LP Relaxations of Some NP-Hard Problems Are as Hard as Any LP[*]

Daniel Průša[†] Tomáš Werner[†]

Abstract

We show that solving linear programming (LP) relaxations of many classical NP-hard combinatorial optimization problems is as hard as solving the general LP problem. Precisely, the general LP can be reduced in linear time to the LP relaxation of each of these problems. This result poses a fundamental limitation for designing efficient algorithms to solve the LP relaxations, because finding such an algorithm might improve the complexity of best known algorithms for the general LP. Besides linear-time reductions, we show that the LP relaxations of the considered problems are P-complete under log-space reduction, therefore also hard to parallelize.

1 Introduction

NP-hard problems in combinatorial optimization can usually be expressed as 0-1 integer linear programs with natural LP relaxations. Solutions to these relaxations are useful for computing exact optimal solutions (by branch-and-bound methods using the LP relaxation to compute lower bounds), exact optimal solutions of a subclass of instances (namely the instances with zero integrality gap), or approximate solutions (by rounding schemes). Despite that the LP relaxation can be solved in polynomial time, in practice its solution can be inefficient or impossible for large instances. Applications leading to large-scale combinatorial optimization problems nowadays appear more and more frequently in disciplines dealing with 'big data', such as computer vision, machine learning, artificial intelligence, data mining, or data science.

As an example, consider the *uniform metric labeling problem*, which is closely related to the multiway cut problem and a special case of the valued constraint satisfaction problem (VCSP) [18, 8]. It has a useful LP relaxation [7, 1] which leads to good approximations both in theory and practice. The problem has numerous applications in computer vision [7, 5]. In a typical setting, the number of problem variables is comparable to the number of image pixels. That is, the LP can easily have as many as 10^7 variables and a similar number of constraints. The simplex or interior point methods cannot be applied: one reason is that the LP simply does not fit in the computer memory. It does not help that the constraints are initially very sparse because these algorithms do not maintain sparsity. Alternatively, the problem structure allows us to use subgradient methods or ADMM, but these converge unacceptably slowly.

It is natural to ask if any LP relaxations are easier to solve than others. In other words, if there are algorithms, tailored to particular problems, to solve the LP relaxations more efficiently than a general LP solver. An example is LP relaxations that can be reduced in linear time to the max-flow problem. This class includes linear programs with up to two non-zeros per column, which furthermore have half-integral solutions [4]. Another example is positive linear programs (PLPs). There are algorithms that compute approximate solutions to PLPs much faster that the general LP solvers compute exact solutions, and such approximations often allow constructing good approximations to the original combinatorial problem [14, 15].

In this paper, we focus on the hardness of exactly solving LP relaxations. We show that the LP relaxations of several classical NP-hard combinatorial optimization problems are hard to solve. Precisely, there exists a linear-time reduction of the general linear program to each LP relaxation. This result poses a fundamental limitation for designing efficient algorithms to solve the LP relaxations because finding such an algorithm might improve the complexity of best known algorithms for general LP.

Though our main focus in this paper is on linear-time reductions, we also show that the considered LP relaxations are P-complete under log-space reductions. We prove this by reduction from the special LP representing the Boolean circuit value problem, which was formulated in [16] to prove P-completeness of PLP.

This paper is a continuation of our previous work in which we showed the above result for the VCSP [11] and the uniform metric labeling problem [12]. To the best of our knowledge, there is not much other literature on the hardness of linear programs resulting from LP relaxations. Somewhat related is the reduction of LP to a slim 3-way transportation program presented in [3].

The paper is organized as follows. In §2, we define

[*] Supported by Czech Science Foundation grant no. 16-05872S.

[†] Czech Technical University, Faculty of Electrical Engineering, Karlovo náměstí 13, 121 35 Prague, Czech Republic

the linear feasibility problem in equality form (LFE), which is equivalent to the decision version of the linear programming problem, and define its size. In §2.1, we show that LFE with rational coefficients can be reduced in linear time to LFE with binary (0-1) coefficients and not more than three variables per equality. In several subsequent sections we then show that this special form of LFE can be reduced in linear time to the LP relaxations of several classical combinatorial problems.

2 Linear Feasibility Problem

The *linear feasibility problem* seeks to decide if a system of linear inequalities is feasible. Using auxiliary variables, it can be transformed, in linear time, to a *linear feasibility problem in equality form* (LFE)

$$a_{i1}x_1 + \cdots + a_{in}x_n = b_i, \qquad i = 1,\ldots,m, \tag{2.1a}$$

$$x_j \geq 0, \qquad j = 1,\ldots,n. \tag{2.1b}$$

Here, $a_{ij}, b_i \in \mathbb{Q}$ are the *coefficients* of the problem. In matrix form, system (2.1a) reads

$$\mathbf{Ax} = \mathbf{b}$$

where $\mathbf{A} \in \mathbb{Q}^{m\times n}$ and $\mathbf{b} \in \mathbb{Q}^m$. We denote $\bar{\mathbf{A}} = [\bar{a}_{ij}] = [\mathbf{A}\,|\,\mathbf{b}] \in \mathbb{Q}^{m\times(n+1)}$ the extended matrix of the system. We assume that each equality involves at least one variable and each variable is involved in at least one equality, that is, $\mathbf{A}$ has no zero column or zero row.

To speak about the complexity of an algorithm to solve the LFE problem, we need to define the size of its instance. For a scalar $a \in \mathbb{Q}$, we define

$$\operatorname{size}(a) = \lceil \log_2(|pq| + 1) \rceil \tag{2.2}$$

where $p, q \in \mathbb{Z}$ are such that $a = p/q$ assuming that q does not divide p unless $q = 1$ or $p = 0$. For a matrix $\mathbf{A} \in \mathbb{Q}^{m\times n}$, we define

$$\operatorname{size}(\mathbf{A}) = \sum_{j=1}^{n}\sum_{i=1}^{m} \operatorname{size}(a_{ij}). \tag{2.3}$$

We define the size of a LFE instance to be

$$\operatorname{size}(\bar{\mathbf{A}}) = \operatorname{size}(\mathbf{A}) + \operatorname{size}(\mathbf{b}).$$

As $\operatorname{size}(a) = 0$ for $a = 0$, (2.3) underestimates the true size of matrix $\mathbf{A}$ by neglecting the space needed, e.g., for storing the indices of zero entries. This does not matter because if the time complexity of an algorithm is linear in $\operatorname{size}(\mathbf{A})$, it is linear also in the true size of $\mathbf{A}$. By not counting zero entries, we take explicitly into account the sparsity of the problem[1]. This will make our results stronger, because it can happen than an algorithm solves the problem in linear time with a dense definition of the instance size but in superlinear time with our sparse definition.

2.1 From Rational to Binary Coefficients.

Here we transform LFE with rational coefficients to a restricted form, with binary coefficients and at most three variables per equality. We do it by composing three transformations described by the following theorems.

