A Combinatorial Certifying Algorithm for Linear Programming Problems with Gainfree Leontief Substitution Systems

Kei Kimura
Faculty of Information Science and Electrical Engineering, Kyushu University, Fukuoka, Japan

Kazuhisa Makino
Research Institute for Mathematical Sciences, Kyoto University, Japan

Abstract

Linear programming (LP) problems with gainfree Leontief substitution systems have been intensively studied in economics and operations research, and include the feasibility problem of a class of Horn systems, which arises in, e.g., polyhedral combinatorics and logic. This subclass of LP problems admits a strongly polynomial time algorithm, where devising such an algorithm for general LP problems is one of the major theoretical open questions in mathematical optimization and computer science. Recently, much attention has been paid to devising certifying algorithms in software engineering, since those algorithms enable one to confirm the correctness of outputs of programs with simple computations. Devising a combinatorial certifying algorithm for the feasibility of the fundamental class of Horn systems remains open for almost a decade. In this paper, we provide the first combinatorial (and strongly polynomial time) certifying algorithm for LP problems with gainfree Leontief substitution systems. As a by-product, we resolve the open question on the feasibility of the class of Horn systems.

2012 ACM Subject Classification Theory of computation → Mathematical optimization

Keywords and phrases linear programming problem, certifying algorithm, Horn system

Digital Object Identifier 10.4230/LIPIcs.ISAAC.2023.47

Related Version *Full Version*: `https://arxiv.org/abs/2306.03368` [19]

Funding Partially supported by JSPS KAKENHI Grant Number JP19K22841.
Kei Kimura: Partially supported by JST, ACT-X Grant Number JPMJAX200C, Japan, and JSPS KAKENHI Grant Number JP21K17700.
Kazuhisa Makino: Partially supported by JSPS KAKENHI Grant Number JP20H05967.

1 Introduction

In this paper, we focus on linear programming (LP) problems with Leontief substitution systems. A matrix A is called *Leontief* if each column of A has at most one positive element.[1] A linear system of the form

$$A\boldsymbol{x} = \boldsymbol{b} \quad \text{and} \quad \boldsymbol{x} \geq \boldsymbol{0} \tag{1}$$

is called a *Leontief substitution system* if A is Leontief and $\boldsymbol{b}$ is nonnegative. Leontief matrices and systems were first studied in 1950s within the context of input-output analysis in economics (for which Wassily Leontief was awarded the Nobel Prize in economics in 1973; see, e.g., Leontief [21] and Dantzig [9]), and have attracted much attention in economics and operations research. There exists a line of research on algorithms for LP problems with

[1] Leontief matrices defined in this paper are sometimes called *pre-Leontief* matrices in the literature.

Leontief substitution systems; an $O(m^3 n \log n)$ strongly polynomial algorithm for a special case where A has no more than two nonzero elements in any column [1], an $O(m^2 n)$ strongly polynomial algorithm for a special case of gainfree Leontief substitution systems [16], and a simplex algorithm [4], where m and n respectively denote the number of equations and variables in (1). The gainfree property will be defined in Section 2; it intuitively says that the corresponding network, which will also be defined later, has no *gain* of flow.

We also remark that Leontief substitution systems play an important role in polyhedral combinatorics and logic. For example, Horn systems are related to Leontief substitution systems. A matrix A is called *Horn* if each row of A has at most one positive element, and a linear system $A\boldsymbol{y} \le \boldsymbol{c}$ with Horn matrix A is called *Horn*. Thus, Horn matrices are exactly transposed Leontief matrices, and the feasibility for Horn systems coincides with that of the dual of LP problems with Leontief substitution systems. The feasibility for Horn systems was inspired by the Horn Boolean satisfiability (SAT) problem, a well-studied subclass of SAT in logic and computer science. Horn systems have been intensively studied in the literature [8, 13, 32] because they have applications in diverse areas such as logic programs, econometrics, program verification, and lattice optimization. Subclasses of Horn systems called difference constraint (DC), unit Horn, and unit-positive Horn systems are also extensively investigated, where a matrix A is *difference* if it is a $\{0, \pm 1\}$-matrix having one $+1$ and one -1 in each row [2, 11, 14, 26], *unit Horn* if it is a Horn $\{0, \pm 1\}$-matrix [5, 30], and *unit-positive Horn* if it is an integral Horn matrix with the positive elements being one [30,31][2]. We note that unit and unit-positive Horn systems are sometimes called Horn constraint and extended Horn, respectively. By definition, difference matrices are unit Horn, and unit Horn matrices are unit-positive Horn. All these matrices are transposed gainfree Leontief matrices, which will be discussed in the next section. The feasibility problem is combinatorially solvable in $O(mn)$ for DC systems [2, 11] and $O(m^2 n)$ for unit and unit-positive Horn systems [5], where m and n respectively denote the number of variables and inequalities in the system.

In this paper, we study certifying algorithms for LP problems with gainfree Leontief substitution systems. Recently, much attention has been paid to certifying algorithms in software engineering; see [22] for a survey. Intuitively, an algorithm is called *certifying* if it produces not only an answer but also a certificate with which we can easily confirm that the answer is correct. For the shortest s-t path problem with positive edge length, the potential of vertices (i.e., distances from s) is a certificate of a shortest s-t path. Certifying algorithms have great advantages in practice because many commercial programs are reported to contain bugs [22]. Certifying algorithms have been proposed for various problems in mathematical optimization and computer science [3, 6, 7, 10, 12, 17, 20, 23, 24, 27, 29] in the past few decades.

