OPTIMISING CONSTRAINED PATHS, CYCLES, EDGE-CUTS AND SCHEDULING WITH LOCAL CONSTRAINTS

MARTIN LOEBL, HANS RAJ TIWARY

Abstract. Let $G = (V, E)$ be a graph endowed with a non-negative integer edge-weight function $w : E \to \mathbf{Q}_+$. Given a zero-one vector $c = (c_1, \ldots, c_k)$ and subsets $F_1, \ldots, F_k$ of E, let $F = (F_1, \ldots, F_k)$ and $C(k) = (F, c)$. Let $\mathcal{S}$ be a family of subsets of E. We say that $S \in \mathcal{S}$ is $C(k)$-*constrained* if for each $i \leqslant k$, $|S \cap F_i| = c_i \mod 2$. We show that the following problems admit a polynomial extension complexity for k fixed: (1) Find min weight $C(k)$-constrained s, t-path and (2) Find min weight $C(k)$-constrained cycle. This extends recently obtained strongly polynomial algorithms for these problems. We next generalise these results to show that a general scheduling problem with local constraints admits polynomial extension complexity.

1. Introduction

Let $G = (V, E)$ be a graph. We let $n = |V|$ and $m = |E|$. A subset $E' \subset E$ is called a *circuit* if the graph (V, E') has only one non-trivial component and each vertex-degree zero or two. A subset $E' \subset E$ is called a *cycle* if it is a disjoint union of circuts; equivalently, if each degree of the graph (V, E') is even. A subset $E' \subset E$ is called a *perfect matching* if each degree of the graph (V, E') is one. A subset $E' \subset E$ is called a *edge-cut* if there is $V' \subset V$ such that $E' = \{e \in E; |e \cap V'| = 1\}$. Let $T \subset V$ have an even cardinality. We say that $E' \subset E$ is a *T-join* if $v \in V$ has an odd degree in the graph (V, E') if and only if $v \in T$. Let G be endowed with a non-negative integer edge-weight function $w : E \to \mathbf{Q}_+$. Given a zero-one vector $c = (c_1, \ldots, c_k)$ and subsets $F_1, \ldots, F_k$ of E, let $F = (F_1, \ldots, F_k)$ and $C(k) = (F, c)$. Let $\mathcal{S}$ be a family of subsets of E. We say that $S \in \mathcal{S}$ is $C(k)$-*constrained* if for each $i \leqslant k$, $|S \cap F_i| = c_i \mod 2$. We consider the following problem:

OPT Constrained T-join (OPT CT). Given graph $G = (V, E)$, a weight function $w : E \to \mathbf{Q}_+$ and $C(k)$, find optimum (min or max) total weight of a $C(k)$-constrained T-join.

Analogously one defines *OPT Constrained Cycle (OPT CC), OPT Constrained s, t-Path (OPT CP), OPT Constrained Perfect Matching (OPT CPM) and OPT Constrained Edge-Cut (OPT CE-C).* An important special case of *OPT CC* is the Max Cut problem for the graphs embeddable in a fixed Riemann surface.

Let $P \subset \mathbb{R}^d$ be a polytope. A polytope $Q \subset \mathbb{R}^{d+r}$ is called an *extension* or *extended formulation* of P if P can be obtained by some linear projection of Q. The *extension complexity* of a polytope P is defined as the minimum number of facets over all extensions of P.

1.1. Motivation and Main Results.

Two facts (explained in more detail in 1.2) are known for decades: (1) the Isolation Lemma implies for k fixed there is a weakly polynomial randomised algorithm for OPT CT and (2) the theory of Kasteleyn orientations implies a weakly polynomial

Date: October 18, 2017.

The authors were partially supported by the Czech Science Foundation under the contract number P202-13-21988S (M. L.) and by GA15-11559S (H.R. T.).

deterministic algorithm for OPT CT, k fixed, and the graphs embeddable in any fixed Riemann surface. It has been a well-known open problem to find strongly polynomial algorithms.

