Flow Augmentation and it's applications in Parameterized Algorithms

By

Shashank Rustagi

Under the supervision of Dr. Diptapriyo Majumdar

Indraprastha Institute of Information Technology Delhi

March, 2024

Flow Augmentation and it's applications in Parameterized Algorithms

By

Shashank Rustagi

Submitted in partial fulfillment of the requirements for the degree of

Master of Technology

Computer Science and Engineering

To

Indraprastha Institute of Information Technology Delhi

March, 2024

Certificate

This is to certify that the scholarly paper titled *Flow Augmentation* being submitted by *Shashank Rustagi* to Indraprastha Institute of Information Technology Delhi, for the award of the degree of Master of Technology, is an original research work carried out by him under my supervision. In my opinion, the scholarly paper has reached the standards fulfilling the requirements of the regulations relating to the degree.

The results contained in this thesis have not been submitted in part or full to any other university or institute for the award of any degree/diploma.

March, 2024

Dr. Diptapriyo Majumdar

Assistant Professor (CSE)

Indraprastha Institute of Information Technology Delhi

New Delhi 110020

Contents

Certificate **1**

Acknowledgements **1**

Abstract **2**

Introduction **3**

Preliminaries **6**

Solving hard cut problems via Flow Augmentation in Undirected Graphs **11**

Directed Flow Augmentation in weighted almost 2-SAT problem **18**

FPT of DMC-3 parameterized by the size of the cutset **27**

References **1**

Acknowledgements

We would like to sincerely thank our advisor, whose constant guidance, availability, and support have enabled me to navigate every obstacle encountered in the course of this project. I have learned and grown immensely through their constructive criticism and feedback on my work.

We are also grateful for our time at IIIT Delhi, which has proved to be a turning point in my life and has provided me with the opportunity to learn from exceptionally knowledgeable professors to better myself at my craft. Every lecture has been a privilege.

Abstract

The graph cut/graph separation problems in directed and undirected graphs are being designed using FPT(Fixed-Parameter Tractable) algorithms which state that an NP-hard problem could be converted to polynomial time using k as a parameter. This is called flow augmentation. The researchers state that given a graph directed/undirected G, the time taken to find the min s-t cut where s,$t \in V(G)$ will be polynomial time. For undirected graphs, paper 1 states that there is a solution $Z \subseteq E(G)$, which is of size $\leq$ k s.t if we remove Z from G, the s and t become two connected components individually i.e distinct. Now, for every edge in Z, the endpoints are connected in either of the two components. This can be proven in randomized time $k^{O(1)}(|V(G)| + |(E(G))|)$ edges can be added to the graph, and with the probability $2^{-O(klogk)}$ no added edge could connect the distinct components s and t in $G - Z$. Similarly, paper 2 states that in a directed graph G, the number of arcs are added to G such that the $s - t$ cut in G of size $\leq k$ becomes minimum with probability $2^{-poly(k)}$. This states that the flow augmentation technique benefits FPT(fixed-parameter tractable) problems and helps reduce NP-hard problems to polynomial time.

We show flow augmentation algorithm in directed graphs. And we show that there exist a polynomial time algorithm that given a directed graph G, two integers $s, t \in V(G)$ and an integer k,adds randomly to G, a number of arcs such that for every minimal (s, t)-cut in G of size at most k, with probability $2^{-poly(k)}$, the set becomes minimum (s, t)-cut in the resulting graph.

In the third research paper, the researchers have shown the fixed parameter tractability of directed multi cut problem with three terminal vertices pairs $(s_1, t_1), (s_2, t_2)$ and (s_3, t_3) and an integer k and a set of at most k non terminal vertices that intersects all s_1t_1, s_2t_2 and s_3t_3 paths. Now, using the directed flow augmentation which we studied in chapter 2, we will cast this problem into Constraint Satisfaction problem and we will use twin width as a tool and get the FPT for this problem.

1. Introduction

Graph cut/graph separation problems are considered necessary while doing parameterized complexity. So if a graph is given (directed/undirected), the researchers aim at obtaining a cut/separation with deletion of $\leq$ k vertices or edges. [1]

As discussed above, a min $s-t$ cut problem focuses on the deletion of k vertices such that there remains no path between s and t in the resultant graph $G-Z$. [2][3]

Whereas the Feedback Vertex Set focuses on the deletion of k vertices such that there is no cycle in the graph $G-Z$. Similarly, for a coupled min-cut problem which is also called as the Constraint Satisfaction Problem(CSP), we have a graph $G(V,E)$ as the input graph, where s,t $\in V(G)$, we need to find the $s-t$ cut such that the graph gets partitioned in $\leq$ k pairs.

Flow augmentation takes a graph $G=(V,E)$ with s,t $\in V(G)$ and edges(A) are added to G in such a way that the probability with which new added edge does not interfere with $Z(s-t$ cut) is $\frac{1}{f(k)}$ and Z becomes the $s-t$ min cut in resulting graph $G+A$.

The main result described in paper 1 is based on: If an undirected graph G is given with $s,t \in V(G)$ and Z as an $s-t$ cut. From $G-Z$ we have $Z_{s,t} \subseteq Z$, these edges are reachable from s and t in $G-Z$. It is being observed that if Z is having the end points of each of the edges present in $G-Z$, and Z_s,t is an $(s-t)$ cut and eligible for (s-t), then Z is a special (s-t) cut.

But it should be also understood that why there is need of flow augmentation techniques and why the methods such as greedy methods, shadow removal techniques, tree width reduction fail.

In greedy methods and shadow removal some cuts are chosen in a greedy manner but they fail to apply for coupled min-cut problems due to some edge coupling

constraints. Various recursive methods are used for graph decomposition problems as if sparse cut is found in G, then recursive methods cannot be applied as they contradict with the edge coupling which says that there is no dependency between the two parts of the cut.

Now, in the above scenario it is being discussed that using treewidth reduction the separations lie between the treewidth of $2^{O(k)}$ counters in directed graph. Here, greedy methods are sometimes used along with FPT but there are no methods developed till date which proves the lower bounds for FPT algorithms in directed graph settings.

We discuss a new technique for FPT algorithms for graph cut problems in undirected graphs. This is called as flow augmentation. In these type of problems, we generally look for an edge (s, t) cut problem of cardinality at most k where s and t are terminal vertices and we talk about a solution set where that solution set Z is a proper subset of edge set of the graph G i.e. $E(G)$ of size at most k. and when we delete this solution set Z from the graph, then there are two disconnected components and s is present in one component and t is present in another one. Also, every edge of Z will connect two distinct connected components of $G - Z$ and if we say there is a set $Z_{s,t}$ subset of Z as the edges e of the solution set Z for which there exist an (s, t) path P_e with $E(P_e)$ upon intersection with Z will give a singleton set having only one edge e and then this $Z_{s,t}$ will separate s from t.

In the second research paper, we talk about directed flow augmentation. It has been shown that there is a polynomial time algorithm which says that if we have a directed graph G, and two designated vertices s and t, just as in the case of first research paper, which belongs to the vertex set of the graph and an integer k, it will add randomly to our graph a number of arcs such that for every minimal st cut Z in G of size at most k, then with probability $2^{-poly(k)}$, the set Z becomes a minimum st cut in the resulting graph. This directed flow augmentation also allows us to prove the fixed parameter tractability of a lot of problems parameterised by the cardinality of the deletion set, whose parameterised complexity status was an open issue in the community. These problems are Chain SAT and Weighted st CUT, Weighted Directed Feedback Vertex Set or Weighted Almost 2-SAT.

In graph separation problems, our aim is to find one edge or one vertex such that it is of importance in some graph separation requirements. We need

to understand what is the multicut in order to understand the fixed parameter tractability of Directed multi cut with 3 pairs. So, a multicut in a graph G, with a family T of pair of terminal vertices, our goal is to delete a minimum number of non terminal vertices so that there is no path from s to t in the entire graph. The parameterised complexity of this problem in the case of undirected graph has been obtained in the year 2010, but in the case of directed graphs, it was observed to be $W-1\ hard$. Directed multicut with three terminal pairs is one of the hardest open problems in terms of tractability. In these type of problems, in our research paper, a new technique has been found out by Kim and group [4] to solve the cut problem in the directed graph setting namely flow augmentation.

Theorem 1.1. D*irected Multicut with three terminal pairs is fixed-parameter tractable when parameterized by the size of the cutset (with a randomized algorithm). [4]*

Theorem 1.2. *There exists a polynomial time randomised algorithm that given a directed graph G, two designated vertices s and t ∈ V(G), and an integer k, outputs a set A ⊆ $V(G) \times V(G)$ called augmentation edges such that for every minimal edge st-cut Z of cardinality at most $2^{-O(k^4 \log k)}$ [4]*

We got the graph $G + A$ by adding arcs A in G. Thus, by adding some arcs A in the graph, with some probability, these arcs connects the s side to the t side and since we have added more edges, thereby it is evident that the connectivity of the graph also has been increased significantly. So, Let G be a directed graph, $s, t \in V(G)$, and let k be $|\lambda|$ where λ is the size of the minimum edge st-cut. Suppose, P is any maximum st-flow consisting of k edge disjoint paths from s to t. So, if we take any minimum edge st-cut, it has only one edge from each path in P.

