Blocking problems in combinatorial optimization

K120254

Final report

The project examined various types of blocking problems in combinatorial optimization, as well as related optimization problems, and made considerable progress in several of these. In order to describe the results, we group them into four topics: "Generalizations of the mininum cut problem", "Blocking paths, arborescences, and trees", "Matroid intersection and related problems", and "Popular matchings and branchings". Most of the hardness results presented here are true under the assumption that the Unique Games Conjecture is true. We will use the abbreviation *UGC* in these cases.

1 Generalizations of the minimum cut problem

1.1 Undirected graphs

In undirected graphs, the most studied generalization of the minimum cut problem is minimum multiway cut, where terminal nodes and edge weights are given in the input, and the aim is to find a partition of the node set such that each class contains exactly 1 terminal node, and the total weight of edges going between classes is minimized. It is known that the problem is APX-hard, and the integrality gap of the so-called CKR relaxation (an LP relaxation named after the authors Călinescu, Karloff and Rabani) is a lower bound on the approximation factor if we assume UGC. There had been a sequence of papers establishing lower bounds on the integrality gap, with the 1.2 of Angelidakis, Makarychev and Manurangsi [1] seen as a possible candidate for the true value. In [3], we ruled out this option by showing an integrality gap strictly larger than 1.2. The main novelty of our approach is to use and extend the technique of Mirzakhani and Vondrák [15] to analyze minimum multiway cuts in higher-dimensional simplices. This technique has the potential to lead to even further improvements.

We also studied a different generalization of minimum cut, called the *$s-t$-separating k-cut* problem for fixed k [2, 5]. In this problem, we are given an edge-weighted undirected graph with specified terminal nodes s and t, and the aim is to remove a minimum weight edge set so that the resulting graph has at least k components with s and t being in different components. This is an intermediate problem between the polynomial-time solvable k-cut problem for fixed k (where no terminal nodes are given) and the NP-hard 3-terminal cut problem (where 3 terminals are given). In [2], we gave a polynomial-time algorithm for this problem.

Vertex cut problems in undirected graphs have a rich history too, but –somewhat surprisingly– the complexity of vertex k-cut for fixed k (where no terminals are given) has been open. In [2], we have shown that the node-weighted 3-cut problem is NP-complete, and is not approximable to a factor better than 4/3 assuming UGC (a 4/3-approximation can be obtained by guessing 3 terminals and using the known 4/3-approximation algorithm for 3-terminal vertex cut).

1.2 Directed graphs

Directed minimum cut problems are generally more difficult than their undirected counterparts: while the undirected $s-t$-separating k-cut problem for fixed k is polynomial-time solvable, even the following seemingly simpler $s-t$-bicut problem is NP-hard in directed graphs: for given terminal nodes s and t, remove a smallest subset of edges such that there is no $s-t$ path and no $t-s$ path in the remaining graph. Furthermore, it has been shown by Chekuri, Madan and Lee that the problem cannot be approximated by a factor less than 2 assuming the unique games conjecture (this result settled a question posed in the original research plan of this project). In [5], we showed that the global variant of the problem (where s and t are not given in the input) can be approximated to a factor strictly less than 2.

Another way to define multiway cut problems in directed graphs is to consider *linear cut* problems, where an ordering is given on the terminal nodes. This class of problems has applications in the theory of network security and network coding. In the linear (s, r, t)-cut problem, we want to remove edges of minimum total weight such that no $s-t$, $s-r$, and $r-t$ path remains. This problem has a simple 2-approximation algorithm, while a lower bound of $4/3$ on approximability under UGC had been known. In [4], we managed to completely close the gap, giving a $\sqrt{2}$-approximation algorithm, while showing an integrality gap example of $\sqrt{2}$, which implies that no approximation of factor better than $\sqrt{2}$ is possible assuming UGC. This is one of the few known cases where an LP integrality gap leads to an irrational approximation factor. The result also implies a lower bound of $\sqrt{2}$ on the approximability of $(s, *)$-bicut, which is the version of bicut where only one of the terminals is given, i.e., we want to remove edges such that there is a node t with no $s-t$ path and no $t-s$ path in the remaining digraph.

