Minimum violation vertex maps and their applications to cut problems*

Ken-ichi Kawarabayashi† Chao Xu‡

July 24, 2020

Abstract

The minimum violation problem asks for a vertex map from a digraph to a pattern digraph that minimizes violation, the total weight of the edges not mapped to an edge. We are interested in surjective mappings. We characterize all patterns where a minimum violation map that fixes some vertices can be computed in polynomial time. We also make progress in the case where we do not fix any vertex in the mapping, including when the digraph is disconnected, when the graph is in the variety of finite paths. Moreover, we obtain a dichotomy result for trees. We apply the result to some cut problems, including k-cut with size lower bounds and length bounded k-cuts.

1 Introduction

The graph homomorphism problem is a classic decision problem that asks if there exists a homomorphism from a graph G to H. That is, is there a mapping of the vertices, such that edges in G map to edges in H.

Many problems can be modeled as graph homomorphism problems, for example, graph coloring [30]. The problem also generalizes to the directed case. In optimization contexts, knowing if there exists a homomorphism is not enough. Therefore, there are generalizations to finding homomorphisms of minimum cost [28, 31].

In this paper, we consider another generalization of the graph homomorphism problem. A special case was considered by Elem, Hassin and Monnot [18]. Instead of asking for a homomorphism of minimum cost, we are interested in how close a map can be to a homomorphism. It captures the idea as to how many edges to remove to obtain a homomorphism. We establish the result on more general digraphs.

For two digraphs $G = (V, E)$ and $H = (U, F)$, a *vertex map* from G to H is a mapping $f : V \to U$. We call the digraph H the *pattern*. An edge $uv \in E$ is a *violated edge* under f if $f(u)f(v) \notin F$. The *violation* of a map is the number of violated edges. If the digraph G is weighted, then the violation is the total weight of the violated edges. That is, for a weight function $w : E \to \mathbb{N}$, the violation of the vertex map $f : V \to U$ is

$$\sum_{uv\in E, f(u)f(v)\notin F} w(uv).$$

We consider the following problems, where $H = (U, F)$ is a fixed digraph.

- The *surjective minimum violation problem for H*, SVIO(H): Given a digraph $G = (V, E)$ with weight $w : E \to \mathbb{N}$, find a surjective vertex map from G to H that minimizes the violation.

*Part of this work was done while Chao Xu was visiting National Institute of Informatics. Work supported in part by NSF EAPSI fellowship (award abstract 1714027) and JSPS KAKENHI Grant Number JP18H05291.

†National Institute of Informatics, Tokyo, Japan k_keniti@nii.ac.jp

‡Previous Affiliation:Department of Computer Science, University of Illinois at Urbana-Champaign, Current Affiliation: The Voleon Group, Berkeley, CA the.chao.xu@gmail.com

- The *retraction minimum violation problem for H*, RVIO(H): Given a digraph $G = (V, E)$ such that $U \subseteq V$, and a weight function $w : E \to \mathbb{N}$. Find a vertex map f from G to H, such that the restriction $f|_U$ of f to U is the identity function and the violation is minimized.

For SVIO(H), we assume that $|V| \geq |U|$ as it is necessary and sufficient for the existence of a feasible surjective vertex map. A problem is *tractable* if every instance can be solved in polynomial time with respect to the instance size. A digraph H is *s-tractable* if SVIO(H) is tractable and *r-tractable* if RVIO(H) is tractable. In particular, s-tractable is a strictly weaker property than r-tractable, since there are digraphs that are s-tractable but not r-tractable, the simplest is a graph consisting of 3 isolated vertices. A digraph is *reflexive*, if every vertex has a self-loop. SVIO(H) and RVIO(H) on an appropriate reflexive digraph H models a variety of graph cut problems (see section 1.2). We are interested in classifying which digraphs are r-tractable, and which are s-tractable. The r-tractable digraphs have been characterized through the algebraic approach [49]. However, given a digraph H, it is unclear if one can test if H is r-tractable in polynomial time. A digraph H is *s-intractable* if SVIO(H) is NP-hard. A dichotomy result would classify each digraph H as either s-tractable or s-intractable. Note we differentiate between s-intractable and not s-tractable. A digraph H could be neither s-tractable nor s-intractable. Indeed, SVIO(H) could be a NP-intermediate problem, which exists assuming $P \neq NP$ [40].

We give an *efficient* characterization of the r-tractable digraphs and make progress in classifying s-tractable reflexive digraphs. We also give some s-tractability and s-intractability results for particular classes of digraphs.

1.1 Previous and related work

A vertex map f from G to H is a *homomorphism* if every edge maps to an edge. The violation is 0 if and only if the mapping is a homomorphism. The minimum violation problem is an optimization version of the homomorphism problem. Various homomorphism problems have been studied (see [30] for a survey). We will focus on the variations most relevant to our study. The *H-retraction problem*, RHOM(H), can be defined as deciding if the input digraph G has a homomorphism from G to H that fixes the vertices in H. The digraphs where RHOM(H) is tractable are called *r_0-tractable*. Feder and Vardi showed for each H, RHOM(H) is equivalent to some constraint satisfaction problem (CSP) depending on H [20]. Since the resolution of CSP dichotomy conjecture, the r_0-tractable digraph have been completely classified [6, 52]. However, it is not clear that there is an efficient algorithm for recognizing r_0-tractable digraphs. Note RHOM(H) is not a special case of RVIO(H). In the *H-surjective homomorphism problem*, denoted by SHOM(H), the input is a digraph G, and the goal is to verify if there is a surjective homomorphism from G to H. The problem is also known as surjective H-coloring. A digraph H is *s_0-tractable* if SHOM(H) is tractable. There are many works on s_0-tractable digraphs. We refer to the recent survey by Bodirsky *et al.* [5]. A recent breakthrough showed that a length 4 reflexive cycle is not s_0-tractable [45]. Chen studied an algebraic criterion for deciding the equivalence of SHOM(H) and RHOM(H) [12]. It was used to show that directed reflexive cycles and non-transitive tournaments are not s_0-tractable [42]. The approximation aspect of graph homomorphism was studied in [41]

To the best of our knowledge, the only work on s-tractability of graphs is the work by Elem *et al.* [19]. They used the terminology of G_c-cut and G_c-multiway-cut for SVIO($\bar{G}$) and RVIO($\bar{G}$), respectively. Here $\bar{G}$ is the complement graph of G. They showed approximation results of finding the minimum violation is NP-hard when the edge weight of the graph forms a metric. One important result is that a graph is r-tractable if the set of vertices with maximal neighborhoods forms a clique.

For s-tractability, Elem *et al.* showed various results, all of which follow from the fact that an r-tractable graph is s-tractable, and s_0-intractable graphs are s-intractable. In addition, they claimed that when the graph is a disjoint union of reflexive complete graphs, it is s-tractable. However, their proof sketch is missing details, and only the case where each complete graph is a single vertex can be verified.

In the *minimum cost homomorphism problem* with respect to H, we are given a digraph $G = (V, E)$ and a cost function $c : V \times V(H) \to \mathbb{N}$. The *cost* of a map f is $\sum_{v \in V} c(v, f(v))$. The problem asks for a homomorphism with minimum cost. A complete classification of pattern H that gives tractable problems for graphs and digraphs is known [28, 31]. For the approximation aspects of minimum cost homomorphism, Hell *et al.*showed that for non-reflexive graphs, co-circular arc bigraphs are the only ones with constant factor approximation [29]. Rafiey, Rafiey and Santos extended the result to a dichotomy of all graph, and obtained some progress on digraphs [47]. Note that the previous authors allowed infinite cost, but it can be simulated by having cost with much larger value, as since the cost itself is part of the input. The *minimum cost and violation problem*, CVIO(H), does not require the violation to be 0. Instead, it asks for a map that minimizes the sum of the cost and violation. H is *c-tractable* if CVIO(H) is tractable. Deineko *et al.* gave an efficient classification of c-tractable digraphs as a consequence of MAXCSP theory. H is c-tractable if and only if the adjacency matrix of H is permuted anti-monge [15]. The approximation aspects were also considered.

Finally, there is another closely related problem. Given a graph $G = (V, E)$, a weight function $w : E \to \mathbb{N}$, a metric $d : U \times U \to \mathbb{N}$, and cost $c : V \times U \to \mathbb{N}$. We want to find a function $f : V \to U$, such that it minimizes $\sum_{v \in V} c(v, f(v)) + \sum_{uv \in E} w(uv) d(f(u), f(v))$. The problem is studied under the name the metric labeling problem [38]. If c is 0 everywhere, it is called the 0-*extension problem* [8]. The specific metrics d where 0-extension problem can be solved in polynomial time was characterized by Hirai [33].

Valued CSP The problem RVIO and CVIO are special cases of Valued CSPs (VCSP) over weighted constraint languages. That is, for each H, there exists a weighted constraint language Γ, such that RVIO(H) (CVIO(H)) is equivalent to VCSP(Γ). The complexity dichotomy of VCSP result was completely resolved [49]. Given Γ, testing if VCSP(Γ) is polynomial time solvable is NP-complete [49], but can be solved in polynomial time when the size of the domain is a constant [39]. Similarly, each SVIO(H) is equivalent to a Surjective Valued CSP on some constraint language Γ, denoted SVCSP(Γ). However, SVCSP is much less well understood. SVCSP for the Boolean case are completely understood [21], which implies the case for SVIO(H) where H consists of 2 vertices. In particular, in the same paper, they showed examples of graph H that is s-tractable but not r-tractable. Recently, Matl and Živný have made additional progress to SVCSP, characterizing more cases where the problem can be solved in polynomial time [46]. However, it does not provide new insights to SVIO.

1.2 Applications to cut problems

A cut problem is a problem where one removes the minimum number of edges to "disconnect" some set of k terminals. Here the definition of disconnect varies among the problems. In the fixed-terminal cut problems, we fix the set of terminals to disconnect. The global cut problems take the minimum over all fixed-terminal cuts. If the fixed-terminal cut problem is equivalent to RVIO(H), then the global cut problem is usually equivalent to SVIO(H). In cut applications, the pattern H is always reflexive. SVIO(H) is closer to a coloring problem if H does not have any self-loop.

A *k-partition of V* is a k-tuple of pairwise disjoint non-empty sets $(V_1, \ldots, V_k)$ such that their union is V. Each V_i is a *partition class*. An edge *crosses* the partition if its two endpoints are in different partition classes. The *value* of a set of edges is the sum of the weight of the edges. The *value* of a partition is the value of the edges crossing the partition. Many cut problems often have an equivalent formulation as finding a k-partition satisfying a certain property.

Here we survey some cut problems that can be modeled by the minimum violation framework.

***k*-way-cut and *k*-cut** Given a graph G and k terminal vertices T, a *k-way-cut* is a set of edges C, such that each vertex in T is in a different component of $G - C$. A *k-cut* is a k-way-cut for some set of k terminals. The kWAYCUT problem asks for a minimum k-way-cut, given the input graph and k terminals. kWAYCUT is the same as finding a k-partition where each terminal is in a different partition class, and

Find a minimum	Directed?	Property P	Equivalent pattern H
k-cut	undirected	no path between terminals	k isolated vertices
s-size-k-cut	undirected	a k-cut with s-size constraints	$K_{s_1} \cup \cdots \cup K_{s_k}$
(ℓ, k)-cut	undirected	no length $\leq \ell$ path between terminals	$B_{\ell+1,k}$
k-reach-cut	directed	no vertex can reach two terminals	S_k
linear-k-cut	directed	t_i cannot reach t_j for $j > i$	T_k
bicut	directed	no $t_1 t_2$-path nor $t_2 t_1$-path	H_{bicut}

Table 1: The cut problems. How to read the table: Finding a minimum weight set of edges C in a undirected(directed) graph such that $G - C$ has property P over some sequence of terminals $(t_1, \ldots, t_k)$ is equivalent to SVIO(H).

the number of edges crossing partitions is minimized. The kCUT problem asks for the minimum k-cut in the graph. Hence kCUT is the global version of kWAYCUT. Let kK_1 be the graph that consists of k isolated vertices with self-loops. kWAYCUT is equivalent to RVIO(kK_1) and kCUT is equivalent to SVIO(kK_1). kCUT is solvable in polynomial time for all fixed k [23]. kCUT is W[1]-hard in terms of k [16], but it is in FPT for cut-size [37]. kWAYCUT is NP-hard for $k \geq 3$ [13]. kCUT is an example of the global problem strictly easier than the fixed-terminal problem.