THEOREM 2.1. *LFE with rational coefficients can be reduced in linear time to LFE with integer coefficients.*

Proof. For[2] each non-zero input coefficient $a_{ij} = p_{ij}/q_{ij}$ with $p_{ij}, q_{ij} \in \mathbb{Z}$, we introduce an auxiliary variable y_{ij} and the equation

$$|q_{ij}|y_{ij} = |p_{ij}|x_j. \tag{2.4}$$

The size of this equation is $O(\operatorname{size}(a_{ij}))$. Then in the input system we replace each non-zero term $a_{ij}x_j$ with $\operatorname{sgn}(a_{ij})y_{ij}$. The coefficients b_i are handled similarly. The size of all these equations is at most the size of the input. Therefore the size of the output (and, obviously, the reduction time) is $O(\operatorname{size}(\bar{\mathbf{A}}))$. □

EXAMPLE 1. *Let equation system (2.1a) be*

$$\tfrac{2}{7}x_1 + \tfrac{3}{5}x_2 = 2x_3,$$
$$\tfrac{7}{3}x_1 - \tfrac{1}{2}x_2 = 0.$$

This system is transformed to

$$2x_1 = 7y_{11}, \qquad 3x_2 = 5y_{12}, \qquad 2x_3 = y_{13},$$
$$7x_1 = 3y_{21}, \qquad x_2 = 2y_{22},$$
$$y_{11} + y_{12} = y_{13},$$
$$y_{21} - y_{22} = 0.$$

THEOREM 2.2. *LFE with integer coefficients can be reduced in linear time to LFE with coefficients* $\{-1, 0, 1\}$.

Proof. The idea is similar to [3, §3.1]. Suppose we want to construct the product $a_{ij}x_j$ for a coefficient $a_{ij} \in \mathbb{N}$ and a variable $x_j \geq 0$. Renaming $x_j = x_{j0}$, we create the equation system

$$\begin{aligned} x_{j1} &= x_{j0} + y_{j0}, & y_{j0} &= x_{j0}, \\ x_{j2} &= x_{j1} + y_{j1}, & y_{j1} &= x_{j1}, \\ &\vdots & &\vdots \\ x_{j,d_j} &= x_{j,d_j-1} + y_{j,d_j-1}, & y_{j,d_j-1} &= x_{j,d_j-1}. \end{aligned} \tag{2.5}$$

[1] Results on complexity of linear programming usually assume dense encoding of the LP matrix [6]. To the best of our knowledge, the complexity of solving sparse LPs is largely open [10].

[2] Note that the most obvious reduction, multiplying all coefficients of each equation by the least common multiple of their denominators, needs superlinear time.

The first line of the system enforces $x_{j1} = 2x_{j0}$, the second line enforces $x_{j2} = 2x_{j1}$, etc., therefore

$$x_{jk} = 2^k x_j.$$

The product $a_{ij}x_j$ can be now obtained by adding appropriate bits of the binary encoding of a_{ij}. E.g., $11x_j = x_{j0} + x_{j1} + x_{j3}$ because $11 = 2^0 + 2^1 + 2^3$.

The whole reduction proceeds as follows:

1. For each $j = 1, \dots, n+1$, create system (2.5) with $d_j = \lfloor \log_2 \max_{i=1}^m |a_{ij}| \rfloor$. Add the equations $x_{j0} = x_j$ $(j = 1, \dots, n)$ and $x_{n+1,0} = 1$.
2. For each $i = 1, \dots, m$, construct non-zero terms $a_{ij}x_j$ and compose the ith equation of the input system (2.1a) from them.

The size of system (2.5) created in Step 1 for one j is $O(d_j)$. The total size of the terms created in Step 1 is $\sum_j d_j = O(\text{size}(\bar{\mathbf{A}}))$ because

$$\sum_j \log_2 \max_{i=1}^m |a_{ij}| \le \sum_j \sum_i \log_2(|a_{ij}| + 1).$$

In Step 2, the total size of the terms representing one term $a_{ij}x_j$ is the number of bits of a_{ij}. We conclude that the size of the output (and the reduction time) is $O(\text{size}(\bar{\mathbf{A}}))$. □

Example 2. *Let equation system (2.1a) be*

$$\begin{aligned} 2x_1 + 11x_2 &= 1, \\ 3x_1 - \ \ 6x_2 &= 5. \end{aligned}$$

This system is transformed to

$$\begin{array}{lll}
x_{10} = x_1, & x_{11} = x_{10} + y_{10}, & y_{10} = x_{10}, \\[4pt]
x_{20} = x_2, & x_{21} = x_{20} + y_{20}, & y_{20} = x_{20}, \\
 & x_{22} = x_{21} + y_{21} & y_{21}, = x_{21}, \\
 & x_{23} = x_{22} + y_{22} & y_{22}, = x_{22}, \\[4pt]
x_{30} = 1, & x_{31} = x_{30} + y_{30}, & y_{30} = x_{30}, \\
 & x_{32} = x_{31} + y_{31} & y_{31}, = x_{31},
\end{array}$$

$$\begin{gathered} x_{11} + (x_{20} + x_{21} + x_{23}) = x_{30}, \\ (x_{10} + x_{11}) - (x_{21} + x_{22}) = x_{30} + x_{32}. \end{gathered}$$

Lemma 2.1. *For any matrix $\mathbf{A} \in \mathbb{R}^{m \times m}$,*

$$|\det \mathbf{A}| \le \prod_{j=1}^m \sum_{i=1}^m |a_{ij}|.$$

Proof. By Hadamard's inequality,

$$|\det \mathbf{A}| \le \prod_{j=1}^m \|\mathbf{a}_j\|_2$$

where $\mathbf{a}_j$ are the columns of $\mathbf{A}$. Now we use $\|\mathbf{a}\|_2 \le \|\mathbf{a}\|_1 = \sum_{i=1}^m |a_{ij}|$. □

Lemma 2.2. *The coordinates of every vertex $(x_1, \dots, x_n)$ of polyhedron (2.1) with integer coefficients satisfy $x_j \le 2^B$ where*

$$B = \sum_{j=1}^{n+1} \left\lceil \log_2 \sum_{i=1}^m |\bar{a}_{ij}| \right\rceil. \tag{2.6}$$

Moreover, $B = O(\text{size}(\bar{\mathbf{A}}))$.

Proof. It is well-known from the theory of linear programming that the vector $\mathbf{x}' = (x'_1, \dots, x'_p)$ of the non-zero variables of any basic solution to system (2.1) is the solution of the system $\mathbf{A}'\mathbf{x}' = \mathbf{b}'$ where $\mathbf{A}'$ is an invertible submatrix of $\mathbf{A}$ and $\mathbf{b}'$ is a subvector of $\mathbf{b}$. By Cramer's rule,

$$x'_j = \frac{\det \mathbf{A}'_j}{\det \mathbf{A}'}$$

where $\mathbf{A}'_j$ denotes $\mathbf{A}'$ with the jth column replaced by $\mathbf{b}'$. Since $\mathbf{A}'$ is invertible and has integer entries, $|\det \mathbf{A}'| \ge 1$. By Lemma 2.1,

$$|\det \mathbf{A}'_j| \le \prod_{j=1}^{n+1} \sum_{i=1}^m |\bar{a}_{ij}| \le 2^B.$$

This proves the first part of the theorem.