2 Blocking paths, arborescences and trees

A spanning arborescence in a directed graph is a rooted directed spanning tree, where there is a unique path to each node from the root. In the following, we will call this an out-arborescence, and in-arborescence will be used for a spanning tree where there is a unique path from each node to the root. Blocking problems for arborescences have several variations based on which types are to be blocked and what kind of edge or node weights are given. The problem of blocking all in-arborescences (or that of blocking all out-arborescences, by reversing the edges) is a well-known polynomial-time solvable problem.

An interesting NP-complete variant is the problem where a root r and node weights are given, and we want to remove nodes of minimum total weight such that no in-arborescence and no out-arborescence exists. We show in [4] that this problem is approximation-equivalent to (s, r, t)-linear cut, so there is a polynomial-time $\sqrt{2}$-approximation algorithm, and this is best possible assuming UGC. The same results hold for the edge-weighted variant.

Another variant is the *node double cut* problem, where no root is fixed; the aim is to remove nodes of minimum weight so that no out-arborescence exists in the remaining digraph (from any root). This problem relates to network security, since the minimum number of node failures preventing network consensus coincides with the minimum node double cut. In [2], we have shown that node double cut has a 2-approximation algorithm, and has no $(3/2-\epsilon)$-approximation for any positive ϵ, assuming UGC. The problem has a fixed-terminal variant called (s,t)-double cut, where the aim is to remove nodes such that there is no node in the remaining graph that can reach both s and t. Node double cut can be solved by solving (s,t)-double cut for every pair s,t. For this problem, we have been able to provide a tight approximation bound of 2 in [2] (again, inapproximability holds assuming UGC). Note that here the edge-weighted variants are solvable in polynomial time using network flows.

As mentioned before, given a root r, blocking all out-arborescences using edges of minimum total weight is also solvable in polynomial time. In [6], we considered the generalization where we only want to block k-arborescences, where a k-arborescence is the edge-disjoint union of k out-arborescences. We showed that this problem is NP-complete if k is part of the input (it was already known that it is in P if k is fixed.)

The paper [6] also considered problems related to blocking k-braids (edge-disjoint unions of k $s-t$ paths) in both undirected and directed graphs. We proved that the weighted problem is NP-complete if k is part of the input. We also considered the problem when not all k-braids have to be blocked, only the optimal ones with respect to some cost function. We showed that computing the minimum number of blocking edges is possible in polynomial time in both the undirected and the directed case. We also gave a polynomial-time algorithm for the problem of blocking optimal k-spanning trees (disjoint unions of k spanning trees) in undirected graphs.

3 Matroid intersection and related problems

Matroid intersection problems are ubiquitous in graph theory, since structures like branchings in directed graphs or matchings in bipartite graphs can be expressed as common independent sets of two matroids. However, matroid intersection encompasses much more than these two cases: for example, k-arborescences can also be expressed as common bases of two matroids, so general matroid results can be applied to a broad class of problems.

In [8], we examined the guaranteed approximation factor of lexicographically optimal solutions compared to the maximum weight ones – note that for independent sets of matroids, these two coincide. We showed that for both matroid intersection problems and for matchings in general graphs, the approximation factor can be bounded using the minimum ratio of two distinct weight values, and an optimal solution is guaranteed if this ratio is at least two.

Matchings in general graphs and branchings in directed graphs (the latter a matroid intersection problem) have a common generalization in mixed graphs (graphs with both undirected and directed edges), called *matching forests*. In [14], we investigated whether the disjoint union of k matching forests can be re-partitioned into k matching forests in an equitable way – where equitability means that the number of edges (undirected, directed or the union of the two) should be almost equal in all matching forests. The motivation for this question is that such an equitable decomposition is not possible for matroid intersection in general; moreover, no upper bound can be given for the difference between the sizes. In contrast, we showed that for matching forests, we can find a decomposition that is simultaneously equitable for undirected and directed edges, with a difference of at most 2 between the sizes. We also introduced a covering version of matching forests, called *mixed edge covers*, and proved similar equitable decomposition results for them.

