 *NP*-completeness

We present now a well-known *NP*-complete problem that will be reduced to SOCp. Its planar special case is known to be one of the simplest open disjoint paths problems.

BACK AND FORTH PATHS (BFP)
Input: A digraph $\hat{G} = (\hat{V}, \hat{E})$ and $s \neq t \in \hat{V}$.
Question: Are there two openly disjoint paths, one from s to t, the other from t to s?

Theorem 4.1 ([11, Theorem 2], see also [23, p. 1225, footnote 6])**.** *BFP is NP-complete.*

Before proving the main *NP*-completeness results we are interested in, it will be useful to deduce the *NP*-completeness of the directed versions of SOCp_+, SOP_+, that immediately follow from this theorem.

Proposition 4.2 ([20, 26])**.** *The directed variants of the $SOCp_+$, $SECp_+$, SOP_+, and SEP_+ problems are all NP-complete.*

The proof of Lapaugh and Papadimitriou [20], that of Thomassen [26], and ours were found independently: the *NP*-complete problems used for the reductions slightly differ from one another, but they are all from [11]. We include our version of the proof to show, in a simpler situation, the starting step of our *NP*-completeness proof for undirected graphs.

Proof. The directed variant of SOCp_+ is NP-complete because given an instance (G, s, t) of BFP, subdividing each edge of G and then splitting t into an in-copy t_{in} and an out-copy t_{out} in the usual way (with all incoming edges arriving at t_{in}, all outgoing edges leaving from t_{out}, and a new edge $t_{\mathrm{in}}t_{\mathrm{out}}$), there exists an odd cycle going through s in the constructed digraph if and only if there is a pair of back and forth paths between s and t in the original digraph. Now, the directed version of SEP_+ is also *NP*-complete, since finding a shortest odd cycle through s can be done by finding a shortest even path from s to an in-neighbor of s. Clearly, the directed variants of SOP_+ and SEP_+ are equivalent, because we can flip the parity of all paths starting at a vertex s by subdividing each edge leaving s; the same trick shows that the directed variants of SOCp_+ and SECp_+ are equivalent. □

The proof shows that already the existence versions of the problems in Proposition 4.2 are *NP*-hard. However, we remark that for planar graphs the complexity of BFP is open [23, p. 1225, footnote 8] and so seems to be the complexity of SOCp for conservative planar undirected graphs or SOC_+ for directed planar graphs.

The polynomial-time solvability of SOC for nonnegative weights is straightforward (Proposition 5.1), but SOC in conservative directed graphs is a more difficult problem because neither of the tentative generalizations of Lemmas 3.4 and 3.6 hold for directed graphs. There is also no relevant indication that these problems could be polynomial-time solvable, contrary to the undirected case. Are they *NP*-hard?

We now focus on undirected graphs and switch to the statement and proof of one of our main messages.

Theorem 4.3. *$SOCp_{\pm 1}$ is NP-complete, even when the negative edges form a matching, $k = 1$, and there exists a vertex t so that $G - t$ is bipartite.*

Proof. $\mathrm{SOCp}_{\pm 1}$ is clearly in *NP*. Let the digraph $\hat{G} = (\hat{V}, \hat{E})$ with vertices $s, t \in \hat{V}$ be an instance of BFP, and construct from it an undirected graph as follows. Split each vertex $v \in \hat{V} \setminus \{t\}$ into an *out-copy* v_1 and an *in-copy* v_2, except for leaving t as it is, but defining $t_1 := t_2 := t$. For each arc $uv \in \hat{E}$ define an edge u_1v_2 with $w(u_1v_2) := 1$. Furthermore, add an edge v_1v_2 for each $v \in \hat{V} \setminus \{t\}$ with $w(v_1v_2) := -1$.