To prove $B = O(\text{size}(\bar{\mathbf{A}}))$, write

$$\begin{aligned} B - n - 1 &\le \sum_{j=1}^{n+1} \log_2 \sum_{i=1}^m |\bar{a}_{ij}| \\ &\le \sum_{j=1}^{n+1} \sum_{i=1}^m \log_2(|\bar{a}_{ij}| + 1) \le \text{size}(\bar{\mathbf{A}}) \end{aligned}$$

where the second inequality follows from the obvious fact that any non-negative numbers $c_i = |\bar{a}_{ij}|$ satisfy $\sum_{i=1}^m c_i \le \prod_{i=1}^m (c_i + 1)$. □

Remark 1. *Other bounds have been proposed on the vertex coordinates of convex polyhedra with integer coefficients. E.g., [9, Lemma 2.1] derived $x_j \le m!\,\alpha^{m-1}\beta$ where $\alpha = \max_{i,j} |a_{ij}|$ and $\beta = \max_i |b_i|$. Unfortunately, in contrast to our bound 2^B, the size of this bound is in general superlinear in $\text{size}(\bar{\mathbf{A}})$.*

Theorem 2.3. *LFE with coefficients $\{-1, 0, 1\}$ can be reduced in linear time to LFE with binary (i.e., $\{0, 1\}$) coefficients in which each equality involves one or three variables.*

Proof. As the first step, we scale down the input polyhedron (2.1) so that the coordinates of its vertices satisfy $x_j \leq \frac{1}{n}$. This is done by replacing the system $\mathbf{Ax} = \mathbf{b}$ of (2.1) with the system $\mathbf{Ax} = \mathbf{b}\sigma$ where $0 < \sigma \leq 1$ is a suitable scale. Applying Lemma 2.2, we set $\sigma = 2^{-\lceil \log_2 n \rceil - B}$. The number σ is constructed, using equations with coefficients $\{-1, 0, 1\}$, by repetitively halving the number 1 similarly as in system (2.5). By Lemma 2.2, the number of added equations is $O(\text{size}(\bar{\mathbf{A}}))$. Expression (2.6) can be computed in time $O(\text{size}(\bar{\mathbf{A}}))$ using bit-wise arithmetic operations.

As the second step, the resulting LFE is transformed, using auxiliary variables, to a LFE containing only equations of the following three types:

- $x_i = 1$,
- $x_i = x_j$,
- $x_i + x_j = x_k$ (where i, j, k are different).

This is indeed possible in linear time. E.g., let one of the equalities (2.1a) be $x_1 + x_2 - x_3 + x_4 = 1$. By moving negative terms to the other side, this is first rewritten as $x_1 + x_2 + x_4 = x_3 + 1$. This is now replaced by $x_1 + x_2 = x_5$, $x_5 + x_4 = x_6$, $x_3 + x_7 = x_6$, $x_7 = 1$ where $x_5, x_6, x_7 \geq 0$. Note that the above scaling causes the coordinates x_j of every vertex of the resulting polyhedron (2.1) to satisfy $x_j \leq 1$.

As the third step, the resulting LFE is transformed to a LFE with coefficients $\{0, 1\}$ in which each equality (2.1a) has one or three variables:

- Equations of the type $x_i = 1$ are already in this form.
- Each equation of the type $x_i = x_j$ (where $x_i \leq 1$, by the scaling) is replaced by the system
$$\begin{aligned} x_i + x_k + x_l &= 1, \\ x_j + x_k + x_l &= 1, \\ x_k, x_l &\geq 0. \end{aligned}$$
- Each equation of the type $x_i + x_j = x_k$ (where again $x_k \leq 1$) is replaced by the system
$$\begin{aligned} x_i + x_j + x_l &= 1, \\ x_k + x_l + x_p &= 1, \\ x_l &\geq 0, \\ x_p &= 0. \end{aligned}$$

In the above three steps, the input LFE has been transformed to a LFE with coefficients $\{0, 1\}$. As the coefficients of the output problem are $\{0, 1\}$, the output polyhedron is bounded, contained in the box $[0, 1]^n$. Therefore, if the input polyhedron is unbounded, the transformation cuts off some solutions. However, thanks to the scaling, the cut-off part contains no vertices. Since every polyhedron in the form (2.1) has at least one vertex, this means that the input LFE is feasible iff the output LFE is feasible. □

Theorem 2.4. *LFE with rational coefficients can be reduced in linear time to LFE with binary coefficients in which each equality involves at one or three variables.*

Proof. Compose the reductions described in Theorems 2.1, 2.2, and 2.3. □

The resulting LFE, with binary coefficients and one or three variables per equality, can be transformed to the yet simpler form with $b_i = 1$ for all $i = 1, \ldots, m$. Indeed, $x_i = 0$ is implied by $x_i + x_j + x_k = 1$, $x_j = 1$, $x_k \geq 0$. We will write this LFE as

$$\sum_{i \in s} x_i = 1, \qquad s \in S, \tag{2.7a}$$

$$x_i \geq 0, \qquad i \in I \tag{2.7b}$$

where $I = \{1, \ldots, n\}$ is the set of variables and $S \subseteq 2^I$ is a collection of variable subsets (a hypergraph over I) such that $|s| \in \{1, 3\}$ for all $s \in S$.

This problem can be no longer simplified, in the following sense. If $|s| = 3$ for all $s \in S$, then system (2.7) is trivially satisfied by setting $x_i = \frac{1}{3}$ for each $i \in I$. If $|s| \in \{1, 2\}$ for all $s \in S$ and system (2.7) is feasible, then it has a half-integral solution [4].

2.2 P-completeness. Although our primary interest in this paper is in linear-time reductions, we will also consider log-space reductions. For that we state the following result.

Theorem 2.5. *LFE with binary coefficients and one or three variables per equality is P-complete under logarithmic space reduction.*

Proof. Consider the reduction from the (P-complete) Boolean circuit value problem to linear programming proposed in [16, (LP1)]. There are gates $g_1, \ldots, g_m$ forming a circuit with output gate g_m. In_0 (In_1) denotes the set of indices of input gates with value zero (one). Neg denotes the set of pairs (i, k) such that g_k is a negation gate taking its input from g_j. Or denotes the set of triplets (i, j, k) such that g_k is a disjunction gate with taking its input from g_i and g_j. Variable x_i represents the output of gate g_i. The circuit outputs one iff the following system is feasible:

$$
\begin{array}{lrl}
(2.8a) & x_m = 1, & \\
(2.8b) & x_k = 1, & k \in \mathsf{In}_1, \\
(2.8c) & x_k = 0, & k \in \mathsf{In}_0, \\
(2.8d) & x_k \geq x_i, & (i,j,k) \in \mathsf{Or}, \\
(2.8e) & x_k \geq x_j, & (i,j,k) \in \mathsf{Or}, \\
(2.8f) & x_k \leq x_i + x_j, & (i,j,k) \in \mathsf{Or}, \\
(2.8g) & x_k + x_j = 1, & (j,k) \in \mathsf{Neg}, \\
(2.8h) & 0 \leq x_i \leq 1, & i \in \{1,\dots,m\}.
\end{array}
$$

By adding slack variables, we replace inequalities by equalities:

$$
\begin{array}{lrl}
(2.9a) & x_m = 1, & \\
(2.9b) & x_k = 1, & k \in \mathsf{In}_1, \\
(2.9c) & x_k = 0, & k \in \mathsf{In}_0, \\
(2.9d) & x_k = x_i + u_{ik}, & (i,j,k) \in \mathsf{Or}, \\
(2.9e) & x_k = x_j + v_{jk}, & (i,j,k) \in \mathsf{Or}, \\
(2.9f) & x_k + w_{ijk} = x_i + x_j, & (i,j,k) \in \mathsf{Or}, \\
(2.9g) & x_k + x_j = 1, & (j,k) \in \mathsf{Neg}, \\
(2.9h) & x_i + t_i = 1, & i \in \{1,\dots,m\}, \\
(2.9i) & x_i, t_i, u_{ik}, v_{jk}, w_{ijk} \geq 0. &
\end{array}
$$

This LFE with coefficients $\{-1,0,1\}$ can be reduced to a LFE with binary coefficients and one or three variables per equality, as described in Theorem 2.3. Since the values on any LHS and RHS are at most 2, it suffices to take the scale $\sigma = \frac{1}{2}$. Such a reduction can be done deterministically in logarithmic space. □

3 Reductions to LP Relaxations

Here we show that LFE with binary coefficients and one or three variables per equality can be reduced to the LP relaxations of several classical combinatorial optimization problems. These reductions will need linear time or logarithmic space (that is, a constant number of counters) – this will mostly be obvious and we will not prove it explicitly.