Let $\mathcal{T}$ be the set of all characteristic vectors of the T-joins of G. Let $\mathcal{E}$ denote the set of the vectors in $\mathbf{Z}_+^E$ where each entry is even. Let $\mathcal{W}$ be the set of the vectors of $\mathcal{T}+\mathcal{E}$ whose sum of entries is at most M. Let $\mathcal{W}^c \subset \mathcal{W}$ be the set of the $C(k)$-constrained elements of $\mathcal{W}$. We show

Theorem 1. *Let k and $|T|$ be fixed positive integers. Then the extension complexity of* conv$\mathcal{W}^c$ *is polynomial in n and M.*

Theorem 1 is proved in section 2. It implies existence of strongly polynomial algorithms:

Corollary 1.1 ([6])**.** *Let k and $|T|$ be fixed positive integers. Then MIN CT, MIN CP and MIN CC admit strongly polynomial algorithms. MIN CE-C admits a strongly polynomial algorithm for the graphs embeddable in any fixed Riemann surface.*

Proof. MIN CT is solved by min $(w^T x : x \in \text{conv}\mathcal{W}^c)$ since the weiths $w(e)$ are assumed to be non-negative. MIN CP is equivalent to MIN CT for $|T| = 2$ and MIN CC is equivalent to MIN CT for $|T| = \emptyset$. Finally, MIN CE-C for a fixed Riemann surface of genus g is by the background result **(4)** below reduced to MIN CC with $2g$ additional parity constraints. □

1.2. **State-of-the-art.** We first write down some background results.

(1) A generalisation of a well-known construction of Fisher (see [3], [9] p.104) reduces OPT CT in the class of graphs embeddable in a fixed Riemann surface S to OPT CPM in the class of graphs embeddable in S. Given graph G embedded in the surface S, the construction produces graph G' embedded in S so that there is a natural weight preserving bijection between the set of the $T-$joins of G and the set of the perfect matchings of G'.

(2) There is a well-known randomised algorithm to solve OPT CPM for k fixed and the edge-weights bounded by a polynomial in n; it goes as follows. Let $x, y_1, \ldots, y_k$ be $k+1$ variables. We define the generating function of perfect matchings by $\mathbf{P}(G, w, C(k)) = \sum_{M \text{ perfect matching}} \prod_{e\in M} z_e$ where $z_e = x^{w(e)} \prod_{j: e\in F_j} y_j$.

Because of our assumptions, the number of *possible* monomials of a non-zero coefficient in $\mathbf{P}(G, w, C(k))$ is polynomial in n. Clearly, we solve OPT CPM if we can determine which of these *potential* monomials of $\mathbf{P}(G, w, C(k))$ indeed have a non-zero coefficient in $\mathbf{P}(G, w, C(k))$. A well-known randomised algorithm to decide if a coefficient of a monomial of the generating function of perfect matchings is non-zero is called **Isolation Lemma** [10]. See also [6] for related considerations.

(3) It is well-known that the generating function of perfect matchings $\mathbf{P}(G, w, C(k))$ of a graph G of genus g in variables $x, y_1, \ldots, y_k$ can be written as a linear combination of 4^g Pfaffians (see [8], [2], [4], [5], [13], [1]); such a Pfaffian is a determinant-type expression in variables $x, y_1, \ldots, y_k$ that can be calculated by a weakly polynomial algorithm. Hence for k fixed and the graphs embeddable in a fixed Riemann surface we can calculate the whole $\mathbf{P}(G, w, C(k))$ in a polynomial time.

(4) A well-known folklore lemma (see e.g. Gerards [7]) relates, for a graph G embedded in a Riemann surface of genus g, the set of edge-cuts of G and certain subset of cycles of the geometric dual of G.