Denote $V_i := \{v_i : v \in \hat{V}\}$ for $i = 1, 2$, and $E := \{u_1v_2 : uv \in \hat{E}\} \cup \{v_1v_2 : v \in \hat{V} \setminus \{t\}\}$, so that the constructed (undirected) graph is $G = (V_1 \cup V_2, E)$, and let $k := 1$. Clearly, the negative-weight edges form a matching, and thus the weight function w is conservative. Note that $G - t$ is *bipartite, so all odd cycles contain t.*

Claim. There exists in $\hat{G}$ a cycle $\hat{C} \subseteq \hat{E}$ containing s and t if and only if there exists a cycle C in G with $w(C) = 1$ containing s_1.

Indeed, let $\hat{C} \subseteq \hat{E}$ be a (directed) cycle in $\hat{G}$ with $s, t \in V(\hat{C})$, and let us associate with it the (undirected) cycle $C := \{u_1v_2 : uv \in \hat{C}\} \cup \{v_1v_2 : v \in V(C) \setminus \{t\}\}$ in G. The cycle C alternates between edges of weight 1 and -1 in every vertex but t, so $w(C) = 1$, and $s_1 \in V(C)$.

Conversely, a cycle $C \subseteq E$ in G with $w(C) = 1$ and $s_1 \in V(C)$ must be an odd cycle due to its weight, so $t \in V(C)$ follows as noted earlier. Moreover, $C - t$ must alternate between edges of weight -1 and 1, so C corresponds to a directed cycle $\hat{C} \subseteq \hat{E}$ containing t. Since $s_1 \in V(C)$ by definition, the cycle $\hat{C}$ contains both s and t, so the claim is proved.

The claim shows that our construction reduces BFP to $\mathrm{SOCp}_{\pm 1}$, since a solution of BFP is exactly a cycle $\hat{C} \subseteq \hat{E}$ in $\hat{G}$ with $s, t \in V(\hat{C})$, and according to the claim such a cycle exists if and only if there exists an odd cycle C in G of weight at most 1 containing s_1; note that an odd cycle of weight at most 1 can have neither weight 0 (due to its parity) nor negative weight (due to conservativeness), so must have weight exactly 1. The instance $(G, p := s_1, k := 1)$ of SOCp$_{\pm 1}$ to which BFP is reduced satisfies the additional assertions, as checked above, so we can conclude that SOCp$_{\pm 1}$ is *NP*-complete and already for the family of the claimed particular instances. $\square$

By simply inspecting the instances of the above proof, the *NP*-hardness of the following problem of Lovász [23, Open Problem 27, p. 517] is an immediate corollary.

Corollary 4.4. *SOP$_{\pm 1}$ is NP-complete, even when the negative edges form a matching, $k = 1$, and there exists a vertex t so that $G - t$ is bipartite.*

Proof. SOCp is the special case of SOP where $s = t$, so we are done. If we want to require $s \neq t$, then with the notation of the proof of Theorem 4.3, observe that the instance $(G, s_1, k = 1)$ of SOCp has a "yes" answer if and only if there exists an odd (s_1, s')-path of weight $k = 1$ in the graph G' obtained from G by replacing the edge $s_1 s_2$ with an $s' s_2$ edge of weight -1 for a new vertex s'. $\square$

Note that the reduction keeps planarity, but the complexity of BFP is open for planar graphs, so we do not know the complexity of SOCp$_+$ for planar graphs.

Let us now consider the following problem which has a strong, although not immediately straightforward, relationship with the problems we study.

DISJOINT SHORTEST PATHS IN CONSERVATIVE GRAPHS (DISP)
Input: An undirected, conservative graph $G = (V, E)$ with $w : E \to \{1, -1\}$, $s_1, s_2, t_1, t_2 \in V$, and $k \in \mathbb{Z}$.
Question: Does G contain two vertex-disjoint $(\{s_1, s_2\}, \{t_1, t_2\})$-paths with total weight at most k?