Each reduction will have the following pattern. For each instance of problem (2.7), we construct an instance of the considered combinatorial problem and a number d such that the optimal value of the LP relaxation of the problem is at least d (assuming the problem is a minimization), and it is equal to d iff (2.7) is feasible. In this way, feasibility of (2.7) is decided by solving the LP relaxation.

3.1 Set Cover. Let (V,E) with $E \subseteq 2^V$ be a hypergraph and $c\colon E \to \mathbb{Z}_+$ be hyperedge costs. The *set cover problem* seeks to find a subset of E with minimal total cost that covers V. This problem has the well-known LP relaxation [17, Chapter 13]

$$
\begin{array}{llrl}
(3.10a) & \min & \displaystyle\sum_{e \in E} c_e x_e, & \\
(3.10b) & \text{s.t.} & \displaystyle\sum_{e\colon i \in e \in E} x_e \geq 1, & i \in V, \\
(3.10c) & & x_e \geq 0, & e \in E.
\end{array}
$$

3.1.1 Weighted Version. First we construct the reduction for the case when the costs c_e are allowed to be arbitrary.

PROPOSITION 3.1. *For each $e \in E$, let $c_e = |e|$. The optimal value of (3.10) is at least $|V|$, which is attained iff all constraints (3.10b) are active.*

Proof. For every feasible solution to (3.10) we have

$$\sum_{e \in E} c_e x_e = \sum_{e \in E} \sum_{i \in e} x_e = \sum_{i \in V} \sum_{e\colon i \in e \in E} x_e \geq \sum_{i \in V} 1 = |V|.$$

This inequality holds with equality iff $\sum_{e\colon i \in e \in E} x_e = 1$ for all $i \in V$. □

The following result is now obvious.

THEOREM 3.1. *Let (V,E) be the dual hypergraph to (I,S) and let $c_e = |e|$ for each $e \in E$. Then linear program (3.10) has the optimal value $|V|$ iff system (2.7) is feasible.*

3.1.2 Unit Weights. The set cover problem has a natural meaning for the case of unit costs, $c_e = 1$ for all $e \in E$. We now construct the reduction for this more difficult situation.

Consider system (2.7) with $|s| \in \{1,3\}$ for all $s \in S$. Let $I_1 = \{\, i \in V \mid (\exists s \in S)(i \in s,\ |s| = 1) \,\}$ denote the set of variables i that occur in some equality (2.7a) of the type $x_i = 1$. Construct the linear program

$$
\begin{array}{llrl}
(3.11a) & \min & \displaystyle\sum_{i \in I_1} x_i + \sum_{i \in I \setminus I_1} (x_i + x_i' + x_i''), & \\
(3.11b) & \text{s.t.} & \displaystyle\sum_{i \in s} x_i \geq 1, & s \in S,\ |s| = 3, \\
(3.11c) & & \displaystyle\sum_{i \in s} x_i' \geq 1, & s \in S,\ |s| = 3, \\
(3.11d) & & \displaystyle\sum_{i \in s} x_i'' \geq 1, & s \in S,\ |s| = 3, \\
(3.11e) & & x_i + x_i' + x_i'' \geq 1, & i \in I, \\
(3.11f) & & x_i \geq 0, & i \in I, \\
(3.11g) & & x_i', x_i'' \geq 0, & i \in I \setminus I_1.
\end{array}
$$

Clearly, this linear program is the LP relaxation of a set cover problem with unit costs.

Proposition 3.2. *The optimal value of linear program (3.11) is at least $|I|$, which is attained iff system (2.7) is feasible.*

Proof. By (3.11e)–(3.11g), the objective value of the linear program is at least $|I|$. This value is attained iff

$$x_i = 1, \qquad i \in I_1, \tag{3.12a}$$

$$x_i + x_i' + x_i'' = 1, \qquad i \in I \setminus I_1. \tag{3.12b}$$

For $s = \{i\}$ (i.e., $|s| = 1$), (3.12) implies $x_i = 1$. For $|s| = 3$, (3.12) implies $\sum_{i\in s} x_i + \sum_{i\in s} x_i' + \sum_{i\in s} x_i'' = 3$. By (3.11b)-(3.11d), each of the three summands is at least 1, hence each summand must be 1. In particular, $\sum_{i\in s} x_i = 1$. Thus x_i form a feasible solution to (2.7).

For the other direction, let x_i $(i \in I)$ be a feasible solution to (2.7). For each $i \in I \setminus I_1$, set $x_i' = x_i'' = (1 - x_i)/2$. These values are feasible to (3.11) and satisfy (3.12). □

3.2 Set Packing. Let (V, E) with $E \subseteq 2^V$ be a hypergraph and c: $E \to \mathbb{Z}_+$ be hyperedge weights. The *set packing problem* seeks to find a disjoint subset of E with maximum total weight. This problem has the LP relaxation

$$\max \quad \sum_{e\in E} c_e x_e, \tag{3.13a}$$

$$\text{s.t.} \quad \sum_{e:\, i\in e\in E} x_e \le 1, \qquad i \in V, \tag{3.13b}$$

$$x_e \ge 0, \qquad e \in E. \tag{3.13c}$$

The reductions, for both the weighted and unweighted version, are entirely analogous to those for the set cover problem, up to the directions of appropriate inequalities.

3.3 Facility Location. Given is a set F of *facilities*, a set C of *cities*, costs f: $F \to \mathbb{Z}_+$ of opening facilities, and costs c: $F \times C \to \mathbb{Z}_+ \cup \{\infty\}$ of connecting cities to facilities. The *uncapacitated facility location problem* seeks to open a subset of facilities and assign each city to an open facility such that the total cost is minimized.

The problem has the well-known LP relaxation [17, Chapter 24]

$$\min \quad \sum_{i\in F}\sum_{j\in C} c_{ij} x_{ij} + \sum_{i\in F} f_i y_i, \tag{3.14a}$$

$$\text{s.t.} \quad \sum_{i\in F} x_{ij} = 1, \qquad j \in C, \tag{3.14b}$$

$$y_i \ge x_{ij}, \qquad i \in F,\ j \in C, \tag{3.14c}$$

$$x_{ij} \ge 0, \qquad i \in F,\ j \in C, \tag{3.14d}$$

$$y_i \ge 0, \qquad i \in F \tag{3.14e}$$

where for $c_{ij} = \infty$ and $x_{ij} = 0$ we define $c_{ij} x_{ij} = 0$.

Problem (2.7) can be reduced to problem (3.14) by setting

$$\begin{aligned} F &= I, \\ C &= S, \\ f_i &= |\{s \in S \mid i \in s\}|, \qquad i \in I, \\ c_{is} &= \begin{cases} a & \text{if } i \in s \\ b & \text{if } i \notin s \end{cases}, \qquad i \in s \in S \end{aligned}$$

where $a, b \ge 0$ are constants satisfying $b > a + 1$.