Let us briefly summarize certifying algorithms related to gainfree Leontief substitution systems. Standard LP solvers output a certificate of the optimality of an optimal solution; however, no combinatorial and strongly polynomial time algorithm for general LP problems is known and algorithms that work for special types of LP problems have been extensively studied. We first note that the well-known Bellman-Ford algorithm for the shortest path problem allowing negative edge length can be regarded as a certifying algorithm for the feasibility of DC systems. In fact, the algorithm computes a feasible solution which correspond to the potential of the associated graph G if it is feasible, and a minimal infeasible subsystem that corresponds to a negative cycle in G if it is infeasible. This result was extended to the unit-two-variable-per-inequality (UTVPI) systems, where a system is called *unit-two-variable-*

[2] The unit-positive matrices coincide with the transposes of integral gainfree Leontief matrices considered in [16], since in [16] any positive element of the matrices is assumed to be one.

per-inequality if each inequality is of the form $\pm x_i \pm x_j \leq c$ for some integer c. Miné [25] proposed a certifying algorithm for the feasibility for UTVPI systems by transforming such systems to DC systems. Therefore, the feasibility for the systems admits combinatorial $O(mn)$ certifying algorithms. Gupta [15] reported that a combinatorial certifying algorithm exists for the feasibility for unit Horn systems with nonpositivity constraints on variables, and mentioned that it is open whether the feasibility problem admits combinatorial certifying algorithms when the systems are unit Horn (without nonpositivity constraints) and unit-positive Horn [15]. For LP problems with gainfree Leontief substitution systems, Jeroslow et al. proposed a combinatorial $O(m^2n)$-time certifying algorithm when it has an optimal solution [16].

In this paper, we propose a combinatorial $O(m^3n)$-time certifying algorithm for LP problems with gainfree Leontief substitution systems when the LP problems have no optimal solution, i.e., when they are unbounded or infeasible. This together with the algorithm by Jeroslow et al. provides a combinatorial $O(m^3n)$-time fully certifying algorithm for LP problems with gainfree substitution systems. As a corollary of our result, we resolve the open problem for the feasibility for unit-positive Horn systems.

Certifying infeasibility draws much attention in, e.g., the field of logic and it was open how to make existing successive-approximation type combinatorial algorithms (e.g., [5,13,16]) certifying for a fundamental class of unit Horn systems. In those algorithms, the values of variables are iteratively updated according to the constraints and for DC systems, it is sufficient to store the previous edge (or constraint) that causes the value update of a variable to obtain a certificate of infeasibility (i.e., a negative cycle). However, in unit Horn systems, this is not enough; we have to store all the history of the value updates of the variables to certify infeasibility. Our algorithm stores in which iteration the values of variables are updated and how the values can be derived by the given constraints and utilizes these data to compute a certificate of infeasibility when the system is infeasible.

Our algorithm is based on the directed hypergraph representation of Leontief substitution systems introduced by Jeroslow et al. [16], and computes a certificate of infeasibility based on Farkas' lemma, called a Farkas' certificate, which was also used by Gupta [15] for unit Horn systems with nonpositivity constraints. Moreover, our algorithm for the dual feasibility can be seen as an extension of the Bellman-Ford algorithm for the feasibility for DC systems. In fact, if a DC system is given, then our algorithm finds a feasible solution if it is feasible, and a minimal infeasible subsystem that corresponds to a negative cycle in the associated graph if it is infeasible, which is the same as the Bellman-Ford algorithm.

The rest of the paper is organized as follows. Section 2 formally defines our problem and introduces the same directed hypergraph representation of Leontief substitution systems as in [16]. Section 3 provides our main algorithm, i.e., a combinatorial certifying algorithm for LP problems with gainfree Leontief substitution systems. Section 4 concludes the paper.

2 Preliminaries

Let $\mathbb{R}$, $\mathbb{R}_+$, and $\mathbb{R}_{++}$ denote the sets of reals, nonnegative reals, and positive reals, respectively. For positive integers m and n, a matrix $A \in \mathbb{R}^{m \times n}$ is called *Leontief* if each column contains at most one positive entry. In this paper, it is always assumed that the positive elements of A are all ones unless otherwise stated, since it is sufficient for our purpose as stated below. Let $A \in \mathbb{R}^{m \times n}$ be an $m \times n$ matrix, and let $\boldsymbol{b} \in \mathbb{R}^m$ be a vector of dimension m. A linear system of the form

$$A\boldsymbol{x} = \boldsymbol{b} \quad \text{and} \quad \boldsymbol{x} \in \mathbb{R}^n_+$$

is called a *Leontief substitution system* if A is Leontief and $\boldsymbol{b} \in \mathbb{R}^m_+$.

In this paper, we consider the following linear programming (LP) problem:

$$\begin{array}{ll} \text{minimize} & \boldsymbol{c}^T \boldsymbol{x} \\ \text{subject to} & A\boldsymbol{x} = \boldsymbol{b} \\ & \boldsymbol{x} \in \mathbb{R}^n_+, \end{array} \tag{2}$$

where the constraint is a Leontief substitution system and $\boldsymbol{c} \in \mathbb{R}^n$. As stated above, we assume throughout the paper that the positive elements of A are all ones unless otherwise stated, since otherwise it can be obtained by scaling the variables in the LP problem (2).