While DISP for nonnegative weights is a special case of the well-known minimum cost flow problem, and for conservative digraphs as well, *it seems the question has not even been asked for conservative undirected graphs!* For these, a tentative reduction to digraphs meets the same obstacle we met for shortest paths in the introduction (Section 1): directing an edge with negative weight in both directions creates a negative cycle consisting of two arcs. However, although the undirected shortest path problem (SP) is still solvable in polynomial time even if the methods are more difficult than those applied for directed graphs, this is not the case for DISP. It turns out to be *NP*-complete, óessentially for the same reason as SOP or SOCp.

Corollary 4.5. *DISP is NP-complete, even when the negative edges form a matching and G is bipartite.*

Proof. We reduce from BFP using the same construction as in the proof of Theorem 4.3 with the only difference being that we split all vertices of the input digraph $\hat{G} = (\hat{V}, \hat{E})$, including t, add the edge $t_1 t_2$ to E, and define $w(t_1 t_2) := -1$. Then the resulting graph G is bipartite, and $(\hat{G}, s, t)$

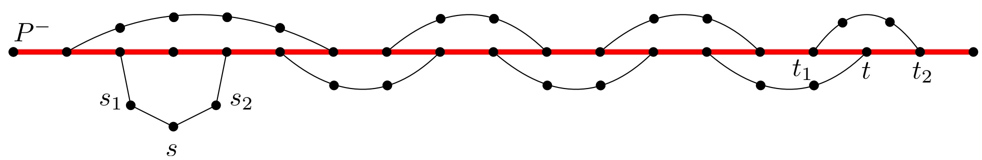


Figure 2: An example where edges of the path P^- have weight -1, shown as red, bold lines, with all remaining edges having weight 1. An odd cycle containing s must also contain t, and the unique such cycle yields also a solution for DISP (with vertices s_1, s_2, t_1, t_2), as well as a shortest odd (s_1, s_2)-path. A shortest odd $\{s_1, s_2\}$-join consists of an (s_1, s_2)-path of 4 edges and weight 0, and the odd cycle of 5 edges through t_1, t, t_2, summing up to 9 edges with total weight 1.

is a "yes"-instance of BFP if and only if there exists a cycle C of weight 0 in G containing both s_1 and t_1, which in turn holds if and only if there exist two vertex-disjoint $(\{s_1, s_2\}, \{t_1, t_2\})$-paths of total weight $k = 2$ in G. □

Corollary 4.5 contrasts the well-known fact that finding two disjoint $(\{s_1, s_2\}, \{t_1, t_2\})$-paths for some vertices s_1, s_2, t_1, and t_2 with minimum total weight in an undirected graph with nonnegative edge weights is a standard classical minimum-cost flow problem [23]. The example depicted in Figure 2 gives some intuition on the strong connection between DISP and our problems SOCp and SOP.

5 Connections, questions, and conclusion

In this section we establish further connections between the problems we have been studying to some known results and open questions.

5.1 Classical results

Forgetting the parity. As mentioned in Section 1, finding a minimum-weight perfect matching or an undirected shortest path is essentially the same problem, where the complexities are within $O(n)$ multipliers. This fact did not play a role in our discussion but could be useful to understand for capturing the different levels of difficulties:

Indeed, *a shortest (s,t)-path in a conservative graph is obviously the same as an inclusionwise minimal minimum-weight $\{s,t\}$-join; finding a minimum-weight T-join for an arbitrary even vertex set T and any weight function on the edges is, ultimately, a minimum-weight perfect matching problem (see Edmonds [10]; cf. [23, Section 29.1, 29.2]).*

Conversely, and more straightforwardly, a minimum-weight perfect matching in a graph G is a minimum-weight $V(G)$-join in the same graph after adding a sufficiently large number (e.g., the sum of the absolute values of weights) to the weight of each edge. Then a minimum weight T-join can be determined with n shortest path computations with conservative weight functions, i.e., by solving n instances of SP, based on the following idea: fix $s, t \in T$, and observe that *an edge set F is a w-minimum T-join if and only if $F = F'\Delta P$, where F' is a w-minimum $T \setminus \{s,t\}$-join, and P is a $w[F']$-minimum (s,t)-path.* Even though we do not use this observation here, the interested reader may want to check it by playing with symmetric differences as we did before.

