Mathematical Programming (2020) 183:455–481
https://doi.org/10.1007/s10107-020-01498-x

FULL LENGTH PAPER

Series B

A new contraction technique with applications to congruency-constrained cuts

Martin Nägele[1] · Rico Zenklusen[1]

Received: 31 May 2019 / Accepted: 23 March 2020 / Published online: 15 April 2020

Abstract

Minimum cut problems are among the most classical problems in Combinatorial Optimization and are used in a wide set of applications. Some of the best-known efficiently solvable variants include global mininmum cuts, minimum s–t cuts, and minimum odd cuts in undirected graphs. We study a problem class that can be seen to generalize the above variants, namely finding congruency-constrained minimum cuts, i.e., we consider cuts whose number of vertices is congruent to r modulo m, for some integers r and m. Apart from being a natural generalization of odd cuts, congruency-constrained minimum cuts exhibit an interesting link to a long-standing open problem in Integer Programming, namely whether integer programs described by an integer constraint matrix with bounded subdeterminants can be solved efficiently. We develop a new contraction technique inspired by Karger's celebrated contraction algorithm for minimum cuts, which, together with further insights, leads to a polynomial time randomized approximation scheme for congruency-constrained minimum cuts for any constant modulus m. Instead of contracting edges of the original graph, we use splitting-off techniques to create an auxiliary graph on a smaller vertex set, which is used for performing random edge contractions. This way, a well-structured distribution of candidate pairs of vertices to be contracted is obtained, where the involved pairs are generally not connected by an edge. As a byproduct, our technique reveals new structural insights into near-minimum odd cuts, and, more generally, near-minimum congruency-constrained cuts.

Keywords Minimum cuts · Congruency-constrained optimization · Contraction algorithm · Splitting off

A short version of this paper appeared at IPCO 2019.

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (Grant Agreement No. 817750). This work was supported by Swiss National Science Foundation Grants 200021_165866 and 200021_184622.

Extended author information available on the last page of the article

Mathematics Subject Classification 90C27 · 90C35 · 68R05 · 68Q25 · 05C99

1 Introduction

Cuts in undirected graphs are a basic structure in Combinatorial Optimization with a multitude of applications. The global minimum cut problem, the minimum s–t cut problem, and the minimum odd cut problem are among the best known efficiently solvable minimum cut variants, and have been the cradle of many exciting algorithmic techniques. We study a generalization of these problems that we call *congruency-constrained minimum cut* (CCMC), where a congruency constraint on the vertices in the cut is imposed, as described in the box below.[1]

Congruency-Constrained Minimum Cut (CCMC): Let $G = (V, E)$ be an undirected graph with edge weights $w\colon E \to \mathbb{R}_{\geqslant 0}$ and let $\gamma\colon V \to \mathbb{Z}_{\geqslant 0}$. Let $m \in \mathbb{Z}_{>0}$ and $r \in \mathbb{Z}_{\geqslant 0}$. The task is to find a minimizer of

$$\min\left\{ w(\delta(C)) \;\middle|\; \emptyset \subsetneq C \subsetneq V,\ \sum_{v\in C} \gamma(v) \equiv r \pmod{m} \right\}. \tag{CCMC}$$

We call m the *modulus* of the problem, and we will typically consider m to be constant. Moreover, allowing general γ-values—instead of setting $\gamma(v) = 1$ for all $v \in V$, i.e., requiring that $|C| \equiv r \pmod{m}$—is merely for convenience. Indeed, the case of general γ-values can be reduced to the unit case by replacing each vertex v by a clique of ($\gamma(v)$ mod m)-many vertices with large edge values if $\gamma(v) \not\equiv 0 \pmod{m}$, and a clique of size m if $\gamma(v) \equiv 0 \pmod{m}$.[2]

Apart from generalizing well-known cut problems, we are interested in the study of (CCMC) due to a link to an intriguing open question in Integer Programming, namely whether integer linear programs (ILPs) defined by an integer constraint matrix with bounded subdeterminants can be solved efficiently. Recently it was shown in [1] that ILPs of the form $\min\{c^\top x \mid Ax \leqslant b,\ x \in \mathbb{Z}^n\}$ can be solved efficiently if $A \in \mathbb{Z}^{m\times n}$ is *bimodular*, i.e., A has full column-rank and the determinant of every $n \times n$ submatrix of A is in $\{-2, -1, 0, 1, 2\}$. This result implies that if A is totally bimodular, i.e., all subdeterminants of A are in $\{-2, -1, 0, 1, 2\}$, then the corresponding ILP can be solved in polynomial time even without the requirement of A having full column rank (see [1] for details). This extends the well-known fact that ILPs with a totally unimodular constraint matrix can be solved efficiently; here, the absolute value of subdeterminants is bounded by 1. Only very limited techniques are known for attacking the question whether ILPs remain efficiently solvable in the Δ-modular

[1] The minimum odd cut problem is captured by (CCMC) by choosing $m = 2$, $r = 1$, and $\gamma(v) = 1$ for all $v \in V$. Global minimum cuts correspond to $m = 1$, r arbitrary, and $\gamma(v) = 0$ for all $v \in V$, and s–t cuts can be modeled as minimum $\{s, t\}$-odd cuts, i.e., $m = 2$, $r = 1$, $\gamma(s) = \gamma(t) = 1$, and $\gamma(v) = 0$ for all $v \in V \setminus \{s, t\}$.

[2] We denote by ($\gamma(v)$ mod m) the smallest non-negative integer congruent to $\gamma(v)$ modulo m. Reducing modulo m is crucial to obtain a blow-up bounded by m, which, as mentioned, we will typically assume to be constant.

case for some constant $\varDelta > 2$, i.e., $\operatorname{rank}(A) = n$ and any $n \times n$ subdeterminant of A is in $\{-\varDelta, -\varDelta + 1, \ldots, \varDelta\}$. Interestingly, to approach the bimodular case, classical combinatorial optimization problems with congruency constraints play a crucial role, and the problem can be reduced to certain types of congruency-constrained cut and flow problems (see [1]). In particular, (CCMC) with modulus m can be reduced to m-modular ILPs.[3] Hence, if one believes that $\varDelta$-modular ILPs can be solved efficiently for $\varDelta = O(1)$, then (CCMC) should admit an efficient algorithm. Conversely, despite the fact that, for $\varDelta \geq 3$, further gaps have to be overcome to reduce $\varDelta$-modular ILPs to congruency-constrained cut and flow problems, the results in [1] give hope that congruency-constrained cuts may be a useful building block for attacking $\varDelta$-modular ILPs, besides merely being a special case thereof.

Unfortunately, not much is known in terms of techniques to deal with congruency constraints in Combinatorial Optimization beyond parity constraints ($m = 2$). These constraints introduce an algebraic component to the underlying problem, which is a main additional hurdle to overcome. Some progress has been achieved for moduli m that are constant prime powers: it was shown in [30] that submodular function minimization under congruency constraints with such moduli can be solved efficiently. As the cut function is submodular, this implies that (CCMC) can be solved efficiently for m being a constant prime power. However, the techniques in [30] do not extend to general constant moduli m, due to intrinsic additional complications appearing when m has two different prime divisors.

The goal of this paper is to show that contraction techniques, in the spirit of Karger's algorithm for global minimum cuts [17,18], can be employed to approach (CCMC). A naive way of using Karger for (CCMC) faces several hurdles, which we exemplify through the (CCMC) instance in Fig. 1, parameterized by an even number n and a weight $M \geqslant 1$.[4] It consists of n paths of length 2 between two vertices u and w. An optimal cut is highlighted in gray. Karger's algorithm returns any global minimum cut in a graph $G = (V, E)$ with probability $\Omega(|V|^{-2})$, implying that there are at most $O(|V|^2)$ global minimum cuts. However, for $M = 1$, the (CCMC) problem in Fig. 1 has exponentially many optimal solutions. Hence, we cannot hope for an algorithm that returns any optimal solution with probability $\Omega({}^1/_{\mathrm{poly}(|V|)})$. Moreover, if one of the n many u–w paths gets contracted, then the problem turns infeasible. It is not clear how to fix this. Even if we forbid contractions that make the instance infeasible, it is likely that in many of the u–w paths, one would contract the edge of weight 1. It is not hard to verify that Karger-type contractions would with high probability lead to a cut

[3] If M is the incidence matrix of the digraph $H = (V, A)$ obtained by bidirecting G, then

$$\min \left\{ \sum_{a \in A} w_a y_a \;\middle|\; \begin{array}{c} Mx - y \leqslant 0,\ \sum_{v \in V} \gamma(v) x_v + zm = r,\ x_s = 1,\ x_t = 0, \\ x \in \{0,1\}^V,\ y \in \{0,1\}^A,\ z \in \mathbb{Z}\,. \end{array} \right\}$$

solves the congruency-constrained minimum s–t cut problem in G with edge weights w and congruency constraint $\gamma(C) \equiv r \ \ (\mathrm{mod}\ *)m$, where the cut corresponds to the set $C = \{v \in V | x_v = 1\}$. Moreover, the constraint matrix of the above ILP can be seen to be m-modular. Analogously to how global min cut problems can be reduced to min s–t cut problems, every (CCMC) problem can be reduced to solving linearly many problems of the above type.

[4] Even n ensures that $S = \{w, v_1, v_2, \ldots, v_n\}$ is infeasible, i.e., $\gamma(S) \not\equiv 1 \pmod{6}$.

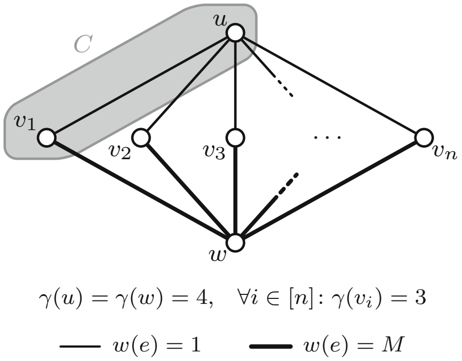
C
u
v_1
v_2
v_3
$\cdots$
v_n
w
$\gamma(u) = \gamma(w) = 4, \quad \forall i \in [n]\colon \gamma(v_i) = 3$
$w(e) = 1$
$w(e) = M$

Fig. 1 A (CCMC) instance with $m = 6$ and $r = 1$. Its optimal value is $n + M - 1$, achieved by the highlighted cut C

that is about twice as large as the minimum cut if M is chosen large (and this factor of 2 can be boosted further).

To overcome these and further hurdles, substantial changes seem necessary, and we introduce new techniques to employ contraction algorithms in our context. A key difference between our method and Karger's algorithm, as well as other contraction algorithms in a similar spirit (see a recent result of Chandrasekaran, Xu, and Yu [8] for a nice adaptation of Karger's algorithm to the hypergraph k-cut problem), is that we do not contract edges of the graph. Instead, we define a distribution over pairs of vertices to contract that may not be connected by an edge. Moreover, we only look for contractions among certain vertices, namely those $v \in V$ fulfilling $\gamma(v) \not\equiv 0 \pmod{m}$. We show that splitting-off techniques from Graph Theory can be leveraged to design an efficient procedure to sample from a distribution of vertex pairs to contract with strong properties.

1.1 Our results

Our main result for (CCMC) via our new contraction technique is the following.

Theorem 1 (CCMC) *with constant modulus m admits a PRAS.*

Recall that a *PRAS* (*polynomial time randomized approximation scheme*) is an efficient procedure that, for any fixed $\varepsilon > 0$, returns a $(1 + \varepsilon)$-approximate solution with high probability, by which we mean with probability at least $1 - {}^{1}\!/\!_{|V|}$. As the focus of this paper is existence of the PRAS claimed by Theorem 1, no efforts were made to optimize its running time. Nevertheless, let us mention that for $\varepsilon < 1$, we can bound the running time of our PRAS by $\log\left(\frac{w_{\max}}{w_{\min}}\right) \cdot |V|^{O\left(\frac{m \log m}{\varepsilon}\right)} \cdot 2^{O(m^2)}$, where $w_{\max}$ and $w_{\min}$ are the maximum and minimum edge weights occurring in the (CCMC) instance, respectively. A short discussion of this bound is given at the end of Sect. 2.

Moreover, for a constant composite modulus m that is the product of only two primes, we obtain an exact procedure.

Theorem 2 (CCMC) *with a constant modulus that is the product of two primes admits an efficient randomized algorithm that w.h.p. returns an optimal solution.*

This is in stark contrast to prior procedures, in particular for congruency-constrained submodular function minimization [30], which employ techniques that face hard barriers for moduli beyond prime powers.