We particularly focus on the subclass of LP with Leontief substitution systems satisfying the *gainfree* property. To define gainfreeness, it is convenient to introduce a directed hypergraph representation [16] of Leontief substitution systems. This representation is also used to state our algorithms.

A directed hypergraph $\mathcal{H}$ is an ordered pair $\mathcal{H} = (V, \mathcal{E})$, where V is a finite set called a vertex set and $\mathcal{E}$ is a set of hyperarcs. A hyperarc $E \in \mathcal{E}$ is an ordered pair $(H(E), T(E))$ of its head and tail sets, where $H(E), T(E) \subseteq V$ and $H(E) \cap T(E) = \emptyset$. In our use, $|H(E)|$ is always at most one, i.e., $|H(E)| \leq 1$. Hence, we denote $H(E)$ by $h(E)$, and when $|h(E)| = 1$, we identify $h(E)$ with the unique element in $h(E)$, e.g., if $v \in h(E)$, then we write $h(E) = v$.

Now, we explain how to define an associated directed hypergraph $\mathcal{H} = (V, \mathcal{E})$ from a given LP problem with a Leontief substitution system (2). For a positive integer k, let $[k] = \{1, \ldots, k\}$. Let $V = \{v_i \mid i \in [m]\}$, where v_i corresponds to the ith row of A in (2) for $i \in [m]$, and let $\mathcal{E} = \{E_j \mid j \in [n]\}$, where for each $j \in [n]$ a hyperarc E_j is defined as $h(E_j) = v_i$ if $A_{ij} = 1$ for some $i \in [m]$ and $h(E_j) = \emptyset$ otherwise (i.e., $A_{ij} \leq 0$ for all $i \in [m]$), and $T(E_j) = \{v_i \in V \mid A_{ij} < 0\}$. Note that for each $j \in [n]$ hyperarc E_j corresponds to variable x_j in (2). We also associate a length function $\ell : \mathcal{E} \to \mathbb{R}$ to the hyperarc set $\mathcal{E}$, where $\ell(E_j) = c_j$ for each $E_j \in \mathcal{E}$. Moreover, we associate a positive value to each element of the tails of the hyperarcs in $\mathcal{E}$, namely, $\gamma : \bigcup_{j \in [n]} (\{E_j\} \times T(E_j)) \to \mathbb{R}_{++}$ defined as $\gamma(E_j, v_i) = -A_{ij}$ (> 0) for each $E_j \in \mathcal{E}$ and $v_i \in T(E_j)$. Note that the directed hypergraph is defined by matrix A and vector $\boldsymbol{c}$ (and $\boldsymbol{b}$ is irrelevant).

▶ **Example 1.** For the following input data, the associated directed hypergraph is drawn in Figure 1.

$$A = \begin{pmatrix} -(1/2) & 0 & 1 & 1 & 0 \\ 1 & -(1/3) & 0 & 0 & 0 \\ 0 & 1 & -9 & 0 & 1 \\ -(1/3) & -3 & -1 & 0 & 0 \end{pmatrix} \textit{ and } c = \begin{pmatrix} -6 \\ 5 \\ 3 \\ -4 \\ 2 \end{pmatrix}. \tag{3}$$

A *directed path* in directed hypergraph $\mathcal{H}$ from vertex v_1 to v_{k+1} is defined by a nonempty sequence $v_1 E_1 v_2 E_2 v_3 \cdots E_k v_{k+1}$, with no intermediate vertex or hyperarc repeated, such that $v_{i+1} = h(E_i)$ and $v_i \in T(E_i)$ for $i = 1, \ldots, k$. A directed path from vertex v_1 to v_{k+1} is a *directed cycle* if $v_1 = v_{k+1}$.

Now, we are ready to define gainfreeness.

▶ **Definition 2** (Gainfreeness). *Let $v_1 E_1 v_2 E_2 v_3 \cdots E_k v_{k+1}$ be a directed cycle, where $v_1 = v_{k+1}$. The gain of this directed cycle is defined by $1/\prod_{i=1}^{k} \gamma(E_i, v_i)$. We term a Leontief substitution system (and its defining matrix)* gainfree *if the gain of every directed cycle in the associated directed hypergraph is at most one.*

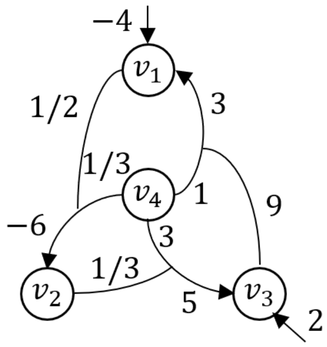

Figure 1 The directed hypergraph representation corresponding to the input (3).

From definition, unit and unit-positive Horn matrices are transpose of gainfree Leontief matrices.

▶ **Example 3.** In Example 1, the unique directed cycle of the directed hypergraph representation is $v_1E_1v_2E_2v_3E_3v_1$, where each E_i corresponds to the ith column of A. The gain of this cycle is $1/(1/2 \cdot 1/3 \cdot 9) = 2/3 \leq 1$. Hence, matrix A in (3) is gainfree.