Nonnegative weights. For polynomial algorithms to various problems see Theorem 2.1. Note that SOC_+ is much easier than SOC for both undirected and directed graphs: there is a well-known method for solving it via n shortest path computations in an auxiliary graph. We state and prove the exact complexity result for comparison and further reference in Proposition 5.1 below. Strangely, the odd path polyhedron (the "dominant" of odd paths, and the related integer minimax theorem [24], see also [23, Chapter 29.11e]) was determined much later.

Proposition 5.1 ([13, Chapter 8.3])**.** *Given an undirected or directed graph with nonnegative weights, a shortest odd cycle can be determined with at most n executions of Dijkstra's algorithm, that is, in $O(mn + n^2 \log n)$ time.*

Proof. Let $G = (V, E)$ be the input graph with edge-weight function w. If G is directed, then double the vertex set and the edge set of G by taking two distinct copies v_1 and v_2 of each $v \in V$ and, for each $uv \in E$, adding edges u_1v_2 and u_2v_1 with the same weight as uv. It is well known and easy to see that the shortest among all (v_1, v_2)-paths, $v \in V$, in the resulting (directed) graph G' yields a shortest odd cycle in G. Finding n shortest paths for nonnegative edge weights takes n executions of Dijkstra's algorithm, proving the claim for directed graphs. We remark that this approach works despite the fact that there is no one-to-one correspondence between shortest odd (v_1, v_2)-paths in G' and shortest odd cycles through v in G, which highlights a crucial difference between SOC and SOCp.

For undirected graphs the problem can be reduced to directed graphs by taking each edge in both directions. ☐

Even though SEC_+ is slightly more difficult than SOC_+, in undirected graphs it can obviously be solved with $|E(G)|$ shortest odd path computations, i.e., by solving polynomially many instances of SOP_+, which can be done in polynomial time (Theorem 2.1).

Digraphs. In Section 4 (mainly after Theorem 4.1) we mentioned the complexity for digraphs of the problems analogous to those we are studying. Similar reductions work for the shortest odd and even cycle problems through a given vertex (or, equivalently, an edge) in a directed graph, proving that these problems are *NP*-hard [20,26]; see Proposition 4.2. While for undirected graphs the shortest odd and shortest even cycles are both similarly easy to determine if all weights are nonnegative, this is not the case for directed graphs. As we have seen in Proposition 5.1, SOC_+ for directed graphs is as easy as in undirected graphs. By contrast, SEC_+ for directed graphs is inherently more difficult, and its complexity is not completely settled: finding any even (directed) cycle had been an open problem for more than two decades before being solved by Robertson, Seymour, and Thomas [22] and McCuaig [21] independently, and the problem of finding a shortest even cycle was solved very recently by Björklund, Husfeldt, and Kaski [5] but only for unweighted digraphs and with a randomized algorithm.

Two problems more closely related to our work also remain open:

Problem 5.2. *What is the complexity of SOC in digraphs with conservative weights?*

The feelings are not really oriented towards polynomial-time solvability, since nothing similar to Lemma 3.4 seems to be true, driving the search towards enumeration.

Problem 5.3. *Is $SOCp_+$ polynomially solvable in planar directed graphs? More generally, what is the complexity for planar graphs of the problems proved to be NP-hard (for undirected and directed graphs) in this article?*

The source of this questioning is that BFP is open for planar graphs [23, p. 1225, footnote 8].

Odd T-joins. Their properties with respect to packing and covering have been intensively studied in terms of the "idealness" (integrality) of their blocking polyhedra. Idealness roughly means that good characterization (minimax) theorems hold for the minimization of odd T-joins *for nonnegative weight functions*.