Finally, in a similar spirit to Karger's algorithm for global minimum cuts, our contraction algorithm allows us to derive structural results on near-minimum congruency-constrained cuts. Whereas Karger's analysis shows that there are only polynomially many cuts of value at most a constant factor higher than the minimum cut, we cannot hope for results of this type: The example in Fig. 1 shows that (CCMC) problems can have exponentially many optimal solutions. For prime moduli, we show that near-minimum (CCMC) cuts are near-minimum cuts (without congruency constraint) in one of only a polynomial number of minimum s–t cut instances. These instances are defined on *contractions of* G, i.e., graphs obtained from $G = (V, E)$ by successively contracting nonempty node sets $S \subseteq V$. When *contracting* a set S, all vertices of S are replaced by a single vertex v_S with $\gamma(v_S) := \sum_{v \in S} \gamma(v)$, all edges with both endpoints in S are deleted, and each edge connecting a vertex in S to a vertex $u \in V \backslash S$ is replaced by an edge between u and v_S of the same weight. By construction, a cut C in a contraction of G naturally corresponds to a cut $\overline{C}$ in G of the same weight with $\gamma(C) = \gamma(\overline{C})$, and thus, we can identify these cuts.

Theorem 3 *Consider a* (CCMC) *problem on* $G = (V, E)$ *with constant prime modulus* m *and nonzero optimal solution value, and let* $\kappa \geq 1$ *be a constant. Then there is an efficient randomized method returning* $\text{poly}(|V|)$ *many minimum* s–t *cut instances defined on contractions of* G *such that the following holds with high probability, where* OPT *denotes the optimal solution value of the* (CCMC) *problem. A cut* $C \subsetneq V$, $C \neq \emptyset$, *is a solution to* (CCMC) *of value at most* $\kappa \cdot \text{OPT}$ *if and only if* C *is a feasible solution of value at most* $\kappa \cdot \text{OPT}$ *in one of the minimum* s–t *cut instances (without congruency constraint).*

Theorems 2 and 3 are in fact consequences of more general structural properties of (CCMC) instances that are exhibited by our contraction algorithm (see Sect. 4 for more details).

1.2 Further discussion on related results

Work on minimum cut problems with constraints of congruency type date back to the early '80s, when Padberg and Rao [32] presented a method to efficiently find a minimum cut among all cuts with an odd number of vertices. Barahona and Conforti [2] later showed that efficient minimization is also possible over all cuts with an even number of vertices. Later works by Grötschel, Lovász, and Schrijver [16], and by Goemans and Ramakrishnan [13] generalized these results, by showing that even any submodular function can be minimized over so-called triple families and, more generally, parity families. Submodular functions generalize cut functions, and triple as well as parity families capture congruency constraints with modulus 2. More generally, these approaches even allow for minimizing over all cuts $C \subseteq V$ of cardinality *not* congruent to r modulo m, for any integers r and m, which turns out to be a much simpler constraint than requiring a cardinality congruent to r modulo m. Indeed, (CCMC) for unbounded m quickly leads to NP-hard problems, as one could model an arbitrary cardinality constraint through a congruency constraint. In particular, if $G = (V, E)$ is a graph with an even number of vertices, then seeking a minimum cut C with $|C| \equiv 0 \pmod{|V|/2}$ captures the well-known minimum bisection problem. Khot [19] showed that, unless NP has randomized sub-exponential time algorithms, the minimum bisection problem does not admit a polynomial time approximation scheme. Hence, it seems unlikely that a PRAS can be obtained for (CCMC) without a bound on the modulus.

We briefly mention further works linked to matrices with bounded subdeterminants. This includes the problem of finding a maximum weight independent set in a graph with constant odd-cycle packing number, for which a PTAS was obtained by Bock, Faenza, Moldenhauer, Vargas, and Jacinto [7]. This problem readily reduces to ILPs with bounded subdeterminants, due to an observation of Grossman, Kulkarni, and Schochetman [15]. Another recent result by Eisenbrand and Vempala [10] is a randomized simplex-type algorithm for linear programming that is strongly polynomial whenever all subdeterminants of the constraint matrix defining the LP are bounded by a polynomial in the dimension of the problem. Furthermore, there has been interesting recent progress on the problem of approximating the largest subdeterminant of a matrix (see Di Summa et al. [9], and Nikolov [31]).

1.3 Organization of the paper

We provide a discussion of the techniques leading to our main contribution (Theorem 1) in Sect. 2. Section 3 expands on how to find a good distribution of vertex pairs to contract through splitting-off techniques, completing the proof of Theorem 1 given in Sect. 2. Section 4 is devoted to the exploration of furter structural properties of (CCMC) instances, leading to proofs of Theorems 2 and 3. Finally, Sect. 5 presents an alternative splitting-off approach for obtaining a suitable distribution of vertex pairs to contract.

2 An overview of our approach

As mentioned, the core of our approach is a contraction procedure inspired by Karger's global minimum cut algorithm, where we sample vertex pairs to be contracted from a certain distribution. In fact, the analysis of Karger's random contraction algorithm exploits that, whenever a random edge is contracted in a graph $G = (V, E)$, this contraction is *bad* with probability at most $k/|V|$ for some constant $k \in \mathbb{Z}_{>0}$. More precisely, in the analysis, an arbitrary minimum cut C is fixed, and a contraction is *bad* if it contracts two vertices on different sides of C. The probability of bad contractions being at most $k/|V|$ implies that by contracting until only k vertices remain, and then enumerating all cuts among those vertices, each minimum cut is found with probability at least $1/\binom{|V|}{k}$.

For (CCMC), an important observation is that it suffices to decide which vertices in

$$V_{\not\equiv 0} := \{v \in V \mid \gamma(v) \not\equiv 0 \pmod{m}\}$$

are part of a solution. Indeed, for any cut C, the value of $\gamma(C)$ is determined by the intersection $C \cap V_{\not\equiv 0}$. Moreover, for any $U \subseteq V_{\not\equiv 0}$, the value

$$\nu(U) := \min\left\{w(\delta(C)) \mid \emptyset \subsetneq C \subsetneq V,\ C \cap V_{\not\equiv 0} = U\right\}$$

and a minimizer C_U can be obtained efficiently by a minimum cut computation in a contraction of G.[5] As $C_U \cap V_{\not\equiv 0} = U$, we have $\gamma(C_U) \equiv \gamma(U) \pmod{m}$.

Due to the above, instead of performing contractions over the full graph, as done in Karger's algorithm, we only contract pairs in $V_{\not\equiv 0}$, with the goal to reduce $V_{\not\equiv 0}$ to a constant-size set. If we achieve this, it suffices to enumerate over all $U \subseteq V_{\not\equiv 0}$ with $\gamma(U) \equiv r \pmod{m}$, minimize $\nu(U)$, and return a corresponding cut C_U. The theorem below is a key technical result of this paper, and shows that a suitable distribution over vertex pairs in $V_{\not\equiv 0}$ to contract exists whenever the sum $\sum_{v \in V_{\not\equiv 0}} \nu(\{v\})$ is large enough.

Theorem 4 *Let $\mathcal{I} = (G, w, \gamma, m, r)$ be a (CCMC) instance on $G = (V, E)$. Let $\alpha \geqslant 0$ and $c > 0$ with $\sum_{v \in V_{\not\equiv 0}} \nu(\{v\}) > ({}^{2\alpha}/_{c}) \cdot |V_{\not\equiv 0}|$. Then, there is a distribution $\mathcal{D}$ over pairs in $V_{\not\equiv 0}$ such that $\Pr_{\{u,v\}\sim\mathcal{D}}\left[|\{u, v\} \cap C| = 1\right] \leqslant {}^{c}/_{|V_{\not\equiv 0}|}$ for any feasible solution C of $\mathcal{I}$ with $w(\delta(C)) \leqslant \alpha$. Moreover, there is a procedure to sample from $\mathcal{D}$ with running time polynomial in $|V|$ (independent of any other input parameters).*

To prove Theorem 4, we use weighted splitting-off techniques on G to construct a weighted auxiliary graph H on the vertex set $V_{\not\equiv 0}$. We show that by choosing edges of H with probabilities proportional to the edge weights, a distribution with the properties highlighted in Theorem 4 is obtained. Details of the proof are discussed in Sect. 3.

Theorem 4 with $\alpha = \text{OPT}$ (or α slightly larger than OPT) implies that, whenever $\sum_{v \in V_{\not\equiv 0}} \nu(\{v\})$ is large compared to OPT, a contraction step has good success probability, similar to Karger's analysis. Otherwise, instead of performing a contraction, we approximately reduce the problem to another (CCMC) instance with smaller modulus. More precisely, if $\sum_{v \in V_{\not\equiv 0}} \nu(\{v\})$ is sufficiently small, then there are many vertices $v \in V_{\not\equiv 0}$ where the smallest cut $C_{\{v\}} \subseteq V$ separating v from $V_{\not\equiv 0} \setminus \{v\}$ has weight no more than $\beta = \kappa \cdot \text{OPT}$ for a small constant κ. Such cuts are useful to modify a cut with wrong residue class. Indeed, consider a cut C with small weight $w(\delta_G(C))$, but $\gamma(C) \not\equiv r \pmod{m}$. Then, $\overline{C} := C \triangle C_{\{v\}}$ satisfies $\gamma(\overline{C}) \equiv \gamma(C) \pm \gamma(v)$ (where the sign depends on whether $v \in C$), while the weight $w(\delta(\overline{C}))$ increased by at most β compared to $w(\delta(C))$; we recall that β is small with respect to OPT. Our plan is that if we have enough small cuts $C_{\{v\}}$, we can simplify the congruency constraint to one with

[5] If $U \notin \{\emptyset, V_{\not\equiv 0}\}$, then C_U can be computed by contracting U and $V_{\not\equiv 0} \setminus U$ in G, and by determining a minimum cut in the contracted graph that separates the two vertices corresponding to the contracted sets. If $U \in \{\emptyset, V_{\not\equiv 0}\}$, then $\nu(U)$ is obtained by contracting $V_{\not\equiv 0}$ and finding a global minimum cut C in the contracted graph, where C is chosen such that $C \cap V_{\not\equiv 0} = U$; this is achieved by replacing the computed global minimum cut by its complement if necessary.

smaller modulus, because the small cuts of type $C_{\{v\}}$ allow for moving solutions into the right residue class. This idea leads to the following notion of a *reduction family*.

Definition 5 (Reduction family) Let $\mathcal{I} = (G, w, \gamma, m, r)$ be a (CCMC) instance on the graph $G = (V, E)$. For $\beta \in \mathbb{R}_{\geqslant 0}$ and $q \in [m-1]$, a family $\mathcal{R}(\beta, q) \subseteq 2^V$ is a *reduction family for* $\mathcal{I}$ if

(i) $\mathcal{R}(\beta, q) = \{R_1, R_2, \ldots, R_{2m_q-1}\}$ with $m_q := \frac{m}{\gcd(m,q)}$,[6]
(ii) for each $i \in [2m_q - 1]$, there is one vertex $u_i \in R_i$ with $\gamma(u_i) \equiv q \pmod{m}$, and $\gamma(u) \equiv 0 \pmod{m}$ for all other $u \in R_i \backslash \{u_i\}$,
(iii) the vertices $u_1, \ldots u_{2m_q-1}$ are distinct, and
(iv) $w(\delta(R_i)) \leqslant \beta$ for all $i \in [2m_q - 1]$.

A reduction family $\mathcal{R}(\beta, q)$ allows for correcting the residue class $\gamma(C)$ of a solution C by any multiple of q modulo m, with losses in terms of cut weight controlled by the parameter β. Given a reduction family $\mathcal{R}(\beta, q)$, it is thus sufficient to find a solution C' satisfying $\gamma(C') \equiv r \pmod{m'}$ for $m' = \gcd(m, q)$. This is formalized in the following lemma.

Lemma 6 (Reduction lemma) *Let $\mathcal{R}(\beta, q)$ be a reduction family for a* (CCMC) *instance* (G, w, γ, m, r), *and let* $m' = \gcd(m, q)$. *Given a cut* $C' \subsetneq V$, $C' \neq \emptyset$, *with* $\gamma(C') \equiv r \pmod{m'}$, *one can efficiently (in running time polynomial in* $|V|$ *and* m*) obtain a cut* $C \subsetneq V$, $C \neq \emptyset$, *such that*

(i) $w(\delta(C)) \leqslant w(\delta(C')) + \left(\frac{m}{m'} - 1\right)\beta$, *and*
(ii) $\gamma(C) \equiv r \pmod{m}$.