In [12], we gave a new characterization for integer base polyhedra, which are a generalization of matroid base polytopes to non-$\{0,1\}$ vectors. We showed that a polyhedron P in the hyperplane $\sum_{j=1}^{n} x_j = k$ is a base polyhedron if and only if it satisfies the following linking property: for any $f \in \mathbb{Z}^n$ and $g \in \mathbb{Z}^n$ with $f \leq g$, P has an integer element between f and g if and only if it has integer elements both above f and below g. Our result means that proving the linking property for a class of polyhedra is sufficient for using all convenient properties of base polyhedra, such as integrality of intersection.

We also considered a partitioning problem which comes from schedulig theory [7]. In scheduling with resource constraints and 0 processing times, we have a list of instantaneous jobs with weights and resource requirements, and a set of resource arrival times with resource quentities. We have to assign the jobs to resource arrival times in such a way that enough resource is available for each job at its assigned date, and the aim is to minimize the weighted completion time. Although a partitioning problem, this cannot be formulated as a matroid problem due to the structure of the resource bounds, and the problem is known to be NP-hard. Our main result is a polynomial-time approximation scheme, which is based on a novel technique for converting a PTAS for a constant number of arrival times into a PTAS for arbitrary number of arrival times – applying this technique to other scheduling problems is a promising future research direction.

4 Popular matchings and branchings

In this section, we describe results related to a different notion of blocking, which is based on preferences of nodes over their incident edges. A well-known example is the popular matching problem: if each node of a graph has a partial order on incident edges, then we can compare two matchings by counting the number of nodes that prefer one to the other. We say that a matching M is *blocked* by matching M' if M loses in such a comparison with M'. A *popular matching* is one that is not blocked by any other matching.

The existence and optimization of popular matchings is a well-studied area. In the project, we considered various generalizations and related concepts, and their computational complexity.

In a bipartite graph $G = (S, T; E)$ with a quota function $b : S \cup T \to \mathbb{Z}_+$, a subgraph $F \subseteq E$ is a b-matching if the degree of each node v is at most $b(v)$. The b-matchings of G can be characterized as common independent sets of two matroids. Popularity of b-matchings can be defined similarly as for matchings; we can also consider *strong popularity*, which requires a b-matching to win a comparison against any other b-matching. In [13], we gave a combinatorial polynomial-time algorithm for finding a strongly popular b-matching (if it exists) for the case when one side has strict preferences, while the other side is indifferent between choices. The relevance of this case is that deciding the existence of a popular b-matching for such a system is NP-complete. Following our work, Brandt and Bullinger [9] gave an LP-based algorithm for a more general class of strongly popular matching problems.

A branching in a directed graph D is a subgraph where every component is an arborescence. In such a subgraph, each node has at most one incoming edge. Thus, if each node has a preference order on the incoming edges in D, then popularity of branchings can be defined similarly as for matchings (being the root is the least preferred option). The popular branching problem is relevant in voting theory, with relation to delegated voting frameworks. In such a framework, voters may either vote on an issue, or delegate their voting right to another person that they consider more knowledgeable on the issue. Delegations should be transitive, so every vote is actually cast. Thus, delegation cycles should be avoided, i.e., the delegation graph should be an in-branching (the reverse of a branching). One may devise a system where voters can indicate multiple possible delegates (with a partial order on them), and the system chooses a feasible delegation graph. It is a natural requirement for this graph to be a popular in-branching, which leads to the problem of deciding whether a popular branching exists at all.

In [10], we gave a polynomial-time algorithm to decide if a popular branching exists, and find one if so. If the preferences are weak orders (rankings with possible ties), then our algorithm is also able to find a minimum cost popular branching according to an arbitrary cost function on the edges. Weak orders also allow us to compute least unpopular branchings if no popular one exists; for general partial orders, we showed that the latter two problems are NP-hard.

We revisited the popular matching problem in [11], a paper that has recently been accepted to the SODA 2022 conference. We again studied a bipartite preference system with one-sided preferences (i.e., the other side is indifferent between options), but now we only considered perfect matchings. A *popular assignment* is a perfect matching that is not blocked by any other perfect matching in the sense of popularity. This setting is relevant to applications where the size of the matching has priority over popularity (e.g., public housing programs, or assignment of students to projects with diversity constraints). Although the problem is formulated in terms of perfect matchings, easy reductions show that we can also solve the popular maximum matching problem, and the popular matching problem with lower and upper degree bounds (where the matching is popular if it is not blocked by another matching that also satisfies the bounds). Our main result is that the popular assignment problem can be solved for arbitrary partial orders. In contrast to the popular branching problem, the minimum cost version is shown in [11] to be NP-hard even for strict preferences. We also show that the existence of an assignment with unpopularity margin at most k can be decided in $O^*(|E|^k)$ time, and this running time is essentially optimal under reasonable complexity assumptions.