Now, we recall some notion from LP theory. A vector $\boldsymbol{x} \in \mathbb{R}^n_+$ is called a *feasible solution* of (2) if it satisfies the constraints in (2). An LP problem is *feasible* if it has a feasible solution, and *infeasible* otherwise. A vector $\boldsymbol{x} \in \mathbb{R}^n_+$ is called an *optimal solution* of (2) if it is feasible and $\boldsymbol{c}^T\boldsymbol{x} \leq \boldsymbol{c}^T\boldsymbol{x}'$ for any feasible solution $\boldsymbol{x}'$. When an LP problem has an optimal solution $\boldsymbol{x}$, the objective value $\boldsymbol{c}^T\boldsymbol{x}$ is called an *optimal value*. An LP problem is either feasible or infeasible, and when it is feasible either it has an optimal solution or it is unbounded (i.e., its optimal value is not bounded below). Since we consider certifying algorithms, we have to produce a certificate in each case. To state what constitutes a certificate in each case, we recall the *dual* LP problem of (2):

$$\begin{array}{ll} \text{maximize} & \boldsymbol{y}^T\boldsymbol{b} \\ \text{subject to} & \boldsymbol{y}^TA \leq \boldsymbol{c}^T \\ & \boldsymbol{y} \in \mathbb{R}^m. \end{array} \tag{4}$$

To contrast, the LP problem (2) is sometimes called the *primal* LP problem in what follows.

The following duality theorem of LP is well-known.

▶ **Theorem 4** (E.g., [28])**.** *For the LP problem* (2) *and its dual problem* (4)*, exactly one of the following holds:*

(i) *both* (2) *and* (4) *have feasible solutions whose objective values are the same,*
(ii) (2) *is infeasible, and* (4) *feasible and unbounded,*
(iii) (2) *is feasible and unbounded, and* (4) *is infeasible;*
(iv) *both* (2) *and* (4) *are infeasible.*

We regard a feasible solution as a certificate of feasibility of an LP problem. For infeasibility we use the following lemma.

▶ **Lemma 5** (E.g., [28])**.** *The LP problem* (2) *is infeasible if and only if*

$$\boldsymbol{z}^TA \leq \boldsymbol{0}, \;\; \boldsymbol{z}^T\boldsymbol{b} > 0, \;\; \text{and } \boldsymbol{z} \in \mathbb{R}^m \tag{5}$$

is feasible. Moreover, the dual LP problem (4) *is infeasible if and only if*

$$A\boldsymbol{r} = \boldsymbol{0}, \;\; \boldsymbol{c}^T\boldsymbol{r} < 0, \;\; \text{and } \boldsymbol{r} \in \mathbb{R}^n_+ \tag{6}$$

is feasible.

Now, we define what constitute certificates for the four possible cases in Theorem 4.

(i) Feasible solutions of (2) and (4) whose objective values are the same,
(ii) a feasible solution of (5) (called a *Farkas' certificate* of infeasibility of (2)) and a feasible solution of (4),
(iii) a feasible solution of (2) and a feasible solution of (6) (called a *Farkas' certificate* of infeasibility of (4)),
(iv) a feasible solution of (5) and a feasible solution of (6).

With those certificates, we can confirm the correctness of the output of our certifying algorithm for solving the LP problem (2) by checking if given vectors satisfy the corresponding linear systems. We note that for case (ii) (resp., (iii)) a feasible solution of (5) (resp., (6)) is a direction of unboundedness.

3 Main algorithms

In this section, we provide a combinatorial certifying algorithm for LP problems with gainfree Leontief substitution systems (2) and show the following theorem. Here, a combinatorial algorithm consists only of additions, subtractions, multiplications, and comparisons. Recall that m is the number of constraints and n is the number of variables in (2).

▶ **Theorem 6** (Main). *The LP problems with gainfree Leontief substitution systems* (2) *admit a combinatorial strongly polynomial time certifying algorithm that runs in* $\mathrm{O}(m^3 n)$ *time.*

Our algorithm extends the non-certifying algorithm in [16]. Let us first summarize the algorithm in [16], which consists of ValueIteration and PrimalRetrieval. ValueIteration determines feasibility of the dual LP problem (4). It starts from a sufficiently large vector and iteratively compute an upper bound of the value of each variable derived from the constraints in (4). For an LP problem with a gainfree Leontief substitution system, m iterations are shown to be sufficient to obtain a feasible solution if the dual LP problem is feasible. Then, feasibility of the primal LP problem (2) can be determined using the data computed in ValueIteration, and when both primal and dual LP problems are feasible, PrimalRetrieval computes a feasible solution of the primal LP problem. This algorithm outputs feasible solutions of the primal and dual LP problems with the same objective values as a certificate of primal and dual feasibility for case (i) in Theorem 4 in Section 2.