Let there be a vertex u_1 on path P_1 and u_2 on path P_2. So, if G contains a path Q from u_1 to u_2 that does not contain any edge of P, then any minimum edge st-cut cannot contain an edge of P_1 after u_1 and any edge of P_2 before u_2 at the same time. This helps us in giving the following constraint satisfaction problem formulation. The constraint added is $(x(P_1) \leq u_1) \vee (x(P_2) \geq u_2)$ where these inequalities have a meaning that being before or after the corresponding vertex along the corresponding path. The space of all instances to such CSP instance is exactly the space of all the minimum edge st-cuts in G. So, we can sub sample the space of all minimal edge st-cuts in G of cardinality at most k, so that every cut is sampled with good probability $2^{-O(k^4 \log k)}$

2. Preliminaries

2.1 Undirected Graphs

The input graph is undirected multigraph without any loops. Here a multigraph is defined as $G = (V, E, \pi)$ where V denotes the vertices, E denotes the number of edges and π denotes $\binom{V}{2}$. A is a multiset of edges on V. So, if we add A to the graph G it becomes $G + A$ and simultaneously upon deleting A from G, we get $G - A$

We have $\delta(S)$ as the multiset of edges from set S which has one endpoint in S, and $\partial(S)$ denotes the incident vertices from set S which have ≥ 1 edge in $\delta(S)$

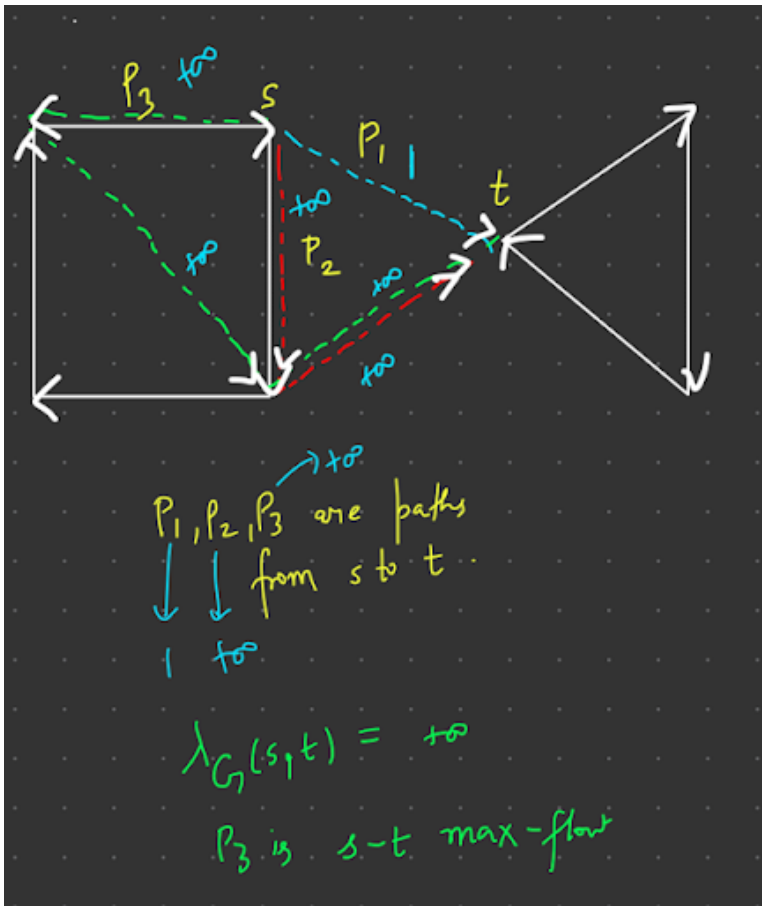
Some notations are S,T $\subseteq$ V and X $\subseteq R_S(X)$ are the set of vertices we can reach from any of the vertex of S in $G - X$. We define X as the $s - t$ cut if $R_S(X) \cap R_T(X) = \phi$. It is defined that X will be the minimal $s - t$ cut if there is no proper subset of X which is an $s - t$ cut.

Overall we define an $s - t$ cut as the path from S to T with maximum capacity and $S - T$ flow is defined as the maximum number of paths from S to T in G with maximum capacity. $\lambda_G(S, T) =$ minimum size of $S - T$ cut in G.

We are having an undirected multi graph without loops and a multigraph could be captured as a tuple $G = (V, E, \pi)$ where V and E are finite sets and $\pi : E- > \binom{v}{2}$ is a mapping. For the multi graph G and A a multi set of edges on V, the graphs $G + A$ and $G - A$ are accordingly understood as starting from G and respectively adding all edges in A that are not yet in G or removing from G all edges that are also in A. For a vertex set S, we denote $\delta(S)$ the multiset of edges that have precisely one endpoint in S and by $\Delta(S)$ the set of vertices in S that are incident with at least one edge in $\delta(S)$. An st-cut is equal to the maximum number of edge disjoint paths from S to T in G or equivalently, to the maximum unit capacity st- flow. By $\lambda_G(s, t)$, we denote the maximum flow

from S to T or the minimum size of an st- CUT in G.

In paper 2, it has been observed that $S - T$ flow is also denoted as the collection of paths P such that there is only one edge of capacity 1 which lies on any one of the path of P.

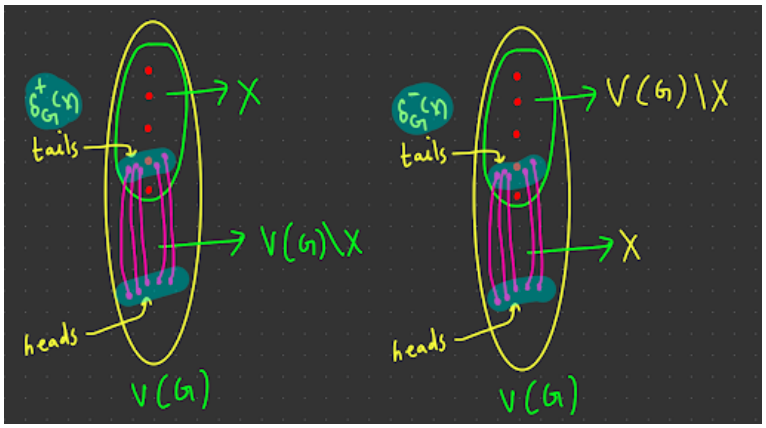


The maximum $s - t$ flow or s-t maxflow is defined as $\lambda_G(S, T) = +\infty$ or it can also be defined as the path where all edges have a capacity $+\infty$. It's been observed that $Z \subseteq E(G)$ becomes an $s - t$ cut if there is no edge of capacity $+\infty$ and there is no path in $G - Z$ from s to t. Further if there is no proper subset of Z then the $s - t$ cut is called as the min-cut. In one of the lemmas it's been stated that $Core(Z)$ is the minimal $s - t$ cut if Z is the star $s - t$ cut

2.2 Directed Graphs

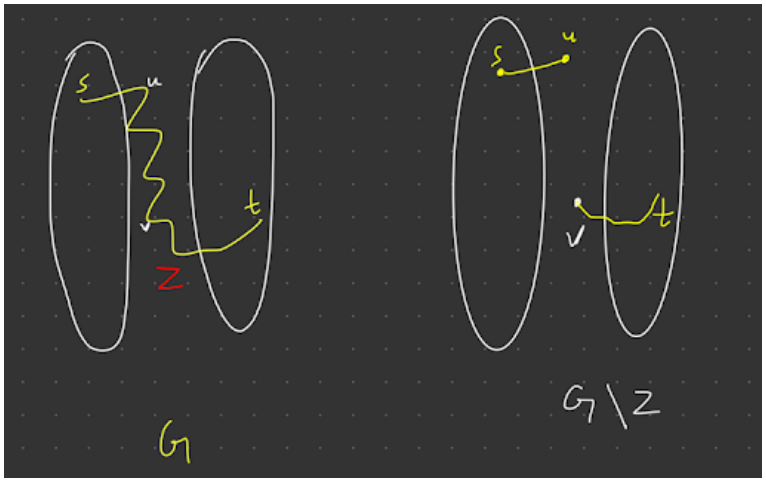
A directed graph H with vertex V induced on the graph H is $= [\lambda]$ i.e. each vertex in this graph, corresponds to path $P_i \in P$ that contains self loop at every vertex is known as reachability pattern. Let (I, P) be an instance with maximum

flow and reachability pattern, if number of vertices induced on the graph H is same as number of edges induced on the graph H, then for every minimal st-cut Z, that is not a minimum cut, there exist $i \in [\lambda]$ such that no edge of $Z \cap E(P_i)$ is a bottleneck edge.



Definition 1. *A bottleneck edge is an edge in a flow network that on being increased, it increases the maximum flow of the network. For a directed graph G and a set X in the vertex set of graph, the set $\delta_G^-(X)$ is the set of arcs with tails in X and heads in $V(G) - X$ and heads in X. We write $\delta_G^+(v)$ and $\delta_G^-(v)$ [4]*

Definition 2. *An st-cut is called a $star$ st-cut for all edges (u, v) in Z, in the graph $G - Z$ there is path from s to u but there is no path from s to v. [5]*



Lemma 2.1. *If, Z is a star st-cut, in the graph G, then $core(Z)$ is a minimal st-cut. [5]*

Lemma 2.2. *If Z is a star st-cut in G and $A \subseteq V(G) \times V(G)$ is compatible with Z, then Z is a star st-cut in $G + A$ as well. [4]*

2.3 Twin Width

In the third research paper, we will discuss about what is twin width and permutation CSP. G is a directed graph. An instance of Directed Multicut (3-DMC for short) is a tuple $(G, k, (s_i, t_i) \in [3], V_\infty)$ consisting of a directed graph G, six distinguished vertices $s_1, s_2, s_3, t_1, t_2, t_3 \in V(G)$, called terminals, an integer k, and a vertex subset $V_\infty \subseteq V(G)$, called undeletable vertices. A solution is a set S of non-terminal vertices of G such that $S \cap V_\infty = \phi$ and for every i = 1, 2, 3 the vertex t_i is not reachable from the vertex $s_i \in G - S$. Directed Multi cut asks for a solution of cardinality at most k.