Theorem 3.2. *The constructed linear program (3.14) has optimal value at least $|S|(a+1)$, which is attained iff system (2.7) is feasible.*

Proof. Define $f_{is} = 1$ if $i \in s$ and $f_{is} = 0$ if $i \notin s$. Now $f_i = \sum_{s\in S} f_{is}$. The objective value of the constructed LP (3.14) satisfies

$$\sum_{i\in I}\sum_{s\in S} c_{is} x_{is} + \sum_{i\in I} f_i y_i = \sum_{s\in S}\sum_{i\in I} (c_{is} x_{is} + f_{is} y_i) \tag{3.15a}$$

$$\ge \sum_{s\in S}\sum_{i\in I} (c_{is} + f_{is}) x_{is} \tag{3.15b}$$

$$\ge \sum_{s\in S}\sum_{i\in I} (a+1) x_{is} \tag{3.15c}$$

$$= |S|(a+1). \tag{3.15d}$$

Inequality (3.15b) becomes equality iff $i \in s$ implies $y_i = x_{is}$. Inequality (3.15c) becomes equality iff $x_{is} > 0$ implies $i \in s$. In this case, for each $s \in S$ we have $1 = \sum_{i\in I} x_{is} = \sum_{i\in s} x_{is} = \sum_{i\in s} y_i$. That is, variables y_i satisfy (2.7). In the other direction, every feasible solution to (2.7) attains the optimal value $|S|(a+1)$. □

3.3.1 Non-metric version. Let us set $a = 0$ and $b = \infty$. In this case, variables x_{ij} with $c_{ij} = \infty$ are inevitably zero, if problem (3.14) is to be feasible. So these variables can be omitted from linear program (3.14). More precisely, this can be done by introducing a bipartite digraph $(F \cup C, E)$ with $(i, j) \in E$ iff $c_{ij} < \infty$, and writing (3.14) in terms of this digraph. The size of this linear program is linear in the size of system (2.7), that is, $O(\sum_{s\in S} |s|)$.

3.3.2 Metric version. If the costs c can be extended to a map c: $(F \cup C)^2 \to \mathbb{Z}_+ \cup \{\infty\}$ that satisfies

$$c_{ij} + c_{jk} \ge c_{ik}, \qquad i, j, k \in F \cup C, \tag{3.16}$$

we speak about *metric* facility location.

The choice $a = 0$ and $b = \infty$ does not result in a metric problem. However, any a, b satisfying $0 \le a \le b \le 2a$ and $b > a + 1$ (e.g., $a = 3$ and $b = 5$) yield a metric problem. Unfortunately, the size of linear program (3.14) is $O(|F| \cdot |C|) = O(|I| \cdot |S|) > O(\sum_{s\in S} |s|)$, so this reduction takes superlinear time.

3.4 Maximum Satisfiability. Let

$$\bigwedge_{j\in C}\Big(\bigvee_{i\in S_j^+} v_i \vee \bigvee_{i\in S_j^-} \neg v_i\Big) \tag{3.17}$$

be a Boolean formula in conjunctive normal form with variables V and clauses C, where S_j^+ (S_j^-) is the set of variables occurring non-negated (negated) in clause $j \in C$. Let $c\colon C \to \mathbb{Z}_+$ be clause weights. The *maximum satisfiability problem* seeks to find the values of the variables to maximize the total weight of satisfied clauses. The classical LP relaxation of this problem [17, Chapter 16] reads

$$\begin{aligned}
&\text{(3.18a)} && \max && \sum_{j\in C} c_j z_j, && \\
&\text{(3.18b)} && \text{s.t.} && \sum_{i\in S_j^+} x_i + \sum_{i\in S_j^-} (1 - x_i) \ge z_j, && j \in C,\\
&\text{(3.18c)} && && 0 \le z_j \le 1, && j \in C,\\
&\text{(3.18d)} && && 0 \le x_i \le 1, && i \in V.
\end{aligned}$$

Problem (2.7) can be reduced to problem (3.18) as follows. Let $V = I$ and define formula (3.17) as

$$\bigwedge_{s\in S}\Big(\Big(\bigvee_{i\in s} v_i\Big) \wedge \bigwedge_{i\in s} \neg v_i\Big), \tag{3.19}$$

where each clause $\bigvee_{i\in s} v_i$ has weight 2 and each clause $\neg v_i$ has weight 1.

Proposition 3.3. *The constructed LP (3.18) has optimal value at most $\sum_{s\in S}(1 + |s|)$, which is attained iff system (2.7) is feasible.*

Proof. By eliminating variables z_j, the LP relaxation (3.18) can be written as the maximization of the concave function

$$\sum_{j\in C} c_j \min\Big\{1, \sum_{i\in S_j^+} x_i + \sum_{i\in S_j^-} (1 - x_i)\Big\} \tag{3.20}$$

subject to (3.18d). For the constructed maximum satisfiability problem, (3.20) reads

$$\begin{aligned}
\sum_{s\in S}\Big(2\min\Big\{1, \sum_{i\in s} x_i\Big\} + \sum_{i\in s}(1 - x_i)\Big)\\
= \sum_{s\in S}\Big(\phi\Big(\sum_{i\in s} x_i\Big) + |s|\Big)
\end{aligned} \tag{3.21}$$

where the function $\phi\colon \mathbb{R} \to \mathbb{R}$ is defined by

$$\phi(t) = 2\min\{1, t\} - t. \tag{3.22}$$

Function ϕ attains its maximum at $t = 1$ with value $\phi(1) = 1$. Therefore, (3.21) attains its maximum iff $\sum_{i\in s} x_i = 1$ for all $s \in S$, with maximum value $\sum_{s\in S}(1 + |s|)$. □

3.5 Maximum Independent Set. Given a graph (V, E) with $E \subseteq \binom{V}{2}$ and vertex weights $c\colon V \to \mathbb{Z}_+$, the *maximum independent set problem* seeks to find a subset $U \subseteq V$ of vertices that is independent (i.e., no edge has both ends in U) and that maximizes its total weight. For this problem, we consider the LP relaxation

$$\begin{aligned}
&\text{(3.23a)} && \max && \sum_{i\in V} c_i x_i, && \\
&\text{(3.23b)} && \text{s.t.} && \sum_{i\in C} x_i \le 1, && C \subseteq V \text{ is a clique},\\
&\text{(3.23c)} && && x_i \ge 0, && i \in V.
\end{aligned}$$

3.5.1 Weighted Version. Let us first consider the easier case when the costs c_i are allowed to be arbitrary.

Proposition 3.4. *Let $\mathcal{C} \subseteq 2^V$ be a set of (not necessarily all) cliques of the graph (V, E). Let $c_i = |\{\, C \in \mathcal{C} \mid i \in C \,\}|$. Then the optimal value of linear program (3.23) is at most $|\mathcal{C}|$, which is attained iff constraint (3.23b) is active for each $C \in \mathcal{C}$.*

Proof. For every feasible solution to (3.23) we have

$$\sum_{i\in V} c_i x_i = \sum_{i\in V} \sum_{C\colon i\in C\in\mathcal{C}} x_i = \sum_{C\in\mathcal{C}} \sum_{i\in C} x_i \le \sum_{C\in\mathcal{C}} 1 = |\mathcal{C}|.$$

This inequality holds with equality iff $\sum_{i\in C} x_i = 1$ for each $C \in \mathcal{C}$. □

Problem (2.7) can be reduced to problem (3.23) by setting

$$\begin{aligned}
V &= I \cup \{\, (i, s) \mid i \in s \in S \,\},\\
E &= \{\, \{(i, s), (j, s)\} \mid i, j \in s \in S \,\}\\
&\quad \cup \{\, \{i, (i, s)\} \mid i \in s \in S \,\},\\
c_i &= |\{\, s \in S \mid i \in s \,\}|, && i \in I,\\
c_{is} &= 2, && i \in s \in S
\end{aligned}$$

where c_{is} is a shortcut for $c_{(i,s)}$. An example is shown in Figure 1. Note that the reduction is valid for any S with arbitrarily sized elements. Hence we present examples where $|s| \in \{2, 3\}$ for $s \in S$.