Lemma 1.2. *Let G be a graph embedded in a Riemann surface of genus g and let $G^* = (V^*, E^*)$ be its geometric dual. Then there are subsets $E_1, \ldots, E_k$ of E^*, $k = 2g$, so that $F \subset E$ is edge-cut of G if and only if the corresponding set of dual edges $F^* \subset E^*$ is a cycle in G^* and for each $i \leqslant k$, $|E_i \cap F^*| = 0 \mod 2$.*

(5) Gerards showed in [7] that the T-joins polyhedron of a graph $G = (V, E)$ embeddable in a fixed Riemann surface have a polynomial extension complexity. The $T-$join polyhedron $P_T(G)$ is the set of all real vectors x such that there exists a convex combination y of elements of $\mathcal{T}$ such that $x_e \geqslant y_e$ for each $e \in E$.

Theorem 2. [7] *Let $G = (V, E)$ be a graph embeddable in a fixed Riemann surface S. Then the $T-$join polyhedron $P_T(G)$ have a polynomial extension complexity.*

(6) Geelen and Kapadia [6] reduce OPT CT for k and $|T|$ fixed to a polynomial number of shortest path algorithm runs.

(7) MIN CE-C was recently proved to be polynomial by Nagele, Sudakov, Zenklusen [12]. Earlier, MIN CE-C with $k \leqslant 2$ and $c_i = 1$ for each $i \leqslant k$ was proved to be polynomial by Padberg and Rao [11].

2. Proof of Theorem 1

Consider a 0/1-vector of length $2|E(G)| * M$ viewed as M blocks of size $2|E(G)|$, where each block of $2|E(G)|$ contains at most one one. A natural interpretation of this encoding is of at most M directed edges of G. Let $P(G, M)$ be the convex hull of all such vectors that satisfy the following properties:

- Each block of $2|E(G)|$ contains at most one one.
- The M edges given in the specific sequence form a collection of walks between disjoint pairs of vertices from T and further closed walks.
- The M edges form a $C(k)$-constrained subset of edges.

Lemma 2.1. *$P(G, M)$ is an extended formulation of* conv$\mathcal{W}^c$.

Proof. The vertices of $\mathcal{W}^c$ are exactly those multisets of edges of G that can be partitioned into walks between pairs of vertices of T and a set of closed walks. The subset of edges must also satisfy the requirement of being $C(k)$-constrained. So the sum over all indices correspoding to a given edge in $P(G, M)$ gives the value for the edge in $\mathcal{W}^c$. □

The following has been shown by Tiwary [14].

Theorem 3. *If the vertices of a d-dimensional 0/1-polytope Q – when viewed as a binary string – can be accepted by a one pass nondeterministic Turing machine requiring space $s(n)$ then the extension complexity of Q is at most $2^{\mathcal{O}(s(n))}d$.*

It is now easy to see that the vertices of $P(G, M)$ can be accepted by a one pass deterministic Turing machine that requires space $\mathcal{O}(\log |E(G)| + |T| + k)$.

Theorem 4. *The vertices of $P(G, M)$ – viewed as binary strings – are accepted by a one pass deterministic Turing machine requiring sapce $\mathcal{O}(\log |E(G)| + |T| + k)$.*

Proof. The accepting Turing machine TM notes down the starting vertex u when reading the first block. For each block it verifies that at most one one is present. From the location of this one TM deduces which edge is being used and hence the next vertex in the walk. Whenever the walk hits some vertex v of T, the vertices u, v are removed from T and the procedure is repeated. When T is empty TM verifies that the remaining stream encodes a walk given in a correct sequence. During processing of each edge $e \in F_i$, TM increases the count of number of edges in F_i modulo 2. At the end of the computation TM verifies that the number of edges seen in F_i equals c_i modulo 2 and so that set of edges in the stream describe a $C(k)$-constrained subset. Since TM only needs to store the current set T, the k modular sums corresponding to each F_i and the identity of the current edge being processed the total space used is as desired. □

This completes the proof of Theorem 1. In the next section, we extend Theorem 1 to provide a general framework to optimally and efficiently schedule jobs with local constraints.