Proof Let $\mathcal{R}(\beta, q) = \{R_1, R_2, \ldots, R_{2m_q-1}\}$ with distinct $u_i \in R_i$ for all $i \in [2m_q - 1]$ as given in item (ii) of Definition 5. We distinguish two cases: Either, there are m_q many vertices among the u_i with $u_i \in C'$, or there are m_q many with $u_i \notin C'$.

In the first case, assume w.l.o.g. that $u_1, \ldots, u_{m_q} \in C'$, and let $U_k := \bigcup_{i=1}^{k} R_i$ for $k \in \{0, \ldots, m_q - 1\}$. We show that for some k, the set $C_k := C' \triangle U_k$ has the desired properties. First observe that all C_k are cuts, as $C_0 = C'$ is a cut, and $u_1 \notin C_k \ni u_{m_q}$ for $k \in [m_q - 1]$. Moreover, $k \leqslant m_q - 1$ implies

$$w(\delta(C_k)) \leqslant w(\delta(C')) + \sum_{i=1}^{k} w(\delta(R_i)) \leqslant w(\delta(C')) + (m_q - 1)\beta. \tag{1}$$

Using $m_q = \frac{m}{\gcd(m,q)} = \frac{m}{m'}$, we see that (1) is precisely point (i) of Lemma 6 for C_k. To conclude, we show that there exists k such that C_k satisfies $\gamma(C_k) \equiv r \pmod{m}$, i.e., point (ii). Using that $\gamma(u) \equiv 0 \pmod{m}$ for all $u \in R_i \backslash \{u_i\}$, and $u_i \in C'$ for all $i \in [m_q]$, we obtain $\gamma(C_k) \equiv \gamma(C') - \sum_{i=1}^{k} \gamma(u_i) \equiv \gamma(C') - kq \pmod{m}$. It thus suffices to find $k \in \{0, \ldots, m_q - 1\}$ with $\gamma(C') - kq \equiv r \pmod{m}$, or equivalently,

$$kq \equiv \gamma(C') - r \pmod{m}. \tag{2}$$

[6] $\gcd(m, q)$ denotes the greatest common divisor of m and q.

By assumption, $\gamma(C') - r \equiv 0 \pmod{m'}$, so $r' := \frac{\gamma(C')-r}{m'} \in \mathbb{Z}$, and $q' := \frac{q}{m'} \in \mathbb{Z}$ because $m' = \gcd(m, q)$. Dividing (2) by m', we obtain the equivalent equation $kq' \equiv r' \pmod{m_q}$, which has a solution $k \in \{0, \ldots, m_q - 1\}$ as $\gcd(q', m_q) = 1$.

The second case, i.e., $u_1, \ldots, u_{m_q} \notin C'$, is similar: C_k always is a cut because $C_0 = C'$ is a cut, and $u_1 \in C_k \not\ni u_{m_q}$ for $k \geq 1$. Equation (1) remains true and implies point (i). For point (ii), we use $\gamma(C_k) \equiv \gamma(C') + \sum_{i=1}^{k} \gamma(u_i)$, and the above analysis results in $kq' \equiv -r' \pmod{m_q}$, admitting a solution $k \in \{0, \ldots, m_q - 1\}$.

Finally, given $\mathcal{R}(\beta, q)$ and C', checking which of the two cases applies can be done efficiently, as well as solving the respective congruence equation for k. Altogether, we conclude that a cut C with the desired properties can be obtained in running time polynomial in $|V|$ and m. □

The above reduction lemma applied with a reduction family $\mathcal{R}(\beta, q)$ allows for reducing the modulus from m to a divisor m' of m, which is strictly smaller than m, as $0 < q < m$. We call such a reduction to a smaller modulus through a reduction family a *reduction step*. Reduction families exist (and can be found efficiently) whenever Theorem 4 fails to guarantee a distribution with the desired properties for Karger-type contraction steps, i.e., whenever $\sum_{v \in V_{\not\equiv 0}} \nu(\{v\})$ is small. In this case, there are many vertices $v \in V_{\not\equiv 0}$ for which $\nu(\{v\})$ is small, i.e., the cut $C_{\{v\}}$ has small value. A subset of these cuts can then be used as a reduction family. This idea is concretized in Theorem 7 below.

Theorem 7 *Let $\mathcal{I}$ be a* (CCMC) *instance with modulus m and let $B > 0$. Assume that $|V_{\not\equiv 0}| \geqslant 4m^2$ and $\sum_{v \in V_{\not\equiv 0}} \nu(\{v\}) \leqslant B \cdot |V_{\not\equiv 0}|$. Then, for some $q \in [m-1]$, one can efficiently (in running time polynomial in $|V|$ and m) obtain a reduction family $\mathcal{R}(2B, q)$ for $\mathcal{I}$.*

Proof The conditions of Theorem 7 imply that at least $|V_{\not\equiv 0}|/2 \geq 2m^2$ many vertices $v \in V_{\not\equiv 0}$ satisfy $\nu(\{v\}) \leqslant 2B$; indeed, for otherwise

$$\sum_{v \in V_{\not\equiv 0}} \nu(\{v\}) > 2B \cdot \frac{|V_{\not\equiv 0}|}{2} = B \cdot |V_{\not\equiv 0}| \ ,$$

However, the above inequality contradicts the second assumption, namely $\sum_{v \in V_{\not\equiv 0}} \nu(\{v\}) \leqslant B \cdot |V_{\not\equiv 0}|$, which implies the claim.

For every one of those $2m^2$ many vertices $v \in V_{\not\equiv 0}$ with $\nu(\{v\}) \leqslant 2B$, there is a corresponding cut $C_{\{v\}}$ in G separating v and $V_{\not\equiv 0} \setminus \{v\}$ of value $w(\delta(C_{\{v\}})) \leqslant 2B$; moreover, $\gamma(C_{\{v\}}) \equiv \gamma(v) \not\equiv 0 \pmod{m}$ because $v \in V_{\not\equiv 0}$. Hence, by the pigeonhole principle, there exists $q \in [m-1]$ such that at least $\frac{2m^2}{m-1} > 2m$ many cuts $C_{\{v\}}$ satisfy $\gamma(C_{\{v\}}) \equiv q \pmod{m}$. Let $v_1, \ldots, v_{2m} \in V_{\not\equiv 0}$ be distinct vertices such that $\{C_{\{v_i\}} | i \in [2m]\}$ are precisely $2m$ such cuts. For $m_q = \frac{m}{\gcd(m,q)}$, the family

$$\mathcal{R} = \left\{C_{\{v_1\}}, \ldots, C_{\{v_{2m_q-1}\}}\right\}$$

is well-defined and fulfils points (i) to (iv) in Definition 5 with parameters $\beta = 2B$ and q. To conclude the proof of Theorem 7, observe that $\mathcal{R}$ can be obtained in running time polynomial in $|V|$ and m by following the above constructive proof. □

A reduction step reduces the modulus m to a divisor strictly smaller than m, hence we can perform at most $\log_2(m)$ many reduction steps, and might end up solving a problem with modulus 1, i.e., an unconstrained minimum cut problem.

Altogether, the ingredients discussed above lead to Algorithm 1. This algorithm requires a guess α for the value of the optimal solution, which we can assume to know up to a factor of $(1+\varepsilon)$ by trying all polynomially many values

$$\alpha \in \{0\} \cup \left\{ (1+\varepsilon)^i \cdot w_{\min} \,\middle|\, 0 \leqslant i \leqslant \lceil \log_{1+\varepsilon}(w_{\text{tot}}/w_{\min}) \rceil \right\} , \tag{3}$$

where $w_{\min} := \min\{w(e) | e \in E,\ w(e) \neq 0\}$ and $w_{\text{tot}} := w(E)$.

Algorithm 1: Contraction-Reduction algorithm for (CCMC).

```
Input: (CCMC) instance $\mathcal{I} = (G, w, \gamma, m, r)$ on $G = (V, E)$, error parameter
       $\rho > 0$, optimal value guess $\alpha \geqslant 0$.
while $|V_{\neq 0}| > \max\left\{4m^2, 2 \cdot \left\lceil \frac{4m}{\rho} \right\rceil\right\}$ and $\sum_{v \in V_{\neq 0}} \nu(\{v\}) > \frac{\rho\alpha}{2m} \cdot |V_{\neq 0}|$ do
    1. Sample a pair $\{u, v\}$ from the distribution $\mathcal{D}$ guaranteed by Theorem 4.
    2. Modify $G$ by contracting the set $\{u, v\}$.
if $|V_{\neq 0}| \leqslant \max\left\{4m^2, 2 \cdot \left\lceil \frac{4m}{\rho} \right\rceil\right\}$ then
    1. For every $S \subseteq V_{\neq 0}$ with $\gamma(S) \equiv r \pmod{m}$, let
           $C_S \in \arg\min\{w(\delta(C)) \mid \emptyset \subsetneq C \subsetneq V,\ C \cap V_{\neq 0} = S\}$ .
    2. Among all cuts $C_S$ obtained in step 1, let $C$ be one of smallest value
       $w(\delta(C))$.
    return Cut corresponding to $C$ in input graph before contractions.
else
    1. Use Theorem 7 to get reduction family $\mathcal{R}(\beta, q)$ for $\beta = \frac{\rho\alpha}{m}$ and some
       $q \in [m-1]$.
    2. Let $m' = \gcd(m, q)$. Apply Algorithm 1 recursively to
       $\mathcal{I}' = (G, w, \gamma, m', r)$ with error parameter $\rho$ and optimal value guess $\alpha$ to
       obtain a solution $C'$ of $\mathcal{I}'$.
    3. Apply Lemma 6 to get a solution $C$ of $\mathcal{I}$ from $C'$ and $\mathcal{R}(\beta, q)$.
    return Cut corresponding to $C$ in input graph before contractions.
```

As long as $|V_{\neq 0}|$ is large, Algorithm 1 contracts two vertices of $V_{\neq 0}$ whenever the conditions of Theorem 4 are met with $c = {}^{4m}/_{\rho}$. Note that every contraction step reduces the number of vertices in $V_{\not\equiv 0}$ by one or two, depending on whether $\gamma(u)+\gamma(v)\not\equiv 0 \pmod{m}$ or not. The if-block in Algorithm 1 performs the enumeration step described earlier once there are at most $\max\left\{4m^2, 2\cdot\left\lceil\frac{4m}{\rho}\right\rceil\right\}$ vertices left in $V_{\not\equiv 0}$. If neither of the above is possible, then Theorem 7 and Lemma 6 allow for a reduction step, which is executed in the else-block, where we recursively invoke Algorithm 1 on an instance with strictly smaller modulus. Combining the above insights, we can prove the following guarantee for Algorithm 1.

Theorem 8 *Consider a* (CCMC) *instance* (G, w, γ, m, r) *with optimal solution value* OPT. *Let* $\alpha \geqslant \mathrm{OPT}$ *and* $\rho > 0$. *Algorithm* 1 *is an efficient procedure with running time bounded by* $|V|^{O(1)} + 2^{O(m^2 + m/\rho)}$ *that, by using* α *as an optimal value guess and* ρ *as error parameter, returns a solution with value at most* $\mathrm{OPT} + \rho\,\alpha \log_2 m$ *with probability at least* $1/\binom{|V|}{\lceil 4m/\rho\rceil}$.

Proof The only randomized step of Algorithm 1 occurs in the while-loop, where pairs $\{u, v\}$ for contraction are sampled. For the analysis, we fix an optimal solution C_0 of $\mathcal{I}$, and first assume that no contraction is bad w.r.t. C_0, i.e., that no contraction step contracts two vertices on different sides of C_0 throughout Algorithm 1. Under this assumption, we prove by induction on m that Algorithm 1 returns a cut C satisfying $w(\delta(C)) \leqslant \mathrm{OPT} + \rho\alpha \log_2 m$.

If $m = 1$, then $V_{\not\equiv 0} = \emptyset$, hence the algorithm directly executes the if-block, where an unconstrained minimum cut problem is solved, giving an exact solution. This reflects that for $m = 1$, (CCMC) is an unconstrained minimum cut problem.