To make the algorithm in [16] also certifying for primal and dual infeasibility (i.e., for cases (ii-iv) in Theorem 4), we modify the algorithm and add several subroutines to it. We first modify ValueIteration to DualFeasibility (Algorithm 2). In DualFeasibility, when the upper bound $\boldsymbol{y}^{(k)}$ for the dual variables is updated in the kth iteration of the for-loop starting from line 2, we store (i) variables changed in the iteration in array $\mathrm{change}^{(k)}$ and (ii) vectors $\boldsymbol{r}^{(k)}$ that represents how an upper bound $\boldsymbol{y}^{(k)}$ is derived from the constraint in (4). This enables us to compute a Farkas' certificate of dual infeasibility in FarkasCertificateOfDualInfeasibility (Algorithm 4) when the dual LP problem is infeasible. This modification also makes our algorithm different from the one in [15]. Since the upper bound $\boldsymbol{y}^{(m)}$ computed in DualFeasibility contains symbol M as described below, we compute in DualSolution (Algorithm 3) a feasible solution of the dual LP problem from $\boldsymbol{y}^{(m)}$ when the dual LP problem is feasible. Intuitively, if we substitute sufficiently large value for M, then $\boldsymbol{y}^{(m)}$ becomes a feasible solution. Then PrimalFeasibility (Algorithm 6) determines the feasibility of the primal LP problem (2) using the same criterion as in (ii) of Theorem 3.6 in [16]. PrimalSolution is almost the same as PrimalRetrieval in [16], however, the former computes a primal feasible solution even when the dual LP problem is infeasible by running DualFeasibility for a feasible dual LP problem where $\boldsymbol{c}$ is set to $\mathbf{0}$. Finally, in DualFeasibility we treat M as a symbol representing an "arbitrary large" number so that we can compute a Farkas' certificate of primal infeasibility in FarkasCertificateOfPrimalInfeasibility (Algorithm 8) by just taking the coefficient of M in $\boldsymbol{y}^{(m)}$. More precisely, for any real numbers $\alpha_1, \alpha_2, \beta_1, \beta_2 \in \mathbb{R}$, we define $\alpha_1 M + \beta_1 > \alpha_2 M + \beta_2$ if and only if $\alpha_1 > \alpha_2$ or ($\alpha_1 = \alpha_2$ and $\beta_1 > \beta_2$)[3]. In what follows, we denote by e_i (resp., e_E) an unit vector of appropriate size, where its ith element (resp., its element indexed by hyperarc E) is 1 and all other elements are 0.

For the readability, we first describe a certifying algorithm for the feasibility for the dual of the LP problems (with gainfree Leontief substitution systems) in Subsection 3.1 and one for the feasibility for the primal LP problems in Subsection 3.2. A proof of Theorem 6 will be given in Subsection 3.3. Due to the space limitation, we omit proofs of most results.

3.1 A certifying algorithm for the feasibility for the dual LP problem

In this subsection, we provide a certifying algorithm for the feasibility for the dual (4) of the LP problem with a gainfree Leontief substitution system. The main algorithm (Algorithm 1) first calls subroutine DualFeasibility (Algorithm 2), which determines the feasibility of the dual LP problem (4). If it is feasible, then subroutine DualSolution (Algorithm 3) is called to compute a feasible solution of the dual LP problem; otherwise, subroutine FarkasCertificateOfDualInfeasibility (Algorithm 4) is called to compute a Farkas' certificate of dual infeasibility.

Algorithm 1 A combinatorial certifying algorithm for the feasibility for the dual of the LP problems with gainfree Leontief substitution systems.

Input: A matrix A and a vector $\boldsymbol{c}$ for the constraint of the dual LP problem (4).
$(\boldsymbol{y}^{(m)}, \boldsymbol{r}^{(k)}(k = 0, ..., m), \mathrm{change}^{(k)}(k = 0, ..., m), p^{(k)}(k = 0, ..., m), \mathrm{nontriv}^{(m)}, \boldsymbol{q}, \mathrm{VALUE}) \leftarrow$ DualFeasibility$(A, \boldsymbol{c})$.
if VALUE = true **then**
  $\boldsymbol{y}^* \leftarrow$ DualSolution$(A, \boldsymbol{c}, \boldsymbol{y}^{(m)})$.
  print "dual-feasible" and **return** $\boldsymbol{y}^*$.
else
  $\boldsymbol{r}^* \leftarrow$ FarkasCertificateOfDualInfeasibility$(A, \boldsymbol{c}, \boldsymbol{y}^{(m)}, \boldsymbol{r}^{(k)}(k = 0, ..., m), \mathrm{change}^{(k)}(k = 0, ..., m), p^{(k)}(k = 0, ..., m))$.
  print "dual-infeasible" and **return** $\boldsymbol{r}^*$.
end

Before going into proofs of correctness of these algorithms, we show an example how these algorithms work. We only describe how upper bound $\boldsymbol{y}^{(k)}$ and vector $\boldsymbol{r}_v^{(k)}$ are updated in each iteration of the for-loop starting from line 2 in DualFeasibility in the example for readability. Also, we omit the input vector $\boldsymbol{b}$ in the example, since $\boldsymbol{b}$ is irrelevant to feasibility of the dual LP problem (4).

[3] We may regard $\alpha M + \beta$ as an element (α, β) of $\mathbb{R}^2$ equipped with a lexicographical order, i.e., $(\alpha_1, \beta_1) > (\alpha_2, \beta_2)$ if and only if $\alpha_1 > \alpha_2$ or ($\alpha_1 = \alpha_2$ and $\beta_1 > \beta_2$). This fact was pointed out by a reviewer. We use notation $\alpha M + \beta$, since we substitute some value for M in our algorithm.

Algorithm 2 DualFeasibility.

Input: A matrix A and a vector $\boldsymbol{c}$ for the constraint of the dual LP problem (4).