The corresponding algorithms and complexity results have been analyzed in Section 3, where we anticipated that nonnegativity is not an essential condition in this case. We provide a precise proof below for the equivalence of arbitrary weight functions with nonnegative ones (Theorem 5.4).

Max Cut. The "min side" of the mentioned minimax theorems concerns transversals of odd T-joins which in the simplest case of odd cycles (i.e., $T = \emptyset$) are easily seen to be exactly the complements of cuts: their minimization is equivalent to the MAXIMUM CUT problem, one of the sample *NP*-hard problems. However, for planar graphs the duality between faces and vertices reduces this problem to the shortest T-join problem [3], solving MAXIMUM CUT for planar graphs; for graphs embeddable into the projective plane the corresponding reduction is to MOTJ, and only a partial solution could be given to the corresponding special case of MOTJ [7].

SOC versus SOCp. SOC can clearly be reduced to SOCp, but the opposite reduction seems to organically resist. This situation is analogous to the problem of finding a minimum-weight odd *hole* (an induced cycle of cardinality at least four) through a given vertex, which is *NP*-complete [4], while without the requirement of containing a given vertex, it has been recently proved to be polynomially solvable [6].

The applications and relevance of the SOC and MOTJ problems and signs of their tractability, mentioned in Section 3 and leading to Conjecture 3.1, makes it interesting to clarify their polynomial equivalence, which we do in Section 5.2.

5.2 Equivalence of SOC and MOTJ

In this section we show that weighted optimization problems on odd T-joins are actually polynomially equivalent to their special case for conservative weight functions, which in turn can be shown to be equivalent to the case where w is restricted to be nonnegative or T to be empty.

Theorem 5.4. *The following problems are polynomially equivalent:*

(i) MOTJ,

(ii) MOTJ with conservative weights,

(iii) $MOTJ_+$,

(iv) MOTJ with conservative weights for $T = \emptyset$,

(v) SOC with conservative weights.

Proof. A polynomial algorithm for (i), i.e., MOTJ in general, clearly implies one for (ii), which, in turn, implies one for (iii).

To prove the polynomial-time solvability of (iv) from that of (iii), consider the input of MOTJ with $T = \emptyset$ consisting of a graph $G = (V, E)$ and a conservative w. We can assume that G contains an even number of negative edges, since otherwise we can simply add to G an edge of weight -1 incident to a new vertex. Define now a nonnegative weighted instance $(G, |w|, T)$ of MOTJ with $T := \{v \in V : d_{E^-}(v) \text{ is odd}\}$ where $E^- := \{e \in E : w(e) < 0\}$. Then E^- is a $|w|$-minimum T-join, and it is even. Now by Proposition 3.3, J is a $|w|$-minimum odd T-join if and only if $C := J \Delta E^-$ is a $w = |w|[E^-]$-minimum odd $\emptyset$-join. Hence, an algorithm for (iii) applied to $(G, |w|, T)$ yields a solution for our instance (G, w) of (iv).

The claim that polynomial-time solvability of (iv) implies the same for (v) follows by noting that a solution for (v) can be obtained from a solution for (iv) with the same input instance by deleting the 0-weight even cycles and possibly all but one 0-weight odd cycle.

We have thus asserted the path of implications from the polynomial-time solvability of (i) to that of (v). A polynomial-time algorithm for (i) follows from one for (v) by Proposition 3.3, since a shortest odd cycle for conservative weights is always a minimum-weight odd $\emptyset$-join. $\square$

Note that (iv) is a specialization of (ii) to the special case $T = \emptyset$, and actually any special T can play this role of implying a polynomial solution to (i)–(v). (Indeed, to prove this, one only has to change w to $w[J]$ in the proof of (iv), where J is a w-minimum T-join, and then E^- is changed to $E^- \Delta J$. In the proof we had $T = \emptyset$, and $\emptyset$ is a shortest $\emptyset$-join in a conservative graph.) Figure 2 illustrates how much easier it is to find a shortest odd $\{s, t\}$-join than to find a shortest odd (s, t)-path in a graph with a conservative weight function.