***s*-size *k*-cut** Let $s = (s_1, \ldots, s_k)$ be a non-decreasing vector of positive integers, where the sum is σ. For a graph G, a k-cut C is a *s-size k-cut* if we can find a k-partition $(V_1, \ldots, V_k)$, such that $|V_i| \geq s_i$ for all $1 \leq i \leq k$, and C are the edges crossing the k-partition. The sSIZEkCUT problem asks to find a s-size k-cut of minimum weight for an input graph G. We recover kCUT when s is the all 1 vector. sSIZEkCUT is equivalent to SVIO($K_{s_1} \cup \ldots \cup K_{s_k}$), where K_n is the *reflexive complete graph* on n vertices. We always assume the number of vertices in the input graph is at least $\sum_{i=1}^{k} s_i$.

When $k = 2$, there are at most $t = \sum_{i=1}^{s_2-1} \binom{n}{i} = O(n^{s_2-1})$ cuts with the smaller side size smaller than s_2. By the pigeonhole principle, one of the smallest $t+1$ cuts is a minimum s-size cut. Hence we can obtain an algorithm with running time $\tilde{O}(mn^{s_2}) = \tilde{O}(mn^{\sigma - s_1})$, where $\tilde{O}$ hides log factors by enumerating the smallest $t+1$ cuts [50]. Here n and m are the number of vertices and edges of the input graph, respectively. Elem *et al.* claimed the algorithm of Goldschmidt and Hochbaum can be modified to solve sSIZEkCUT for fixed k, but we could not verify the proof [19]. A randomized algorithm is known for arbitrary k with running time $\tilde{O}(n^{2\sigma})$ in an unpublished manuscript [24]. The s-size k-cut problem was also noted as the *lower-bounded k-Way Min-Cut problem* [46].

(ℓ, k)-way-cut and (ℓ, k)-cut The length of a path is the number of edges in the path. Consider a graph G with k specified terminal vertices $t_1, \ldots, t_k$. A set of edges C such that in $G - C$, the distance between each pair of terminals is at least $\ell + 1$, is a *(ℓ, k)-way-cut*. A *(ℓ, k)-cut* is an (ℓ, k)-way-cut for some set of k terminals. In the graph with an (ℓ, k)-way-cut removed, the distance between every pair of terminals is at least $\ell + 1$. The (ℓ, k)-way-cut problem, denoted by (ℓ, k)WAYCUT, takes an input graph G and k terminals, and returns an (ℓ, k)-way-cut of minimum weight. Similarly, finding a minimum (ℓ, k)-cut is the problem denoted by (ℓ, k)CUT. Note (∞, k)-way-cut is the standard k-way-cut. For the special case where $k = 2$, $(\ell, 2)$-way-cut is the ℓ-length bounded st-cut. In this case, $(\ell, 2)$WAYCUT is polynomial time solvable for $\ell \leq 3$ and $\ell = \infty$. It is NP-hard if $4 \leq \ell < \infty$, but constant approximation is possible [1, 35, 44]. Note $(\ell, 2)$CUT is equivalent to remove the minimum number of edges to increase the diameter to at least $\ell + 1$. It was shown to be NP-hard if ℓ is not fixed [48]. Recently, there is a study on the FPT aspects of a more general version of (ℓ, k)-way-cut, which specify a length lower bound for each pair of the vertices [17]. However, it is still unknown for which fixed ℓ and $k \geq 3$ the problem is NP-hard. Consider that ℓ is finite, let $B_{k,\ell}$ be the following reflexive graph. If ℓ is even, it consists of k

copies of length $\frac{\ell}{2}$ paths. One endpoint of each path is identified. If ℓ is odd, it consists of k copies of length $\frac{\ell-1}{2}$ paths and a clique defined on one endpoint of each path (See Figure 1.1a and Figure 1.1b). We show $(\ell-1,k)$WAYCUT, $(\ell-1,k)$CUT, RVIO($B_{k,\ell}$) and SVIO($B_{k,\ell}$) are all equivalent.

k-way-reach-cut and k-reach-cut Let G be a digraph. A set of edges C is a *k-way-reach-cut* for a set of k terminals T, if in $G-C$, each vertex can reach at most one terminal in T. C is a *k-reach-cut* if C is a k-way-reach-cut for some set of k terminals. Again, kWAYREACHCUT and kREACHCUT denotes the problem of finding a minimum k-way-reach-cut and k-reach-cut, respectively. When $k=2$, kWAYREACHCUT is studied as the minimum st-double cut problem. Bernáth and Pap showed that minimum st-double cut can be solved in polynomial time using a flow-based approach [3]. The minimum 2-reach-cut is precisely the minimum set of edges such that its removal destroys all arborescences in the digraph [3]. Let S_k be a reflexive directed in-star, that is, a set of k vertices having an outgoing edge to the center vertex. (See Figure 1.1d). It was shown that kWAYREACHCUT is equivalent to RVIO(S_k) and kREACHCUT is equivalent to SVIO(S_k) [9]. We will show S_3 is not r-tractable, but s-tractable.

Linear-k-cut For a digraph G, a set of edges C is a *linear-k-way-cut* for a *tuple of k terminals* $(t_1,\dots,t_k)$, if there is no path from t_i to t_j for all $i<j$ in $G-C$. C is a *linear-k-cut* if it is a linear-k-way-cut for some k-tuple of terminals. LINEARkWAYCUT and LINEARkCUT denote the problem of finding a minimum linear-k-way-cut and a minimum linear-k-cut. LINEARkWAYCUT was studied in approximating multicuts [10]. LINEARkWAYCUT is NP-hard for all $k\geq 3$. A $\sqrt{2}$-approximation algorithm exists for the case when $k=3$, and it is tight assuming the Unique Game Conjecture [7]. It is unknown if LINEARkCUT is tractable. LINEARkWAYCUT is equivalent to finding k nested sets $V_1\subseteq\cdots\subseteq V_k=V$, such that $t_i\in V_i\setminus V_{i-1}$ and the total number of incoming edges to $V_1,\dots,V_k$ is minimized ([2] proved $k=3$ case). It is not hard to see that LINEARkWAYCUT is equivalent to RVIO(T_k) and LINEARkCUT is equivalent to SVIO(T_k), where T_k is the reflexive transitive tournament on k vertices (See Figure 1.1c).

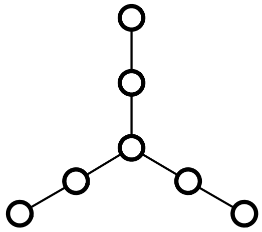
Bicut A set of edges is a *st-bicut* if removing it disconnects s and t in both ways. That is, there is no path from s to t nor from t to s. A set of edges is a *bicut* if it is a st-bicut for some s and t. Finding a minimum st-bicut is NP-hard but a simple 2-approximation algorithm exists [22]. There is no efficient $(2-\epsilon)$-approximation for any constant $\epsilon>0$ assuming the Unique Game Conjecture [10, 43]. However, it is unknown if finding the minimum bicut is NP-hard. A $(2-1/448)$-approximation exists, showing the problem could be easier than finding a minimum st-bicut [2]. Finding a minimum bicut can be reduced to SVIO(H_{bicut}), where H_{bicut} is the graph in Figure 1.1e [2].

1.3 Main contributions

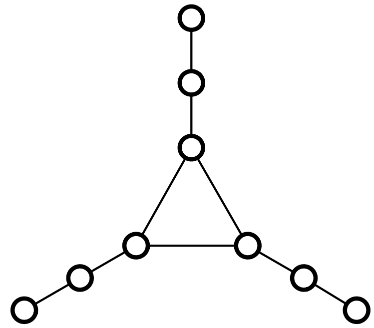
Our main contributions are the following:

- We give an *efficient* classification of r-tractable digraphs. (Theorem 3.3)
- We show that a reflexive digraph is s-tractable if and only if each of its components is s-tractable. (Theorem 4.1)
- We make progress on s-tractability for the variety of reflexive finite paths. In particular, we establish the dichotomy of reflexive trees. (Theorem 5.9)
- We show s-tractability result for star-like digraphs, which shows the first example where r-tractability is not equivalent to s-tractability and the digraph is weakly connected. (Theorem 6.2)

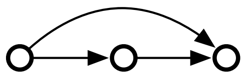
We show a *deterministic* algorithm for sSIZEkCUT faster than the current randomized algorithm [24] (Theorem 4.4), and we also resolve the complexity of (ℓ,k)CUT and (ℓ,k)WAYCUT (Corollary 5.6).



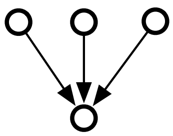
(a) $B_{k,\ell}$, $k = 3$, $\ell = 4$



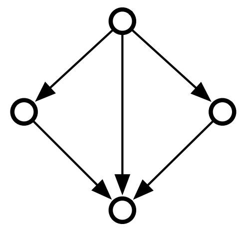
(b) $B_{k,\ell}$, $k = 3$, $\ell = 5$



(c) T_3



(d) S_k, $k = 3$



(e) H_{bicut}

Figure 1.1: Various patterns useful in cut problems. All graphs are reflexive. Self-loops are not shown.

2 Preliminaries

The set of positive integers from 1 to n is denoted by $[n]$. Let $G = (V, E)$ and $H = (U, F)$ be digraphs. $G \cup H$ is a new digraph $(V \cup U, E \cup F)$. $G[V']$ is an induced subgraph of G on V', defined as $G[V'] = (V', \{e | e \subseteq V', e \in E\})$. G is *trivial* if it does not contain any edge, *reflexive* if every vertex has a self-loop. The *out neighbors* and *in neighbors* of v are defined as $N_G^+(v) = \{u \mid vu \in E\}$ and $N_G^-(v) = \{u \mid uv \in E\}$, respectively. The set of all neighborhoods is $\mathcal{N}_G = \{N_G^+(v) | v \in V\} \cup \{N_G^-(v) | v \in V\}$. For any set A, we define the *complement characteristic function* $\bar{\chi}_A(x) = 0$ if $x \in A$ and 1 otherwise. The universe of $\bar{\chi}_A$ is always non-ambiguous from context. We define $h_G : V^2 \to \{0, 1\}$ as $h_G = \bar{\chi}_E$. That is, $h_G(u, v) = 0$ if and only if uv is an edge in G. A problem is *polynomial-time equivalent* (or *equivalent* for short) to another, if there is a polynomial-time Turing reduction between them. We use *components* in digraphs to refer to the weakly connected components. The *distance* between u and v in G, denoted by $d_G(u, v)$, is the length of the shortest path from u to v. The distance $d_G(u, v) = \infty$ if u cannot reach v. In the entire paper, k and ℓ are assumed to be constants.

Recall that if a digraph is r-tractable, then it is s-tractable. If it is not s_0-tractable, then it is not s-tractable.

We use n and m to denote the number of vertices and number of edges, respectively, in the input digraph G.

Let H be a digraph. We always assume there is some total ordering $\prec$ of the vertices, e.g., based on vertex labels. A vertex v *dominates* u if

1. $N_H^-(u) \subseteq N_H^-(v)$ and $N_H^+(u) \subsetneq N_H^+(v)$, or
2. $N_H^-(u) \subsetneq N_H^-(v)$ and $N_H^+(u) \subseteq N_H^+(v)$, or
3. $N_H^-(u) = N_H^-(v)$, $N_H^+(u) = N_H^+(v)$ and $u \prec v$.

Domination relation is an inclusion relation of the neighborhoods with ties broken by the vertex total order. A vertex is a *dominated vertex* if it is dominated by some other vertex. There is always a minimum violation surjective map from G to H such that the preimage of each dominated vertex is a single vertex [19].

We introduce some notions to describe the minimum violation problem in both digraphs and graphs uniformly. A digraph is *symmetric* if (u,v) is an edge iff (v,u) is an edge. For simplicity, we formally define a *graph* to be a symmetric digraph. However, we will still use standard graph terminology. For example, removing an edge in a graph is equivalent to removing the two opposing edges (or a self-loop) in the symmetric digraph. Other notions follow similarly.

If H is a graph, we define $\text{SVIO}_u(H)$ to be $\text{SVIO}(H)$ but with input restricted to graphs. It is easy to see that $\text{SVIO}(H)$ is tractable if and only if $\text{SVIO}_u(H)$ is tractable. Therefore, in the paper, the input of the problem $\text{SVIO}(H)$ is assumed to be a graph if H is a graph. The same statement holds for every variation of the minimum violation problem described in this paper.