References

[1] H. Angelidakis, Y. Makarychev, P. Manurangsi, *An Improved Integrality Gap for the Călinescu-Karloff-Rabani Relaxation for Multiway Cut*, Integer Programming and Combinatorial Optimization 2017, 39–50 (2017)

[2] K. Bérczi, K. Chandrasekaran, T. Király, E. Lee, C. Xu, *Global and fixed-terminal cuts in digraphs*, Proceedings of Approximation, Randomization, and Combinatorial Optimization (APPROX/RANDOM 2017), https://arxiv.org/abs/1612.00156

[3] K. Bérczi, K. Chandrasekaran, T. Király, V. Madan, *Improving the Integrality Gap for Multiway Cut*, Mathematical Programming 183:171-193 (2020), https://doi.org/10.1007/s10107-020-01485-2

[4] K. Bérczi, K. Chandrasekaran, T. Király, V. Madan, *A tight $\sqrt{2}$-approximation for Linear 3-Cut*, Mathematical Programming 184:411-443 (2020), https://doi.org/10.1007/s10107-019-01417-9

[5] K. Bérczi, K. Chandrasekaran, T. Király, E. Lee, C. Xu, *Beating the 2-approximation factor for global bicut*, Mathematical Programming Ser. A 177:291-320 (2019), https://doi.org/10.1007/s10107-018-1270-8

[6] K. Bérczi, A. Bernáth, T. Király, Gy. Pap, *Blocking optimal structures*, Discrete Mathematics 341:1864-1872 (2018), https://doi.org/10.1016/j.disc.2018.03.024

[7] K. Bérczi, T. Király, S. Omlor, *Scheduling with Non-Renewable Resources: Minimizing the Sum of Completion Times*, Proceedings of ISCO 2020, LNCS, volume 12176, 167-178 (2020), https://doi.org/10.1007/978-3-030-53262-8_14

[8] K. Bérczi, T. Király, Y. Yamaguchi, Y. Yokoi, *Approximation by Lexicographically Maximal Solutions in Matching and Matroid Intersection Problems*, arxiv:2107.09897 (2021, közlésre elfogadva: Theoretical Computer Science), https://arxiv.org/abs/2107.09897

[9] F. Brandt, M. Bullinger, *Finding and Recognizing Popular Coalition Structures*, In: AAMAS 2020, 195-203, https://www.ifaamas.org/Proceedings/aamas2020/pdfs/p195.pdf

[10] T. Kavitha, T. Király, J. Matuschke, I. Schlotter, U. Schmidt-Kraepelin, *Popular Branchings and Their Dual Certificates*, Mathematical Programming, available online (2021), https://doi.org/10.1007/s10107-021-01659-6

[11] T. Kavitha, T. Király, J. Matuschke, I. Schlotter, U. Schmidt-Kraepelin, *The popular assignment problem: when cardinality is more important than popularity*, arXiv:2110.10984 (közlésre elfogadva: SODA 2022), https://arxiv.org/abs/2110.10984

[12] T. Király, *Base polyhedra and the linking property*, Journal of Combinatorial Optimization 36:671-677 (2018), https://doi.org/10.1007/s10878-017-0133-1

[13] T. Király, Zs. Mészáros-Karkus, *Finding strongly popular b-matchings in bipartite graphs*, European Journal of Combinatorics Volume 88, August 2020, https://doi.org/10.1016/j.ejc.2020.103105

[14] T. Király, Y. Yokoi, *Equitable Partitions into Matchings and Coverings in Mixed Graphs*, arxiv:1811.07856 (2019, közlésre elfogadva: Discrete Mathematics), https://arxiv.org/abs/1811.07856

[15] M. Mirzakhani, J. Vondrák, *Sperner's Colorings, Hypergraph Labeling Problems and Fair Division*, Proceedings of the twenty-sixth annual ACM-SIAM Symposium on Discrete Algorithms, 873–886 (2015)