Restricting MOTJ with $|T| \leq 2$ and requiring at the same time nonnegative weights results in an easy problem, as we have shown in Section 3.1. However, if only one of T and w is restricted, then the general problem can be reduced to these (seemingly) more special ones, as stated by Theorem 5.4. The cases where the absolute values of the weights are 1 are not proved to be essentially easier than general weights for any of the problems, *so they also remain wide open.*

5.3 Conclusion

The MOTJ problem is a relevant combinatorial optimization problem that may be solvable in polynomial time. The complexity of this problem remains open, but we proved that SOC, polynomially equivalent to $MOTJ_+$ or MOTJ, is fixed-parameter tractable when parameterized by the number of components of negative edges. If negative weights are also allowed, then finding a minimum-weight odd $\{s,t\}$-join is already equivalent to general MOTJ.

We also proved that the related SOCp, SECp, SOP, SEP, and DISP problems in conservative undirected graphs are *NP*-complete, answering a long-standing question of Lovász [23, Problem 27], and we exhibited some related, polynomial algorithms. At the same time, we pointed out three open challenges for undirected and directed graphs, one of which is MOTJ itself.

Another interesting research direction is now to study the parameterized complexity and approximability of the SOP problem for both directed and undirected graphs and its other *NP*-hard variants. Some initial FPT results have been achieved by part of our research group formed during the 12th Emléktábla Workshop held in Gárdony, Hungary in 2022 [17].

Acknowledgement

The collaboration of the authors has been hosted and supported by the 12th Emléktábla Workshop, Gárdony, Hungary, July 2022; we would like to express our gratitude for the organizers. Also, we would like to thank Yutaro Yamaguchi for initiating a research group on the problem of finding shortest odd paths in conservative graphs, and Alpár Jüttner, Csaba Király, Lydia Mendoza, Gyula Pap as well as Yutaro for the enjoyable discussions. Ildikó Schlotter acknowledges the support of the Hungarian Academy of Sciences under its Momentum Programme (LP2021-2), and the Hungarian Scientific Research Fund (OTKA grants K128611 and K124171).

We thank a very thorough referee for his pertinent comments that induced important corrections.

References

[1] A. Abdi. The cycling property for the clutter of odd *st*-walks. Master's thesis, University of Waterloo, 2014.

[2] A. Abdi and B. Guenin. Packing odd *T*-joins with at most two terminals, *Journal of Graph Theory*, 87(4):587–652, 2018. https://doi.org/10.1002/jgt.22178

[3] F. Barahona. On the complexity of max cut. Rapport de recherche no. 186, Mathématiques Appliqués et Informatique, Université Scientifique et Médicale de Grenoble, 1980.

[4] D. Bienstock. On the complexity of testing for odd holes and induced odd paths. *Discrete Mathematics*, 90(1):85–92, 1991. https://doi.org/10.1016/0012-365X(91)90098-M

[5] A. Björklund, T. Husfeldt and P. Kaski. The shortest even cycle problem is tractable. In *Proc. of the 54th Annual ACM SIGACT Symposium on Theory of Computing (STOC 2022)*, ACM Press, pp. 117–130, 2022. https://dl.acm.org/doi/10.1145/3519935.3520030

[6] M. Chudnovsky, A. Scott, P. Seymour, and S. Spirkl. Detecting an odd hole. *Journal of the ACM*, 67(1), Article 5, 2020. https://doi.org/10.1145/3375720

[7] É. Colin de Verdière, F. Meunier, A. Sebő, and M. Stehlík. Max Cut in projective planar graphs: homologic T-joins and the matching lattice. In preparation.