Every graph G is a trigraph with only black edges and reg edges being empty. Black edges are represented by $E(G)$ and reg edges are represented as $R(G)$. When we contract two vertices $u, v \in V(G)$, if we merge them into a single vertex w, then possibly color the edges incident to the new vertex w. Every exiting wz remains black if and only if uz and vz were previously black edges. All other edges incident to w are colored in red.

Definition 3. *Contraction sequence of an n vertex trigraph G is a sequence of trigraphs $G = G_n, ..., G_1 = K_1$ such that G_i is obtained from G_{i+1} by contracting two vertices. [6]*

Definition 4. *A $d - sequence$ is a contraction sequence if all the trigraphs in it have the red degree at most d. [6]*

Definition 5. *Twin width of the graph G is denoted by $dww(G)$, it is the minumum such integer d such that G admits a $d - sequence$. [6]*

Now, let us talk about permutation CSP, there is a a constraint $C \in C$ is satisfied by α if

Lemma 2.3. *Let G be a directed graph and let $C \subseteq$ V(G). Let G' be obtained from G after bypassing C and let $S \subseteq$ V(G) \ C. For any $a, b \in V(G) \backslash (C \cup S)$, $G - S$ has an ab-path if and only if $G' - S$ has an ab-path. [4]*

The above lemma states that if we have an instance of graph G, our parameter k and all the 6 designated terminal vertices and the vertex set V^∞ of 3-DMC instance and a set $X \subseteq V(G) \backslash V^\infty$ and a vertex $v \in V(G) \backslash (X \cup V^\infty)$ is in the forward shadow of X if v is reachable from neither of the three starting terminal vertices in $G - X$. Likewise, v is in the reverse shadow of X if neither of the

ending terminal vertices is reachable from v in $G - X$. Set of vertices in the forward shadow of X and reverse shadow of X are denoted by $f_G(X)$ and $r_G(X)$ respectively. Any vertex v is in the shadow of X if it is in either of the forward or reverse shadow of X. A set X is shadowless if $f_G(X) \cup r_G(X) = \Phi$. It follows directly from [7] and [8].

Theorem 2.1. *Given an instance $(G, k, (s_i, ti)_{i\in[3]}, V^\infty)$ of $3 - DMC$, there is an algorithm that runs in time $2^{O(k^2 \log k)} n^{O(1)}$, and outputs a family Z of subsets of $V(G) \backslash V^\infty$ such that $|Z| = 2^{O(k^2 \log k)} \log^2 n$ and, if the input instance is a Yes-instance, then there exists $Z \in Z$ such that $(G, k, (s_i, ti)_{i\in[3]}, V^\infty \backslash Z)$ is a Yes-instance that admits a shadowless solution of cardinality at most k, where G' is the result of bypassing Z in G. [6]*

3. Solving hard cut problems via Flow Augmentation in Undirected Graphs

We are given one undirected graph $G = (V, E)$ with two designated vertices $s, t \in V(G)$ and there is one st-cut which is not known as of now, and let $Z(s, t) \subseteq Z$ be those edges whose endpoints are reachable from both s, t in $G - Z$.

We discuss a new technique for FPT algorithms for graph cut problems in undirected graphs. This is called as flow augmentation.[3] In these type of problems, we generally look for an edge (s, t) cut problem of cardinality at most k where s and t are terminal vertices and we talk about a solution set where that solution set Z is a proper subset of edge set of the graph G i.e. $E(G)$ of size at most k. and when we delete this solution set Z from the graph, then there are two disconnected components and s is present in one component and t is present in another one. Also, every edge of Z will connect two distinct connected components of $G - Z$ and if we say there is a set $Z_{s,t}$ subset of Z as the edges e of the solution set Z for which there exist an (s, t) path P_e with $E(P_e)$ upon intersection with Z will give a singleton set having only one edge e and then this $Z_{s,t}$ will separate s from t.

There exist a proof that in $2^{-O(k \log(k)}$ probability no added edge can connect two components of $G - Z$ and $Z_{s,t}$ will be the minimum $s - t$ cut.

In this research paper 1, we are studying Fixed Parameter Tractable Algorithm for undirected graph cuts. The flow augmentation technique is there for the construction of FPT algorithms for these graph separation problems. Let us say we have the problems like l-chain SAT, Coupled MIN-CUT problem or any other constraint satisfaction problem which are w[1] hard in general. And assume we know the solution Z of any of these problems which is also an (s, t) min cut of minimum cardinality. Then if we see that there exist an FPT algorithm exist for any of the problems which is mentioned above.

We have also discussed a randomised procedure for flow augmentation and definition of generalise coupled min cut i.e. GCMC problem which indeed has a generalisation of Coupled Min Cut into a maximal tractable problem in some sense, and an FPT algorithm for GCMC which is also a very good application of flow augmentation.

Now, let us briefly describe the readers about the flow augmentation technique.We consider an undirected graph G which two designated vertices s and t which belong to the vertex set of the graph $V(G)$. And there exist an unknown solution $Z_{s,t}$ subset of Z and these are the edges with all the endpoints reachable from both s and t in the graph $G - Z$.

Definition 1. *Special (s,t) cut is a cut where $Z_{s,t}$ has to be an (s,t) cut and also every edge of Z has endpoints in different connected components of $G - Z$. [5]*

3.1 Why is flow augmentation Necessary?

Flow augmentation is necessary as graph cut algorithms fail to apply to the coupled min cut and generalise coupled min cut problems. There were many techniques like greedy methods and shadow removal techniques. Also these methods fail in the weighted problems like Bi-Objective $(s - t)$ cut. And, then comes the turn of graph decomposition.

Theorem 3.1. *There is a randomised algorithm that, given an undirected graph $G = (V, E)$ with $s, t \in V$ and two integers $k \geq \lambda^* \geq \lambda_G(s,t)$, in time $k^{O(1)}(|V| + |E|)$ outputs an edge multiset A with $\lambda_{G+A}(s,t) \geq \lambda^*$ and a flow P in $G + A$ of cardinality λ^* such that for any (s,t)-cut Z eligible for (s,t) with $|Z| = k$ and $|Z_{s,t}| = \lambda^*$, with probability $2^{-O(k \log k)}$, the following holds : for every $uv \in A$, u and v are connected in $G - Z$; and for every path $P \in P, |E(P) \cap Z| = 1$. [5]*

Theorem 3.2. *The $GENERALIZED\ COUPLED\ MINCUT$ problem, restricted to $2K_2 - free - b - bounded$ instances, is fixed parameter tractable when parameterised by b and k. [5]*

So, we assume that the graph G is connected and we have a solution S that we are looking for in this paper has a solution cost of k. The violated clauses will be at most $\kappa \models kb$. Let us assume that solution Z has to be a $min - (s,t) - cut$ whose cardinality equals λ. For a max $(s-t) - flow\ P$, each path P of P, must

intersect with Z, precisely once and all vertices on P before (after) the edge of $P \cap Z$ is reachable from $s(fromt)$ in $G - Z$. Each path $P \in P$ going from s to t is known as $flow - path$. If we have a directed cycle of flow path, then they must be present in the same connected components of $G - Z$.

We are going to do some major simplification of instances by applying some changes in the instances. We will contract the edges in the graph who are not a part of any flow path. Then after contraction of the instance, the resulting flow path is called $tidy$. Suppose Z has k pair of edges and let $\lambda = 2k$. So, consider two paths $P, P' \in P$ and two edge pairs $(e, e'), (f, f') \in E(P) \times E(P')$. So, here one edge pair dominate the others., then no solution Z will take the other pair as this will leave the other set of edge pair reachable from s in $G - Z$. This further can be mapped to find an assignment of the vertex to either 0 or 1 with $\phi(s) = 1, \phi(t) = 0$ such that the following condition does not increase our flow. This is just like Almost 2-SAT problem. which is FPT by [9]

In the generalised couple min cut, we have our main algorithmic contribution and it is a FPT undirected graph separation problem generalised couple min cut and the input instance for the GCMC problem is an undirected multigraph G and there are two designated vertices s and t such that s and t are not the same. There is also a multiset C of pairs of vertices of $V(G) - s, t$ called henceforth pairs. And a family B of disjoint subsets of C and multi set union with $E(G)$ which is called blocks. Also, the input integer k.

Minimum $(s - t)$ cuts are very crucial for flow augmentation. Even simple graphs have exponential minimum $(s - t)$ cut. There is a notion of min $(s - t)$ cut closest to s or t. There exist a sequence of non crossing minimum $(s - t)$ cut.

Definition 2. *Blocks are the parts between the consecutive cuts. Bundles are partition of blocks into consecutive groups. The way we decompose graph G into the bundles will tell the type of edges for the flow augmenting set A in the algorithm and will tell what parts of G we have to recurse on.[5]*

Now we will discuss the flow augmentation technique in which we will see that there are many NP hard cut problems that become tractable i.e. every point becomes computable when the allowable cut size k matches the maximum $(s-t)$ cut in G. It is NP hard to find an $(s - t)$ cut in an edge weighted graph that is of minimum total weight and of cardinality at most k but it easily reduces to min cut and max flow computation if we assume that k is the minimum such

cardinality of an $(s-t)$ cut in the graph.