For each $v \in V$, $y^{(0)}(v) \leftarrow M$, $\boldsymbol{r}_v^{(0)} \leftarrow \boldsymbol{0}$, $\text{change}^{(0)}(v) \leftarrow \text{false}$, $p^{(0)}(v) \leftarrow \emptyset$, $\text{nontriv}^{(0)}(v) \leftarrow \text{false}$, and $q(v) \leftarrow 0$.
for $k = 1, \ldots, m$ **do**
  for $v \in V$ **do**
    if $y^{(k-1)}(v) > \min\left\{\ell(E) + \sum_{u \in T(E)} \gamma(E,u) y^{(k-1)}(u) \mid E \in \mathcal{E}, h(E) = v\right\}$ **then**
      Choose an arbitrary $E \in \operatorname{argmin}\left\{\ell(E) + \sum_{u \in T(E)} \gamma(E,u) y^{(k-1)}(u) \mid E \in \mathcal{E}, h(E) = v\right\}$.
      $y^{(k)}(v) \leftarrow \ell(E) + \sum_{u \in T(E)} \gamma(E,u) y^{(k-1)}(u)$.
      $p^{(k)}(v) \leftarrow E$.
      $\boldsymbol{r}_v^{(k)} \leftarrow e_E + \sum_{u \in T(E)} \gamma(E,u) \boldsymbol{r}_u^{(k-1)}$.
      $\text{change}^{(k)}(v) \leftarrow \text{true}$.
      if *for every* $u \in T(E)$ $\text{nontriv}^{(k-1)}(u) = \text{true}$ *(this includes the case that* $T(E) = \emptyset$*)* **then**
        $\text{nontriv}^{(k)}(v) \leftarrow \text{true}$ and $q(v) \leftarrow k$.
      else
        $\text{nontriv}^{(k)}(v) \leftarrow \text{nontriv}^{(k-1)}(v)$.
      end
    else
      $y^{(k)}(v) \leftarrow y^{(k-1)}(v)$, $p^{(k)}(v) \leftarrow \emptyset$, $\boldsymbol{r}_v^{(k)} \leftarrow \boldsymbol{r}_v^{(k-1)}$, $\text{change}^{(k)}(v) \leftarrow \text{false}$, and $\text{nontriv}^{(k)}(v) \leftarrow \text{nontriv}^{(k-1)}(v)$.
    end
  end
end
if $y^{(m)}(v) > \min\left\{\ell(E) + \sum_{u \in T(E)} \gamma(E,u) y^{(m)}(u) \mid E \in \mathcal{E}, h(E) = v\right\}$ for some $v \in V$ **then**
  VALUE $\leftarrow$ false.
else if $0 > \ell(E) + \sum_{u \in T(E)} \gamma(E,u) y^{(m)}(u)$ *for some* $E \in \mathcal{E}$ *with* $h(E) = \emptyset$ **then**
  VALUE $\leftarrow$ false.
else
  VALUE $\leftarrow$ true.
end
return $(\boldsymbol{y}^{(m)}, \boldsymbol{r}^{(k)}(k = 0, ..., m), \text{change}^{(k)}(k = 0, ..., m), p^{(k)}(k = 0, ..., m), \text{nontriv}^{(m)}, \boldsymbol{q}, \text{VALUE})$.

▶ **Example 7.** For the following matrix A (whose transpose is unit Horn) and vector $\boldsymbol{c}$

$$A = \begin{pmatrix} -1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 1 & 0 \\ -1 & -1 & -1 & 0 & 0 & 0 & 1 \end{pmatrix} \text{ and } \boldsymbol{c} = \begin{pmatrix} -1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix},$$

Algorithm 3 DualSolution.

Input: A matrix A and a vector $\boldsymbol{c}$ for the constraint of the dual LP problem (4), and an n-dimensional vector $\boldsymbol{y}^{(m)}$ with each entry being a linear function of M.

```
for each E ∈ 𝓔 do
    Define two integers α(E) and β(E) such that
      α(E)M + β(E) = y^(m)(h(E)) − ℓ(E) − Σ_{u∈T(E)} γ(E,u) y^(m)(u) , if where we
      define y^(m)(∅) = 0.
end
if all E ∈ 𝓔 satisfy α(E) ≥ 0 then
    λ ← 0.
else
    λ ← max { β(E)/(−α(E)) | E ∈ 𝓔, α(E) < 0 }.
end
Let y* be the vector obtained from y^(m) by substituting λ for M.
return y*.
```

$\boldsymbol{y}^{(0)} = (M, M, M, M)^T$ and $\boldsymbol{r}_{v_i}^{(0)} = \mathbf{0}$ for $i = 1, 2, 3, 4$.
Iteration 1: $\boldsymbol{y}^{(1)} = (0, 0, 0, 0)^T$ and $\boldsymbol{r}_{v_i}^{(1)} = e_{i+3}$ $(i = 1, 2, 3, 4)$.
Iteration 2: $\boldsymbol{y}^{(2)} = (0, -1, 0, 0)^T$ and $\boldsymbol{r}_{v_2}^{(2)} = e_1 + e_4 + e_7$.
Iteration 3: $\boldsymbol{y}^{(3)} = (0, -1, -1, 0)^T$ and $\boldsymbol{r}_{v_3}^{(3)} = e_1 + e_2 + e_4 + 2e_7$.
Iteration 4: $\boldsymbol{y}^{(4)} = (-1, -1, -1, 0)^T$ and $\boldsymbol{r}_{v_1}^{(4)} = e_1 + e_2 + e_3 + e_4 + 3e_7$.
Now, the first inequality is violated by $\boldsymbol{y}^{(4)}$ as $(-1, 0, 1, -1)\boldsymbol{y}^{(4)} = 0 > -1$. Then, by running FarkasCertificateOfDualInfeasibility, we have $\boldsymbol{r}^* = \boldsymbol{r}_{v_2}^{(5)} - \boldsymbol{r}_{v_2}^{(2)} = e_1 + e_2 + e_3 + 3e_7$. Here, $A\boldsymbol{r}^* = (0, 0, 0, 0)^T$ and $c^T\boldsymbol{r}^* = -1$. Hence, $\boldsymbol{r}^*$ is a Farkas' certificate of infeasibility of the dual LP problem (4).