2.1 Distances and retraction

Let G be a graph, and H a subgraph of G. A homomorphism from G to H is a *retraction* if it is the identity function when the domain is restricted to vertices on H. H is a *retract* of G if there is a retraction from G to H. H is a *isometric subgraph* of G if $d_H(u,v) = d_G(u,v)$ for all $u,v \in U$. Namely, $d_H = d_G|_{U\times U}$.

Theorem 2.1 *For any homomorphism f from G to H, $d_G(x,y) \geq d_H(f(x),f(y))$.*

Theorem 2.2 *Let H be a retract of H'. If H' is r-tractable then H is r-tractable.*

Proof: Let ϕ be a retraction from H' to H. Consider an input graph G to $\text{RVIO}(H)$, we construct $G' = G \cup H'$.

Let f be the optimal solution of $\text{RVIO}(H)$ with input graph G. f has value α. Let f' be the optimal solution of $\text{RVIO}(H')$ with input graph G'. f' has value α'.

We will show that $\alpha = \alpha'$. Construct a map $g : V(G) \to V(H)$, such that $g(v) = \phi(f'(v))$. The map g is a feasible solution of $\text{RVIO}(H)$ because ϕ is a retraction. Since ϕ is a homomorphism, g has violation no larger than f', therefore this shows $\alpha' \geq \alpha$. We construct g' from f such that g' is a feasible solution to $\text{RVIO}(H')$ with input G'. $g'(v) = v$ if $v \in V(H')$ and $g'(v) = f(v)$ otherwise. The violation of g' is at most the violation of f, hence this shows $\alpha \geq \alpha'$. □

The *(direct) product* of two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ is $G_1 \times G_2 = (V_1 \times V_2, E)$, where $E = \{\{(u_1,u_2),(v_1,v_2)\} \mid u_1v_1 \in E_1 \text{ and } u_2v_2 \in E_2\}$. One can generalize it to product of finite number of graphs. $\prod_{i=1}^{k}(V_i,E_i) = (V,E)$ where $V = V_1 \times \cdots \times V_k$, and $\{(u_1,\ldots,u_k),(v_1,\ldots,v_k)\}$ is an edge if and only if $u_iv_i \in E_i$ for each $1 \leq i \leq k$. P_n is the reflexive *path graph* of length n, where the vertices are $\{0,\ldots,n\}$ and the edges are $\{\{i,i+1\} \mid 0 \leq i \leq n-1\} \cup \{\{i\} \mid 0 \leq i \leq n\}$. The *variety of finite paths*, denoted $\mathcal{FP}$, is the set of retracts of product of finite paths. That is, $G \in \mathcal{FP}$ if and only if it is a retract of $\prod_{i=1}^{k} P_{a_i}$ for some sequence of non-negative integers $a_1,\ldots,a_k$.

Theorem 2.3 ([32]) *Let $H \in \mathcal{FP}$. There is a retract from G to H if and only if H is an isometric subgraph of G.*

2.2 Minimum CSP

Our main result on r-tractability requires tools from VCSP theory. In particular, we describe a simpler special case of VCSP, the *minimum constraint satisfaction problem* (MINCSP). We formally define MINCSP using the notations in [15]. Note [15] actually defines MAXCSP, but it is equivalent to MINCSP.

Let the *domain* D be a finite set. The set of all m-ary $\{0,1\}$-valued functions over the domain D is $R_D^{(m)}$. That is, $f \in R_D^{(m)}$ if and only if $f : D^m \to \{0,1\}$. The set of all $\{0,1\}$-valued functions over the product space of D is $R_D = \bigcup_{m=1}^{\infty} R_D^{(m)}$. For $f \in R_D^{(m)}$, $w \in \mathbb{N}$ and $\pi : [m] \to [n]$, a 3-tuple (f, w, π) is a D^n-*constraint*. A finite set $\Gamma \subseteq R_D$ is a *constraint language*. We define an instance of the problem MINCSP(Γ) as follows. The input consists of an integer n and a sequence of D^n-constraints $(f_1, w_1, \pi_1), \ldots, (f_k, w_k, \pi_k)$, where either each $f_i \in \Gamma$ or $f_i \in R_D^{(0)}$. The case of $f_i \in R_D^{(0)}$ allows f_i to be a constant. Assume that the arity of f_i is m_i. The output of the MINCSP(Γ) instance is the value of

$$\min_{(x_1,\ldots,x_n)\in D^n} \sum_{i=1}^{k} w_i f_i(x_{\pi_i(1)}, \ldots, x_{\pi_i(m_i)}).$$

For $D' \subseteq D$ and $f : D^m \to \{0,1\}$, we define $f[D']$ to be $f|_{D'^m}$. For Γ a constraint language, we define $\Gamma[D']$ to be $\left\{f[D'] \mid f \in \Gamma\right\}$, the *constraint language induced on* D'. A constraint language Γ is *tractable* if every instance of MINCSP(Γ) can be solved in polynomial time. The set Γ_c consists of functions obtained from Γ by fixing a subset of variables. For example, if $f \in \Gamma$ and it is a binary function, then functions of the form $g_b(a) = f(a,b)$ and $h_a(b) = f(a,b)$ are in Γ_c. In fact, if Γ consists of a binary function h and some unary functions, then Γ_c also consists of the same binary function h and some (possibly a larger set of) unary functions.

Let $H = (U, F)$ be a digraph, we define $\Gamma_H = \{h_H\}_c$. Observe that

$$\Gamma_H = \{h_H\} \cup \left\{\bar{\chi}_{N_H^+(u)} \mid u \in U\right\} \cup \left\{\bar{\chi}_{N_H^-(u)} \mid u \in U\right\}.$$

Indeed, for an arbitrary $u \in U$, define the function $g_u(v) = h_H(u,v)$. We have $g_u(v) = 0$ if and only if $v \in N_H^+(u)$. Therefore $g_u = \bar{\chi}_{N_H^+(u)}$. The case of $\bar{\chi}_{N_H^-(u)}$ can be handled similarly.

Let Γ be a constraint language on D. Then $\phi : D \to D$ is an *endomorphism* of Γ if for every $f \in \Gamma$ and $f(x) = 0$ implies $f(\phi(x)) = 0$. An injective endomorphism is an *automorphism*. Γ is a *core* if every endomorphism is an automorphism. We say a core Γ' is a *core of* Γ, if there exists an endomorphism ϕ of Γ, such that $\Gamma' = \Gamma[\phi(D)]$.

We state a sequence of theorems.

Theorem 2.4 ([15]) *If constraint language Γ' is a core of constraint language Γ, then Γ' is tractable if and only if Γ is tractable.*

Theorem 2.5 ([34]) *Let the constraint language Γ on D be a core. Γ is tractable if and only if $\Gamma_c \cup \left\{\bar{\chi}_{\{a\}} \mid a \in D\right\}$ is tractable.*

Theorem 2.6 ([15]) *Γ is a constraint language on D. $\Gamma \cup \left\{\bar{\chi}_{\{a\}} \mid a \in D\right\}$ is tractable if and only if $\Gamma \cup \{\bar{\chi}_U \mid U \subseteq D\}$ is tractable.*

The problems in CVIO can be modeled by MINCSP.

Theorem 2.7 ([15]) *A digraph H is c-tractable if and only if $\{h_H\} \cup \{\bar{\chi}_U \mid U \subseteq V(H)\}$ is tractable.*

Theorem 2.8 *Let H be a digraph on vertices D and let Γ be a constraint language on the domain D consisting of h_H and some unary functions. If $\Gamma[D']$ is a core of Γ, then Γ is tractable if and only if $H[D']$ is c-tractable.*

Proof: $\Gamma[D']$ is a core of Γ and therefore by Theorem 2.4 and Theorem 2.5, Γ is tractable if and only if $\Gamma_c[D'] \cup \left\{\bar{\chi}_{\{a\}} \mid a \in D'\right\}$ is tractable. By Theorem 2.6, $\Gamma_c[D'] \cup \left\{\bar{\chi}_{\{a\}} \mid a \in D'\right\}$ is tractable if and only if $\Gamma_c[D'] \cup \left\{\bar{\chi}_U \mid U \subseteq D'\right\}$ is tractable. Note that $\Gamma_c[D'] \cup \left\{\bar{\chi}_U \mid U \subseteq D'\right\} = \left\{h_H[D']\right\} \cup \left\{\bar{\chi}_U \mid U \subseteq D'\right\}$. Hence by Theorem 2.7, Γ is tractable if and only if $H[D']$ is c-tractable. □

2.3 c-tractable digraphs

A matrix M is *anti-monge* if $M_{i,j'} + M_{i',j} \leq M_{i,j} + M_{i',j'}$ for all $i < i'$ and $j < j'$. The matrix M is *permuted anti-monge*, if there exists some permutation π, such that π applied to both the rows and columns, we obtain an anti-monge matrix.

Theorem 2.9 (Classification of c-tractable digraphs [14, 15]) *H is c-tractable if and only if the adjacency matrix of H is a permuted anti-monge matrix. Moreover, c-tractability can be decided in $O(n^2)$ time, where n is the number of vertices in H.*

A set is called a *rectangle* if it is $A \times B$ for two sets A and B such that $A \subseteq B$ or $B \subseteq A$. For a $n \times n$ matrix, an *L-anchored rectangle* is $[a] \times [b]$ for some $a, b \in [n]$. An *R-anchored rectangle* is $([n] \setminus [a]) \times ([n] \setminus [b])$ for some $a, b \in [n]$.

Lemma 2.10 (Lemma 4.4 [15]) *An $n \times n$ $\{0,1\}$ matrix M without all-ones rows and columns is anti-monge if and only if the indices of the 1 entries is a union of an L-anchored and an R-anchored rectangles that are disjoint.*

Lemma 2.11 (Lemma 3.3 [36]) *Let M and N be two $\{0,1\}$-matrices such that all entries are the same except one single row (column), such that the row (column) is all 1 in M and all 0 in N. N is anti-monge if and only if M is anti-monge.*

The reflexive c-tractable digraphs have an easy combinatorial description.

Theorem 2.12 (combinatorial classification of c-tractable reflexive digraphs) *Consider a reflexive digraph $H = (V, E)$ with $S \subseteq V$ the set of vertices with outgoing edges to all vertices, and $T \subseteq V$ the set of vertices with incoming edges from all vertices.*

H is c-tractable if and only if there are 4 sets A, A', B, B' such that E is the union of disjoint sets $A \times B$, $A' \times B'$ and E', where

1. *if T is non-empty, then $E' = V \times T$ and $T = V \setminus (B \cup B')$,*
2. *otherwise, $E' = S \times V$ and $S = V \setminus (A \cup A')$.*

Proof: First we prove one direction. Consider a reflexive digraph $H = (V, E)$ where there are 4 sets A, A', B and B' with the desired property, then H is c-tractable.

Let M be the adjacency matrix of H. There is a permutation of M, M', such that $A \times B$ is an L-anchored rectangle, $A' \times B'$ is an R-anchored rectangle. E' is either the all-ones rows or all-ones columns. By Lemma 2.11 and Lemma 2.10, we know M' is anti-monge, and M is permuted anti-monge. Therefore H is c-tractable.

Now, we prove the other direction. Let H be a reflexive c-tractable digraph and M be the adjacency matrix of H. Every column has at least one 1. There are two cases, either there is an all-ones column or there is no all-ones column. We consider the case of having an all-ones column.

We change each all-ones column into an all-zeros column. The new matrix is M', and has no all-ones column or rows. By Lemma 2.11, M' has the permuted anti-monge property. M' is permuted anti-monge if and only if we can permute the matrix into the form in Lemma 2.10. Let N' be such a permuted matrix. We reverse the change of the all-ones columns in N', which must consist of all columns indexed by $T = V \setminus (B \cup B')$ due to the fact every column has at least one 1. Let the new matrix be N. N consists of 3 different parts, an L-anchored rectangle of the form $A \times B$, an R-anchored rectangle of the form $A' \times B'$, and a rectangle $E' = V \times T$.

Otherwise, if there is no all-ones column. We consider the above operation again but using all-ones rows (if it exists) instead. The only difference is the last rectangle E' in N is $(V \setminus (A \cup A')) \times V$. □

A graph $G = (V, E)$ is a *double clique*, if there exists $A, B \subseteq V$ such that $E = (A \times A) \cup (B \times B)$. Next we use the previous theorem on *digraphs* to obtain a much simplified characterization for c-tractable reflexive *graphs*.