Sometimes it happens that the edges have to be paired up so that there are l pairs and k=2l, then we ask the question that whether maximum $(s-t)$ flow of exactly k makes the problem tractable or at least FPT.

Definition 3. *Special cut : We say that $(s-t)$ cut Z is a special cut if Z induced on s and t in an $(s-t)$ cut that is the set of edges are subset of Z with one endpoint in s size of Z i.e. the first component i.e. $R_s(Z)$ and other endpoint in $R_t(Z)$ is also an $(s-t)$ cut. Special $(s-t)$ cut also generalises the minimal $(s-t)$ cut. [5]*

Definition 4. *Eligible Cut : Z has to be special and each edge of Z has its endpoints in different connected components of $G-Z$.[5]*

Definition 5. *Compatible set : A multiset A of $\binom{V}{2}$ is compatible with a set Z subset E if for every uv belongs to A and u and v are connected in $G-Z$.[5]*

Definition 6. *Flow augmenting set is a set where for integer $\lambda^* \geq \lambda_G(s,t)$ a multi subset A of $\binom{V}{2}$ is λ^* - flow augmenting if $\lambda_{G+A}(s,t) \geq \lambda^*$. [5]*

Edges in A are undeletable and unbounded i.e. of infinite capacity A witnessing flow is a flow where we return an $(s-t)$ max flow in the augmented graph which intersects $Z_{s,t}$

If A is flow augmenting and compatible with an eligible set Z, then if we add number of copies of any edges in A, this will not violate this property. So, if we have total k+1 copies of every edge in A, then we will not be able to delete those edges for sets Z of size k. So, the endpoints of any edge would not be separated by Z.

Definition 7. *A flow f is known to be a witnessing flow iff Z is an $(s-t)$ cut in G which contains an $(s-t)$ min cut and a witnessing flow has to be a $(s-t)$ max flow P in G such that every edge of $Z_{s,t}$ occurs on a path of P and every path P intersects Z in only one edge. [5]*

Suppose we have a multi graph G and let s,t be two designated vertices such that s is not equal to t. Let Z be a subset of E be a special s-t cut of size k then let $\lambda^* = |Z_{s,t}|$ and then there exist an λ^* flow augmenting set A compatible with Z and a witnessing flow P for Z in graph $G+A$.

Definition 8. *A compatible pair is a pair of A and path P such that it is compatible with a special $(s-t)$ cut Z if A is λ^* augmenting set compatible for $\lambda^* = |Z_{s,t}|$ and P would be a witnessing flow for Z in graph $G + A$. [5]*

Here, we are given with an instance (G, s, t, k, λ^*) consisting of an undirected multi graph G such that the two designated vertices s and t are there and the integer k and λ^* is there such that $k \geq \lambda^* \geq \lambda := \lambda_G(s,t)$. We have to find a probabilistic polynomial time multiset A which is a subset of all the possible number of edges in the graph and an (s,t) flow P in $G + A$ which says that $\lambda_{G+A}(s,t) \geq \lambda^*$ and the length of the augmenting path P would be strictly equal to λ^* and for each λ^* eligible (s,t) cut Z of size exactly k will output an (A, P) is compatible with Z for probability at least p. This p is also known as success probability.

Theorem 3.3. *There exists a polynomial time algorithm and it is also randomised and it says that given a flow augmentation sampling instance (G, s, t, k, λ^*) with $\lambda_G(s,t) \leq \lambda^* \leq k$ outputs a set A with $\lambda_G(s,t) \geq \lambda^*$ and a flow P in $G + A$ of cardinality λ^*, such that the following holds : for each set Z subset of E of size k that is lambda star eligible for (G, s, t, k), the output A is compatible with Z and P is a witnessing flow for Z in $G + A$ with probability $2^{-O(k \log k)}$. The algorithm take $k^{O(1}O(m)$ time to be run and implemented. [5]*

As we have already discussed about blocks and bundles, we shall incorporate its knowledge here in the below mentioned propositions.

Proposition 1. *Each block V_i has at most λ^* connected components and each connected component in a block with i ranging from 1 to p is incident with $c \geq 1$ edges in C_{i-1} and with exactly c edges in C_i [5]*

Proposition 2. *Given a multi graph G and vertices s and t, there is a unique sequence of cuts $C_0, C_1, ..., C_p$ and decomposition of blocks $V_0, V_1, ..., V_{p+1}$ can be computed in polynomial time. [5]*

Proposition 3. *Given a multi graph G and vertices s and t, the unique sequence of cuts, $C_0^{'}, C_1^{'}, ..., C_q^{'}$ and decomposition of bundles $W_0, ..., W_{q+1}$ can be computed in polynomial time. [5]*

3.2 Algorithm for Flow augmentation

So, here the flow augmentation algorithm talks about the outer loop and inner loop, which is fed to the input instance (G, s, t, k, λ^*) and also to certain instances in recursive calls, and an inner loop which is called short separation on the latter. The outer loop talks about color coding approach to guess week as well as strongly affected stretches of bundles and then calls the inner loop subroutine on the other half.

Each recursive call to this algorithm will return a pair (A, P) for the instance in the question where the pair may or may not be compatible. If the stretch is unaffected, then it will always be compatible with Z.

The algorithm sample takes input instance (G, s, t, k, λ^*) and it says that if it does not hold that the $\lambda_G(s, t) \leq \lambda^* \leq k$ then we have to set A to be max of either of k+1 or λ^* copies of s, t, P to be any of these copies and we have to return (A, P).

Then we have to initialise A as null set and P to be a set of λ^* zero length paths starting in s. Then we have to compute the partition V = $W_0 \cup ... \cup W_{q+1}$ of G into bundles. Now either we can go in single mode or we can go in multiple mode with equal probability. In single mode $P_{blue} = P_{red} = \frac{1}{2}$ and in multiple mode $P_{blue} = \frac{1}{k}$ and $P_{red} = 1 - \frac{1}{k}$

Now, uniformly at random, color each bundle either blue with P_{blue} or red with P_{red} Now, do a recursive call of short separation on single mode and a recursive call of short separation in multiple mode and store in some variables. Now, update A and add to A all the edges which were not incident with s' or t' to A' likewise for every edge s'v belong to A' add to A a separate edge uv for each edge add sv to A. Update p as follows, for every path in P', if the first or last edge of P'.

Otherwise add to A with multiplicity k+1 all the edges with u in right side of W_{a-1} and w to the left side of W_{b+1} where w=t and $b = q + 1$. This is the outer loop algorithm.

There were some other important lemmas which are not related to the outer loop and inner loop algorithm.

There exists a randomized fixed-parameter algorithm that, given a directed multigraph G with two designated vertices s, t $\in$ V (G) and a parameter k,

samples a multiset A of arcs such that the size of a maximum (s, t)-flow in $G + A$ is strictly larger than in G and for every minimal (s, t)-cut Z of size at most k that is not a minimum (s, t)-cut, Z remains an (s, t)-cut in G + A with probability bounded from below by $1/f(k)$ for a computable function f.

4. Directed Flow Augmentation in weighted almost 2-SAT problem

In the second research paper, we talk about directed flow augmentation. It has been shown that there is a polynomial time algorithm which says that if we have a directed graph G, and two designated vertices s and t, just as in the case of first research paper, which belongs to the vertex set of the graph and an integer k, it will add randomly to our graph a number of arcs such that for every minimal st cut Z in G of size at most k, then with probability $2^{-poly(k)}$, the set Z becomes a minimum st cut in the resulting graph.

This directed flow augmentation also allows us to prove the fixed parameter tractability of a lot of problems parameterised by the cardinality of the deletion set, whose parameterised complexity status was an open issue in the community. These problems are Chain SAT and Weighted st CUT, Weighted Directed Feedback Vertex Set or Weighted Almost 2-SAT.

If we are able to prove that CHAIN SAT is FPT, then for any graph H, the LIST-H colouring problem is polynomial time solvable, then the vertex deletion problem is also FPT. Generally graph separation problems have a cut budget usually denoted by k. And sometimes, the terminal vertices like s, t are annotated. We wish to achieve separation via at most k edges or vertex deletion.

4.1 The classic s-t cut problem

We have to delete at most k edges so that there is no s-t path in the resulting graph and the feedback vertex set asks to remove at most k vertices so that the graph does not contain any cycle in it. Using graph separation, we invented several techniques like shadow removal and tree width reduction and randomised contractions. In the last research paper, we studied flow augmentation in undirected graphs, here we are motivated to study some important separators and shadow removal technique to study flow augmentation in directed graphs as well.

This also helps us to prove the FTP of the directed feedback vertex set and some other important problems like directed multiway cut, directed subset feedback vertex set, directed multicut and directed odd cycle transversal(DOCT)

All separators of size at most k between two fixed terminals s and t live in a part of the graph with treewidth bounded by $2^{O(k)}$.