[8] W. Cook, D.G. Espinoza, and M. Goycoolea. Generalized domino-parity inequalities for the symmetric traveling salesman problem, *Mathematics of Operations Research*, 35(2):479–493, 2010. https://doi.org/10.1287/moor.1100.0451

[9] U. Derigs. An efficient Dijkstra-like labeling method for computing shortest odd/even paths. *Information Processing Letters*, 21(5):253–258, 1985. https://doi.org/10.1016/0020-0190(85)90094-8

[10] J. Edmonds. The Chinese postman's problem. *Bulletin of the Operations Research Society of America*, 13, B-73, 1965.

[11] S. Fortune, J. Hopcroft, and J. Wyllie. The directed subgraph homeomorphism problem, *Theoretical Computer Science,* 10(2):111–121, 1980. https://doi.org/10.1016/0304-3975(80)90009-2

[12] J. Geelen and R. Kapadia. Computing girth and cogirth in perturbed graphic matroids. *Combinatorica*, 38(1):167–191, 2018. https://doi.org/10.1007/s00493-016-3445-3

[13] M. Grötschel, L. Lovász, and A.J. Schrijver. *Geometric Algorithms and Combinatorial Optimization.* Springer, Berlin, 1988.

[14] M. Grötschel and W. Pulleyblank. Weakly bipartite graphs and the Max-cut problem. *Operations Research Letters*, 1(1):23–27, 1981. https://doi.org/10.1016/0167-6377(81)90020-1

[15] B. Guenin. A characterization of weakly bipartite graphs. *Journal of Combinatorial Theory*, Series B (83):112–168, 2001. https://doi.org/10.1006/jctb.2001.2051

[16] B. Guenin. Integral polyhedra related to even-cycle and even-cut matroids. *Mathematics of Operations Research*, 27(4):693–710, 2002. https://doi.org/10.1287/moor.27.4.693.299

[17] A. Jüttner, Cs. Király, L.M. Mendoza-Cadena, Gy. Pap, I. Schlotter, and Y. Yamaguchi. Shortest odd paths in undirected graphs with conservative weight functions. *Discrete Applied Mathematics*, 357:34–50, 2024. https://doi.org/10.1016/j.dam.2024.05.044

[18] B. Korte and J. Vygen. *Combinatorial Optimization.* 6th edition, Springer Berlin, Heidelberg, 2018.

[19] M.-K. Kwan. Graphic programming using odd or even points. *Acta Mathematica Sinica* (in Chinese), 10:263–266, 1960. Translated in *Chinese Mathematics*, 1:273–277, 1962.

[20] A.S. Lapaugh and C.H. Papadimitriou. The even-path problem for graphs and digraphs. *Networks*, 14(4):507–513, 1984. https://doi.org/10.1002/net.3230140403

[21] W. McCuaig. Pólya's permanent problem. *Electronic Journal of Combinatorics*, 11(1), Article R79, 2004. https://doi.org/10.37236/1832

[22] N. Robertson, P.D. Seymour and R. Thomas. Permanents, Pfaffian orientations, and even directed circuits. *Annals of Mathematics*, 150(3):929–975, 1999. https://doi.org/10.2307/121059

[23] A. Schrijver. *Combinatorial Optimization*. Springer-Verlag Berlin Heidelberg, 2003.

[24] A. Schrijver and P.D. Seymour. Packing odd paths. *Journal of Combinatorial Theory, Series B*, 62:280–288, 1994. https://doi.org/10.1006/jctb.1994.1070

[25] P.D. Seymour. Matroids and multicommodity flows. *European Journal of Combinatorics*, 2:257–290, 1981. https://doi.org/10.1016/S0195-6698(81)80033-9

[26] C. Thomassen. Even cycles in directed graphs. *European Journal of Combinatorics*, 6(1):85–89, 1985. https://doi.org/10.1016/S0195-6698(85)80025-1