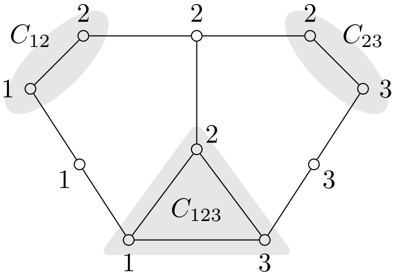

Figure 1: Graph (V, E) of the constructed weighted maximum independent set problem with $I = \{1, 2, 3\}$ and $S = \{\{1, 2, 3\}, \{1, 2\}, \{2, 3\}\}$.

Proposition 3.5. *The constructed linear program (3.23) has optimal value at most $\sum_{s\in S}(1 + |s|)$, which is attained iff system (2.7) is feasible.*

Proof. The weights c_i, c_{is} satisfy the assumption of Proposition 3.4 for

$$\mathcal{C} = \{\, \{\, (i, s) \mid i \in s \,\} \mid s \in S \,\} \cup \{\, \{\, i, (i, s) \,\} \mid i \in s \in S \,\}.$$

Therefore the constructed LP has optimal value at most $|\mathcal{C}| = \sum_{s\in S}(1 + |s|)$, which is attained iff its variables x_i, x_{is} are feasible and satisfy

$$\text{(3.24a)} \qquad \sum_{i\in s} x_{is} = 1, \qquad s \in S,$$

$$\text{(3.24b)} \qquad x_i + x_{is} = 1, \qquad i \in s \in S.$$

But (3.24b) implies $x_{is} = x_i$ for all $s \in S$, hence (3.24) is equivalent to (2.7a).

If inequality (3.23b) holds for some set C then it clearly holds for every subset of C. As each clique of graph (V, E) that is not in $\mathcal{C}$ is a subset of some clique in $\mathcal{C}$, (3.24b) implies (3.23b) for all cliques of (V, E). Therefore, (2.7) implies feasibility of x_i, x_{is}. □

For general graph (V, E), linear program (3.23) can have exponentially many inequalities (3.23b) and solving it is in fact NP-hard[3]. However, the linear program (3.23) resulting from our reduction is small. If the input problem (2.7) has $|s| \leq 3$ for all $s \in S$, the number of all cliques in the graph (V, E) is $O(\sum_{s\in S} |s|)$. If $|s|$ in (2.7) are allowed to be unbounded, the graph (V, E) has $O(\sum_{s\in S} |s|)$ maximal cliques (though the number of all cliques is exponential). But in (3.23) it suffices to impose inequalities (3.23b) only for maximal cliques because this implies (3.23b) for all cliques. In both cases, the size of the constructed linear program is linear.

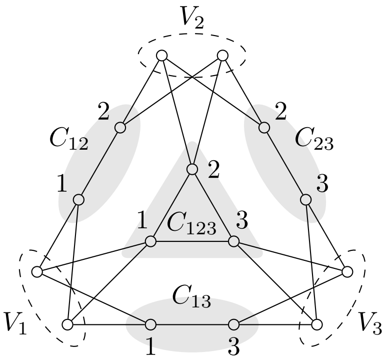

Figure 2: Graph (V, E) of the constructed unweighted maximum independent set problem with $I = \{1, 2, 3\}$ and $S = \{\{1, 2, 3\}, \{1, 2\}, \{1, 3\}, \{2, 3\}\}$.

3.5.2 Unit Weights.

Now we construct the reduction for the more difficult case of unit weights, $c_i = 1$ for all $i \in V$.

Lemma 3.1. *Any system (2.7) can be extended to an equivalent system (2.7) in which for each $i \in I$, the number $|\{s \in S \mid i \in s\}|$ of occurrences of variable i is divisible by 3.*

Proof. For each variable x_i ($i \in I$) in the input system (2.7), create new variables x'_i, x''_i. For each equation $\sum_{i\in s} x_i = 1$ ($s \in S$), create new equations $\sum_{i\in s} x'_i = 1$ and $\sum_{i\in s} x''_i = 1$. Now we have three copies of the input system (2.7). Therefore, for each $i \in I$ the number of occurrences of variables x_i, x'_i, x''_i is now the same.

If for some $i \in I$ the number of occurrences is not divisible by 3, we increase it by 1 or 2 using the first or both columns of the following systems, respectively:

$$\begin{aligned} x_i + y_{i1} + y_{i2} &= 1, & x_i + y_{i3} + y_{i4} &= 1, \\ x'_i + y_{i1} + y_{i2} &= 1, & x'_i + y_{i3} + y_{i4} &= 1, \\ x''_i + y_{i1} + y_{i2} &= 1, & x''_i + y_{i3} + y_{i4} &= 1. \end{aligned}$$

Note that $y_{i1}, y_{i2}, y_{i3}, y_{i4}$ are new variables.

Clearly, the input problem is feasible iff the output problem is feasible. □

Lemma 3.2. *For every $k \in \mathbb{N}$ there exists a connected bipartite graph $(U \cup V, E)$ with partitions U and V such that $|U| = 3k$, $|V| = 2k$, $\deg(u) = 2$ for all $u \in U$, $\deg(v) = 3$ for all $v \in V$.*

Proof. Partition the set $U \cup V$ into k subsets of size 5, each of them containing 3 vertices from U and 2 vertices from V. Connect the vertices in each group by the scheme in Figure 3. Chain the groups one by one into a cycle (using the trailing edges). The resulting graph has the desired properties. □

[3] For unit weights $c_i = 1$, the optimal value of (3.23) is known as the *fractional clique number* of the graph, which by duality is the same as its *fractional chromatic number* [13].

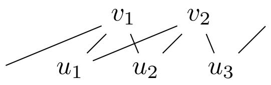

Figure 3: The basic building block of the bipartite graph construction. It consists of vertices $\{u_1, u_2, u_3\} \subseteq U$ and $\{v_1, v_2\} \subseteq V$. All the incident edges are displayed. The trailing edges connect to neighboring groups.

Consider problem (2.7) in which we assume, by Lemma 3.1, that the number of occurrences of each variable is divisible by 3. We now reduce this problem to linear program (3.23) with unit costs. For $i \in I$, denote

$$U_i = \{\, (i,s) \mid i \in s \in S \,\},$$
$$V_i = \{\, (i,j) \mid j \in \{1, \dots, \tfrac{2}{3}|U_i|\} \,\}$$

and E_i to be the edge set of bipartite graph $(U_i \cup V_i, E_i)$ constructed as in Lemma 3.2. Now set

$$V = \bigcup_{i \in I} (U_i \cup V_i),$$
$$E = \{\, \{(i,s), (j,s)\} \mid i, j \in s \in S \,\} \cup \bigcup_{i \in I} E_i.$$

An example is shown in Figure 2.

Proposition 3.6. *The constructed linear program (3.23) has optimal value at most*

$$\frac{1}{3}(|S| + 2\sum_{i \in I} |U_i|),$$

which is attained iff system (2.7) is feasible.

Proof. For each $s \in S$, the set $C_s = \{\, (i,s) \mid i \in s \,\}$ is a clique in graph (V, E). Therefore

$$\sum_{i \in I} \sum_{v \in U_i} x_v = \sum_{s \in S} \sum_{v \in C_s} x_v \le |S|,$$

where the inequality becomes equality iff $\sum_{v \in C_s} x_v = \sum_{i \in s} x_{is} = 1$.