Theorem 4.1. *There exists a randomised polynomial time algorithm which says that if we have a graph G and two designated vertices s and t, and an integer k, it will output a set A which is subset of $V(G) \times V(G)$ such that it should hold that for every minimal st cut Z of size at most k, with probability $2^{-O(k^4 \log k}$, Z remains an st-cut in $G + A$ and also Z is a minimum st-cut . Also, here this set Z is an st-cut if there is no path from s to t in $G - Z$ [4]*

Lemma 4.1. *Let P be an st-maxflow in Instance $I = (G, s, t, k)$ and assume $\lambda_G(s, t) > 0$. Then for every $v \in V(G)$ exactly one of the following options hold.*

- $s \in RReach(v), t \notin RReach(V)$ *and* $\delta^+_G(RReach(V))$ *is an st-mincut.*
- $s, t \in RReach(v)$, *v is on the t side of every st-mincut and* $\delta^+_G(RReach(v)) = \phi$
- $s, t \notin RReach(v)$, *t is not reachable from v in G and* $\delta^+_G(RReach(v)) = \phi$

[4]

Lemma 4.2. *Let (I,P) be an instance with a maximum flow such that I has proper boundaries and let H be the associated reachability pattern. Let $C_1, C_2, ..., C_l$ be the H-sequence of mincuts Then C1 = $\delta^+(S)$ and $l \geq 2$, that is, C2 is defined. [4]*

Lemma 4.3. *Let (I, P) be an instance with a maximum flow such that I has proper boundaries and let H be the associated reachability pattern. Let $C_1, C_2, ..., C_l$be the $H - sequence$ of mincuts. If $l \geq 3$, then H is transitive. [4]*

Lemma 4.4. *Let (I, P) be an instance with maximum flow with proper boundaries and let H be its reachability pattern. Assume $|V(H)| = |E(H)|$. For every $i \in [\lambda], v \in V(P_i) \backslash t$, and a bottleneck edge e on P_i, if e lies before v on P_i, then any path from s to v in G visits e. Consequently, any set consisting of one bottleneck edge from each path P_i is an st-mincut in G. [4]*

Now, we can say that $WEIGHTED$ s-t-CUT can be solved in randomised time $2^{O(k^4 \log k)} n^{O(1)}$. This is in addition to the monte carlo stimulation, i.e. we run these many times, this algorithm, so that we can say that our problem of weighted s-t cut can be solved easily. We will take instance of bundled cut which consists of a directed graph G, two designated vertices s and t and a non negative integer k and a family B of pairwise disjoint subsets of edge set of the graph G. These elements of B are called as $bundles$ and the cut in this bundled cut instance $I = (G, s, t, k, B)$ is an st-cut Z with Z subset of all the bundles.

Weighted bundled cut with order is randomised FPT when it is parameterised by k and maximum size of the bundle. So, here in addition to our FPT parameter k, we used some other parameter to parameterise it successfully.

Also, $l - CHAIN\ SAT$ is randomised FPT when it is parameterised by l and k, even in the weighted setting. Also, weighted directed feedback vertex set parameterised by the cardinality of deletion set is also randomised FPT. We should know that, before starting off that all our graphs can be multigraphs and edges may have capacity of 1 or $+\infty$

In paper 2, we saw directed flow augmentation where we had directed graphs in our exposure. We shall study the flow augmentation in directed graphs. There is a poly time Algorithm that if we have a graph G and two integers s, t and integer k, we will add to G the number of $arcs(A)$, s.t. for every minimal s-t cut in G of size at most$k$, with probability of $2^{-poly(k)}$, the set Z becomes the minimum s-t cut in the resulting graph $G + A$.

This helps us to prove FPT of a number of problems parameterised by cardinality of the deletion set. We will discuss $ChainSAT$ and Weighted s-t cut, Weighted directed feedback vertex set, weighted almost 2-SAT

If we are able to prove that CHAIN SAT is FPT, then for any graph H, the

LIST-H coloring problem is polynomial time solvable, then the vertex deletion problem is also FPT.

Generally graph separation problems have a cut budget usually denoted by k. And sometimes, the terminal vertices like s, t are annotated. We wish to achieve separation via at most k edges or vertex deletion.

The classic s-t cut problem

We have to delete at most k edges so that there is no s-t path in the resulting graph and the feedback vertex set asks to remove at most k vertices so that the graph does not contain any cycle in it.

Using graph separation, we invented several techniques like shadow removal and tree width reduction and randomized contractions. In the last research paper, we studied flow augmentation in undirected graphs.

Here we are motivated to study some important separators and shadow removal techniques to study flow augmentation in directed graphs as well.

This also helps us to prove the FTP of the directed feedback vertex set and some other important problems like directed multiway cut, directed subset feedback vertex set, directed multicut and directed odd cycle transversal(DOCT).

All separators of size at most k between two fixed terminals s and t live in a part of the graph with treewidth bounded by $2^{O(k)}$

Theorem 4.2. *If we have a directed graph G and there are two designated vertices s and t and an integer k, then it outputs an arc set which is a proper subset of $V(G) \times V(G)$ such that every minimal cut $Z \subseteq E(G)$ remains an s-t cut in Graph $G + A$ with success probability $2^{-O(k^4 \log k)}$.*

Also Z remains a minimal s-t cut in Graph G Z is the minimum s-t cut in the graph. There is no path from s to t in $G - Z$ hence Z is the minimum s-t cut. A is added with infinity capacity arcs. With the monte-carlo stimulation in place, the Weighted s-t cut problem can be solved in $2^{O(k^4 \log k)} \times n^{O(1)}$ [4]

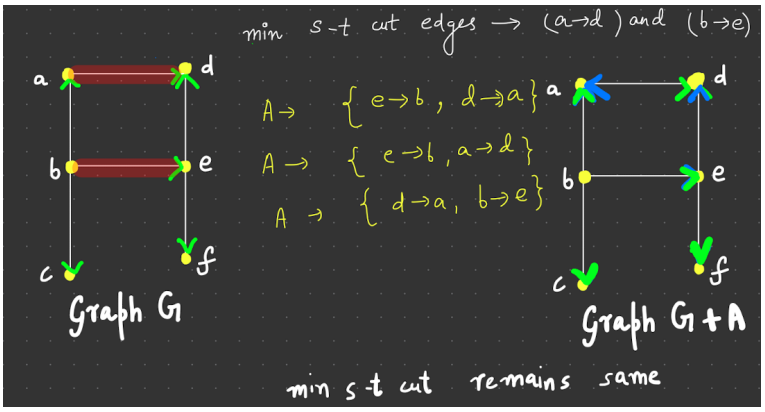
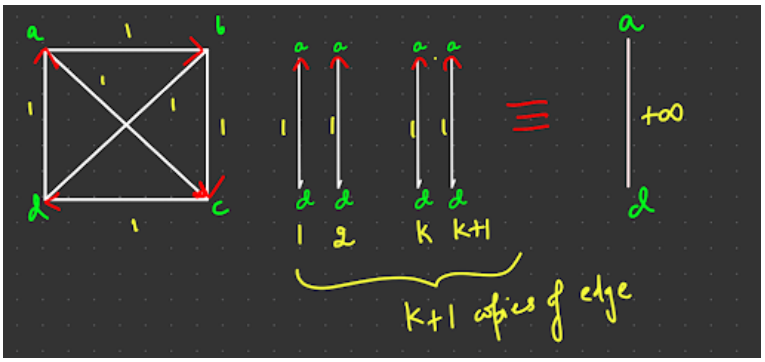


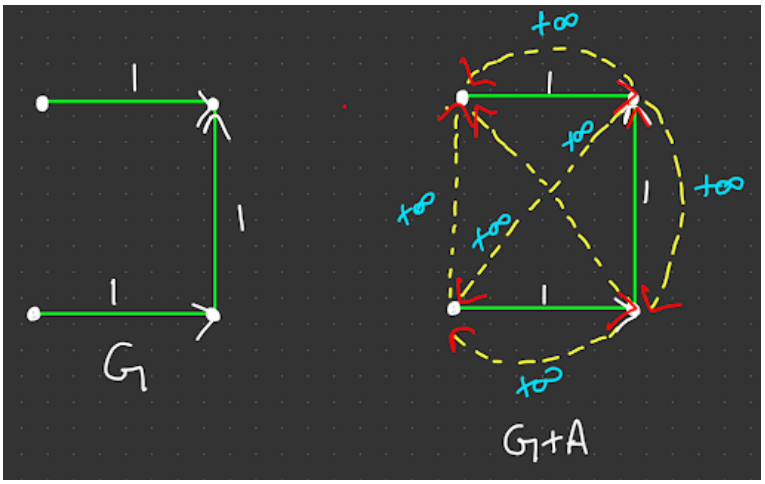
Figure 1: Theorem 1



In the directed graph case, as per research paper 1, we can say that we can have a case that our graphs can be multigraphs as well. All the edges can be given a capacity of either 1 or $+\infty$.

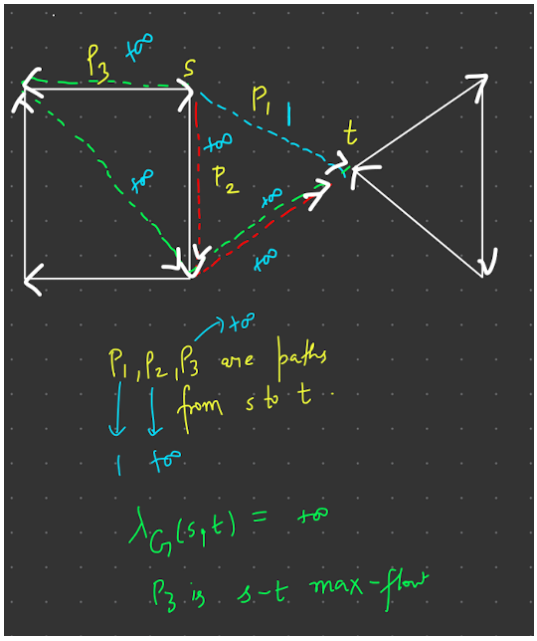
Initially all the edges of the input graph have capacity of 1. If we take $k+1$ such flows of capacity 1 , that would be equivalent of one flow of capacity $+\infty$.