Now let $m > 1$. If no bad contraction is performed, C_0 remains feasible after the termination of the while-loop, and α remains an upper bound on the optimal solution value in the new contracted graph. If $|V_{\not\equiv 0}| \leqslant \max\left\{4m^2, 2\cdot\left\lceil\frac{4m}{\rho}\right\rceil\right\}$, then, in the if-block, all remaining options are enumerated, and an optimal solution is found. Else, we have $|V_{\not\equiv 0}| \geqslant 4m^2$ and $\sum_{v\in V_{\not\equiv 0}} \nu(\{v\}) \leqslant \frac{\rho\alpha}{2m}\cdot|V_{\not\equiv 0}|$, hence by Theorem 7 with $B = \frac{\rho\alpha}{2m}$, a reduction family $\mathcal{R}(\frac{\rho\alpha}{m}, q)$ can be found efficiently. We have $q \in [m-1]$ by Theorem 7, so $m' = \gcd(m, q) < m$. Thus, by the inductive assumption, the recursive application of Algorithm 1 in step 2 of the else-block returns a solution $C' \subsetneq V$, $C' \neq \emptyset$, of $\mathcal{I}'$ with

$$\gamma(C') \equiv r \pmod{m'} \quad \text{and} \quad w(\delta(C')) \leqslant \mathrm{OPT} + \rho\alpha\log_2(m')\ . \tag{4}$$

Note that in the inequality, we used $\mathrm{OPT}(\mathcal{I}') \leqslant \mathrm{OPT}$, which follows from the fact that C_0 remains feasible for $\mathcal{I}'$. By (4) and Lemma 6, the solution C of $\mathcal{I}$ constructed in step 3 is a cut, satisfies $\gamma(C) \equiv r \pmod{m}$, and

$$\begin{aligned} w(\delta(C)) &\leqslant w(\delta(C')) + \left(\frac{m}{m'} - 1\right)\frac{\rho\alpha}{m} \\ &\leqslant \mathrm{OPT} + \rho\alpha(\log_2 m' + 1) \leqslant \mathrm{OPT} + \rho\alpha\log_2 m\ , \end{aligned}$$

where the last inequality follows from $m' \leqslant m/2$, as m' is a divisor of m and strictly smaller than m. This concludes the induction. Thus, if no bad contraction steps are performed, a solution of value at most $\mathrm{OPT} + \rho\alpha\log_2 m$ is returned.

We now show that with probability at least $1/\binom{|V|}{\lceil 4m/\rho \rceil}$, no contraction step is bad w.r.t. C_0 throughout all recursive calls of Algorithm 1. Assume that overall, there are q contraction steps, and let $m_1, \dots, m_q \in \mathbb{Z}_{\geq 0}$ and $s_1, \dots, s_q \in \mathbb{Z}_{\geq 0}$ be such that when the i^{th} contraction step is performed, the modulus is m_i and $|V_{\not\equiv 0}| = s_i$. By the condition in the while-loop, we know that when the i^{th} contraction is performed, then $\sum_{v \in V_{\not\equiv 0}} \nu(\{v\}) > \frac{\rho\alpha}{2m_i} \cdot |V_{\not\equiv 0}|$. Hence, by Theorem 4 with $c = \frac{4m_i}{\rho}$,

$$\Pr[\text{contraction } i \text{ is bad w.r.t. } C_0] \leqslant \frac{4m_i}{\rho \cdot |V_{\not\equiv 0}|} \leqslant \frac{k_i}{s_i} \ ,$$

where $k_i := \left\lceil \frac{4m_i}{\rho} \right\rceil$, holds for all $i \in [q]$. Consequently,

$$\Pr[\text{no contraction is bad w.r.t. } C_0] \geqslant \prod_{i=1}^{q} \left(1 - \frac{k_i}{s_i}\right) \ . \tag{5}$$

To bound the latter product from below, we exploit the following three facts.

(i) For $x \in \mathbb{Z}$, let $\kappa(x) := \max\{k_i | i \in [q] \colon x > 2k_i\}$, which is finite if $x > \min\{2k_1, \dots, 2k_q\}$. As $s_i > 2k_i$ by the contraction conditions, $\kappa(s_i) \geqslant k_i$.

(ii) The sequence $(s_i)_{i \in [q]}$ measures the size of $V_{\not\equiv 0}$, which decreases in each contraction step, and never increases. In particular,

$$|V| \geq s_1 > s_2 > \ldots > s_q \geq 2k_q + 1 \ .$$

(iii) The sequence $(k_i)_{i \in [q]}$ is decreasing, and it drops precisely when reduction steps are performed. Thus, there exists $p \leqslant q$ and $a_1, \dots, a_p \in [q]$ such that $k_{a_1} > k_{a_2} > \ldots > k_{a_p}$ are all values taken by the sequence $(k_i)_{i \in [q]}$. Note that in particular, $k_{a_1} = k_1 = \left\lceil \frac{4m}{\rho} \right\rceil$.

Using this, we get

$$\begin{aligned}
\prod_{i=1}^{q} \left(1 - \frac{k_i}{s_i}\right) &\overset{\text{(i)}}{\geqslant} \prod_{i=1}^{q} \left(1 - \frac{\kappa(s_i)}{s_i}\right) \overset{\text{(ii)}}{\geqslant} \prod_{i=2k_q+1}^{|V|} \left(1 - \frac{\kappa(i)}{i}\right) \\
&\overset{\text{(iii)}}{\geqslant} \prod_{i=2k_{a_p}+1}^{2k_{a_{p-1}}} \left(1 - \frac{k_{a_p}}{i}\right) \cdot \ldots \cdot \prod_{i=2k_{a_2}+1}^{2k_{a_1}} \left(1 - \frac{k_{a_2}}{i}\right) \cdot \prod_{i=2k_{a_1}+1}^{|V|} \left(1 - \frac{k_{a_1}}{i}\right) \\
&= \frac{\binom{2k_{a_p}}{k_{a_p}}}{\binom{2k_{a_{p-1}}}{k_{a_p}}} \cdot \ldots \cdot \frac{\binom{2k_{a_2}}{k_{a_2}}}{\binom{2k_{a_1}}{k_{a_2}}} \cdot \frac{\binom{2k_{a_1}}{k_{a_1}}}{\binom{|V|}{k_{a_1}}} \geqslant \frac{\binom{2k_{a_p}}{k_{a_p}}}{\binom{|V|}{k_{a_1}}} \geqslant \frac{1}{\binom{|V|}{\lceil 4m/\rho \rceil}} \ .
\end{aligned} \tag{6}$$

For the penultimate inequality we use $\binom{2k_{a_j}}{k_{a_j}} \geq \binom{2k_{a_j}}{k_{a_{j+1}}}$ for $j \in [p-1]$, and the last inequality follows from $\binom{2k_{a_p}}{k_{a_p}} \geq 1$. Combining (5) and (6), we get the desired bound.

To prove the running time guarantee, first assume that $|V| \leqslant m$, in which case the enumeration step is executed directly, giving a running time of $2^{O(|V|)}$, which is less than the claimed bound. Thus, assume that $|V| > m$ from now on. Overall, there are less than $|V|$ many contraction steps, each with a running time polynomial in $|V|$ by Theorem 4, giving a bound of the form $|V|^{O(1)}$ for all contraction steps together. Moreover, there are at most $\log_2(m)$ many reduction steps as observed earlier, each with running time polynomial in m and $|V|$ by Lemma 6 and Theorem 7. As $|V| > m$ by assumption, this shows that reduction steps take time bounded by $|V|^{O(1)}$, as well. Finally, the algorithm enumerates subsets of a set of size at most $\max\{4m^2, 2 \cdot \lceil {}^{4m}/_{\rho} \rceil\}$, i.e., $2^{O(m^2 + {}^{m}/_{\rho})}$ many sets. Adding these bounds gives the result. □

Guessing the optimal solution value up to a factor $(1+\varepsilon)$ and repeating Algorithm 1 polynomially often independently implies our main result, Theorem 1.

Proof of Theorem 1 For all polynomially many values of α given in (3), we run Algorithm 1 with $\rho = \frac{\varepsilon}{(1+\varepsilon)\log_2(m)}$ for $\binom{|V|}{\lceil {}^{4m}/_{\rho} \rceil} \log |V|$ many times independently, and we return the best solution found over all iterations. By Theorem 8, for $\alpha \in [\mathrm{OPT}, (1+\varepsilon)\mathrm{OPT})$, a single iteration returns a $(1+\varepsilon)$-approximate solution with probability at least ${}^{1}/_{\binom{|V|}{\lceil {}^{4m}/_{\rho} \rceil}}$. Hence, among all iterations with this α, a $(1+\varepsilon)$-approximate solution is found with probability at least

$$1 - \left(1 - \frac{1}{\binom{|V|}{\lceil {}^{4m}/_{\rho} \rceil}}\right)^{\binom{|V|}{\lceil {}^{4m}/_{\rho} \rceil} \cdot \log |V|} \geqslant 1 - \mathrm{e}^{-\log |V|} = 1 - \frac{1}{|V|} \,. \qquad \square$$

□

From the above proof of Theorem 1, we can immediately obtain a bound on the running time of our PRAS for (CCMC): There are

$$\left\lceil \log_{1+\varepsilon}\left(\frac{w_{\mathrm{tot}}}{w_{\min}}\right)\right\rceil + 2 = O\left(\frac{\log |V| + \log(\frac{w_{\max}}{w_{\min}})}{\log(1+\varepsilon)}\right)$$

many guesses for the value of α, where we exploit that $w_{\mathrm{tot}} \leqslant |V|^2 \cdot w_{\max}$. For each of these guesses, we run Algorithm 1 for $\binom{|V|}{\lceil {}^{4m}/_{\rho} \rceil} \log |V| = |V|^{O\left(\frac{1+\varepsilon}{\varepsilon} m \log m\right)}$ many times independently. Finally, by Theorem 8, every such run takes time $|V|^{O(1)} + 2^{O\left(m^2 + \frac{1+\varepsilon}{\varepsilon} m \log m\right)}$. Together, this gives a running time bound of the form

$$\frac{\log(\frac{w_{\max}}{w_{\min}})}{\log(1+\varepsilon)} \cdot |V|^{O\left(\frac{1+\varepsilon}{\varepsilon} m \log m\right)} \cdot 2^{O(m^2)} \,,$$

which can be simplified to $\log(\frac{w_{\max}}{w_{\min}}) \cdot |V|^{O\left(\frac{m \log m}{\varepsilon}\right)} \cdot 2^{O(m^2)}$ for $\varepsilon < 1$. We remark that, by using Megiddo's parametric search technique [25,26], we can get rid of the factor $\log(\frac{w_{\max}}{w_{\min}})$ and thus obtain a strongly polynomial time algorithm at the expense of a larger constant in the exponent of $|V|$ that is hidden by the O-notation.

3 Good contraction distributions through splitting-off

To obtain a good distribution for Karger-type contractions (Theorem 4), we construct a weighted auxiliary graph $H = (V_{\neq 0}, F)$, and then select a pair of vertices $\{u, v\} \in F$ for contraction in G with probabilities proportional to the edge weights in H. The construction of H is based on splitting-off techniques, which, loosely speaking, allow for modifying a given graph such that certain connectivity properties are preserved. Our interest lies in preserving the values $\nu(\{v\}) = \mu_{G,w}(\{v\}, V_{\neq 0}\setminus\{v\})$ for all $v \in V_{\neq 0}$, where we use the notation $\mu_{G,w}(A, B){:=}\min\{w(\delta(C)) | A \subseteq C \subseteq V\setminus B\}$. This is achieved by the following theorem.

Theorem 9 *Let $G = (V, E)$ be a graph with edge weights $w\colon E \to \mathbb{R}_{\geqslant 0}$, and let $Q \subseteq V$. There is a strongly polynomial time algorithm to obtain a graph $H = (Q, F)$ and edge weights $w_H\colon F \to \mathbb{R}_{\geqslant 0}$ such that*

(i) $w_H(\delta_H(q)) = \mu_{G,w}(\{q\}, Q\setminus\{q\})$ for all $q \in Q$, and
(ii) $w_H(\delta_H(C \cap Q)) \leqslant w(\delta_G(C))$ for all $C \subseteq V$.

We remark that similar theorems are known in literature, and there are various ways to derive the version above, which we need for our purposes. A splitting-off theorem of Lovász [22] gives the existential result in an unweighted setting, and allows an immediate generalization to the weighted setting. Alternatively, the non-algorithmic version of Theorem 9 can also be seen to be an implication of a result on weakly parsimonious set functions by Bertsimas and Teo [5], which uses splitting-off, as well. In order to obtain strongly polynomial algorithms complementing the existential results, ideas of Frank [11] can be used. A full proof of Theorem 9 combining Lovász' splitting-off result and Frank's ideas is given in Sect. 3.1. Let us now show how Theorem 9 is used to prove Theorem 4.