3. Scheduling jobs with local constraints

Let $G = (V, E)$ be a graph, $n = |V|$, $m = |E|$ and let $\overline{E}$ be the edges of the symmetric orientation of G. Let $w : \overline{E} \to \mathbf{Q}$. Finally let $A_1, \ldots, A_d$ be subsets of $\overline{E}$.

A *walk* on graph G is a sequence $(a_1, \ldots, a_l)$ such that each a_i is an orientation of an edge of G, and the head of a_i is the tail of a_{i+1} for each $i < l$. We say that walk X is *closed* if its terminal vertex is equal to its starting vertex. A walk will be a 'trajectory' of a job in the 'production network' given by graph G. We are interested in multiple jobs that satisfy some constraints. Some jobs may have constraints depending on other jobs, such as at time 3 no two jobs share an edge. We call such jobs *special* jobs. A job whose constraints do not depend on any other job is called a *normal* job. An example of a normal job is a job with constraint "whenever an edge of A_i is used, another edge of A_i is not used for 10 steps".

For each $t \in \mathbf{Z}$ we use notation $s(X, t)$ for a vector of $\{0, \ldots, \Delta\}^{d+4}$. The meaning of $s(X, t)$ is that it contains complete 'local information' of walk X at time t; each component $s(X, t)_i, i \leqslant d$, concerns set $A_i \subset \overline{E}$; $s(X, t)_{d+1}$ is some global information for X such as "the (parity of) total number of edges in X", $s(X, t)_{d+2}$ encodes the starting vertex of walk X, $s(X, t)_{d+3}$ encodes the desired end vertex of the job whose partial trajectory is walk X and $s(X, t)_{d+4} \in \{0, 1\}$ indicates whether X is faulty (at time t).

Scheduling jobs with local constraints
Input: $G, w, \Delta, \delta, K, A_1, \ldots, A_d$ and further:

- An algorithm $A(i, s(X, t), a)$ takes the information vector of walk X representing partial trajectory of job i and an edge a to be added to the walk X, and produces information vector $s(X + a, t + 1)$ for the new walk. The index i allows for choosing different such algorithms for different walks. The algorithm is allowed to use only linear space. Apart from producing the information vector $s(X + a, t + 1)$ the algorithm is also allowed to manipulate the fixed number of global registers $r_1, \ldots, r_c$.
- Another algorithm $A'(s(X_1, t), \ldots, s(X_\delta, t))$ that takes the information vectors of δ special walks and outputs whether the walks satisfy some mutual constraints. This algorithm is also allowed to use only linear space.

Output: Special walks $X_1, \ldots, X_\delta$ and normal walks $X_{\delta+1}, \ldots, X_K$ *consistent* with the input algorithms and such that for each $j \leqslant K$, $s(X_j, M)_{d+4} = 1$ where M is an upper bound on the length of all admissible walks. Moreover we require that the total weight of the walks is optimised.

Theorem 5. *Let $\mathcal{C}$ be the set of all characteristic vectors of multi-sets of edges of valid collections of walks in the problem defined above. Then the extension complexity of* conv$\mathcal{C}$ *is polynomial in* $\Delta^{\delta \cdot d}|V|^\delta KM|E|$.

Proof. The K walks are considered to be given as a 0/1 string of length $2|E|MK$ with the following interpretation. For the first $2|E|\delta M$ bits each consective block of $2|E|\delta$ bits represents the δ next edges of the special walks. Within these blocks each block of length $2|E|$ contains exactly one one corresponding to which directed edge is the next edge in the correpsonding walk. After the first $2|E|\delta M$ bits the next bits are considered in groups of $2|E|M$ bits specifying a walk of length at most M by representing one edge by a vector of length $2|E|$ with at most one one corresponding to the next edge in the walk.

The Turing machine accepting only the valid jobs works as follows. It creates the information vectors $s(X_i, M)$ for $1 \leqslant i \leqslant \delta$ by applying algorithm $A(\cdot)$ on the current information vectors after reading the current edge. At each step it also applies algorithm A' on these information vectors to decide whether some constraint is violated or not. After this the algorithm reads each walk separately and uses the same space on the work tape to construct $s(X, M)$ deciding at the end whether the walk is valid or not.