If we have a graph G with arc $A \subseteq V(G) \times V(G)$, then G+A is a graph with every arc of A added to G with capacity of $+\infty$. For the designated vertices s and $t \in V(G)$



4.2 Directed flow augmentation technique

Lemma 4.5. *If A is compatible with Z, then Z will also be a star s-t cut in $G+A$ There is no guarantee that A is compatible with Z with good probability and core Z in graph $G+A$ is an s-t min cut in graph $G+A$. Also P^- is a witnessing flow Usage of theorem 4.1 applies on even the cuts Z with $core_{G+}A(Z)$ being an s-t min cut. We may be required to add some of the edges to G, so that G+A is also a witnessing flow. [4]*



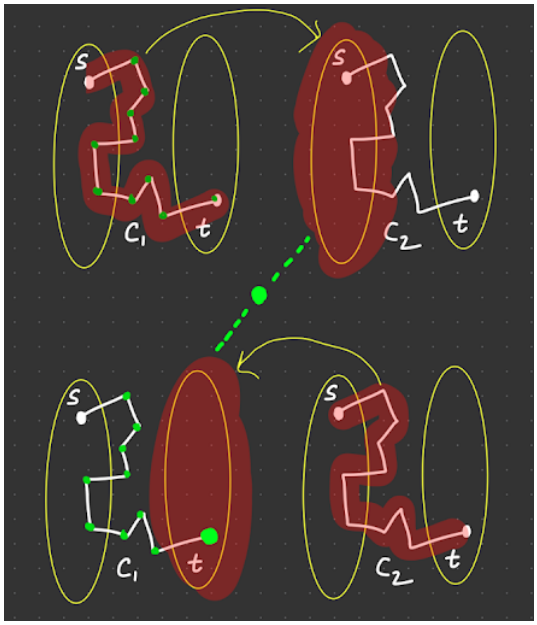
A set Z subset of $E(G)$ is an st-cut if it contains no edge of capacity $+\infty$ and there is no path from s to t in $G - Z$.

An st-cut Z is minimal if no proper subset of Z is an st-cut and minimum (or

st-mincut) if it has minimum possible cardinality.

By Menger's theorem, if $\lambda_{G(s,t)} < +\infty$ then the size of every st-mincut is exactly $\lambda_{G(s,t)}$ and there are no st-cuts if $\lambda_{G(s,t)} = +\infty$ For an st-cut Z, the s-side of Z is the set of vertices reachable from s in $G - Z$, and the t-side of Z is the complement of the s-side.

Note that this is not symmetric as we do not mandate that t is reachable from all elements of the t-side in $G - Z$. If we have two (inclusion-wise) minimal st-cuts C_1 and C_2 such that both endpoints of every edge of C_1 lie in the s-side of C_2 (and thus both endpoints of every edge of C_2 lie in the t-side of C_1), then a vertex is between C_1 and C_2 if it is both in the t-side of C_1 and s-side of C_2. One flow-augmentation step,If we do not pay particular attention to the exact bound on the probability of a success and we are happy with any $\frac{1}{f(k)}$ bound for a computable function f.



4.3 Tractable case of Weighted Bundled Cut

In this, we have a directed graph G, with designated vertices s and t in the vertex set and we have a non negative integer k and we have a family B of pairwise disjoint subsets of $E(G)$. The element B $\in B$ is called a bundle.

An edge that is part of bundle can be deletable, otherwise it is undeletable. A cut in this bundled cut instance $I = (G, s, t, k, B)$ is an st-cut Z such that does not

contain any undeletable edge, $Z \subseteq \cup B$.

A cut Z will touch the bundle if the intersection between Z and B is not ϕ . When we parameterise this by k, then it is known as bundled cut and it is $W[1]$ hard even if all of these bundles are of size 2.

Lemma 4.6. *Let (I, P) be an instance with maximum flow with proper boundaries and H be its reachability pattern. Assume $|V(H)| = |E(H)|$ and let Z be a star $(s-t)$-cut in G. If $|Core(Z)| \geq \lambda_G(s,t)$ then there existsi $\in [\lambda]$ such that no edge of $Z \cap E(P_i)$ is a bottleneck edge. Otherwise if $core(Z) = \lambda_G(s,t)$, then no edge of $Z - core(Z)$ is a bottleneck edge of Z. [4]*

Theorem 4.3. *Weighted bundled cut with order is radomised FPT when parameterised by k and maximum size of a bundle. Suppose you have a solution Z to the input $WEIGHTED - BUNDLED - CUT$ instance $I = ((G, s, t, k, B), \omega, W)$ [4]*

Proof. We can prove lemma 5.1 using lemma 4.7. So, consider b is your maximum size of your bundle. So, there is a randomised algorithm that will running n time FPT in b and k, and if you have YES instance, answers yes with probability $2^{-O((bk)^4 \log(bk))}$ and will never answers YES to a NO-instance and if we repeat the algorithm $2^{O((bk)^4 \log(bk))}$ times gives the desired algorithm. [4] □

Lemma 4.7. *Assume we are given a weighted bundle cut with order instance $I = ((G, s, t, k, B), w, W)$. Then one can in time FPT with parameters $\lambda_G(s,t)$, maximum size of the bundle, and k check if there is solution to I that is an st-mincut, where for $\lambda_G(s,t)$ we treat every deletable edge of capacity 1 and undeletable edge as capacity $+\infty$ [4]*

Proof. For every $B \in B$, fix an order $(e_1^B, ..., e_{|B|}^B)$ which is the definition of the weighted bundled cut. It is given that b is the maximum size of the bundle. We have to find the maximum $(s-t)$-flow. $P = P_1, P_2, ..., P_\lambda$ where $\lambda \models \lambda_G(s,t)$. Let us fix one hypothetical solution Z such that there exists only one deletable edge on every path P_i also let us denote f_i as the unique edges of $E(P_i) \cap Z$. Then apply branching. At first we will guess $\kappa \leq k$ bundles which are touching Z. Let those k bundles be in B namely $F_1, F_2, ..., F_k \in B$. For every $i \in [\lambda]$, guess the indices $\alpha(i) \in [\kappa], \beta(i) \in [b]$.Start guessing the partitions of B into $B_1, ..., B_k$. If every $\alpha(i) = j$ and edge does not lie on P_i, then remove B from B_j. F_j remains in B_j, where we have correct guesses in the branch. Now, in the main step iterate over all the indices where all the $\alpha_i's$ are same but $\beta_{i1} < \beta_{i2}$ and so

on.So, by the property of the $weighted\ BUNDLED\ CUT$,$WITH\ ORDER$ instance, G contains a path Q that uses only edges which are undeletable in nature. Then filtering step is performed exhaustively. If,$\lambda \neq \lambda(G)_{s,t}$, then we terminate the branch.

Definition 1. *An edge is said to be vulnerable if* $e = e^{B}_{\beta(i)}$ *for some* $i \in [\lambda]$ *and* $B \in B_{\alpha(i)}$. *[4]*

Now, we construct the auxiliary weighted graph. We start with only s and t. For every $j \in [1, \kappa]$ add a path P^H_j to the graph from s to t with $|B_j|$ edges. Graph will contain path from u_1 to u_2 containing only non-vulnerable edges. Add an edge of weight $+\infty$ to H. We will see that Z' is an $(s-t)$-cut in H. Also in the reverse direction Y' is an $(s-t)$-min cut in H, then

$$Y = \left\{ e^{B_{\alpha(i),a\alpha(i)}}_{\beta(i)} | i \in [\lambda] \right.$$

is a solution to I. Hence, it suffices to find in H an st-cut of cardinality κ and minimum possible weight. Since $\kappa \leq \lambda_H(s,t)$, this can be done in polynomial time.

□

Theorem 4.4. *Weighted CUT is randomised FPT when parameterised by k. [4]*

Theorem 4.5. *Weighted b-CHAIN SAT is randomised FPT when parameterised by* k *and* b. *[4]*

Theorem 4.6. *Given a Weighted Skew Multicut instance* $I = (G, (s_i, t_i)^b_{i=1}, \omega, k, W)$, *one can in polynomial time can construct an equivalent WEIGHTED BUNDLED CUT with order instance* $I' = (G', B, \omega, k, W)$ *with the same k and W and bundles of size b each. [4]*

Theorem 4.7. *Weighted DFAS and Weighted DFVS are randomized FPT when parameterized by k. [4]*

Weighted Almost 2 SAT is also one another application for directed flow augmentation. It is randomised FPT which is also parameterised by k with running time of $2^{k^{O(1)}} n^{O(1)}$

5. FPT of DMC-3 parameterized by the size of the cutset

In order to understand the fixed parameter tractability of directed multi cut with three pairs, we should have a sound knowledge of what exactly a directed multicut is. So, according to [10], a directed multicut is a set of arcs $M \subseteq A$ such that from any directed path P_i from some s_i to its corresponding t_i, $P \cap M = \phi$

So, one may observe that flow augmentation speaks about edge cut while DIRECTED MULTICUT asks for a vertex cut. Talking about directed multi cuts, with three terminal pairs, we can say that flow augmentation can be used in the following way:

Let S be an inclusion wise minimal solution to the input instance $(G, k(s_i, t_i)_{i=1,2,3})$. Now, S_i is minimal vertex $s_i t_i$ cut(st-mincut). Hence we can apply flow augmentation separately to all these three, and we receive augmentation edges A_1, A_2 and A_3

With good probability, we get three minimum vertex st-cuts in $G + A_i, i \in [1, 3]$. The difficulty of directed multicut lies in the fact that we may not get S_i which are pairwise disjoint.