Proof of Theorem 4 Apply Theorem 9 to (G, w) with $Q = V_{\neq 0}$ to obtain the graph $H = (V_{\neq 0}, F)$ with weights w_H. The distribution $\mathcal{D}$ over vertex pairs $\{u, v\}$ we use is given by choosing $\{u, v\} \in F$ with probability proportional to $w_H(\{u, v\})$. By Theorem 9 (i),

$$2 \cdot w_H(F) = \sum_{v \in V_{\neq 0}} w_H(\delta_H(v)) = \sum_{v \in V_{\neq 0}} \mu_{G,w}(\{v\}, Q\setminus\{v\}) = \sum_{v \in V_{\neq 0}} \nu(\{v\}).$$

If C is a solution of $\mathcal{I}$ with $w(\delta(C)) \leqslant \alpha$, then by choice of $\mathcal{D}$ and the above,

$$\Pr_{\{u,v\}\sim\mathcal{D}}\big[|\{u, v\} \cap C| = 1\big] = \frac{w_H(\delta_H(C \cap V_{\neq 0}))}{w_H(F)} \leqslant \frac{2 \cdot w(\delta_G(C))}{\sum\limits_{v \in V_{\neq 0}} \nu(\{v\})} \leqslant \frac{c}{|V_{\neq 0}|},$$

as desired, where the inequalities are due to Theorem 9 (ii), $w(\delta_G(C)) \leqslant \alpha$, and the assumption $\sum_{v \in V_{\neq 0}} \nu(\{v\}) > \frac{2\alpha}{c} \cdot |V_{\neq 0}|$ in Theorem 4. To finish the proof, observe that the auxiliary graph H can be constructed in time polynomial in $|V|$ by Theorem 9, and the sampling procedure can be realized in running time polynomial in $|V|$, too. □

3.1 Proof of Theorem 9

As indicated above, Theorem 9 is a consequence of splitting-off techniques from Graph Theory, a fundamental tool dating back to the ’70s [22–24]. In this context, a graph is typically modified by repeatedly *splitting off* two edges from a vertex v, i.e., replacing two non-parallel edges $\{v, x\}$ and $\{v, y\}$ by a new edge $\{x, y\}$, or deleting two parallel edges incident to v. Denoting $\mu_G(A, B) := \min\{|\delta_G(C)| \mid A \subseteq C \subseteq V\backslash B\}$ for a graph $G = (V, E)$ and $A, B \subseteq V$, Lovász proved the following.

Theorem 10 (Lovász [22]) *Let $G = (V, E)$ be Eulerian, let $Q \subseteq V$, and let $v \in V\backslash Q$. For every edge $\{v, x\} \in E$, there exists another edge $\{v, y\} \in E$ such that the graph G' arising from G by splitting off $\{v, x\}$ and $\{v, y\}$ from v satisfies*

$$\mu_G(\{q\}, Q\backslash\{q\}) = \mu_{G'}(\{q\}, Q\backslash\{q\}) \qquad \forall q \in Q \ .$$

Iterative applications of Theorem 10 for fixed $Q \subseteq V$ and $v \in V\backslash Q$ result in a new graph on the vertex set $V\backslash v$ only, without changing the value of minimum cuts separating a single vertex q from $Q\backslash\{q\}$, for all $q \in Q$. We aim for a generalization of this statement to a weighted setting, where the graph $G = (V, E)$ has edge weights $w\colon E \to \mathbb{R}_{\geqslant 0}$, a splitting operation consists of decreasing the weight on two edges $\{v, x\}$ and $\{v, y\}$ by some $\varepsilon > 0$ while increasing the weight on the edge $\{x, y\}$ by ε, and we want the weighted cut values $\mu_{G,w}(\{q\}, Q \setminus \{q\})$ to be invariant. We claim that this is achieved by Algorithm 2. We highlight that efficient weighted versions of other splitting-off results (than Theorem 10) have already been studied extensively (see [3,4,6,11,12,21,27–29]), and our method is heavily inspired by an approach of Frank [11].

```
Algorithm 2: Fractionally splitting off a single vertex.
  Input: Graph $G = (V, E)$ with edge weights $w\colon E \to \mathbb{R}_{\geqslant 0}$, $Q \subseteq V$,
         $v \in V \setminus Q$.
  foreach $x, y \in N_G(v) := \{z \in V \setminus \{v\} \mid \{v, z\} \in E\}$, $x \neq y$ do
      foreach $q \in Q$ do
          Calculate the min cut sizes
            $c_1^q = \mu_{G,w}(\{q\}, Q \setminus \{q\})$, $c_2^q = \mu_{G,w}(\{q, v\}, (Q \setminus \{q\}) \cup \{x, y\})$,[7]
                  and $c_3^q = \mu_{G,w}(\{q, x, y\}, (Q \setminus \{q\}) \cup \{v\})$.
      Split off $\varepsilon$ from $e_1 = \{v, x\}$ and $e_2 = \{v, y\}$, where
            $\varepsilon = \min_{q \in Q} \min \left\{ (c_2^q - c_1^q)/2, (c_3^q - c_1^q)/2, w(e_1), w(e_2) \right\}$ .
  return Modified graph G with v deleted and modified weights w.
```

[7] If $q \in \{x, y\}$, then $c_2^q = \mu_{G,w}(\{q, v\}, (Q\backslash\{q\}) \cup \{x, y\})$ is the value of an infeasible cut problem (because both arguments of $\mu_{G,w}$ contain q), which we interpret as ∞.

In each iteration of the outer for-loop in Algorithm 2, we split off $\varepsilon \geqslant 0$ from $\{v, x\}$ and $\{v, y\}$, with ε chosen maximal so that all weights remain non-negative and the connectivities of interest are preserved. This choice of ε implies that once the outer for-loop terminated, there is no pair of edges incident to v from which a positive weight can be split off. Uniformly scaling all weights of this remaining graph to even integral weights (which we interpret as edge multiplicities) and employing Theorem 10, we can prove that there can only be a single edge with positive weight incident to v in the remaining graph, which we can thus safely delete without affecting connectivities within $V \backslash \{v\}$.

The following lemma summarizes the guarantees that we thereby obtain for Algorithm 2.

Lemma 11 *Let $G = (V, E)$ be a graph with edge weights $w \colon E \to \mathbb{R}_{\geqslant 0}$, let $Q \subsetneq V$ and $v \in V \backslash Q$. On this input, Algorithm 2 returns, in running time dominated by $\mathcal{O}(|V|^3)$ many minimum s–t cut computations in contractions of (G, w), a graph $H = (V \backslash \{v\}, F)$ with edge weights $w_H \colon F \to \mathbb{R}_{\geqslant 0}$ such that*

(i) $\mu_{H,w_H}(\{q\}, Q \backslash \{q\}) = \mu_{G,w}(\{q\}, Q \backslash \{q\})$ for all $q \in Q$, and
(ii) $w_H(\delta_H(C \backslash \{v\})) \leqslant w(\delta_G(C))$ for all $C \subseteq V$.

Proof Consider a splitting operation performed in Algorithm 2 on edges $e_1 = \{v, x\}$ and $e_2 = \{v, y\}$ for $x \neq y$, i.e., the weights on e_1 and e_2 are decreased by ε while the weight on $\{x, y\}$ is increased by ε. Such an operation changes the values of precisely those cuts that separate v from $\{x, y\}$, and their values all decrease by 2ε. Thus, cut values never increase in splitting steps, and neither do they when deleting v at the end of Algorithm 2, implying point (ii).

Moreover, observe that c_2^q and c_3^q computed in Algorithm 2 are precisely the minimum values of cuts separating q from $Q \backslash \{q\}$ and v from $\{x, y\}$. Thus, choosing $\varepsilon \leqslant \min \left\{ (c_2^q - c_1^q)/2, (c_3^q - c_1^q)/2 \right\}$ guarantees that the values of these cuts do not decrease below $\mu(\{q\}, Q \backslash \{q\})$. In other words, $\mu(\{q\}, Q \backslash \{q\})$ remains invariant under all splitting operations in Algorithm 2. Additionally, $\varepsilon \leqslant \min\{w(e_1), w(e_2)\}$ ensures that edge weights are always non-negative.

The extremal choice of ε implies that after the splitting operation is applied to a pair of edges (e_1, e_2), either one of $w(e_1)$ and $w(e_2)$ is zero, or there is a vertex $q \in Q$ and a cut $C \subseteq V$ with the following property: C separates q from $Q \backslash \{q\}$ as well as v from $\{x, y\}$, and $w(\delta(C)) = \mu(q, Q \backslash \{q\})$. We call such a cut *tight* for the pair (e_1, e_2), as any further reduction of $w(e_1)$ or $w(e_2)$ would reduce the value of C and hence also $\mu(q, Q \backslash \{q\})$. Observe that once a cut is tight for a pair of edges, it remains tight under all subsequent splitting operations.

Let (G', w') be the weighted graph obtained from (G, w) after performing all $\mathcal{O}(n^2)$ splitting operations in Algorithm 2. We claim that (G', w') has at most one edge with non-negative weight incident to v. If so, deleting v (and all its incident edges) from G' does not reduce $\mu(q, Q \backslash \{q\})$ for any $q \in Q$, hence the resulting graph has the desired properties. To see the claim, assume by contradiction that in (G', w'), there is more than one edge with positive weight incident to v. Then, there is a tight cut for each pair of such edges, implying that none of the edge weights can be reduced without reducing $\mu(q, Q \backslash \{q\})$ for some $q \in Q$. Now scale w' by an integer $M > 0$

such that all edge weights become even integers, and interpret these edge weights as edge multiplicities. Doing so, we obtain an Eulerian graph to which Theorem 10 is applicable, resulting in a pair of (potentially parallel) edges that can be split off from v without affecting $\mu(q, Q\backslash\{q\})$. But deleting an edge incident to v in this new graph corresponds to reducing the weight of the corresponding edge in G' by $^1/_M$. By assumption, the latter does reduce $\mu(q, Q\backslash\{q\})$, a contradiction.

Finally, observe that the values $\mu_{G,w}(A, B)$ needed in Algorithm 2 are infinite if $A \cap B \neq \emptyset$, and else they can be computed as the values of minimum s–t cuts in the contraction of G where A and B are contracted to vertices s and t, respectively. Three such values are computed for every triple (x, y, q) consisting of $x, y \in N_G(v)$ with $x \neq y$ and $q \in Q$, and these $\mathcal{O}(|V|^3)$ many minimum s–t cut computations indeed dominate the overall running time. □

Applying Lemma 11 iteratively for all $v \in V\backslash Q$ reduces the graph G to the vertex set Q while maintaining the desired cut sizes, and thus immediately yields Theorem 9.

As indicated earlier in this section, there are different versions of splitting-off techniques. Some better known ones, for which strongly polynomial algorithms are already known, preserve pairwise connectivities among vertices in $Q \subseteq V$ instead of fulfilling the guarantees stated in Theorem 10. In Section 5, we show that under slightly stronger assumptions, these standard splitting-off techniques can also be used to obtain contraction distributions with the properties given in Theorem 4, with the necessary stronger assumptions leading to a weaker running time guarantee for Algorithm 1.

4 Further structural properties and their implications

Karger's mininum cut algorithm also provides a means of proving that a minimum cut problem has only polynomially many optimal solutions, and repeated applications of Karger's algorithm can find all these solutions with high probability. As discussed, analogous results cannot hold for (CCMC) problems. Note that in contrast to Karger's algorithm, our Contraction-Reduction Algorithm does not contract pairs of vertices until only two of them are left, but it stops early and terminates in an enumeration phase, solving reduced s–t cut problems. Following the spirit of the above-mentioned implications of Karger's algorithm, we obtain a structural result on these s–t cut instances. To state this result in full generality, we need the following congruency-constrained version of minimum s–t cut problems.

Congruency-Constrained Minimum s–t Cut (s–t CCMC): Let $G = (V, E)$ be an undirected graph with edge weights $w\colon E \to \mathbb{R}_{\geqslant 0}$ and let $\gamma\colon V \to \mathbb{Z}_{\geqslant 0}$. Let $m \in \mathbb{Z}_{>0}$ and $r \in \mathbb{Z}_{\geqslant 0}$, and let $s, t \in V$ be two distinct vertices. The task is to find a minimizer of

$$\min\left\{ w(\delta(C)) \;\middle|\; \begin{array}{c} s \in C \subseteq V\backslash\{t\}, \\ \sum_{v\in C}\gamma(v) \equiv r \pmod{m} \end{array} \right\}. \qquad (s\text{–}t \text{ CCMC})$$

Note that (s–t CCMC) problems can easily be modeled by (CCMC) problems if one allows to increase the modulus by an additional factor.[7] The subsequent theorem shows that the opposite reduction can be done, as well: Every (CCMC) problem can be reduced to polynomially many (s–t CCMC) problems with a smaller modulus.