In the remainder of this subsection, we will prove correctness of Algorithm 1. We show correctness of subroutines DualFeasibility, DualSolution[4], and FarkasCertificateOfDualInfeasibility, and show the following proposition.

▶ **Proposition 8.** *Algorithm 1 is a combinatorial strongly polynomial time certifying algorithm that runs in* $\mathrm{O}(m^3 n)$ *time for the feasibility for the dual* (4) *of the LP problem with a gainfree Leontief substitution system.*

To show Proposition 8, we first deal with the case where Algorithm 1 prints "dual-feasible" (or, equivalently, DualFeasibility returns true) in Lemma 9 below. Then, we deal with the case where Algorithm 1 prints "dual-infeasible" (or, equivalently, DualFeasibility returns false) in Lemma 10 below.

▶ **Lemma 9.** *If DualFeasibility returns* true*, then the dual LP problem* (4) *is feasible and DualSolution outputs a feasible solution to it.*

Proof. We show that the output $\boldsymbol{y}^*$ of DualSolution is a feasible solution of the dual LP problem (4). We divide the proof into cases according to the conditions in the definition of λ in DualSolution.

[4] DualSolution uses a division, however, we can avoid the division by using ValueIteration in [16] to obtain a feasible dual solution.

Algorithm 4 FarkasCertificateOfDualInfeasibility.

Input: A matrix A and a vector $\boldsymbol{c}$ for the constraint of the dual LP problem (2), $\boldsymbol{y}^{(m)}$, and $\boldsymbol{r}^{(k)}$, change$^{(k)}$, and $p^{(k)}$ for $k = 0, ..., m$.

if $y^{(m)}(v) > \min\left\{\ell(E) + \sum_{u\in T(E)} \gamma(E,u)y^{(m)}(u) \mid E \in \mathcal{E}, h(E) = v\right\}$ for some $v \in V$ **then**
    Choose one $v \in V$ such that $y^{(m)}(v) > \min\left\{\ell(E) + \sum_{u\in T(E)} \gamma(E,u)y^{(m)}(u) \mid E \in \mathcal{E}, h(E) = v\right\}$.
    Choose an arbitrary $E \in \mathcal{E}$ with $h(E) = v$ that minimizes $\ell(E) + \sum_{u\in T(E)} \gamma(E,u)y^{(m)}(u)$.
    $w_{m+1} \leftarrow v$.
    $\boldsymbol{r}^{(m+1)}_{w_{m+1}} \leftarrow e_E + \sum_{u\in T(E)} \gamma(E,u)\boldsymbol{r}^{(m)}_u$.
    $E^{(m+1)} \leftarrow E$.
    /* Find a cycle */
    for $k = m+1, \ldots, 2$ **do**
        Choose an arbitrary $u \in T(E^{(k)})$ such that change$^{(k-1)}(u) =$ true.
        $w_{k-1} \leftarrow u$.
        $E^{(k-1)} \leftarrow p^{(k-1)}(w_{k-1})$.
        if $w_{k-1} = w_q$ *for some* $q \geq k$ **then**
            $t \leftarrow q$.
            $s \leftarrow k-1$.
            Break.
        end
    end
    $\boldsymbol{r}^* \leftarrow \boldsymbol{r}^{(t)}_{w_t} - \boldsymbol{r}^{(s)}_{w_s}$.
    return $\boldsymbol{r}^*$.
else
    Choose one $E \in \mathcal{E}$ with $h(E) = \emptyset$ such that $0 > \ell(E) + \sum_{u\in T(E)} \gamma(E,u)y^{(m)}(u)$.
    $\boldsymbol{r}^* \leftarrow \boldsymbol{e}_E + \sum_{u\in T(E)} \gamma(E,u)\boldsymbol{r}^{(m)}_u$.
    return $\boldsymbol{r}^*$.
end

Fix $E \in \mathcal{E}$. Note that we have

$$y^{(m)}(h(E)) \leq \ell(E) + \sum_{u\in T(E)} \gamma(E,u)y^{(m)}(u),$$

since the conditions of "if " and "else if " in lines 20 and 22, respectively, are false in DualFeasibility, where we define $y^{(m)}(\emptyset) = 0$. Hence, we have $\alpha(E)M + \beta(E) \leq 0$. It follows that $\alpha(E) \leq 0$. If $\alpha(E) = 0$, then $\beta(E) \leq 0$ and $\boldsymbol{y}^*$ satisfy the constraint in the dual LP problem (4) corresponding to E. If $\alpha(E) < 0$, then $\boldsymbol{y}^*$ also satisfies the inequality in the dual LP problem (4) corresponding to E, since $\lambda \geq \frac{\beta(E)}{-\alpha(E)}$ by definition. This completes the proof. ◀

Next, we treat the case where DualFeasibility returns false and show the following.

▶ **Lemma 10.** *If* DualFeasibility *returns* false*, then the dual LP problem* (4) *is infeasible and* FarkasCertificateOfDualInfeasibility *returns a Farkas' certificate of dual infeasibility.*

The proof of Lemma 10 is the most technical part of our results. Intuitively, when DualFeasibility returns false, we can find a "negative cycle" as in the case of difference constraint (DC) systems. Here, the gainfree property ensures that such a negative cycle, together with paths to the tails of hyperarcs in the cycle, corresponds to an infeasible subsystem of (4). The vectors $\boldsymbol{r}_v^{(k)}$ store how the negative cycle is derived from constraints in (4) and help to compute such a subsystem (with multiplicity) in FarkasCertificateOfDualInfeasibility.