Theorem 2.13 (combinatorial classification of c-tractable reflexive graphs) *A reflexive graph H is c-tractable if and only if it is a reflexive double clique.*

Proof: One direction is trivial as adjacency matrix of reflexive double cliques consists of an L-anchored rectangle, an R-anchored rectangle and some all-ones columns, so it is c-tractable.

The other direction is straightforward but tedious. Let $H = (V, E)$ be a c-tractable reflexive digraph. S is the set of vertices with outgoing edges to all vertices and T is the set of vertices with incoming edges from all vertices. If $E = V \times V$ then we are done, the two cliques are V and V. Now we consider when $E \neq V \times V$. By Theorem 2.12, when H is c-tractable, there are 4 sets A, A', B, B' such that $E = (A \times B) \cup (A' \times B') \cup E'$. Because $S = T$, there are only two cases. If T is empty, then we have $A = B$ and $A' = B'$ to be the only solution. The desired cliques are A and A'. If T is non-empty, we will show that $A \cap A'$ are the index of all-ones rows. Assume $a \in A \cap A'$. Consider a $b \in V$, if $b \in B$, then $(a, b) \in A \times B$. If $b \in B'$, then $(a, b) \in A' \times B'$. Otherwise, $(a, b) \in V \times T$. On the other hand, if $a \notin A \cap A'$, then there is a $b \in B \cup B'$ where $(a, b) \notin E$. So $A \cap A' = S = T$.

If $b \in B$ and $(a, b) \in E$, then $a \in A$. Since $(b, b) \in E$, we also know $b \in A$. This shows $B \subseteq A$. Similarly, $B' \subseteq A'$.

We claim $E \subseteq (A \times A) \cup (A' \times A')$. Note that $E = (A \times B) \cup (A' \times B') \cup (V \times T)$.

- $A \times B \subseteq A \times A$,
- $A' \times B' \subseteq A' \times A'$,
- and

$$\begin{aligned}
&V \times T \\
=&V \times (A \cap A') \\
=&(A \cup A') \times (A \cap A') \\
=&(A \times (A \cap A')) \cup (A' \times (A \cap A')) \\
\subseteq&(A \times A) \cup (A' \times A').
\end{aligned}$$

We show $(A \times A) \cup (A' \times A') \subseteq E$. Consider $(a, b) \in A \times A$.

1. $b \in B$, then $(a, b) \in A \times B$,
2. $b \in T$, then $(a, b) \in V \times T$,
3. $b \in B'$, then $b \in A \cap A' = S$, so $(a, b) \in S \times V$.

One can show the above for $(a, b) \in A' \times A'$. This shows $(A \times A) \cup (A' \times A') \subseteq E$. Therefore A and A' are the desired cliques. □

3 Classification of r-tractable digraphs

In this section, we characterize the r-tractable digraphs by giving a simple condition that can be checked in polynomial time. The idea is to reduce r-tractability to c-tractability. One can observe RVIO(H) and MINCSP(Γ_H) are equivalent problems. The proof is just a direct translation between terminologies.

Theorem 3.1 *For a digraph H,* RVIO(H) *and* MINCSP(Γ_H) *are equivalent problems.*

Proof: Let $H = (U, F)$. Let $G = (U \cup V, E)$ be the input to RVIO(H), here $U \cap V = \emptyset$. Also, we assume $V = [n]$. That is, the non-fixed vertices in G are integers from 1 to n. Assume G is simple, and there is a weight function w on the edges. Let f be a minimum violation map from G to H.

We construct an instance of MINCSP(Γ_H). Let π_{uv} be the function $\pi_{uv}(1) = u$ and $\pi_{uv}(2) = v$, which simulates the edge uv. We define π_v to be the function $\pi_v(1) = v$, which can be used to simulate both edges of the form uv and vu, where $u \in U$. π is a function of 0 arity.

1. For each edge $uv \in E$ where $u, v \in V$, we create a tuple $(h_H, w(uv), \pi_{uv})$.
2. For each edge $uv \in E$ where $u \in U$ and $v \in V$, we create a tuple $(\bar{\chi}_{N^+_H(u)}, w(uv), \pi_v)$.
3. For each edge $vu \in E$ where $u \in U$ and $v \in V$, we create a tuple $(\bar{\chi}_{N^-_H(u)}, w(vu), \pi_v)$.
4. For each edge $uv \in E$ where $u, v \in U$ and $uv \notin F$ we create a tuple $(1, w(uv), \pi)$.

Let the instance consist of all the tuples above and n. It is not hard to see the optimal value of the instance is equal to the minimum violation of the optimal f.

The above construction can be reversed to build input digraph G and its weight. Indeed, for example, we can replace (h_H, w, π_{uv}) with an edge with weight w from u to v in digraph G. □

To study the tractability of Γ_H, we need to introduce the notion of apex vertices. A vertex v is an *apex* if no vertex dominates v. The set of apex vertices is the *apex set*. Consider a digraph $G = (V, E)$ with apex set A. $G[A]$ is the *apex subgraph*. The *apex map* α is the following vertex map.

$$\alpha(v) = \begin{cases} v & \text{if } v \in A \\ u & \text{if } u \in A \text{ and is the smallest vertex that dominates } v \end{cases}$$

Elem *et al.* observed that, in our terminology, an *undirected graph* with a reflexive complete graph as its apex subgraph is r-tractable [19]. Reflexive complete graphs are c-tractable. It is natural to conjecture that if the apex subgraph of a digraph is c-tractable, then the digraph itself is r-tractable. We will show that it is both necessary and sufficient.

Recall, that, by Theorem 3.1, we are interested in the tractability of Γ_H. To this end, we find the core of Γ_H. The next lemma shows that the core of Γ_H is induced on the the apex set of H.

Lemma 3.2 *Let $H = (V, E)$ be a graph and A be the apex set of H. Then $\Gamma_H[A]$ is the core of Γ_H.*

Proof: Let α be the apex map of H. Clearly, $A = \alpha(V)$.

First, we show that α is an endomorphism of Γ_H. v is dominated by $\alpha(v)$ and $v \in N^-_H(u)$ implies $\alpha(v) \in N^-_H(u)$. Therefore α is an endomorphism for $\bar{\chi}_{N^-_H(u)}$ for all u. Similarly it is an endomorphism for all $\bar{\chi}_{N^+_H(u)}$ for all u. For each edge uv in E, $v \in N^-_H(u)$ implies $\alpha(v) \in N^-_H(u)$. Therefore $u\alpha(v) \in E$. Apply the same argument for u, we get $\alpha(u)\alpha(v) \in E$. α is therefore an endomorphism for h_H.

Second, we show $\Gamma_H[A]$ is a core. Consider an endomorphism $g : A \to A$ of $\Gamma_H[A]$. Consider an arbitrary $u \in A$. For each $v \in V$, if $u \in N^-_H(v)$, then $g(u) \in N^-_H(v) \cap A$. Indeed, this is because $g(u) \in \bar{\chi}_{N^-_H(v)}[A]$. Similarly, if $u \in N^+_H(v)$, then $g(u) \in N^+_H(v) \cap A$. Let $B_u = \bigcap \{B | B \in \mathcal{N}_H, u \in B\} \cap A$. Note that by definition $g(u) \in B_u$. But B_u consists of precisely the single element u, since B_u consists of elements that have neighborhoods containing the neighborhoods of u. Thus we have $g(u) = u$ for all $u \in A$, therefore g is an automorphism of $\Gamma_H[A]$. □

Theorem 3.3 (classification of r-tractable digraphs) *A digraph H is r-tractable if and only if its apex subgraph is c-tractable. Moreover, r-tractability can be decided in polynomial time.*

Proof: Let A be the apex vertices of H. By Lemma 3.2, $\Gamma_H[A]$ is the core of Γ_H. By Theorem 2.8, we have H is r-tractable if and only if $H[A]$ is c-tractable. Finding A is equivalent to finding the vertices with maximal neighborhood. For a graph on k vertices and ℓ edges, a $O(k\ell)$ time algorithm exists for finding the set of maximal neighborhoods [51]. Since A can be found in $O(nm)$ time, and c-tractability can be decided in $O(n^2)$ time [14, 15]. We can decide if H is r-tractable in $O(nm)$ time. □

Remark There is no forbidden graph characterization for r-tractable digraphs. One can modify any graph by adding a single vertex that incident to all other vertices. The new graph is r-tractable because the apex subgraph is a single vertex.

As an application of the theorem, we show the following two results on r-tractability which we also use later. In particular, the one on $B_{k,\ell}$ can be used for hardness of (ℓ, k)WAYCUT.

Corollary 3.4 *S_k is r-tractable if and only if $k \leq 2$.*

Proof: The apex subgraph of S_k is itself. By applying Theorem 3.3, we just have to show the c-tractability of S_k. All permutations of the adjacency matrix of S_k are essentially the same: one column and the diagonal consists of all 1, all other entries are 0. By Lemma 2.11, we can replace the entries of the 1 column to all 0s. It's easy to see that except for S_1 and S_2, the 1 entries cannot equal to a union of an L-anchored and an R-anchored rectangle. We obtain the corollary by Lemma 2.10. □

Corollary 3.5 *$B_{k,\ell}$ is r-tractable if and only if $k = 2$ and $\ell \leq 4$, or $k \geq 3$ and $\ell \leq 3$.*

Proof: By Theorem 3.3, we have to find for which k and ℓ the apex subgraph of $B_{k,\ell}$ is c-tractable. The apex subgraph of $B_{k,\ell}$ is $B_{k,\ell-2}$. By Theorem 2.13, we can see that $B_{2,2}$ is a path of length 2, a double clique. $B_{k,1}$ is is a clique of size k. No other $B_{k,\ell}$ is a double clique. □

4 s-tractability of disconnected reflexive digraphs

In this section, we show that the s-tractability of a reflexive digraph is determined by the s-tractability of its components. There were no general techniques to show a digraph H is s-tractable beyond showing H is r-tractable. But s-tractable graphs are much richer. The first result in this area is the solution to the k-cut problem, which demonstrates a reflexive graph that consists of k isolated vertices is s-tractable. Here we extend the result. If each component of a reflexive digraph is s-tractable, then the digraph is s-tractable. The converse is also true. We introduce the main theorem established in this section.

Theorem 4.1 *A reflexive digraph H is s-tractable if and only if every component of H is s-tractable.*

We demonstrate a polynomial time algorithm for SVIO(H) knowing that each component of H is s-tractable. We use the fact that the minimum violation can be bounded by the value of a min-k-cut, and partitions with value no more than min-k-cut can be found effectively.

Recall a k-cut is a set of edges such that after its removal, the graph has at least k components. Let $\lambda_k(G)$ denote the weight of the minimum k-cut in G.

Theorem 4.2 ([11]) *Given a graph G of n nodes and m edges. There are $O(n^{2(k-1)})$ partitions with value no larger than $\lambda_k(G)$. Such partitions, together with their values, can be computed in $\tilde{O}(mn^{2(k-1)})$ time.*

Gupta, Lee and Li showed the number of partitions with value no larger than $\lambda_k(G)$ is $n^{(1.981+o(1))k}$, and subsequently improved it to $n^k 2^{O(\log\log n)^2}$ [26, 27]. Recently with the addition of Harris, the optimum $O(n^k)$ bound was established [25]. The partitions can also be found in the same time using a randomized algorithm. However, the deterministic algorithm in the above theorem is not able to obtain the same result.

Theorem 4.3 *If each component of a reflexive digraph H is s-tractable then H is s-tractable.*

Proof: Let $H = (U, F)$ be a k vertex digraph consisting of components $U_1, \ldots, U_t$. For each i, let $H_i = H[U_i]$. Assume SVIO(H_i) can be solved in $T_i(n)$ time for graphs of size n for some polynomial T_i. Let $G = (V, E)$ be a digraph. Consider a minimum violation surjective vertex map $f : V \to U$ from G to H. Define $V_i = f^{-1}(U_i)$. The edges crossing the t-partition $P = (V_1, \ldots, V_t)$ are violated edges of f. Let G' be the graph where we undirect each edge in G. The total weight of the edges crossing P in G is at most the weight of the min k-cut of G'. In particular, by Theorem 4.2, a collection $\mathcal{X}$ of partitions where P is one of them can be found in polynomial time.