Each vertex in U_i has two and each vertex in V_i three incident edges in E_i. Each edge $\{u, v\} \in E_i$ is a clique in graph (V, E). Therefore, for each $i \in I$ we have

$$2 \sum_{u \in U_i} x_u + 3 \sum_{v \in V_i} x_v = \sum_{\{u,v\} \in E_i} (x_u + x_v) \le 2|U_i|,$$

where the inequality becomes equality iff $x_u + x_v = 1$ for each $\{u, v\} \in E_i$, that is, $x_{is} + x_{ij} = 1$ for each $s \ni i$ and $j = 1, \dots, \frac{2}{3}|U_i|$. Since the graph $(U_i \cup V_i, E_i)$ is connected, this implies that all variables x_{is} ($s \ni i$) are equal, that is, there exists y_i such that $x_{is} = y_i$ for each $s \ni i$.

Putting the above together, the objective value of LP (3.23) can be expressed as

$$\sum_{v \in V} x_v = \frac{1}{3}\Big(\sum_{s \in S} \sum_{v \in C_s} x_v + \sum_{i \in I} \sum_{\{u,v\} \in E_i} (x_u + x_v) \Big)$$
$$\le \frac{1}{3}\Big(|S| + 2 \sum_{i \in I} |U_i| \Big)$$

where the inequality becomes equality iff $\sum_{i \in s} x_{is} = 1$ for all $s \in S$ and $x_{is} = y_i$ for all $i \in I$ and $s \in S$, that is, iff the variables y_i satisfy (2.7). □

3.6 Multiway Cut. Let (V, E) with $E \subseteq \binom{V}{2}$ be an undirected graph with edge costs $c\colon E \to \mathbb{Z}_+$, and $T \subseteq V$ be a set of terminals. The *minimum multiway cut problem* seeks to find a subset $F \subseteq E$ of edges with minimum total cost such that in the graph $(V, E \setminus F)$ each terminal is in a different component. We consider its relaxation proposed in [2] (see also [17, Chapter 19]):

$$\text{(3.25a)} \quad \min \quad \frac{1}{2} \sum_{\{i,j\} \in E} c_{ij} \sum_{t \in T} |x_{it} - x_{jt}|,$$
$$\text{(3.25b)} \quad \text{s.t.} \quad \sum_{t \in T} x_{it} = 1, \qquad i \in V,$$
$$\text{(3.25c)} \quad x_{tt} = 1, \qquad t \in T,$$
$$\text{(3.25d)} \quad x_{it} \ge 0, \qquad i \in V,\ t \in T.$$

This is not a linear program but it is readily transformed to one. An instance of linear program (3.25) is defined by a tuple (V, E, T, c).

We have presented a linear-time reduction from the general LP to (3.25) in [12]. Here we present a more direct reduction from problem (2.7) to problem (3.25).

The output multiway cut problem will be constructed by gluing small multiway cut problems (gadgets). First we show that gluing gadgets cannot increase the LP optimum.

Proposition 3.7. *Let two instances (V_1, E_1, T, c_1) and (V_2, E_2, T, c_2) of problem (3.25) have optimal values* Opt_1 *and* Opt_2*, respectively. Define $V = V_1 \cup V_2$, $E = E_1 \cup E_2$ and $c\colon E \to \mathbb{Z}_+$ where*

$$c_{ij} = \begin{cases} c_{1,ij} & \textit{if } \{i,j\} \in E_1 \setminus E_2, \\ c_{2,ij} & \textit{if } \{i,j\} \in E_2 \setminus E_1, \\ c_{1,ij} + c_{2,ij} & \textit{if } \{i,j\} \in E_1 \cap E_2. \end{cases}$$

Let Opt *be the optimal value of instance (V, E, T, c). Then,* $\mathrm{Opt} \ge \mathrm{Opt}_1 + \mathrm{Opt}_2$.

Proof. Clearly,

$$\text{(3.26)}\qquad \sum_{\{i,j\}\in E} c_{ij} \sum_{t\in T} |x_{it} - x_{jt}|$$
$$= \sum_{\{i,j\}\in E_1} c_{1,ij} \sum_{t\in T} |x_{it} - x_{jt}| + \sum_{\{i,j\}\in E_2} c_{2,ij} \sum_{t\in T} |x_{it} - x_{jt}|.$$

Let $X \subseteq \mathbb{R}^{V\times T}$ denote the set of feasible solutions to (3.25). Now minimize each side of (3.26) over X, using that $\min_{\mathbf{x}\in X}[f(\mathbf{x}) + g(\mathbf{x})] \geq \min_{\mathbf{x}\in X} f(\mathbf{x}) + \min_{\mathbf{x}\in X} g(\mathbf{x})$ for any functions f, g. □

We now describe the gadgets, which are defined in Figure 4. From now on, we fix the terminals to be $T = \{1, 2, 3\}$.

Proposition 3.8. *The LP relaxation of the multiway cut problem in Figure 4(a) has optimal value 0, attained iff $x_{u1} = 1$ and $x_{u2} = x_{u3} = 0$.*

Proof. Obvious: (3.25) minimizes $\frac{1}{2}(|1-x_{u1}|+x_{u2}+x_{u3})$ subject to $x_{u1} + x_{u2} + x_{u3} = 1$. □

Proposition 3.9. *The LP relaxation of the multiway cut problem in Figure 4(b) has optimal value 3, attained iff $x_{v2} = x_{u2} + x_{u3}$ and $x_{v1} = x_{u1}$.*

Proof. Taking into account constraint (3.25b), the objective value of (3.25) is

$$\begin{aligned}
&(x_{u2} + x_{u3}) + 2(x_{u1} + x_{u2}) + 3(x_{v2} + x_{v3}) \\
&\quad +4(x_{v1} + x_{v3}) + \textstyle\sum_{i=1}^{3} |x_{ui} - x_{vi}| = \\
&3 + (x_{u2} - x_{v2} + |x_{u2} - x_{v2}|) \\
&\quad + (x_{v3} - x_{u3} + |x_{u3} - x_{v3}|) + 2x_{v3} + |x_{u1} - x_{v1}|.
\end{aligned}$$

This attains the minimum value, 3, iff $x_{u2} \leq x_{v2}$, $x_{v3} = 0$ and $x_{u1} = x_{v1}$, which is equivalent to $x_{v2} = x_{u2} + x_{u3}$ and $x_{v1} = x_{u1}$. □

By permuting the terminals, we obtain gadgets enforcing $x_{vj} = x_{uj} + x_{uk}$ and $x_{vi} = x_{ui}$ for any $\{i, j, k\} = \{1, 2, 3\}$. We denote this gadget by $\mathsf{Add}(u, v, i, j, k)$.

Proposition 3.10. *The LP relaxation of the multiway cut problem in Figure 4(c) has optimal value 4, attained iff $x_{u2} = x_{v1} = 0$, $x_{u3} = x_{v3}$, and $x_{u1} = x_{v2}$.*

Proof. Taking into account constraint (3.25b), the objective value of (3.25) is

$$\begin{aligned}
&4(x_{u2} + x_{u3}) + 3(x_{u1} + x_{u2}) + 4(x_{v1} + x_{v3}) \\
&\quad +3(x_{v1} + x_{v2}) + \textstyle\sum_{i=1}^{3} |x_{ui} - x_{vi}| = \\
&4 + (x_{v1} - x_{u1} + |x_{u1} - x_{v1}|) + 2x_{u2} + 2x_{v1} \\
&\quad + (x_{u2} - x_{v2} + |x_{u2} - x_{v2}|) + |x_{u3} - x_{v3}|.
\end{aligned}$$

This attains the minimum value, 4, iff $x_{u2} = x_{v1} = 0$ and $x_{u3} = x_{v3}$, which implies $x_{u1} = x_{v2}$. □

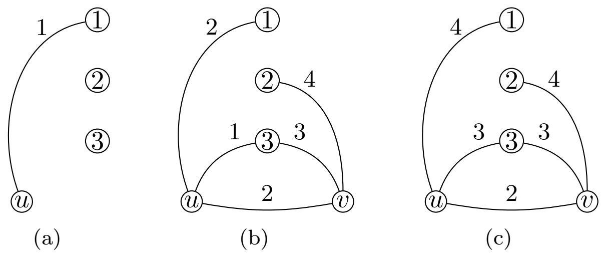

Figure 4: Gadgets enforcing specific equalities among the variables of LP (3.25) at its optimum: (a) $x_{u1} = 1$, $x_{u2} = x_{u3} = 0$; (b) $x_{v2} = x_{u2} + x_{u3}$, $x_{v1} = x_{u1}$; (c) $x_{u2} = x_{v1} = 0$, $x_{u1} = x_{v2}$, $x_{u3} = x_{v3}$.