Theorem 12 *Consider a* (CCMC) *problem on* $G = (V, E)$ *with constant modulus* $m > 1$ *and nonzero optimal value, and let* $\kappa \geq 1$ *be a constant. Then there is an efficient randomized algorithm returning* $\text{poly}(|V|)$ *many* (s–t CCMC) *instances that (i) are defined on contractions of* G *with modified vertex multiplicities, and (ii) have a modulus that is a divisor of* m *strictly smaller than* m*, such that the following holds with high probability, where* OPT *denotes the optimal solution value of the initial* (CCMC) *problem: A cut* $C \subsetneq V$*,* $C \neq \emptyset$*, is a solution to the initial* (CCMC) *problem of value at most* $\kappa \cdot \text{OPT}$ *if and only if* C *is a feasible solution of value at most* $\kappa \cdot \text{OPT}$ *in one of the returned* (s–t CCMC) *instances.*

While the reduction of the modulus m obtained in the above theorem looks promising, the additional hurdle introduced by the transition to (s–t CCMC) instances seems to be substantial. We know efficient algorithms only for very special cases of (s–t CCMC) instances, one of them being the case of prime moduli, where a reduction to congruency-constrained submodular minimization is possible. This can be exploited to prove Theorem 2.

Proof of Theorem 2 Let $\mathcal{I}$ be the given (CCMC) instance with modulus m that is a product of two primes. Let C be an optimal solution of $\mathcal{I}$ and denote its value by OPT. If $\text{OPT} = 0$, an optimal solution can be found easily by contracting the components and finding a union of them satisfying the congruency constraint. Else, an application of Theorem 12 to $\mathcal{I}$ with $\kappa = 1$ results in a polynomial number of (s–t CCMC) instances $\mathcal{I}_1, \ldots, \mathcal{I}_\ell$. By Theorem 12, C is a feasible solution to at least one of these instances with high probability. On the other hand, Theorem 12 also asserts that the instances $\mathcal{I}_1, \ldots, \mathcal{I}_\ell$ are defined on contractions of the initial graph G with weights induced by the initial weights, hence their optimal values are all at least OPT. Thus, we conclude that with high probability, C is an optimal solution to at least one of the instances $\mathcal{I}_1, \ldots, \mathcal{I}_\ell$.

To conclude Theorem 2, it is enough to show that the instances $\mathcal{I}_1, \ldots, \mathcal{I}_\ell$ can all be solved in polynomial time. To see this, fix an instance I_k. By Theorem 12, its modulus m' is a divisor of m that is strictly smaller than m. As m is a product of two primes, m' equals 1 or is a prime number. In the first case, $\mathcal{I}_k$ is an unconstrained minimum s–t cut problem, which can be solved efficiently. In the other case, $\mathcal{I}_k$ is a (s–t CCMC) instance with modulus equal to a prime number. This problem can easily be seen to be a special case of congruency-constrained submodular function minimization with prime modulus, which can be solved efficiently and to optimality as shown in [30]. □

Finally, if the modulus of the input problem in Theorem 12 is a prime number, the only feasible reduction of the modulus to one of its divisors is a reduction to modulus

[7] An (s–t CCMC) instance $(G, w, \gamma, m, r, s, t)$ is captured by the (CCMC) instance $(G, w, \widehat{\gamma}, \widehat{m}, \widehat{r})$, where $\widehat{\gamma}(v) = 3 \cdot \gamma(v)$ for $v \notin \{s, t\}$, $\widehat{\gamma}(s) = 3 \cdot \gamma(s) + 1$, $\widehat{\gamma}(t) = 3 \cdot \gamma(t) + 2$, $\widehat{m} = 3 \cdot m$, and $\widehat{r} = 3 \cdot r + 1$, for example.

1—and hence, to s–t cut problems without congruency constraint, which we exploit to prove Theorem 3.

Proof of Theorem 3 An application of Theorem 12 to the given instance with the given parameter κ results in polynomially many (s–t CCMC) instances. As the modulus of the given instance is a prime number, Theorem 12 implies that all the returned instances have modulus 1, i.e., they are in fact minimum s–t cut problems (without congruency constraint). Thus, Theorem 12 asserts that these instances have precisely the properties claimed by Theorem 3. □

4.1 Proof of Theorem 12

In this section, we show that Theorem 12 can be deduced from Algorithm 1. To this end, we add some further insights to the discussion of Algorithm 1 given in Sect. 2. In particular, observe that during a call of the algorithm on a (CCMC) instance $\mathcal{I} = (G, w, \gamma, m, r)$, the input graph G is repeatedly modified by random contractions, until the if-block of Algorithm 1 is reached (potentially only after several recursive calls to itself). Within the if-block, problems of the form

$$\min\{w(\delta(C)) | \emptyset \subsetneq C \subsetneq V,\ C \cap V_{\not\equiv 0} = S\} \tag{7}$$

are solved for certain sets $S \subseteq V_{\not\equiv 0}$. Problems of this type can be immediately reduced to minimum s–t cut problems in a further contracted graph: If $S \notin \{\emptyset, V_{\not\equiv 0}\}$, then contract S and $V_{\not\equiv 0} \backslash S$ to vertices s and t, respectively, and the problem in (7) is equivalent to the minimum s–t cut problem in the contracted graph. If $S = \emptyset$, contract $V_{\not\equiv 0}$ to a vertex t, and a solution of (7) can be obtained by solving the minimum v–t cut problems for all $v \in V \backslash V_{\not\equiv 0}$ and returning the solution of minimum value. Similarly, if $S = V_{\not\equiv 0}$, contract $V_{\not\equiv 0}$ to a vertex s, and the best solution among all solutions to the minimum s–v cut problems for $v \in V \backslash V_{\not\equiv 0}$ solves (7).

Recall that these minimum s–t cut instances on contractions of G come with weights and vertex multiplicities induced by the original weights w and vertex multiplicities γ such that any cut C in the contracted graph has the same weight $w(\delta(C))$ and value $\gamma(C)$ as the corresponding cut in the initial graph. Hence, after imposing the original congruency constraint $\gamma(C) \equiv r \pmod{m}$, we obtain ($s$–$t$ CCMC) instances which we call the instances *reached by* Algorithm 1. Note that these instances have modulus m equal to the input modulus, and not the potentially smaller modulus that is used in the call where the if-block is reached. The instances defined this way have several useful properties, and we will see that they are essentially the instances claimed by Theorem 12. More precisely, the only missing property compared to the instances in Theorem 12 is that they still have modulus m. In Lemma 17, we will see that their structure allows for reducing the modulus to a divisor of m that is strictly smaller than m. Finally, analogous to the proof of Theorem 1, to obtain a sufficiently high success probability, we will run Algorithm 1 multiple times independently, and consider all (s–t CCMC) problems reached by these runs to construct the desired family of (s–t CCMC) problems as claimed in Theorem 12.

We start with two quick observations. First, note that the enumeration is done only if $V_{\not\equiv 0}$ reaches constant size, hence there are at most $\operatorname{poly}(|V|)$ many candidates $S \subseteq V_{\not\equiv 0}$ to be enumerated over. For every such choice of S, Algorithm 1 reaches either a single or linearly many (s–t CCMC) instances. Combining these arguments, we obtain Observation 13.

Observation 13 *The family of* (s–t CCMC) *instances reached by Algorithm* 1 *in a single run has size at most* $\operatorname{poly}(|V|)$.

Additionally, note that contractions and the transition from a global cut problem to an s–t cut problem for certain vertices s and t of the graph only reduce the set of feasible solutions, implying Observation 14.

Observation 14 *A cut C that is feasible for an* (s–t CCMC) *instance reached by Algorithm 1 is feasible for the input problem, and the weight of the cut is the same with respect to the two instances.*

For the other direction, we saw in the proof of Theorem 8 that for a suitable guess α of the optimal solution value, an optimal solution C of the input (CCMC) problem is not destroyed in the random contraction phase of Algorithm 1 (i.e., none of the contractions are applied to two vertices lying on different sides of C) with probability at least $1/\operatorname{poly}(|V|)$. The following lemma shows that with a slightly larger choice of the optimal solution value guess, this result extends to almost-minimum cuts.

Lemma 15 *Let $\mathcal{I}$ be a* (CCMC) *instance with optimal solution value denoted by* OPT, *let $\kappa \geqslant 1$, and let $\mathcal{F}$ be the family of* (s–t CCMC) *instances reached by Algorithm* 1 *on input $\mathcal{I}$ with optimal value guess $\alpha \geqslant \kappa \cdot \mathrm{OPT}$ and error parameter $\rho > 0$. Then, the probability that a feasible solution C of $\mathcal{I}$ with value at most $\kappa \cdot \mathrm{OPT}$ is also feasible for at least one of the instances in $\mathcal{F}$ is at least* $1/\binom{|V|}{\lceil 4m/\rho \rceil}$.

Proof Fix a feasible solution C of $\mathcal{I}$ with $w(\delta(C)) \leqslant \kappa \cdot \mathrm{OPT}$. Let $m_i \in \mathbb{Z}_{\geq 0}$ and $s_i \in \mathbb{Z}_{\geq 0}$ denote the modulus and the size of $V_{\not\equiv 0}$, respectively, when the i^{th} contraction step is performed. By assumption, $\alpha \geqslant \kappa \cdot \mathrm{OPT}$, hence Theorem 4 with $c = 2m_i/\rho$ guarantees that

$$\Pr[\text{contraction } i \text{ is bad w.r.t. } C] \leqslant \frac{4m_i}{\rho \cdot |V_{\not\equiv 0}|} \leqslant \frac{k_i}{s_i} \ ,$$

where we define $k_i := \left\lceil \frac{4m_i}{\rho} \right\rceil$. Following the very same reasoning as in the proof of Theorem 8, we get

$$\Pr\left[\begin{array}{c}\text{no random contraction is bad w.r.t. } C\\ \text{throughout a full run of Algorithm 1 on } \mathcal{I}\end{array}\right] \geqslant \frac{1}{\binom{|V|}{k_1}} = \frac{1}{\binom{|V|}{4m/\rho}} \ .$$

Thus, with the above probability, C is still feasible once Algorithm 1 reaches the if-block. In this case, among all (s–t CCMC) instances reached by Algorithm 1 in the if-block, the cut C is feasible for at least the one instance reached when choosing $S = V_{\not\equiv 0} \cap C$. This concludes the proof. □

Besides preserving almost-minimum cuts with inverse polynomial probability, the (s–t CCMC) instances reached by Algorithm 1 have structured vertex multiplicities γ as stated by Lemma 16 below. This structure directly reflects the enumeration step performed in the if-block of Algorithm 1, where the choice of a suitable subset $S \subseteq V_{\not\equiv 0}$ guarantees that the remaining congruency constraint (note that at this stage, the modulus may have reduced to a divisor m_0 of m) is satisfied for every feasible solution of the resulting s–t cut problem. Showing that we must have $m_0 > 1$ under a mild assumption on the algorithm parameters α and ρ, we obtain the following.

Lemma 16 *Let $\mathcal{I} = (G, w, \gamma, m, r)$ be a* (CCMC) *instance such that $m > 1$ and denote its optimal value by* OPT. *Let $\mathcal{I}' = (G', w', \gamma', m, r, s, t)$ be an* (s–t CCMC) *instance reached by Algorithm* 1 *in a call on $\mathcal{I}$ with error parameter $\rho > 0$ and optimal solution guess $\alpha \geqslant 0$ such that $\rho\alpha <$* OPT. *Then, there exists a divisor m_0 of m with $m_0 > 1$ such that $\gamma(v) \equiv 0 \pmod{m_0}$ for all vertices v of G' with $v \notin \{s, t\}$.*

Proof When processing a (CCMC) instance $\mathcal{I}$, Algorithm 1 starts with repeatedly doing random contractions and reduction steps, with each of the latter ones issuing a recursive call to Algorithm 1 using a modulus that is a divisor of the input modulus m. When there are only few vertices in $V_{\not\equiv 0}$ left, an enumeration over subsets of $V_{\not\equiv 0}$ is performed. Let m_0 be the modulus used when the if-block is started, and let $S \subseteq V_{\not\equiv 0}$ be the subset used in the enumeration step to reach $\mathcal{I}'$. If $S \in \{\emptyset, V_{\not\equiv 0}\}$, then the set $V_{\not\equiv 0}$ is contracted and used in $\mathcal{I}'$ as vertex s or t. In the other case, S and $V_{\not\equiv 0} \backslash S$ get contracted to the vertices s and t. Thus, in both cases, a vertex v of the contracted graph different from s and t lies in $V \backslash V_{\not\equiv 0}$, so by definition of $V_{\not\equiv 0}$, it satisfies $\gamma(v) \equiv 0 \pmod{m_0}$.