For $(V_1, \ldots, V_t) \in \mathcal{X}$, we solve SVIO($H_i$) on $G[V_i]$ to obtain f_i in $O(T_i(n))$ time. Gluing together f_i in the natural way gives us a candidate solution. We return the candidate with the minimum violation. The total running time is therefore $O(mn^{2(k-1)} + n^{2(k-1)} \sum_{i=1}^{k} T_i(n))$, which is a polynomial. □

As a consequence, we obtain a faster deterministic algorithm for sSIZEkCUT.

Theorem 4.4 *Let $s = (s_1, \ldots, s_k)$ and $s_i \geq s_{i+1}$ for all $i \leq k-1$. sSIZEkCUT for graph G can be solved in $O(mn^{2(\sigma - s_1)})$ time, where $\sigma = \sum_{i=1}^{k} s_i$ and n is the number of vertices of G.*

Proof: Let $\sigma' = \sigma - s_1$. Assume that n is at least $s_1(\sigma'+1)$, otherwise brute force takes $O(1)$ time because both σ' and s_1 are constants. We show that the value of the min s-size k-cut is at most the value of a min-$(\sigma'+1)$-cut when $n \geq s_1(\sigma'+1)$. Consider any $(\sigma'+1)$-cut with vertex partition $V_1, \ldots, V_{\sigma'+1}$, each has size $n_1, \ldots, n_{\sigma'+1}$ respectively. Assume that $n_i \geq n_{i+1}$ for all i. We will find a k-partition $(U_1, \ldots, U_k)$ that induces a s-size constrained k-cut. By the pigeonhole principle, $|V_1| \geq s_1$. Let $U_1 = V_1$. We consider an arbitrary partition of the remaining σ' sets into $k-1$ partition classes, such that the ith partition class contain s_{i+1} of the sets. This is feasible since $\sigma' = \sum_{i=2}^{k} s_i$. Let U_{i+1} be the union of the sets in the ith partition class. The resulting $\{U_1, \ldots, U_k\}$ induces a s-size k-cut that only uses edges in the minimum $(\sigma'+1)$-cut. Hence we have shown that the minimum s-size k-cut is bounded above by minimum $(\sigma'+1)$-cut. Therefore, by Theorem 4.2, the optimal partition is one of the $O(n^{2(\sigma - s_1)})$ k-partitions that can be computed in $\tilde{O}(mn^{2(\sigma - s_1)})$ time. It takes $O(k) = O(1)$ time to verify if the k-partition is a s-size k-partition. Hence the total running time is $\tilde{O}(mn^{2(\sigma - s_1)})$. □

One can show the converse of Theorem 4.3.

Theorem 4.5 *If any component of a reflexive digraph H is not s-tractable, then H is not s-tractable.*

Proof: Consider $H = (U, F)$ consisting of k components. The components are subgraphs $H_1, \ldots, H_k$. The vertices in H_i is U_i.

We want to reduce SVIO(H_1) to SVIO(H) in polynomial time. Let the input of SVIO(H_1) be $G_1 = (V_1, E_1)$. Let M be a value larger than the sum of weights of edges in G_1. We create digraph G that is the union of G_1 and $G_2, \ldots, G_k$. Here for $i \geq 2$, G_i and H_i are exactly the same digraph except each edge in G_i have weight M. We will use V_i to denote the vertices in G_i, although it is clear that $V_i = U_i$. If there is a surjective map f_1 of violation t from G_1 to H_1, then there is a surjective map f of violation t from G to H. Indeed, we extend the map f_1 to f, such that $f|_{V \setminus V_1}$ is the identity function. We want to show this is true in the other direction. That is, if there is a *minimum violation* surjective map f of violation t from G to H, then there is a surjective map f' of violation at most t from G_1 to H_1.

For a surjective map f from G to H, an index i is *good* if $f(V_i) = U_i$ and $f^{-1}(U_i) = V_i$, otherwise it is bad. If f^* is a minimum violation surjective map from G to H with no bad indices, then $f^*|_{V_1}$ is the minimum violation map from G_1 to H_1. We will show that there always exists an optimal surjective map with no bad indices. We show that if f has at least one bad index, then there exists an optimal solution with strictly fewer bad indices.

First, the number of bad indices cannot be 1. Let i be a bad index. If $f(V_i)\setminus U_i$ is non-empty, then $f(V_i)\cap U_j$ is non-empty for some $j\neq i$, and therefore j is a bad index. If $U_i\setminus f(V_i)$ is non-empty, then $f^{-1}(U_i)\neq V_i$. There is some $j\neq i$ such that there is a vertex $v\in V_j$ and $f(v)\in U_i$. Hence j is a bad index. In either case, there are at least two bad indices.

Any optimal solution to SVIO(H) has violation smaller than M. Indeed, a map that acts as an identity map G_i to H_i for all $i\geq 2$ has violation smaller than M. Let f be a minimum violation map from G to H. Let B be the set of bad indices of f. If B is empty, then we are done. Let $i\in B$ and $i\neq 1$. Such i must exist, since $|B|\geq 2$. No edge in G_i can be a violated edge, since it would incur violation at least M. Therefore, $f(V_i)$ is contained in one of $U_1,\ldots,U_k$.

Case 1: $f(V_i)\subseteq U_i$. Pick $j\in B$ such that $j\neq i$. Let u be any vertex in U_j.

$$f'(v)=\begin{cases} v & \text{if } v\in V_i\\ u & \text{if } v\in f^{-1}(U_i)\setminus V_i\\ f(v) & \text{otherwise}\end{cases}$$

We show that any non-violated edge with respect to f is also an non-violated edge with respect to f'. An edge not in G_i maps to an edge not in H_i under f, and still maps to the same edge under f'. If an edge not in G_i maps to an edge H_i by f, then it maps to a self-loop of u under f'. An edge in G_i maps to an edge in H_i under f'. The violation of f' is no larger than the violation of f. Also, i is not a bad index for f'. No new bad index was generated in the process. The number of bad indices for f' is strictly smaller since i is no longer in B.

Case 2: $f(V_i)\cap U_i=\emptyset$.

$$f'(v)=\begin{cases} v & \text{if } v\in V_i\\ f(f(v)) & \text{if } v\in f^{-1}(U_i)\\ f(v) & \text{otherwise}\end{cases}$$

A similar argument as case 1 holds. The number of bad indices for f' is strictly smaller than the number of bad indices for f. We have to argue that the non-violated edges not in G_i that gets mapped to H_i are still non-violated. Indeed, if uv is an edge, then $f(u)f(v)$ is an edge for all $u,v\in V_i$. Now if $f(a)f(b)$ is an edge in H_i then $a,b\in f^{-1}(U_i)$. If additionally ab is an edge in G, then $f(f(a))f(f(b))$ is an edge.

We showed we can find an optimal mapping with strictly smaller number of bad indices. Hence there exists an optimal mapping with no bad indices. □

The exact same proof also shows that if any component of H is s-intractable, then H is s-intractable.

5 s-tractability for variety of finite paths and dichotomy of trees

In this section we show that if $H\in\mathcal{FP}$, then SVIO(H) is intractable if $\operatorname{diam}(H)\geq 5$, or at least 3 vertices has pairwise distance 4. Every reflexive tree is a variety of finite paths [32], the result is used to show a s-tractability dichotomy theorem for reflexive trees. The reduction is not a direct one. It relies on the intermediate results on (ℓ,k)WAYCUT and existence of c-connectors, which we will describe first.

5.1 c-connector

We first introduce c-connectors, which is a useful tool to force some vertices to have a certain distance.

Definition Let $c: U^2\to\{\ell,\ell+1\}$ and $\ell\geq 3$. A graph $G=(V,E)$ is a *c-connector* if $U\subseteq V$, and the following distance properties hold.

1. $d_G(x,y)=c(x,y)$ for all distinct $x,y\in U$,
2. $d_G(x,y)\leq\ell$ for all $x,y\in V$ such that $x\in V\setminus U$

Theorem 5.1 *For any $c : U^2 \to \{\ell, \ell+1\}$ such that $c(x,y) = c(y,x)$ and $\ell \geq 3$. There exists a c-connector of $O(\ell|U|^2)$ vertices and edges.*

Proof: Let $k = \lfloor \ell/2 \rfloor - 1$. Note that $k+3 \leq \ell$ for $\ell \geq 3$. We build the *c-connector base graph* $G = (V,E)$ using the following construction. For each vertex $u \in U$, we create a path with a sequence of vertices $u_0, \ldots, u_k$, where $u_0 = u$ and all other vertices are new. These vertices are $U' = \{u_i | u \in U, 0 \leq i \leq k\}$. For each $u \in U$, create a star centered at u_k, with new vertices u_v for each $v \in U$ and $v \neq u$. Finally, we add a vertex p where p is adjacent to each vertex u_v for distinct $u, v \in U$.

See Figure 5.1a for an example of the c-connector base.

Observe we already have some nice properties for the c-connector base G.

- $d_G(p,v) \leq k+2 \leq \ell$ for all $v \in V$.
- $d_G(u_v, w_x) = 2 \leq \ell$ for $u,v,w,x \in U$.
- $d_G(u_i, u_j) \leq j - i \leq k \leq \ell$ for $u \in U$ and $0 \leq i,j \leq k$.
- $d_G(u, v_w) \leq k+3 \leq \ell$ for $u,v,w \in U$.

We will construct a c-connector H by adding edges to G. If ℓ is odd, then H is obtained from G by adding edges $u_v v_u$ if $c(u,v) = \ell$. If ℓ is even, then H is obtained from G by adding edges $u_v v_k$ where $c(u,v) = \ell$, and adding edges $u_v v_u$ if $c(u,v) = \ell + 1$.

This establishes $d_H \leq d_G$. Each newly added edge incidents to some vertex u_k where $u \in U$, or u_v where $u,v \in U$. Therefore, every path in H containing $u \in U$ either contains u_k, or it consists of only $u_0, \ldots, u_i$ for some $i \leq k$. In particular, this shows $d_H(a, u_i) \leq d_H(a,u)$ for $a \in V \setminus U'$ and $u \in U$. Together with the distance facts of G, this shows that $d_H(u,v) \leq \ell$ for all $u \in V$ and $v \in V \setminus U$. Therefore, in order to show H is a c-connector, we just have to show $d_H(u,v) = c(u,v)$ for $u,v \in U$.

Now, we argue depending on the parity of ℓ by consider distinct $u,v \in U$.

$\ell \geq 3$ and is odd. Any path from u to v must contain $u_0, u_1, \ldots, u_k, u_x$ and $v_y, v_k, \ldots, v_0$ for some x and y, which shows $d_H(u,v) \geq \ell$. If $c(u,v) = \ell+1$, then $d_H(u,v) = \ell + 1 = c(u,v)$ since there does not exist x and y such that $u_x v_y$ is an edge. If $c(u,v) = \ell$, then $d_H(u,v) = \ell = c(u,v)$ since there is a path u_k, u_v, v_u, v_k.

$\ell \geq 4$ and is even. Any path from u to v must contain $u_0, u_1, \ldots, u_k$ and $v_k, \ldots, v_0$. If $c(u,v) = \ell$, then there is a path from v_k and u_k using u_k, u_v, v_k. This is the shortest u_k-v_k path, since there is no edge $u_k v_k$. This shows $d_H(u,v) = c(u,v)$. If $c(u,v) = \ell+1$, then there is a path from v_k and u_k using u_k, u_v, v_u, v_k. There is no shorter path, since there is no edge $u_k v_k$, and all neighbors of u_k is not a neighbor of v_k. Therefore we have $d_H(u,v) = c(u,v)$.

It is clear the number of vertices and edges in H is $O(\ell |U|^2)$. □

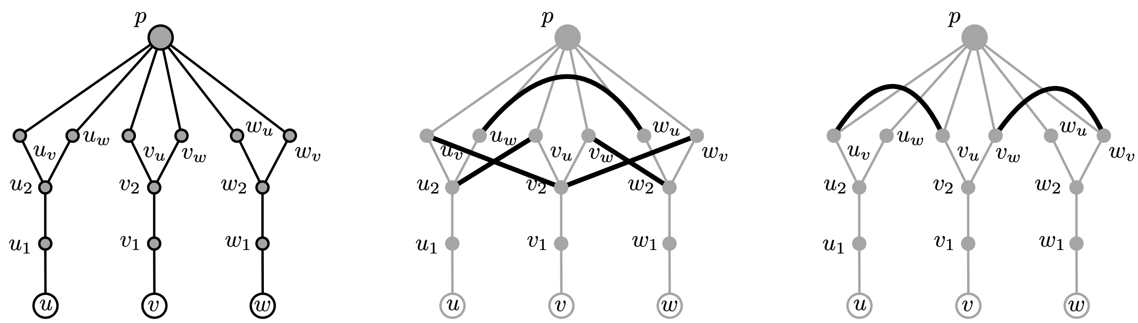
5.2 (ℓ,k)WAYCUT and $B_{k,\ell}$

In this section, we show that (ℓ,k)WAYCUT and (ℓ,k)CUT are equivalent problems. Then, we show RVIO$(B_{k,\ell})$ and $(\ell-1,k)$WAYCUT are equivalent.