In CSP world, it means that for some $i, j \in 1, 2, 3$ and $P \in P_i$ and $Q \in Q_j$, the variables $x_i(P)$ and $x_j(Q)$ describe the same vertex of S. There are $2^{O(k \log k)}$ options of what pair of vertices describe the same vertex, we can guess the set of k pairs exhaustively.

It is important to see here that even though we assume that the flow augmentation steps are successful and in the branching step, a correct choice is made of which variables describe the same vertex, the final CSP instance is the reformulation of the original Directed Multicut instance which means that no matter what choice we make, every solution to the obtained CSP instance will give a set of non terminal vertices that cuts all paths from s_i to t_i for all $i = 1, 2, 3$.

So, given a directed multi cut instance $(G, k, (s_i, t_i)_{i=1,2,3})$, we have to find a solution if there exists a shadowless solution i.e. it does not matter if our algorithm will fail, but it should fail iff it is not shadowless.

The problem to be discussed in this paper is to prove that Three Terminal Pair DIRECTED MULTICUT is fixed-parameter tractable. Here is the theorem for above.

Theorem 5.1. *Directed Multicut with three terminal pairs is fixed- tractable when parameterized by the size of the cutset (with a randomized algorithm). [6]*

To prove this theorem, we need to follow four steps which we will see in the upcoming sections.

Theorem 5.2. *WEIGHTED DIRECTED MULTICUT, parameterised by the cardinality of the cut set is W[1]-hard even with two terminal pairs. [6]*

So, we do not have an FPT for two pair Terminal vertices DMC, but we will see how we are able to generate FPT for 3-DMC.

Theorem 5.3. *Given an instance $(G, k, (s_i, t_i)_{i\in[3]}, V^\infty)$ of $3 - DMC$, there is an algorithm that runs in $2^{O(k^2 \log k)} n^{O(1)}$,and outputs a family Z, of subsets of $V(G) - V^\infty$ such that $|Z| = 2^{O(k^2 \log k)} \log^2(n)$ and if the input instance is a YES-instance, then there exists $Z \in Z$ such that $(G, k, (s_i, t_i)_{i\in[3]}, V^\infty) - Z$ is a YES instance that admits a shadowless solution of cardinality at most k, Where G' is the result of bypassing Z in G. [6]*

Theorem 5.4. *There is a computable function f: $N \to N$ such that the following hold. Let G be a graph*

- *For any total order $\prec$ of $V(G)$, if $gr(Adj_\prec(G)) \leq k$, then $tww(G) \leq f(k)$*
- *If $tww(G) \leq k$, then there is a total order $\prec$ of $V(G)$ such that gr($Adj_\prec(G)$) $\leq f(k)$.*

[6]

Theorem 5.5. *There are two computable functions $c : N \times N \to N$ and $q : N \times N \to N$ such that the following holds : There exists a polynomial-time randomized algorithm that, given a directed graph G (with possibly some vertices marked as undeletable), vertices s, $t \in V(G)$, and an integer k, returns*

an arc set $A \subseteq V(G) \times V(G)$ and an st-maxflow P' in $G + A$ such that for every minimal st-separator Z of size at most k, with probability $2^{-O(k^4 \log k)}$, the tuple (A, P') is compatible with Z. Additionally, the algorithm returns a partition B of the deletable vertices of $\cup_{P \in P'}$V(P) into at most c(k) sets such that for every P ∈ P', every integer p ∈ N, every B ∈ B and every two disjoint sets C, D of size at least $q(k, p)$, consisting of vertices of $B \cap V(P)$ that are interlaced on P, the graph G contains a family of p pairwise vertex-disjoint CD-soybeans. Finally, one can take c and q such that $c(k) = 2^{O(k^3 \log k)}$ and $q(k, p) = 2^{O(k^3 \log(kp))}$. [6]

Theorem 5.6. *Twin-Width -w Permutation CSP parameterised by the number of constraints plus w is fixed parameter tractable. [6]*

Proof. We take instance I of tww-Permutation-CSP and transform it into ordered,vertex and edge colored graph G and we note that it's twin width depends on w and all the constraints in I. In the proposed research paper, the researchers have provided an FO-formula [11] ϕ $s.t. G \models \phi$ iff I is satisfiable. This FO model checking algorithm is then applied onto ordered graphs running in Fixed polynomial tractable time parameterised by the twin width of the input graph and length of the formula. [[12],[13]].

$x_1, x_2, ..., x_k$ are the variables of I and set of downward closed constrains and set of permutation constraints by R and Π respectively. All the i's belong to the closed set $[k]$. Elements of D_i are denoted by $d^i_1, d^i_2, ..., d^i_{n_i}$. A set V_i of n_i vertices have been introduced to G having color i and they are denoted by $v^i_1, ..., v^i_{n_i}$. So, we have to verify whether the vertex assigned to y has color i. There is an existential guess associated to it and it is denoted by
$\phi_\exists \equiv \exists_{y1} ... \exists_{yk} \wedge_{i \in [k]} col(y_i) = i$.

Encoding of the constraints can be done in (G, ϕ). So, few claims can be made in this regard

Claim 1. *I is satisfiable if and only if $G \models \phi$ [6]*

Claim 2. *Let (V,$\preceq$) be an ordered set of vertices and let X_i be disjoint consecutive subsets of V . Let G be a graph on the vertex set V that only has edges between x_i and X_j. For each $k \geq 1$, if $gr(Adj_\preceq(G) \geq k + 2)$, then $gr(Adj_\preceq(G)[X_i, X_j]) \geq k$. [6]*

Claim 3. *Let $R \in R$ be a downwards-closed constraint, let $E_R \subseteq E(G)$ be the set of edges colored R in G and let $G_R = (V(G), E_R)$. Then $gr(Adj_\preceq(G)[R]) \leq 3$. [6]*

Claim 4. *There is a computable function* $h : N \times N \to N$*, such that* $tww(G) \leq h(w, |R \cup \Pi|)$*. [6]*

□

Now, let us take some formal understanding of twin width [13]. Every graph here in this paper is a trigraph [14] in which we contract or merge two vertices into single vertex w and then color the edges incident to the new vertex w. The edge wz remains black, if uz and vz were previously black edges. Let us also briefly discuss about the notion of contraction sequence. It is a sequence of n vertices trigraphs $G = G_n, ..., G_1 = K_1$ such that G_i is obtained from G_{i+1} by contracting two of the vertices. A $d - sequence$ is a contraction sequence in which every trigraph in it have degree at least d. Twin width of the graph is denoted as $tww(G)$ is the minimum such integer d such that G admits a $d - sequence$.

In order to further understand how the FPT of the 3-DMC problem is parameterised by the size of the cut set, we need to understand what are permutation CSP [15]. So, in permutation CSP, V is the set of variables and C is the set of constraints in which tuple of elements of V are the constraints. We have to find total ordering of the variables which satisfies as many constraints as possible.

Theorem 5.1. *Directed Multicut with three terminal pairs is fixed-parameter tractable when parameterized by the size of the cutset (with a randomized algorithm). [6]*

We need to show how to reduce $3 - DMC$ to $PERMUTATION - CSP$ so that the twin width of each constraint is bounded by some function of favourite separator k.

We have to heavily rely on flow augmentation for the proof. We have already discussed in theorem 5.4 how to apply the flow augmentation technique, so we have to take 3 terminal pairs and them apply flow augmentation to each of the terminal pairs.

It will give one one augmented graph each, three of a kind and also, we will get value of k for each terminal pair. If there is a solution S, we will have solution preserved with large probability as mentioned above.

We will have selection of vertices as the solution each for one flow path, three of

a kind. Now, if we need to achieve a reduction to the permutation CSP problem, then we need to introduce one variable x for each flow P .

We get an instance of $PERMUTATION - CSP$ with $O(k^2$ constraints. Permutation CSP is $W - 1$ hard in general. So, for remainder section, we will fix our instance to be $(G, k, (s_i, t_i)_{i\in[3]}, k)$ of $3 - DMC$.

5.1 Shadow Removal technique

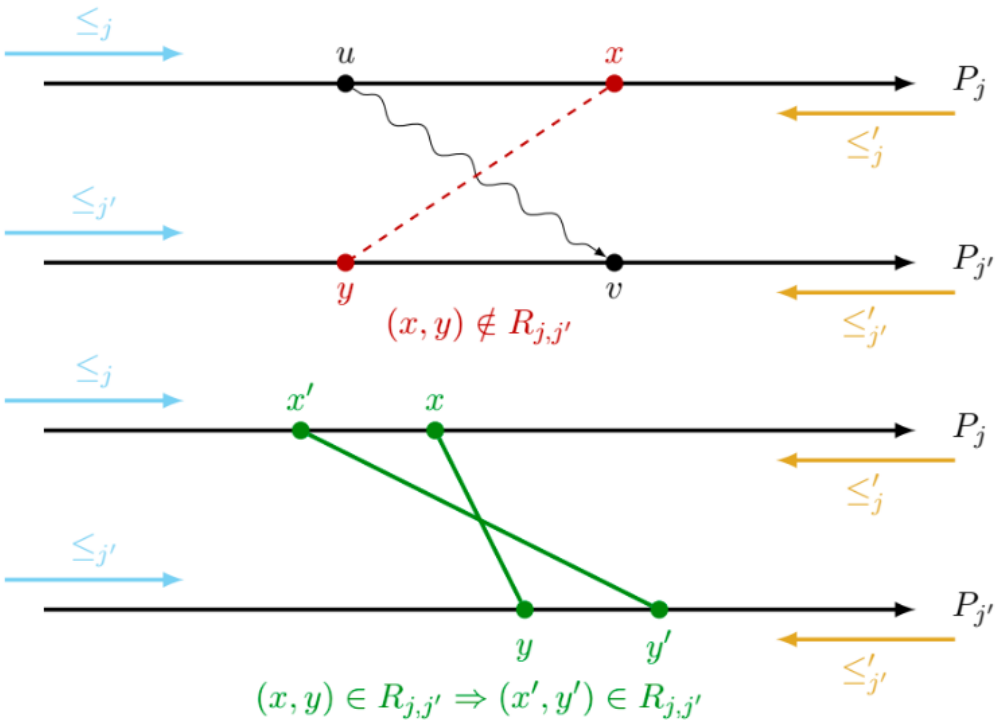
Let us also see how to solve this FPT with large enough probability. We have partitioned the proof of algorithm in four main parts, as you can see below. First step to our proof is the shadow removal technique which we have already discussed in the introduction section itself.