By permuting the terminals, we obtain gadgets enforcing $x_{uj} = x_{vi} = 0$, $x_{uk} = x_{vk}$ and $x_{ui} = x_{vj}$ for any $\{i, j, k\} = \{1, 2, 3\}$. We denote this gadget by $\mathsf{Perm}(u, v, i, j, k)$.

Theorem 3.3. *For problem (2.7) with $|s| \in \{1, 3\}$, one can in linear time construct problem (3.25) that has optimal value at least $\sum_{i\in I} 10(|K_i| - 1)$ where $K_i = \{\, s \in S \mid i \in s \,\}$, which is attained iff problem (2.7) is feasible.*

Proof. In input problem (2.7), denote $S_3 = \{\, s \in S \mid |s| = 3 \,\}$. The desired multiway cut problem (V, E, T, c) with terminals $T = \{1, 2, 3\}$ is constructed as follows. Initially, set $V = S_3 \cup T$ and $E = \emptyset$. The variables x_{st} ($s \in S_3$, $t \in T$) of (3.25) are intended to represent the variables x_i of (2.7). By adding suitable gadgets to this initial problem (which enlarges sets V and E and defines new costs c_{ij}), it is now necessary to enforce equalities (2.7a) for $|s| = 1$, and to enforce equalities of different variables x_{st} of (3.25) that represent the same variable x_i in (2.7a).

The former is easy, using the gadget in Figure 4(a).

The latter is achieved by combining the gadgets Add and Perm. Let $\{i, j, k\} = \{1, 2, 3\}$. Suppose x_{si} and $x_{s'j}$ represent the same variable of (2.7a). To enforce $x_{si} = x_{s'j}$, introduce gadgets $\mathsf{Add}(s, u, i, j, k)$, $\mathsf{Perm}(u, v, i, j, k)$ and $\mathsf{Add}(s', v, j, k, i)$ where u, v are new vertices. By Proposition 3.7, the optimal value attained over this combination of the three gadgets is 10. Enforcing equality $x_{si} = x_{s'i}$ is even easier, by introducing gadgets $\mathsf{Add}(s, u, i, j, k)$ and $\mathsf{Add}(s, u, i, j, k)$ where u is a new vertex. This combination attains minimal value 6. To attain 10 instead, simply increase the costs c_{u1}, c_{u2}, c_{u3} by 4.

For a group of $|K_i|$ variables of (3.25) that represent the same variable x_i of (2.7a), it is required to enforce $|K_i| - 1$ equalities to ensure that all the representatives attain the same value in the minimum. Hence, by

Proposition 3.7, the LP relaxation attains minimum $\sum_{i \in I} 10(|K_i| - 1)$ iff the problem (2.7) is feasible. □

4 Final Hardness Results

Now we are able to combine the material from §2 and §3 to obtain the main results of this paper.

The *linear programming problem* seeks to decide if a linear function can attain a value not greater than a threshold, subject to a set of linear inequalities.

Theorem 4.1. *The linear programming problem can be reduced in linear time to each LP relaxation from §3 (except for metric facility location, §3.3.2, which takes quadratic time).*

Proof. Using auxiliary variables, LP reduces in linear time to LFE with rational coefficients. By Theorem 2.4, this problem reduces in linear time to LFE with binary coefficients and one or three variables per equality. As shown in §3, this problem is reduced in linear time to each LP relaxation. □

Theorem 4.2. *Each LP relaxation from §3 is P-complete under logarithmic space reduction.*

Proof. Each reduction in §3 can be performed in a logarithmic space. Composing this reduction with that in Theorem 2.5 gives the result. □

References

[1] Chandra Chekuri, Sanjeev Khanna, Joseph Naor, and Leonid Zosin. A linear programming formulation and approximation algorithms for the metric labeling problem. *SIAM J. on Discrete Mathematics*, 18(3):608–625, 2005.

[2] Gruia Călinescu, Howard Karloff, and Yuval Rabani. An improved approximation algorithm for multiway cut. In *13th Annual ACM Symposium on Theory of Computing (STOC)*, pages 48–52, 1998.

[3] Jesús A. De Loera and Shmuel Onn. All linear and integer programs are slim 3-way transportation programs. *SIAM J. on Optimization*, 17(3):806–821, 2006.

[4] Dorit S. Hochbaum. Monotonizing linear programs with up to two nonzeroes per column. *Operations Research Lett.*, 32(1):49–58, 2004.

[5] Jörg H. Kappes, Bjoern Andres, Fred A. Hamprecht, Christoph Schnörr, Sebastian Nowozin, Dhruv Batra, Sungwoong Kim, Bernhard X. Kausler, Thorben Kröger, Jan Lellmann, Nikos Komodakis, Bogdan Savchynskyy, and Carsten Rother. A comparative study of modern inference techniques for structured discrete energy minimization problems. *Intl. J. of Computer Vision*, 115(2):155–184, 2015.

[6] Narendra Karmarkar. A new polynomial-time algorithm for linear programming. In *ACM Symp. on Theory of Computing*, pages 302–311, 1984.

[7] Jon Kleinberg and Eva Tardos. Approximation algorithms for classification problems with pairwise relationships: Metric labeling and Markov random fields. *J. of ACM*, 49(5):616–639, 2002.

[8] Vladimir Kolmogorov, Johan Thapper, and Stanislav Živný. The power of linear programming for general-valued CSPs. *SIAM J. on Comp.*, 44(1):1–36, 2015.

[9] Christos H. Papadimitriou and Kenneth Steiglitz. *Combinatorial Optimization: Algorithms and Complexity.* Dover Publications, 1998.

[10] Panos M. Pardalos and Stephen A. Vavasis. Open questions in complexity theory for numerical optimization. *Math. Program.*, 57:337–339, 1992.

[11] Daniel Průša and Tomáš Werner. Universality of the local marginal polytope. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 37(4):898–904, April 2015.

[12] Daniel Průša and Tomáš Werner. LP relaxation of the Potts labeling problem is as hard as any linear program. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, PP(99), 2016.

[13] Edward R. Scheinerman and Daniel H. Ullman. *Fractional Graph Theory: a Rational Approach to the Theory of Graphs.* Dover Publications, 2013.

[14] Luca Trevisan. Parallel approximation algorithms by positive linear programming. *Algorithmica*, 21(1):72–88, 1998.

[15] Luca Trevisan. Erratum: A correction to ”parallel approximation algorithms by positive linear programming”. *Algorithmica*, 27(2):115–119, 2000.

[16] Luca Trevisan and Fatos Xhafa. The parallel complexity of positive linear programming. *Parallel Processing Lett.*, 8(4):527–533, 1998.

[17] Vijay V. Vazirani. *Approximation Algorithms.* Springer-Verlag New York, 2001.

[18] Stanislav Živný. *The Complexity of Valued Constraint Satisfaction Problems.* Cognitive Technologies. Springer, 2012.