Consequently, it remains to prove that $m_0 > 1$. To this end, assume $m_0 = 1$ and consider the reduction step that leads to the recursive call of Algorithm 1 with modulus 1. This reduction step can only be performed if there is a reduction family $\mathcal{R}(\beta, q)$ for $\beta = \frac{\rho\alpha}{m}$ and some $q \in [m-1]$ with $\gcd(m, q) = 1$, where the latter condition comes from the assumption that the modulus is reduced to 1. By definition of a reduction family, we have $|\mathcal{R}(\beta, q)| = 2m - 1$, i.e., $\mathcal{R}(\beta, q) = \{R_1, \dots, R_{2m-1}\}$, and every set R_i contains one vertex u_i with $\gamma(u_i) \equiv q \pmod{m}$, while all other vertices have γ-value $0 \pmod{m}$. Furthermore, the vertices $u_1, \dots, u_{2m-1}$ are all distinct. Let $k \in [m]$ be such that $qk \equiv r \pmod{m}$ (such a k exists as $\gcd(m, q) = 1$), and let $C := R_1 \cup \ldots \cup R_k$. Observe that C is feasible for $\mathcal{I}$, as $\emptyset \neq C \subsetneq V$ because $u_1 \in C$ and $u_{2m-1} \notin C$, and $\gamma(C) = \sum_{i=1}^{k} \gamma(u_i) \equiv k \cdot q \equiv r \pmod{m}$. But

$$w(\delta(C)) \leqslant \sum_{i \in [k]} w(\delta(R_i)) \leqslant k \cdot \beta \leqslant \rho\alpha < \mathrm{OPT} \ ,$$

contradicting that OPT is the optimal solution value of $\mathcal{I}$. Thus indeed, we must have $m_0 > 1$. □

We remark that the assumptions of Lemma 16 imply OPT > 0, which is inevitable. Indeed, consider an instance only consisting of isolated vertices with $\gamma(v) \equiv 1 \pmod{m}$ and congruency constraint $\gamma(C) \equiv 1 \pmod{m}$. As there are no edges at all, no contractions can be applied, and an efficient enumeration is not possible either, leaving only a reduction to modulus 1—making an argument as in the previous proof impossible. This also explains why we have to assume that the optimal solution value is nonzero in Theorem 12.

Furthermore, note that in the proof of Lemma 16, we show that the modulus m_0 that is used when reaching the enumeration phase satisfies $m_0 > 1$ if $\rho\alpha < \mathrm{OPT}$. Equivalently, Algorithm 1 never reduces to global minimum cut instances for input moduli $m > 1$. In particular, no reduction steps are performed at all if m is a prime, hence in this case and under the given condition on ρ and α, we see that Algorithm 1 is exact.

Finally, it is easy to observe that the structured (CCMC) instances as specified in Lemma 16 can be transformed to equivalent (s–t CCMC) instances with strictly smaller modulus.

Lemma 17 *Let $\mathcal{I} = (G, w, \gamma, m, r, s, t)$ be an (s–t CCMC) instance on a graph $G = (V, E)$ such that there exists a divisor m_0 of m satisfying $\gamma(v) \equiv 0 \pmod{m_0}$ for all vertices $v \in V\backslash\{s, t\}$. Then, we can efficiently obtain an (s–t CCMC) instance $\mathcal{I}'$ on the same edge-weighted graph G with modulus ${}^{m}/_{m_0}$ such that a cut $C \subseteq V$ is feasible for $\mathcal{I}$ if and only if C is feasible for $\mathcal{I}'$.*

Proof If $\mathcal{I}$ is infeasible, there is nothing to show (and feasibility can be checked efficiently). In the other case, we must have $\gamma(s) \equiv r \pmod{m_0}$, hence $r' := \frac{r-(\gamma(s) \bmod m_0)}{m_0} \in \mathbb{Z}$, where for $v \in V$, $(\gamma(v) \bmod m_0)$ denotes the smallest non-negative integer k such that $\gamma(v) \equiv k \pmod{m_0}$. Moreover, let $m' := \frac{m}{m_0} \in \mathbb{Z}$, and define $\gamma' \colon V \to \mathbb{Z}_{\geqslant 0}$ by

$$\gamma'(v) = \frac{\gamma(v) - (\gamma(v) \bmod m_0)}{m_0}$$

for all $v \in V$. We claim that $\mathcal{I}' = (G, w, \gamma', m', r', s, t)$ has the desired properties.

To see this, let $C \subseteq V$ be a feasible solution for $\mathcal{I}$. Then $\gamma(C) \equiv r \pmod{m}$, and thus by definition of γ' and as $\gamma(v) \equiv 0 \pmod{m_0}$ for all $v \in C\backslash\{s\}$,

$$\begin{aligned} m_0 \cdot \gamma'(C) &= \gamma(C) - (\gamma(s) \bmod m_0) \\ &\equiv r - (\gamma(s) \bmod m_0) \\ &\equiv m_0 \cdot r' \qquad\qquad \pmod{m}\ , \end{aligned}$$

and thus, after division by m_0, $\gamma'(C) \equiv r' \pmod{m'}$, so C is feasible for $\mathcal{I}'$. For the other direction, let $C \subseteq V$ be feasible for $\mathcal{I}'$, i.e., $\gamma'(C) \equiv r' \pmod{m'}$. After multiplication by m_0, the latter gives $m_0 \cdot \gamma'(C) \equiv r - (\gamma(s) \bmod m_0) \pmod{m}$, hence

$$\gamma(C) = m_0 \cdot \gamma'(C) + (\gamma(s) \bmod m_0) \equiv r \pmod{m}\ ,$$

so C is feasible for $\mathcal{I}$. Finally, observe that $\mathcal{I}'$ can obviously be obtained from $\mathcal{I}$ efficiently. □

From the above findings, we can now complete a proof of Theorem 12.

Proof of Theorem 12 Let $\mathcal{I}$ be the given (CCMC) instance. For all polynomially many values

$$\alpha \in \left\{ \kappa \cdot 2^j \cdot w_{\min} \,\middle|\, 0 \leqslant j \leqslant \lceil \log_2 ({}^{w_{\text{tot}}}/_{w_{\min}}) \rceil \right\} \ ,$$

we run Algorithm 1 on $\mathcal{I}$ with $\rho = \frac{1}{2\kappa}$ for $\binom{|V|}{\lceil 8\kappa m \rceil} \log |V|$ times independently, obtaining families $\mathcal{F}_i^\alpha$ of (s–t CCMC) instances reached by the algorithm with guess α in the i^{th} run. By Observation 14, any cut C that is a solution to one of the problems in $\bigcup_{\alpha,i} \mathcal{F}_i^\alpha$ is a solution to $\mathcal{I}$, as well, and the solutions have the same value.

For the other direction, fix a solution C of $\mathcal{I}$ with value at most $\kappa \cdot \mathrm{OPT}$, and consider the families $\mathcal{F}_i^\alpha$ obtained from a run of Algorithm 1 with $\alpha \in [\kappa \cdot \mathrm{OPT}, 2\kappa \cdot \mathrm{OPT})$. By Lemma 15, for each of these $\binom{|V|}{\lceil 8\kappa m \rceil} \log |V|$ many families $\mathcal{F}_i^\alpha$, the following is true: With probability ${}^1/_{\binom{|V|}{\lceil 8\kappa m \rceil}}$, it contains an ($s$–$t$ CCMC) instance for which C is feasible. Consequently, with probability at least

$$1 - \left(1 - \frac{1}{\binom{|V|}{\lceil 8\kappa m \rceil}}\right)^{\binom{|V|}{\lceil 8\kappa m \rceil} \cdot \log |V|} \geqslant 1 - \mathrm{e}^{-\log |V|} = 1 - \frac{1}{|V|} \ ,$$

the cut C is feasible for at least one instance in $\bigcup_i \mathcal{F}_i^\alpha$. As by Observation 13, every family $\mathcal{F}_i^\alpha$ has polynomial size, and because we generated polynomially many such families, $\mathcal{F} := \bigcup_{i,\alpha} \mathcal{F}_i^\alpha$ has polynomial size. Thus, the (s–t CCMC) instances in $\mathcal{F}$ have all properties stated in Theorem 12 except for the fact that their modulus is still m. But, by Lemma 16 (note that $\rho\alpha < \mathrm{OPT}$ as $\alpha < 2\kappa \cdot \mathrm{OPT}$), the instances in $\mathcal{F}$ satisfy the assumptions of Lemma 17, and thus can be transformed to equivalent instances on the same edge-weighted graphs with moduli that are divisors of m and strictly smaller than m, as desired. □

5 Weaker contraction distributions from standard splittings

The proof of Theorem 4 given in Sect. 3 is built on an algorithmic version of the splitting-off theorem by Lovász (Theorem 10), which allows for reducing a graph $G = (V, E)$ while preserving the sizes of minimum cuts separating the sets $\{q\}$ and $Q \backslash \{q\}$ for a fixed $Q \subseteq V$ and all $q \in Q$. This splitting-off version is much less known and studied than the most common variant which preserves all pairwise connectivities among vertices in a subset $Q \subseteq V$. In this section, we show that these better-known splitting-off versions, for which strongly polynomial algorithms are already known, also allow for constructing contraction distributions for our purposes that are only slightly weaker than those obtained in Sect. 3.

Let us start by stating one of the above-mentioned standard versions of splitting-off results, namely a theorem of Frank [11]. We write $\mu_{G,w}(s,t)$ instead of $\mu_{G,w}(\{s\},\{t\})$ for the value of a minimum cut separating distinct vertices s and t.

Theorem 18 (Frank [11])[8] *Let $G = (V, E)$ be a graph with edge weights $w \colon E \to \mathbb{R}_{\geqslant 0}$, and let $Q \subseteq V$. Then there is a strongly polynomial time algorithm to obtain a graph $H = (Q, F)$ and edge weights $w_H \colon F \to \mathbb{R}_{\geqslant 0}$ satisfying*

(i) $\mu_{G,w}(s,t) = \mu_{H,w_H}(s,t)$ *for all* $s, t \in Q$ *with* $s \neq t$, *and*
(ii) $w_H(\delta_H(C \cap Q)) \leqslant w(\delta_G(C))$ *for all* $C \subseteq V$.

For the rest of this section, let us fix a (CCMC) instance $\mathcal{I} = (G, w, \gamma, m, r)$ with graph $G = (V, E)$, and let $H = (V_{\neq 0}, F)$ with edge weights $w_H \colon F \to \mathbb{R}_{\geqslant 0}$ be a graph obtained by applying Theorem 18 to (G, w) with $Q = V_{\neq 0}$. Moreover, let $\mathcal{D}$ be the distribution over vertex pairs $\{u, v\} \subseteq V$ given by choosing $\{u, v\} \in F$ with probability proportional to $w_H(\{u, v\})$. For this distribution $\mathcal{D}$, we show the following theorem, which is a weaker version of Theorem 4.

Theorem 19 *Let $\alpha \geqslant 0$ and $c > 0$ with $\sum_{v \in V_{\neq 0}} \nu(\{v\}) > \frac{4\alpha}{c} \cdot |V_{\neq 0}|$. Then, for any feasible solution C of the instance $\mathcal{I}$ with $w(\delta(C)) \leqslant \alpha$, the distribution $\mathcal{D}$ satisfies* $\Pr_{\{u,v\} \sim \mathcal{D}}\big[|\{u, v\} \cap C| = 1\big] \leqslant {}^{c}/_{|V_{\neq 0}|}$.

Note that compared to Theorem 4, the assumption in Theorem 19 is stronger by a factor of 2. This implies that the analogue of Algorithm 1 which contracts vertices based on the distribution $\mathcal{D}$ obtained in this section can perform contraction steps only if stronger assumptions are satisfied, and thus has to fall back on reduction or enumeration steps earlier. This leads to an increase in running time.