Second step is using flow augmentation and reducing to permutation CSP. Third step is reducing the twin width of the constraints in the permutation CSP instance. Fourth step is solving permutation CSP instance.

Step 1 is Theorem 3.3 We will apply theorem 5.3 and we will get family Z. We will do iteration on all the instances of $3 - DMC$ and this would result in bypassing a set Z in the graph G. Our aim is to find a shadowless solution.

Now, we will see how we use flow augmentation and its reduction to Permutation CSP. We will run algorithm from Theorem 5.5 and inputs will be $s = s_i$ and $t = t_i$, the graph G and k.

All we have in our hands is a triple (A_i, P_i, B_i) consisting of an arc set $A_i \subseteq V(G) \times V(G)$ and an s_it_i maxflow in $G + A_i$ and partition B_i of deletable arcs on P_i.



Lemma 5.1. *Let S be an arbitrary fixed solution to $(G, k, (s_i, t_i)_{i\in[3]}, V^\infty)$. Then with probability at least $2^{-O(k^4 \log k)}$ we have safely augmented wrt. S. [6]*

Lemma 5.2. *Assume we have safely augmented and let $S_i, i \in [3]$ be witnesses to that fact, then for each $i \in [3]$ and for each $j \in [3]$, we have $|V(P^i_j) \cap S_i| = 1$. Then we define mapping ϕ by defining $\phi(x^i_j)$ and $\phi(x'^i_j)$ both as a single vertex in in $V(P^i_j) \cap S_i$ for each $i \in [3]$ and $j \in k_i$. Then ϕ is the solution to PERMUTATION CSP instance C_1. [6]*

Lemma 5.3. *If ϕ is a solution to the PERMUTATION CSP instance C_1 then for each $i \in [3]$ the set $\phi(x^i_j)|j \in [k_i]$ is an s_it_i- separator in G. [6]*

Lemma 5.4. *Let S be a solution to $(G, k, (s_i, t_i)_{i\in[3]}, V^\infty)$. and assume we have safely augmented with witnesses S_i. Then one of the partitions considered in the consistency iterations complies. [6]*

Lemma 5.5. *Let S be a solution to $(G, k, (s_i, t_i)_{i\in[3]}, V^\infty)$., assume we have safely augmented with witnesses S_i and that X complies, Then we have $|V(P^i_j) \cap S_i| = 1$. for each $i \in [3]$ and for each $j \in k_i$. define mapping ϕ by defining $\phi(x^i_j)$ and $\phi(x'^i_j)$ both as a single vertex in in $V(P^i_j) \cap S_i$ for each $i \in [3]$ and $j \in k_i$. Then ϕ is the solution to PERMUTATION CSP instance C_2. [6]*

Lemma 5.6. *If ϕ is the solution to PERMUTATION CSP instance C_2, then the set $\phi(x^i_j)|i \in [3], j \in [k_i]$ is a solution to $(G, k, (s_i, t_i)_{i\in[3]}, V^\infty)$. [6]*

Lemma 5.7. *For each $i \in [3]$ and $j \in k_i$ we have $gr(Adj(p^i_j)) \leq 1$. [6]*

Lemma 5.8. *There exists a computable function $h : N \to N$ and an algorithm that given an instance $(G, k, (s_i, t_i)_{i\in[3]}, V^\infty)$, the constraint $\pi^{i,i'}_{j,j'}$ in C_2 for $i, i' \in [3]$ distinct, $j \in [k_i]$, and $j' \in [k_{i'}]$, and the augmented paths $P^i_j, P^{i'}_{j'}$ together with the partitions $B_i, B_{i'}$ certifies that $gr(Adj(p^i_j)) \leq h(k)$ or finds a vertex $v \in D^i_j \cap D^{i'}_{j'}$ such that there is no shadowless solution S with $v \in S$ and all vertices before v on P^i_j do not reach t_i and all vertices after v on P^i_j are not reachable from s_i in $G - S$;all vertices before v on $P^{i'}_{j'}$ do not reach $t_{i'}$ and all vertices after v on $P^{i'}_{j'}$ are not reachable from $s_{i'}$ in $G - S$; and the algorithm runs in fixed parameter time w.r.t. k. [6]*

5.2 Directed Multicut for Three Terminal Pairs

Theorem 5.1. *Directed Multicut with three terminal pairs is fixed-parameter tractable when parameterized by the size of the cutset (with a randomized algorithm). [6]*

Proof. We claim that the algorithm described in this section solves $(G, k, (s_i, t_i)_{i\in[3]}, V^\infty)$in fixed-parameter time with respect to k with large-enough probability. By the arguments given throughout the section, the algorithm indeed runs in fixed-parameter time. If the algorithm returns a vertex set S, then S is a solution to $(G, k, (s_i, t_i)_{i\in[3]}, V^\infty)$ by Lemma 5.3. Assume that there is a solution S to $(G, k, (s_i, t_i)_{i\in[3]}, V^\infty)$. So, by Theorem 5.3, we can assume that S is shadowless. By Lemma 5.1, with at least probability $2^{-O(k^4 \log k)}$, the algorithm is safely augmented with respect to S. So, by Lemma 5.5, X complies with S. Thus by lemma 5.3, The PERMUTATION CSP instance C_2 has a solution. By Lemma 5.8, C_2 maintains having a solution even after removing the vertices computed in Lemma 5.8 from the respective domains. Algorithm of theorem 5.6, has a solution. □

Bibliography

[1] M. Cygan, D. Lokshtanov, M. Pilipczuk, M. Pilipczuk, and S. Saurabh, "Minimum bisection is fixed parameter tractable," in *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, 2014, pp. 323–332.

[2] N. Bousquet, J. Daligault, and S. Thomassé, "Multicut is fpt," in *Proceedings of the forty-third annual ACM symposium on Theory of computing*, 2011, pp. 459–468.

[3] D. Marx and I. Razgon, "Fixed-parameter tractability of multicut parameterized by the size of the cutset," in *Proceedings of the forty-third annual ACM symposium on Theory of computing*, 2011, pp. 469–478.

[4] E. J. Kim, S. Kratsch, M. Pilipczuk, and M. Wahlström, "Directed flow-augmentation," in *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, 2022, pp. 938–947.

[5] ——, "Solving hard cut problems via flow-augmentation," in *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*. SIAM, 2021, pp. 149–168.

[6] M. Hatzel, L. Jaffke, P. T. Lima, T. Masařík, M. Pilipczuk, R. Sharma, and M. Sorge, "Fixed-parameter tractability of directed multicut with three terminal pairs parameterized by the size of the cutset: twin-width meets flow-augmentation," *arXiv preprint arXiv:2207.07425*, 2022.

[7] R. Chitnis, M. Cygan, M. Hajiaghayi, and D. Marx, "Directed subset feedback vertex set is fixed-parameter tractable," *ACM Transactions on Algorithms (TALG)*, vol. 11, no. 4, pp. 1–28, 2015.

[8] R. Chitnis, M. Hajiaghayi, and D. Marx, "Fixed-parameter tractability of directed multiway cut parameterized by the size of the cutset," *SIAM Journal on Computing*, vol. 42, no. 4, pp. 1674–1696, 2013.

[9] I. Razgon and B. O'Sullivan, "Almost 2-sat is fixed-parameter tractable," *Journal of computer and system sciences*, vol. 75, no. 8, pp. 435–450, 2009.

[10] A. Gupta, "Improved results for directed multicut," in *SODA*, vol. 3, 2003, pp. 454–455.

[11] A. Dawar and S. Kreutzer, "Parameterized complexity of first-order logic correction," 2013.

[12] É. Bonnet, U. Giocanti, P. Ossona de Mendez, P. Simon, S. Thomassé, and S. Toruńczyk, "Twin-width iv: ordered graphs and matrices," in *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, 2022, pp. 924–937.

[13] É. Bonnet, E. J. Kim, S. Thomassé, and R. Watrigant, "Twin-width i: tractable fo model checking," *ACM Journal of the ACM (JACM)*, vol. 69, no. 1, pp. 1–46, 2021.

[14] J. A. Bondy, "Trigraphs," in *Annals of Discrete Mathematics*. Elsevier, 1989, vol. 43, pp. 69–79.

[15] E. J. Kim and D. Gonçalves, "On exact algorithms for the permutation csp," *Theoretical Computer Science*, vol. 511, pp. 109–116, 2013.