This is the first time we demonstrate the use of c-connector.

Theorem 5.2 *(ℓ,k)WAYCUT and (ℓ,k)CUT are equivalent for $\ell \geq 3$.*

Proof: In the easy direction, (ℓ,k)CUT reduces to $\binom{n}{k}$ instances of (ℓ,k)WAYCUT. We consider the harder direction. Let $G = (V,E)$ be the graph and T the set of terminals. Consider a c-connector H with $c : V^2 \to \mathbb{N}$ such that $c(u,v) = \ell+1$ if $u,v \in T$ and $c(u,v) = \ell$ otherwise. Let $G' = G \cup H$, where the



(a) The connector base for $k = 2$. (b) The case where $\ell = 6$. (c) The case where $\ell = 7$.

Figure 5.1: The c-connector for the specific example of 3 nodes u, v, w and $c(u, v) = c(v, w) = \ell$ and $c(u, w) = \ell + 1$.

edges in H has infinite capacity. We show that every finite value (ℓ, k)-cut in G' is a (ℓ, k)-way-cut in G, and every (ℓ, k)-way-cut in G is a (ℓ, k)-cut in G'.

A finite value (ℓ, k)-cut C' in G' only use edges in E. Since $d_{G'-E}(u, v) = \ell$ unless $u, v \in T$, we have $\ell + 1 = d_{G'-C'}(u, v) \leq d_{G-C'}(u, v)$ for all $u, v \in T$, and it shows C' is a (ℓ, k)-way-cut in G.

Let C be a (ℓ, k)-way-cut in G. Consider $d_{G'-C}(u, v)$ for distinct $u, v \in T$. Consider a shortest uv-path P in $G' - C$. There are two cases. If P contains no edge in H, then $d_{G'-C}(u, v) = d_{G-C}(u, v) = \ell + 1$. If P contains an edge in H, then it has to contain a path P' between two vertices in V using only edges in H. Note length of P' is at least ℓ, and if P' has length ℓ, then it cannot be a uv-path since $d_H(u, v) = \ell + 1$. Hence we have $d_{G'-C}(u, v) > \ell$. Together, we have $d_{G'-C}(u, v) \geq \ell + 1$ for all distinct $u, v \in T$, therefore C is a (ℓ, k)-cut in G'.

There always exists a finite value (ℓ, k)-cut (i.e. E). This shows (ℓ, k)WAYCUT reduces to (ℓ, k)CUT. □

Note that $B_{k,\ell} \in \mathcal{FP}$ for all k, ℓ. Indeed, when ℓ is even, $B_{k,\ell}$ is a tree, and trees are in $\mathcal{FP}$ [32]. When ℓ is odd, then it takes a bit more to see. Let $\ell' = \lceil \ell/2 \rceil$. Consider $H = \prod_{i=1}^{k} P_{\ell'}$. $B_{k,\ell}$ is an isometric subgraph of H. In particular, let $w_{i,j}$ be the tuple in $\{0, 1, \ldots, \ell'\}^k$ that is 0 at all coordinates except the ith coordinate has value j. $B_{k,\ell}$ is isomorphic and isometric to the induced subgraph of H on vertices $\{w_{i,j} | 1 \leq i \leq k, 1 \leq j \leq \ell'\}$.

We name the leaves of $B_{k,\ell}$ as $t_1, \ldots, t_k$ in order to refer to them easier.

Theorem 5.3 RVIO$(B_{k,\ell})$ *reduces to* $(\ell - 1, k)$WAYCUT.

Proof: Let $G = (V, E)$ be the input graph to RVIO$(B_{k,\ell})$. Let M be a value strictly larger than the total weight of the edges of G. We take $G' = G \cup B_{k,\ell}$, where the edges in $B_{k,\ell}$ has weight adjusted to M. Let the terminals in G' be t_i for all $i \in [k]$.

Let α' be the value of the minimum $(\ell - 1, k)$-way-cut in G'. Let α be the optimal violation for RVIO$(B_{k,\ell})$ with input G. We show that $\alpha' = \alpha$.

Consider a minimum $(\ell - 1, k)$-way-cut C in G'. It does not contain edges in $B_{k,\ell}$. $d_{G'-C}(t_i, t_j) = \ell$ for all $i \neq j$. The unique path in $B_{k,\ell}$ is a shortest $t_i t_j$-path, and $B_{k,\ell}$ is a subgraph of $G' - C$ because it is a subgraph of $G' - E$. This shows $B_{k,\ell}$ is an isometric subgraph of $G' - C$. Hence there is a retraction from $G' - C$ to $B_{k,\ell}$ by Theorem 2.3. The minimum violation map has violation at most weight of C. Therefore $\alpha' \geq \alpha$.

Consider a minimum solution f of RVIO$(B_{k,\ell})$ with input G. Consider when all violated edges are removed, the distance between t_i and t_j is at least ℓ due to Theorem 2.1. The violated edges of f form an $(\ell - 1)$-length-bounded k-way-cut in G'. Hence $\alpha \geq \alpha'$. □

Theorem 5.4 *$(\ell-1,k)$*WAYCUT *reduces to* RVIO$(B_{k,\ell})$.

Proof: Let $G=(V,E)$ be the input graph to $(\ell-1,k)$WAYCUT, and $T=\{t_1,\dots,t_k\}$ be the set of terminals. Let the optimal solution to $(\ell-1,k)$WAYCUT be C, and it has value α.

Let G' be obtained by identifying G with $B_{k,\ell}$. That is, take the disjoint union of G and $B_{k,\ell}$ and identifying $t_1,\dots,t_k$ with terminals of G. Let f' be the solution to RVIO$(B_{k,\ell})$ with input G'. f' has violation α'. Let C' be the violated edges.

We show that $\alpha=\alpha'$. C' is a $(\ell-1,k)$-way-cut of G. Indeed, this is because $d_{G-C'}(t_i,t_j)\geq d_{G'-C'}(t_i,t_j)\geq \ell$. Hence this shows $\alpha'\geq\alpha$. We show $G'-C$ has a retraction to $B_{k,\ell}$ by showing $B_{k,\ell}$ is an isometric subgraph of $G'-C$. Consider two vertices u,v in $B_{k,\ell}$, and let P be a path between u and v and is in $G'-C$. P can either consists of only edges in $B_{k,\ell}$, or it consists of edges outside $B_{k,\ell}$. If P consists of edges outside of $B_{k,\ell}$, it contains a subpath that start from t_i and end at t_j for some $i\neq j$ and containing only edges in $G-C$. This P has length at least ℓ. Hence this shows $d_{B_{k,\ell}}(u,v)\geq d_{G'-C}(u,v)\geq \min(d_{B_{k,\ell}}(u,v),\ell)=d_{B_{k,\ell}}(u,v)$. Hence $B_{k,\ell}$ is an isometric subgraph of $G'-C$. This shows $\alpha\geq\alpha'$. □

Theorem 5.5 *$(\ell-1,k)$*WAYCUT *is equivalent to* RVIO$(B_{k,\ell})$.

Corollary 5.6 *(ℓ,k)*WAYCUT *and (ℓ,k)*CUT *are tractable if and only if $k=2$ and $\ell\leq 3$ or $k\geq 3$ and $\ell\leq 2$.*

5.3 Variety of finite paths and trees

In this section, we make progress on the s-tractability of $H\in\mathcal{FP}$ using distance information. The results are sufficient to show dichotomy of s-tractability for reflexive trees.

Theorem 5.7 *Let $H\in\mathcal{FP}$. H is s-intractable if*

1. $\operatorname{diam}(H)\geq 5$*, or*
2. $\operatorname{diam}(H)=4$ *and there are* 3 *distinct vertices with pairwise distance* 4*.*

Proof: We first sketch the idea of the proof. Let $\ell=\operatorname{diam}(H)-1$. If $\ell\geq 4$, we reduce $(\ell,2)$WAYCUT to SVIO(H). If $\ell=3$ and we have 3 distinct vertices with pairwise distance 4, we reduce $(3,3)$WAYCUT to SVIO(H). By Theorem 5.3 and Corollary 3.5, $(\ell,2)$WAYCUT for $\ell\geq 4$ and $(3,3)$WAYCUT are NP-hard. Therefore SVIO(H) is NP-hard.

The proof for both cases are identical. In fact, we consider the more general case of (ℓ,k)WAYCUT where there exists k vertices with pairwise distance $\ell+1$ in $H=(U,F)$, and $\ell+1=\operatorname{diam}(H)$. Let those k vertices be T.

Let $G=(V,E)$ and $T\subseteq V$ be the input to (ℓ,k)WAYCUT. We assume $|V|>k$. Let M be a value more than the total weight of the edges of G.

Let $G_1=(V_1,E_1)$ obtained by taking union of G and H (note they share terminals T). Define a function $c: V_1\times V_1\to\{\ell,\ell+1\}$, where $c(u,v)=\max(\ell,d_H(u,v))$ if $u,v\in U$. Otherwise, $c(u,v)=\ell$. Let G' be the graph obtained by taking the union of G_1 and a c-connector, and all the edges in the c-connector and H has weight M.

Let f be the optimal map for SVIO(H) with input graph G', and it has value α'. Let C be the set of violated edges of the map f. C does not include any M weight edges, since $G'-E$ is a feasible solution. Let the value of the minimum (ℓ,k)-way-cut of G be α.

First, we want to show C is a (ℓ,k)-way-cut of G with terminals T. For a graph G, the set of *diametral pairs* $DP(G)$ is $\{\{x,y\}\mid x,y\in V(G), d_G(x,y)=\operatorname{diam}(G)\}$. Because $G'-C$ contains the c-connector, $DP(G'-C)\subseteq DP(H)$. f is a surjective homomorphism from $G'-C$ to H, so $d_{G'-C}(x,y)\geq d_H(f(x),f(y))$, and therefore $|DP(G'-C)|\geq|DP(H)|$. Together it shows $DP(G'-C)=DP(H)$. For each distinct pair

$x, y \in T$, $\{x, y\} \in DP(H) = DP(G' - C)$, so $d_{G-C}(x, y) \geq d_{G'-C}(x, y) = \ell + 1$. Therefore, C is a (ℓ, k)-way-cut of G and $\alpha' \geq \alpha$.

Now, we show $\alpha \geq \alpha'$. Let C be an (ℓ, k)-way-cut in G. We have to show that $G' - C$ has a surjective homomorphism to H. Let U be the vertices of H. We obtain this by showing $d_{G'-C}(u, v) = d_H(u, v)$ for all $u, v \in U$, which shows H is a isometric subgraph of $G' - C$. Note because we already had a copy of H, and paths through the c-connector does not affect distance no larger than $\ell + 1$, hence $d_{G'-E}(u, v) = d_H(u, v)$ for $u, v \in U$. Assume $d_{G'-C}(u, v) < d_H(u, v)$, this implies the shortest path from u to v using some edges in $E \setminus C$. However, any simple path that uses edges in $E \setminus C$ must go through pairs of distinct vertices $x, y \in T$. Therefore, the shortest path has length at least $\ell + 1$, and it shows $\ell + 1 \leq d_{G'-C}(u, v) < d_H(u, v) \leq \ell + 1$, a contradiction. Hence we have $d_{G'-C}(u, v) = d_H(u, v)$ for all $u, v \in U$, namely H is an isometric subgraph of $G' - C$. Therefore $G' - C$ has a retraction to H, and $\alpha \geq \alpha'$. □

Next, we show all diameter 3 graphs in $\mathcal{FP}$ is r-tractable.

Theorem 5.8 *$H \in \mathcal{FP}$ is r-tractable if* $\operatorname{diam}(H) \leq 3$.