The proof of Theorem 19 is similar to the one of Theorem 4. There, we could exploit that by construction, $w_H(F) = \frac{1}{2} \sum_{v \in V_{\neq 0}} \nu(\{v\})$. This is no longer true in the current alternative setting, but we can instead use the following bound.

Theorem 20 *We have* $w_H(F) \geqslant \frac{1}{4} \sum_{v \in V_{\neq 0}} \nu(\{v\})$.

This indeed implies Theorem 19 immediately.

Proof of Theorem 19 If C is a solution of $\mathcal{I}$ with $w(\delta(C)) \leqslant \alpha$, then by the choice of $\mathcal{D}$,

$$\Pr_{\{u,v\} \sim \mathcal{D}}\big[|\{u, v\} \cap C| = 1\big] = \frac{w_H(\delta_H(C \cap V_{\neq 0}))}{w_H(F)} \leqslant \frac{4 \cdot w(\delta_G(C))}{\sum_{v \in V_{\neq 0}} \nu(\{v\})} \leqslant \frac{c}{|V_{\neq 0}|} ,$$

where the first inequality is due to Theorem 18 (ii) and Theorem 20, and the second one follows from $w(\delta_G(C)) \leqslant \alpha$ and $\sum_{v \in V_{\neq 0}} \nu(\{v\}) > \frac{4\alpha}{c} \cdot |V_{\neq 0}|$, which is an assumption in Theorem 19. □

It thus remains to show Theorem 20. To this end, we use the notion of a Gomory–Hu tree [14] (see also [20,33] for two excellent exhibitions of the topic). More precisely,

[8] Frank shows how to get (i). However, (ii) is immediate from the fact that Frank's algorithm performs classical splitting-off operations. More precisely, the method repeatedly considers a pair of edges $\{w, v\}$, $\{w, u\}$ sharing one endpoint w, and reduces their weights by some $\varepsilon > 0$, while increasing the weight of $\{v, u\}$ by ε (if needed, a new edge $\{v, u\}$ is introduced). Clearly, this way of modifying weights will never increase the value of any cut.

we consider a Gomory-Hu tree $T = (V_{\neq 0}, L)$ for $V_{\neq 0}$ in G. This is a spanning tree over $V_{\neq 0}$, where the edges $L \subseteq \binom{V_{\neq 0}}{2}$ of the spanning tree are not necessarily edges of G. Moreover, the edges L of T have weights $w_T : L \to \mathbb{R}_{\geqslant 0}$, such that

$$w_T(\{s,t\}) = \mu_{G,w}(s,t) = \nu(C_{s,t}) \quad \forall \{s,t\} \in L \ , \tag{8}$$

where $C_{s,t} \subseteq V_{\neq 0}$ are all vertices of the graph $(V_{\neq 0}, L \setminus \{s,t\})$ in the connected component that contains s.[9] To prove Theorem 20, the next two lemmas relate both the w_H-weight of F and the values $\nu(\{v\})$, respectively, to weights on the Gomory–Hu tree T, which then allows us to compare them.

Lemma 21 *For all $v \in V_{\neq 0}$, we have $\nu(\{v\}) \leqslant w_T(\delta_T(v))$.*

Proof Let $k := |\delta_T(v)|$, and let $t_1, \dots, t_k \in V_{\neq 0}$ be the neighbors of v in T. The desired result holds due to

$$\nu(\{v\}) \leqslant \sum_{i=1}^{k} \nu(C_{v,t_i}) = \sum_{i=1}^{k} w_T(\{v, t_i\}) = w_T(\delta_T(v)) \ ,$$

where the inequality holds because a cut in G that separates v from $V_{\neq 0} \setminus \{v\}$ can be obtained by removing, for each $i \in [k]$, the minimum cut in G that separates C_{v,t_i} from $V_{\neq 0} \setminus C_{v,t_i}$; moreover, the first equality follows from (8). □

Lemma 22 *We have $w_T(T) \leqslant 2 w_H(F)$.*

Proof We start by showing that

$$w_H(\delta_H(v)) \geqslant w_T(f) \quad \forall f \in \delta_T(v) \ . \tag{9}$$

The above holds because for any $f = \{u, v\} \in \delta_T(v)$, we have

$$w_T(f) = \mu_{G,w}(u,v) = \mu_{H,w_H}(u,v) \leqslant w_H(\delta_H(v)) \ ,$$

where the first equality follows from (8), the second one from Theorem 18(i), and the inequality holds because $w_H(\delta_H(v))$ is the value of the singleton cut $\{v\}$ in H, which is a v–u cut, and $\mu_{H,w_H}(u,v)$ is the value of the smallest v–u cut in H.

Finally, to show the lemma, we choose an arbitrary vertex $r \in V_{\neq 0}$, and direct all edges L of $T = (V_{\neq 0}, L)$ away from r, to obtain an r-arborescence. This arborescence can be interpreted as a bijection between L and $V_{\neq 0} \setminus \{r\}$, where an edge $f \in L$ gets assigned to the vertex in $V_{\neq 0}$ to which it points to. Now for each edge $\{u, v\} \in L$, we have $w_T(f) \leq w_H(\delta_H(v))$ by (9), where v is the vertex to which f points to.

[9] A more classical notion of Gomory-Hu trees considers a spanning tree over *all* vertices of G. However, the generalized version we need, with a tree only over a subset of the vertices, can be readily derived from the more classical version, and often follows as a byproduct when building classical Gomory-Hu trees (see proof of Theorem 15.14 in [33] for a formal proof).

Hence, by summing over all edges in T, we obtain the first inequality in the following relation, which proves the statement:

$$w_T(L) \leqslant \sum_{v \in V_{\neq 0} \setminus \{r\}} w_H(\delta_H(v)) \leqslant \sum_{v \in V_{\neq 0}} w_H(\delta_H(v)) = 2w_H(F) \ ,$$

where the equality is the classical relation that the sum of weighted degrees is equal to twice the total weight. □

With the above two statements at hand, Theorem 20 is a straightforward consequence.

Proof of Theorem 20 We have

$$w_H(F) \geqslant \frac{1}{2} w_T(T) = \frac{1}{4} \sum_{v \in V_{\neq 0}} w_T(\delta_T(v)) \geqslant \frac{1}{4} \sum_{v \in V_{\neq 0}} \nu(\{v\}) \ ,$$

where the first inequality follows from Lemma 22, the equality holds because the sum of weighted degrees is twice the total weight, and the second inequality holds due to Lemma 21. □

References

1. Artmann, S., Weismantel, R., Zenklusen, R.:. A strongly polynomial algorithm for bimodular integer linear programming. In: Proceedings of the 49th Annual ACM Symposium on Theory of Computing (STOC), pp. 1206–1219 (2017)
2. Barahona, F., Conforti, M.: A construction for binary matroids. Discrete Math. **66**(3), 213–218 (1987)
3. Benczúr, A.A.: A representation of cuts within 6/5 times the edge connectivity with applications. In: Proceedings of the 36th Annual Symposium on Foundations of Computer Science (FOCS), pp. 92–102 (1995)
4. Benczúr, A.A., Karger, D.R.: Augmenting undirected edge connectivity in $\mathcal{O}(n^2)$ time. In: Proceedings of the 9th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), pp. 500–509 (1998)
5. Bertsimas, D., Teo, C.: The parsimonious property of cut covering problems and its applications. Oper. Res. Lett. **21**(3), 123–132 (1997)
6. Bhalgat, A., Hariharan, R., Kavitha, T., Panigrahi, D.: Fast edge splitting and Edmonds' arborescence construction for unweighted graphs. In: Proceedings of the 19th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), pp. 455–464 (2008)
7. Bock, A., Faenza, Y., Moldenhauer, C., Ruiz-Vargas, A.J.: Solving the stable set problem in terms of the odd cycle packing number. In: Proceedings of the 34th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS), pp. 187–198 (2014)
8. Chandrasekaran, K., Xu, C., Yu, X.: Hypergraph k-cut in randomized polynomial time. In: Proceedings of the 29th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), pp. 1426–1438 (2018)
9. Di Summa, M., Eisenbrand, F., Faenza, Y., Moldenhauer, C.: On largest volume simplices and subdeterminants. In: Proceedings of the 26th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), pp. 315–323 (2015)
10. Eisenbrand, F., Vempala, S.: Geometric random edge. Math. Program. **164**(1), 325–339 (2017)
11. Frank, A.: Augmenting graphs to meet edge-connectivity requirements. SIAM J. Discrete Math. **5**(1), 25–53 (1992)
12. Gabow, H.N.: Efficient splitting off algorithms for graphs. In: Proceedings of the 26th Annual ACM Symposium on Theory of Computing (STOC), pp. 696–705 (1994)
13. Goemans, M.X., Ramakrishnan, V.S.: Minimizing submodular functions over families of sets. Combinatorica **15**(4), 499–513 (1995)

14. Gomory, R.E., Hu, T.C.: Multi-terminal network flows. J. Soc. Ind. Appl. Math. **9**(4), 551–570 (1961)
15. Grossman, J.W., Kulkarni, D.M., Schochetman, I.E.: On the minors of an incidence matrix and its smith normal form. Linear Algebra Appl. **218**, 213–224 (1995)
16. Grötschel, M., Lovász, L., Schrijver, A.: Corrigendum to our paper "The ellipsoid method and its consequences in combinatorial optimization". Combinatorica **4**(4), 291–295 (1984)
17. Karger, D.R.:. Global min-cuts in $\mathcal{RNC}$, and other ramifications of a simple min-cut algorithm. In: Proceedings of the 4th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), pp. 21–30 (1993)
18. Karger, D.R., Stein, C.: A new approach to the minimum cut problem. J. ACM **43**(4), 601–640 (1996)
19. Khot, S.: Ruling out PTAS for graph min-bisection, dense k-subgraph, and bipartite clique. SIAM J. Comput. **36**(4), 1025–1071 (2006)
20. Korte, B., Vygen, J.: Combinatorial Optimization, Theory and Algorithms, 6th edn. Springer, Berlin (2018)
21. Lau, L., Yung, C.: Efficient edge splitting-off algorithms maintaining all-pairs edge-connectivities. SIAM J. Comput. **42**(3), 1185–1200 (2013)
22. Lovász, L.: On some connectivity properties of Eulerian graphs. Acta Math. Acad. Sci. Hung. **28**(1), 129–138 (1976)
23. Lovász, L.: Combinatorial Problems and Exercises. North-Holland, Amsterdam (1979)
24. Mader, W.: A reduction method for edge-connectivity in graphs. Ann. Discrete Math. **3**, 145–164 (1978)
25. Megiddo, N.: Combinatorial optimization with rational objective functions. Math. Oper. Res. **4**(4), 414–424 (1979)
26. Megiddo, N.: Applying parallel computation algorithms in the design of serial algorithms. J. ACM **30**(4), 852–865 (1983)
27. Nagamochi, H., Ibaraki, T.: Deterministic $\mathcal{O}(nm)$ time edge-splitting in undirected graphs. In: Proceedings of the 28th Annual ACM Symposium on Theory of Computing (STOC), pp. 64–73 (1996)
28. Nagamochi, H., Nakamura, S., Ibaraki, T.: A simplified $\mathcal{O}(nm)$ time edge-splitting algorithm in undirected graphs. Algorithmica **26**(1), 50–67 (2000)
29. Nagamochi, H., Nishimura, K., Ibaraki, T.: Computing all small cuts in undirected networks. In: International Symposium on Algorithms and Computation (ISAAC), pp. 190–198 (1994)
30. Nägele, M., Sudakov, B., Zenklusen, R.: Submodular minimization under congruency constraints. In: Proceedings of the 29th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), pp. 849–866 (2018)
31. Nikolov, A.: Randomized rounding for the largest simplex problem. In: Proceedings of the 47th Annual ACM Symposium on Theory of Computing (STOC), pp. 861–870 (2015)
32. Padberg, M.W., Rao, M.R.: Odd minimum cut-sets and b-matchings. Math. Oper. Res. **7**(1), 67–80 (1982)
33. Schrijver, A.: Combinatorial Optimization, Polyhedra and Efficiency. Springer, Berlin (2003)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Affiliations

Martin Nägele[1] · Rico Zenklusen[1]

✉ Rico Zenklusen
ricoz@math.ethz.ch

Martin Nägele
martin.naegele@ifor.math.ethz.ch

[1] Department of Mathematics, ETH Zurich, Zurich, Switzerland