The total space used is $\mathcal{O}((\log \Delta d + \log |V|)\delta + \log k)$ and so by Theorem 3 the extension complexity of all valid jobs is at most polynomial in $\Delta^{\delta \cdot d}|V|^{\delta} KM|E|$. □

Since any polynomial method for optimizing linear functions over polytopes can then be used to find the optimum collection of jobs, we get the following corollary.

Corollary 3.1. *There is an algorithm to solve the general scheduling problem defined above whose running time is polynomial in* $\Delta^{\delta \cdot d}|V|^{\delta} KM|E|$.

Concluding remark. Theorem 1 is a special case of Theorem 5: $\mathcal{W}^c = \mathcal{C}$ for the instance of the local scheduling problem where $k = d$, $A_i = \overline{F_i}$ for $i = 1, \ldots, d$, $c = |T| + d$, $\delta = 0$, $\Delta = 1$ and $K \leqslant M$. There are no special walks, the global registers $r_1, \ldots, r_d$ describe the parity of the current valid collection in $A_1, \ldots, A_d$ and the remaining global registers $r_{d+1}, \ldots r_c$ describe whether each vertex $t \in T$ has degree zero or one in the current valid collection.

Our approach (Theorem 5) provides polynomial scheduling for a variety of temporal constraints. Moreover, if processors are interpreted as jobs then the algorithms A and A' may also model message passing among the processors, and between a processor and the global memory in parallel and distributive computation.

References

[1] D. Cimasoni, N. Reshetikhin, Dimers on surface graphs and spin structures I. Comm. Math.Phys. 275(1) (2007).
[2] A. Galluccio, M. loebl, On the theory of pfaffian orientations I, Electron. J. Combin. 6 (1999).
[3] A. Galluccio, M. loebl, On the theory of pfaffian orientations II. t-joins, k-cuts, and duality of enumeration, Electron. J. Combin. 6 (1999).
[4] A. Galluccio, M. Loebl, J. Vondrak, A new algorithm for the Ising problem: partition function for finite lattice graphs, Physical Review Letters 84 (2000).
[5] A. Galluccio, M. Loebl, J. Vondrak, Optimization via enumeration: a new algorithm for the Max Cut problem, Mathematical Programming 90 (2001).
[6] J. Geelen, R. Kapadia, Computing girth and cogirth in perturbed graphic matroids, Combinatorica, Available online at http://dx.doi.org/10.1007/s00493-016-3445-3 (2017).
[7] A.M.H. Gerards, Compact System for T-join and perfect matching polyhedra of graphs with bounded genus, Operations Research Letters 10 (1991) 377- 382.
[8] P. Kasteleyn, Graph theory and crystal physics, in: Graph theory and theoretical physics, Academic Press, London (1967).
[9] M. Loebl, Discrete mathematics in statistical physics, Vieweg and Teubner (2010).
[10] K. Mulmuley, U.V. Vazirani, V.V. Vazirani, Matching is as easy as matrix inversion, In Proceedings of the 19th Annual ACM Symposium on Theory of Computing, STOC'87 (1987) 345– 354.
[11] M.W. Padberg, M.R. Rao, Odd minimum cut-sets and b-matchings, Math.Oper. Res. 1 (1982) 67- 80.
[12] M. Nagele, B. Sudakov, R. Zenklusen, Submodular Minimization Under Congruency Constraints, arXiv:1707.06212 (2017)
[13] G. Tesler, Matchings in graphs on Non-orientable Surfaces, Journal of Combinatorial Theory-B, 78 (2000).
[14] H. R. Tiwary, Extension Complexity of Formal Languages, arXiv:1603.07786 (2017)

Dept. of Applied Mathematics, Charles University, Praha, Czech Republic.
E-mail address: loebl@kam.mff.cuni.cz, hansraj@kam.mff.cuni.cz