Proof: H is a retract of the product $\prod_{i=1}^{k} P_3$ for some k. By Theorem 2.2, we need to show the theorem is true for $\prod_{i=1}^{k} P_3$. The apex subgraph of $\prod_{i=1}^{k} P_3$ is a single clique. Hence by Theorem 3.3, H is r-tractable. □

The above two theorems leaves an interesting gap for s-tractability, the cases where the graph in $\mathcal{FP}$ has diameter 4 but without any triple of vertices with pairwise distance 3. Fortunately, if we restrict to trees, such graphs are very simple. We conclude with a dichotomy theorem for reflexive trees.

Theorem 5.9 (s-tractable dichotomy for reflexive trees) *Let T be a tree, then*

1. *T is s-tractable if and only if it is r-tractable.*

2. *T is s-intractable if and only if it is r-intractable.*

3. *T is s-tractable if and only if* $\operatorname{diam}(T) \leq 4$ *and there is no* 3 *distinct vertices with pairwise distance* 4.

Proof: If T has $\operatorname{diam}(T) \geq 5$, or $\operatorname{diam}(T) = 4$ and there are 3 distinct vertices with pairwise distance 4, then it is s-intractable by Theorem 5.7, which implies T is r-intractable. Otherwise, if T has diameter at most 3, then it is r-tractable by Theorem 5.8, which implies T is s-tractable.

The remaining case is when $\operatorname{diam}(T) = 4$ and no 3 distinct vertices has pairwise distance 4. We claim this case is also r-tractable.

Consider any diametral path P of T with the sequence of vertices v_1, v_2, v_3, v_4, v_5. We claim T is a union of 3 stars centered at v_2, v_3 and v_4. It is clear such graph satisfies the desired condition.

We show T must be of this form by proving the following simple statements.

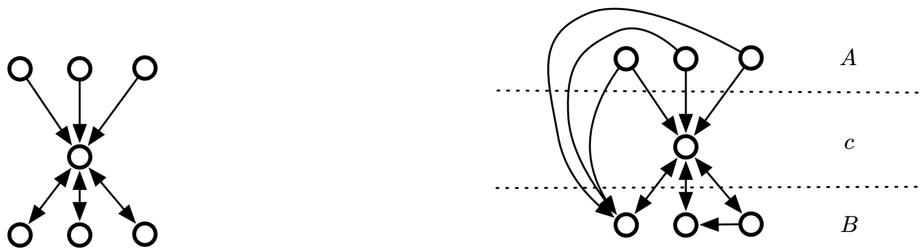
1. v_1 and v_5 must be leaves: If v_1 or v_5 are not leaves, then there exists a path of length 5, a contradiction.

2. every simple path of length 2 not intersecting P from v_2 must end in v_4: If the path ends in $v \neq v_4$, then $d_T(v, v_5) = 5$, a contradiction.

3. every simple path of length 2 not intersecting P from v_4 must end in v_2: Similar to above.

4. every simple path of length 2 not intersecting P from v_3 must end in v_1 or v_5: If the path ends in $v \notin \{v_1, v_5\}$, then $d_T(v, v_1) = d_T(v, v_5) = d_T(v_1, v_5)$, a contradiction.

The apex subgraph of T is the path v_2, v_3, v_4, which is a double clique. By Theorem 3.3, T is r-tractable. □



(a) A simple star-like digraph. (b) A more complicated star-like digraph.

Figure 6.1: Example of star-like digraphs. There can be arbitrary edges between vertices in B. There can also be edges from every vertex in A to some subset of vertices in B.

6 Star-like digraphs

In this section, we show that star-like digraphs are s-tractable. As a consequence, it shows S_k is s-tractable for all k. Because S_3 is both weakly-connected and not r-tractable by Corollary 3.4, it is the first example where one cannot use Theorem 3.3 or Theorem 4.3 to infer s-tractability.

Definition A reflexive digraph is a *star-like* digraph, if the vertices can be partitioned into 3 sets, A, B and $\{c\}$, such that there are no edges between vertices in A, ac is an edge for all $a \in A$, bc and cb are edges for each $b \in B$, and for each $b \in B$, one of the following properties hold.

- For all $a \in A$, ab is an edge and ba is not an edge in H.
- For all $a \in A$, both ab and ba are not edges in H.

We emphasis that A and B are allowed to be empty. One example of star-like digraphs is a reflexive graph where each edge is either an edge to the center, or an undirected edge with the center as an endpoint (See Figure 6.1a). S_k is the special case where there are no undirected edges. A more general star-like digraph is shown in Figure 6.1b.

A *k-subpartition* of a vertex set V is a k-partition of some subset of V. The value of a k-subpartition is the sum of incoming edges to each set in the k-subpartition. The minimum k-subpartition problem takes an input graph G and returns a minimum value k-subpartition. A k-subpartition is *U-avoiding*, if each subset in the k-subpartition is disjoint from U.

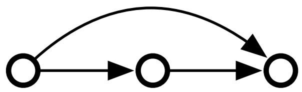
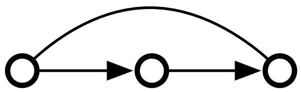
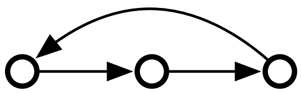
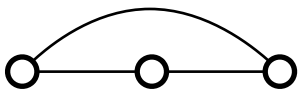
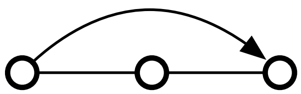
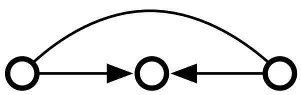
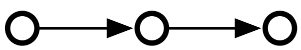
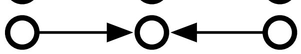
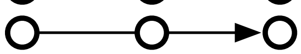
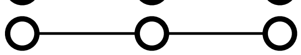
Lemma 6.1 *A minimum value U-avoiding k-subpartition can be found in polynomial time for fixed k.*

Proof: Consider the input digraph $G = (V, E)$ and $U \subseteq V$. Let M be a value that is greater than the total weight of edges in G. For each $v \in V$ and $u \in U$, we add vu as an weight M edge. Let this new graph be G'. Each non-U-avoiding k-subpartition has weight more than M in G'. Let $\mathcal{P}$ be a U-avoiding k-subpartition. In both G' and G, $\mathcal{P}$ has the same value. Hence the minimum value k-subpartition in G' is a minimum value U-avoiding k-subpartition in G. Bernáth and Pap showed the minimum value k-subpartition can be found in polynomial time for fixed k [4]. □

Theorem 6.2 *Star-like digraphs are s-tractable.*

Proof: Let $G = (V, E)$ be the input digraph to SVIO(H) with n vertices and m edges. $H = (U, F)$ is a star-like digraph with the $A, \{c\}, B$ vertex partition in section 6. Let the vertices in $A = \{a_1, \ldots, a_k\}$, and vertices in $B = \{b_1, \ldots, b_\ell\}$. Assume some total order of the vertices so that c is the smallest element. By definition, c dominates all other vertices in B. Hence, there exists an minimum violation surjective map f, where $|f^{-1}(b)| = 1$ for all $b \in B$.

Table 2: Known results for 3 vertex reflexive digraphs. We omit non-weakly connected digraphs and digraphs that can be obtained by flipping the orientation of each edge.

Pattern digraph	s-tractable?	comment
	?	Linear 3-cut
	?	
	No	non-transitive reflexive tournament [42].
	Yes	r-tractable
	Yes	r-tractable
	Yes	r-tractable
	?	
	Yes	r-tractable, double cut
	Yes	r-tractable
	Yes	r-tractable

If $A = \emptyset$, the apex subgraph of H is a single vertex c. H is r-tractable by Theorem 2.12, which implies H is s-tractable. Otherwise, let T be a set of $\ell + 1$ elements, which consists of $f^{-1}(B)$ and an arbitrary element $v \in f^{-1}(c)$.

Let w be the total weight of the edges of the form $f^{-1}(b)f^{-1}(b')$, where bb' is not an edge in H and $b, b' \in B$. For each $b \in B$ such that it is not incident to any vertex in A, we reorient every incoming edge of $f^{-1}(b)$ in G to an outgoing edge. That is, if $xf^{-1}(b)$ is an edge, we remove $xf^{-1}(b)$ and add $f^{-1}(b)x$ with the same weight.

Let the new digraph be G'. We compute a T-avoiding $|A|$-subpartition in G' in polynomial time by Lemma 6.1. Let the weight of this solution be w'. Let the subpartitions be $V_1, \ldots, V_k$, and we define a vertex map $h : V \to U$ as follows.

$$h(v) = \begin{cases} a_i & \text{if } v \in V_i \\ f(v) & \text{if } v \in T \\ c & \text{otherwise} \end{cases}$$

This would obtain a vertex map h with violation $w + w'$. The violation of h is no larger than f. Hence, we can try all possible choices of T, and return the minimum solution. There are $O(n^{\ell+1})$ choices, so we run a polynomial number of polynomial time algorithm, which gives us a polynomial running time. □

7 Conclusive Remarks

Our paper introduces a systematic study of r-tractable and s-tractable digraphs. We give a complete resolution of r-tractability. However, s-tractability is much harder to work with. The classification of s-tractable reflexive digraphs on 3 vertices is still open. The state of the art is documented in Table 2. The remaining 3 unclassified patterns are all r-intractable and s_0-tractable. Hence new techniques are required. It would be interesting to see if there are other cut problems that can be modeled by this framework.

Acknowledgments We like to thank Standa Živný for the pointer to the result of [15], Pavol Hell for discussion on graph retraction, Maxim Sviridenko for pointing to the literatures on 0-extensions and metric labeling, and Devanathan Thiruvenkatachari for discussing on c-connectors. We also like to thank the anonymous reviewers who gave various feedbacks in terms of presentation and correctness.

References

[1] Georg Baier, Thomas Erlebach, Alexander Hall, Ekkehard Köhler, Petr Kolman, Ondřej Pangrác, Heiko Schilling, and Martin Skutella. Length-bounded cuts and flows. *ACM Transactions on Algorithms*, 7(1):1–27, nov 2010. URL: http://portal.acm.org/citation.cfm?doid=1868237.1868241, doi:10.1145/1868237.1868241.

[2] Kristóf Bérczi, Karthekeyan Chandrasekaran, Tamás Király, Euiwoong Lee, and Chao Xu. Beating the 2-approximation factor for global bicut. *Mathematical Programming*, 177(1):291–320, Sep 2019. doi:10.1007/s10107-018-1270-8.

[3] Attila Bernáth and Gyula Pap. Blocking optimal arborescences. *Mathematical Programming*, 161(1):583–601, Jan 2017. doi:10.1007/s10107-016-1025-3.

[4] Attila Bernáth and Gyula Pap. Blocking unions of arborescences. *Discrete Optimization*, 22(Part B):277 – 290, 2016. URL: http://www.sciencedirect.com/science/article/pii/S157252861630055X, doi:https://doi.org/10.1016/j.disopt.2016.08.003.

[5] Manuel Bodirsky, Jan Kára, and Barnaby Martin. The complexity of surjective homomorphism problems—a survey. *Discrete Applied Mathematics*, 160(12):1680 – 1690, 2012. URL: http://www.sciencedirect.com/science/article/pii/S0166218X12001333, doi:https://doi.org/10.1016/j.dam.2012.03.029.

[6] Andrei A. Bulatov. A dichotomy theorem for nonuniform CSPs. In *Proc. FOCS 2017*, 2017.

[7] Kristóf Bérczi, Karthekeyan Chandrasekaran, Tamás Király, and Vivek Madan. A tight $\sqrt{2}$-approximation for linear 3-cut. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '18, pages 1393–1406. URL: http://epubs.siam.org/doi/abs/10.1137/1.9781611975031.92, arXiv:http://epubs.siam.org/doi/pdf/10.1137/1.9781611975031.92, doi:10.1137/1.9781611975031.92.

[8] Gruia Calinescu, Howard Karloff, and Yuval Rabani. Approximation algorithms for the 0-extension problem. In *Proceedings of the Twelfth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '01, pages 8–16, Philadelphia, PA, USA, 2001. Society for Industrial and Applied Mathematics. URL: http://dl.acm.org.proxy2.library.illinois.edu/citation.cfm?id=365411.365413.

[9] Karthekeyan Chandrasekaran and Chao Xu. A note on k-subpartitioning in directed graphs. Unpublished manuscript, 2018.

[10] Chandra Chekuri and Vivek Madan. Approximating multicut and the demand graph. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '17, pages 855–874, Philadelphia, PA, USA, 2017. Society for Industrial and Applied Mathematics. URL: http://dl.acm.org/citation.cfm?id=3039686.3039740.

[11] Chandra Chekuri, Kent Quanrud, and Chao Xu. LP Relaxation and Tree Packing for Minimum k-cuts. In Jeremy T. Fineman and Michael Mitzenmacher, editors, *2nd Symposium on Simplicity in Algorithms (SOSA 2019)*, volume 69 of *OpenAccess Series in Informatics (OASIcs)*, pages 7:1–7:18, Dagstuhl, Germany, 2018. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. URL: http://drops.dagstuhl.de/opus/volltexte/2018/10033, doi:10.4230/OASIcs.SOSA.2019.7.

[12] Hubie Chen. An algebraic hardness criterion for surjective constraint satisfaction. *Algebra universalis*, 72(4):393–401, Dec 2014. doi:10.1007/s00012-014-0308-x.

[13] E. Dahlhaus, D. S. Johnson, C. H. Papadimitriou, P. D. Seymour, and M. Yannakakis. The Complexity of Multiterminal Cuts. *SIAM Journal on Computing*, 23(4):864–894, Aug. 1994. doi:10.1137/S0097539792225297.

[14] V.G. Deineko and V.L. Filonenko. On the reconstruction of specially structured matrices, 1979. (Russian).

[15] Vladimir Deineko, Peter Jonsson, Mikael Klasson, and Andrei Krokhin. The approximability of MAX CSP with fixed-value constraints. *J. ACM*, 55(4):16:1–16:37, Sep 2008. doi:10.1145/1391289.1391290.

[16] Rodney G. Downey, Vladimir Estivill-Castro, Michael Fellows, Elena Prieto, and Frances A. Rosamund. Cutting up is hard to do: The parameterised complexity of k-cut and related problems. *Electronic Notes in Theoretical Computer Science*, 78:209 – 222, 2003. CATS'03, Computing: the Australasian Theory Symposium. URL: http://www.sciencedirect.com/science/article/pii/S1571066104810144, doi:https://doi.org/10.1016/S1571-0661(04)81014-4.

[17] Pavel Dvořák and Dušan Knop. Parameterized complexity of length-bounded cuts and multicuts. *Algorithmica*, Jan 2018. doi:10.1007/s00453-018-0408-7.

[18] Itamar Elem. The minimum G_c cut problem. Master's thesis, Tel-Aviv University, 2011.

[19] Itamar Elem, Rafael Hassin, and Jérôme Monnot. The minimum G_c cut problem. http://www.math.tau.ac.il/~hassin/itamar.pdf. Accessed: Sep-07-2017.

[20] Tomás Feder and Moshe Y. Vardi. The computational structure of monotone monadic snp and constraint satisfaction: A study through datalog and group theory. *SIAM Journal on Computing*, 28(1):57–104, 1998. arXiv:https://doi.org/10.1137/S0097539794266766, doi:10.1137/S0097539794266766.

[21] Peter Fulla, Hannes Uppman, and Stanislav Živný. The complexity of boolean surjective general-valued csps. *ACM Trans. Comput. Theory*, 11(1):4:1–4:31, November 2018. URL: http://doi.acm.org/10.1145/3282429, doi:10.1145/3282429.

[22] N. Garg, V. Vazirani, and M. Yannakakis. Multiway cuts in directed and node weighted graphs. In *Proceedings of the 20th International Colloquium on Automata, Languages and Programming*, ICALP '94, pages 487–498, 1994.

[23] Olivier Goldschmidt and Dorit S. Hochbaum. A Polynomial Algorithm for the k-cut Problem for Fixed k. *Mathematics of Operations Research*, 19(1):24–37, 1994. doi:10.1287/moor.19.1.24.

[24] Flavio Guiñez and Maurice Queyranne. The size-constrained submodular k-partition problem. Manuscript, https://docs.google.com/viewer?a=v&pid=sites&srcid=ZGVmYXVsdGRvbWFpbnxmbGF2aW9ndWluZXpob21lcGFnZXxneDo0NDVlMThkMDg4ZWRlOGI1,

2012. URL: https://docs.google.com/viewer?a=v&pid=sites&srcid=ZGVmYXVsdGRvbWFpbnxmbGF2aW9ndWluZXpob21lcGFnZXxneDo0NDVlMThkMDg4ZWRlOGI1.

[25] Anupam Gupta, David G. Harris, Euiwoong Lee, and Jason Li. Optimal bounds for the k-cut problem, 2020. arXiv:2005.08301.

[26] Anupam Gupta, Euiwoong Lee, and Jason Li. The number of minimum k-cuts: Improving the karger-stein bound. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2019, pages 229–240, New York, NY, USA, 2019. ACM. URL: http://doi.acm.org/10.1145/3313276.3316395, doi:10.1145/3313276.3316395.

[27] Anupam Gupta, Euiwoong Lee, and Jason Li. The karger-stein algorithm is optimal for k-cut. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2020, page 473–484, New York, NY, USA, 2020. Association for Computing Machinery. doi:10.1145/3357713.3384285.

[28] Gregory Gutin, Pavol Hell, Arash Rafiey, and Anders Yeo. A dichotomy for minimum cost graph homomorphisms. *European Journal of Combinatorics*, 29(4):900 – 911, 2008. Homomorphisms: Structure and Highlights. URL: http://www.sciencedirect.com/science/article/pii/S0195669807002053, doi:https://doi.org/10.1016/j.ejc.2007.11.012.

[29] Pavol Hell, Monaldo Mastrolilli, Mayssam Mohammadi Nevisi, and Arash Rafiey. Approximation of minimum cost homomorphisms. In Leah Epstein and Paolo Ferragina, editors, *Algorithms – ESA 2012*, pages 587–598, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[30] Pavol Hell and Jaroslav Nešetřil. *Graphs and Homomorphisms*. Oxford Lecture Series in Mathematics and Its Applications. Oxford University Press, 2004.

[31] Pavol Hell and Arash Rafiey. The dichotomy of minimum cost homomorphism problems for digraphs. *SIAM Journal on Discrete Mathematics*, 26(4):1597–1608, 2012. doi:10.1137/100783856.

[32] Pavol Hell and Ivan Rival. Absolute retracts and varieties of reflexible graphs. *Can. J. Math.*, 39(3):544–567, 1987. doi:10.4153/CJM-1987-025-1.

[33] Hiroshi Hirai. Discrete convexity and polynomial solvability in minimum 0-extension problems. *Mathematical Programming*, 155(1):1–55, Jan 2016. doi:10.1007/s10107-014-0824-7.

[34] Anna Huber, Andrei Krokhin, and Robert Powell. Skew bisubmodularity and valued CSPs. *SIAM Journal on Computing*, 43(3):1064–1084, 2014. doi:10.1137/120893549.

[35] A. Itai, Y. Perl, and Y. Shiloach. The complexity of finding maximum disjoint paths with length constraints. *Networks*, 12(3):277–286, 1982. URL: http://dx.doi.org/10.1002/net.3230120306, doi:10.1002/net.3230120306.

[36] Peter Jonsson, Mikael Klasson, and Andrei Krokhin. The Approximability of Three-valued MAX CSP. *SIAM Journal on Computing*, 35(6):1329, 2006. URL: http://link.aip.org/link/SMJCAT/v35/i6/p1329/s1{&}Agg=doi, doi:10.1137/S009753970444644X.

[37] Ken-ichi Kawarabayashi and Mikkel Thorup. The minimum k-way cut of bounded size is fixed-parameter tractable. In *Proceedings of the 2011 IEEE 52Nd Annual Symposium on Foundations of Computer Science*, FOCS '11, pages 160–169, Washington, DC, USA, 2011. IEEE Computer Society. URL: http://dx.doi.org/10.1109/FOCS.2011.53, doi:10.1109/FOCS.2011.53.

[38] Jon Kleinberg and Éva Tardos. Approximation algorithms for classification problems with pairwise relationships: Metric labeling and markov random fields. *J. ACM*, 49(5):616–639, September 2002. URL: http://doi.acm.org/10.1145/585265.585268, doi:10.1145/585265.585268.

[39] Vladimir Kolmogorov. Testing the Complexity of a Valued CSP Language. In Christel Baier, Ioannis Chatzigiannakis, Paola Flocchini, and Stefano Leonardi, editors, *46th International Colloquium on Automata, Languages, and Programming (ICALP 2019)*, volume 132 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 77:1–77:12, Dagstuhl, Germany, 2019. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. URL: http://drops.dagstuhl.de/opus/volltexte/2019/10653, doi:10.4230/LIPIcs.ICALP.2019.77.

[40] Richard E. Ladner. On the structure of polynomial time reducibility. *J. ACM*, 22(1):155–171, January 1975. URL: http://doi.acm.org/10.1145/321864.321877, doi:10.1145/321864.321877.

[41] Michael Langberg, Yuval Rabani, and Chaitanya Swamy. Approximation algorithms for graph homomorphism problems. In Josep Díaz, Klaus Jansen, José D. P. Rolim, and Uri Zwick, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 176–187, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[42] Benoît Larose, Barnaby Martin, and Daniël Paulusma. Surjective h-colouring over reflexive digraphs. *ACM Trans. Comput. Theory*, 11(1):3:1–3:21, November 2018. URL: http://doi.acm.org/10.1145/3282431, doi:10.1145/3282431.

[43] Euiwoong Lee. Improved Hardness for Cut, Interdiction, and Firefighter Problems. In Ioannis Chatzigiannakis, Piotr Indyk, Fabian Kuhn, and Anca Muscholl, editors, *44th International Colloquium on Automata, Languages, and Programming (ICALP 2017)*, volume 80 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 92:1–92:14, Dagstuhl, Germany, 2017. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. URL: http://drops.dagstuhl.de/opus/volltexte/2017/7485, doi:10.4230/LIPIcs.ICALP.2017.92.

[44] A. Ridha Mahjoub and S. Thomas McCormick. Max flow and min cut with bounded-length paths: complexity, algorithms, and approximation. *Mathematical Programming*, 124(1):271–284, Jul 2010. doi:10.1007/s10107-010-0366-6.

[45] Barnaby Martin and Daniël Paulusma. The computational complexity of disconnected cut and $2K_2$-partition. *Journal of Combinatorial Theory, Series B*, 111(Supplement C):17 – 37, 2015. URL: http://www.sciencedirect.com/science/article/pii/S0095895614001026, doi:https://doi.org/10.1016/j.jctb.2014.09.002.

[46] Gregor Matl and Stanislav Živný. Beyond Boolean Surjective VCSPs. In Rolf Niedermeier and Christophe Paul, editors, *36th International Symposium on Theoretical Aspects of Computer Science (STACS 2019)*, volume 126 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 52:1–52:15, Dagstuhl, Germany, 2019. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. URL: http://drops.dagstuhl.de/opus/volltexte/2019/10291, doi:10.4230/LIPIcs.STACS.2019.52.

[47] Akbar Rafiey, Arash Rafiey, and Thiago Santos. Toward a Dichotomy for Approximation of H-Coloring. In Christel Baier, Ioannis Chatzigiannakis, Paola Flocchini, and Stefano Leonardi, editors, *46th International Colloquium on Automata, Languages, and Programming (ICALP 2019)*, volume 132 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 91:1–91:16, Dagstuhl, Germany, 2019. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. URL: http://drops.dagstuhl.de/opus/volltexte/2019/10667, doi:10.4230/LIPIcs.ICALP.2019.91.

[48] A. A. Schoone, H. L. Bodlaender, and J. Van Leeuwen. Diameter increase caused by edge deletion. *Journal of Graph Theory*, 11(3):409–427, 1987. URL: http://dx.doi.org/10.1002/jgt.3190110315, doi:10.1002/jgt.3190110315.

[49] Johan Thapper and Stanislav Živný. The complexity of finite-valued csps. *J. ACM*, 63(4):37:1–37:33, September 2016. URL: http://doi.acm.org/10.1145/2974019, doi:10.1145/2974019.

[50] Li Pu Yeh, Biing Feng Wang, and Hsin Hao Su. Efficient algorithms for the problems of enumerating cuts by non-decreasing weights. *Algorithmica (New York)*, 56(3):297–312, 2010. doi:10.1007/s00453-009-9284-5.

[51] Daniel M. Yellin. Algorithms for subset testing and finding maximal sets. In *Proceedings of the Third Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '92, pages 386–392, Philadelphia, PA, USA, 1992. Society for Industrial and Applied Mathematics. URL: http://dl.acm.org/citation.cfm?id=139404.139481.

[52] Dmitriy Zhuk. A proof of csp dichotomy conjecture. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 331–342. IEEE, 2017.