We first treat the case where the "if " condition in line 20 is false and the "else if " condition in line 22 is true in DualFeasibility.

▶ **Lemma 11.** *If* DualFeasibility *returns* false *as the "if " condition in line 20 is false and the "else if " condition in line 22 is true, then the dual LP problem* (4) *is infeasible and* FarkasCertificateOfDualInfeasibility *returns a Farkas' certificate of dual infeasibility.*

To show this lemma, we need some auxiliary claims, which can be shown by mathematical induction on k.

▷ Claim 12. In the end of DualFeasibility, for all $k \in \{1, \ldots, m\}$ and $v \in V$, $y^{(k)}(v)$ contains M if and only if $\mathrm{nontriv}^{(k)}(v) = \mathrm{false}$. Moreover, if $y^{(k)}(v)$ contains M, the coefficient of M is positive for all $k = 1, \ldots, m$ and $v \in V$.

▷ Claim 13. In the end of DualFeasibility, for all $k \in \{1, \ldots, m\}$ and $v \in V$, we have $A\boldsymbol{r}_v^{(k)} \leq \boldsymbol{e}_v$, $\boldsymbol{r}_v^{(k)} \geq \boldsymbol{0}$, and $\boldsymbol{c}^T\boldsymbol{r}_v^{(k)}$ equals the constant term of $y^{(k)}(v)$. If $\mathrm{nontriv}^{(k)}(v) = \mathrm{true}$, then $A\boldsymbol{r}_v^{(k)} = \boldsymbol{e}_v$ and $\boldsymbol{c}^T\boldsymbol{r}_v^{(k)} = y^{(k)}(v)$.

Proof of Lemma 11. We show that $\boldsymbol{r}^*$ returned by FarkasCertificateOfDualInfeasibility is actually a Farkas' certificate of dual infeasibility, i.e., (i) $\boldsymbol{r}^* \geq \boldsymbol{0}$, (ii) $A\boldsymbol{r}^* = \boldsymbol{0}$, and (iii) $\boldsymbol{c}^T\boldsymbol{r}^* < 0$ (see Lemma 5).

For (i), from Claim 13, we have that $\boldsymbol{r}^*(= \boldsymbol{e}_E + \sum_{u \in T(E)} \gamma(E, u)\boldsymbol{r}_u^{(m)})$ is a sum of nonnegative vectors. Hence, $\boldsymbol{r}^* \geq \boldsymbol{0}$.

For (ii), observe that to satisfy $0 > \ell(E) + \sum_{u \in T(E)} \gamma(E, u)y^{(m)}(u)$, $y^{(m)}(u)$ must not contain M for each $u \in T(E)$, since otherwise the right-hand side of the inequality contains M with a positive coefficient from Claim 12 and thus greater than zero. Hence, for each $u \in T(E)$ $\mathrm{nontriv}^{(m)}(u) = \mathrm{true}$ from Claim 12, implying that $A\boldsymbol{r}_u^{(m)} = \boldsymbol{e}_u$ from Claim 13. Therefore, we have

$$A\boldsymbol{r}^* = A\boldsymbol{e}_E + \sum_{u \in T(E)} \gamma(E, u)A\boldsymbol{r}_u^{(m)} = -\sum_{u \in T(E)} \gamma(E, u)\boldsymbol{e}_u + \sum_{u \in T(E)} \gamma(E, u)\boldsymbol{e}_u = 0.$$

For (iii), for each $u \in T(E)$ we have $\boldsymbol{c}^T\boldsymbol{r}_u^{(m)} = y^{(m)}(u)$ from Claim 13 since $\mathrm{nontriv}^{(m)}(u) = \mathrm{true}$ as shown above. Hence, we have $\boldsymbol{c}^T\boldsymbol{r}^* = \boldsymbol{c}^T\boldsymbol{e}_E + \sum_{u \in T(E)} \gamma(E, u)\boldsymbol{c}^T\boldsymbol{r}_u^{(m)} = \ell(E) + \sum_{u \in T(E)} \gamma(E, u)y^{(m)}(u) < 0$.

Therefore, $\boldsymbol{r}^*$ is a Farkas' certificate of dual infeasibility and by Lemma 5 the dual LP problem (4) is infeasible. ◀

We then deal with the case where the "if " condition in line 20 is true in DualFeasibility.

▶ **Lemma 14.** *If* DualFeasibility *returns* false *as the "if " condition in line 20 is true, then the dual LP problem* (4) *is infeasible and* FarkasCertificateOfDualInfeasibility *returns a Farkas' certificate of the dual infeasibility.*

To show Lemma 14, we need further auxiliary claims.

▷ Claim 15. In FarkasCertificateOfDualInfeasibility, for each $k = m+1, m, \ldots, s+1$, there exists $u \in T(E^{(k)})$ such that $\mathrm{change}^{(k-1)}(u) = \mathrm{true}$.

Claim 15 implies the following claim.

▷ Claim 16. In FarkasCertificateOfDualInfeasibility, we can always obtain a cycle.

The following claim uses the gainfree property of the LP problem (2).

▷ Claim 17. In the end of FarkasCertificateOfDualInfeasibility, for any $s+1 \leq k \leq t$ and any $u \in T(E^{(k)}) \setminus \{w_{k-1}\}$, we have $\mathrm{nontriv}^{(k-1)}(u) = \mathrm{true}$.

Now, we are ready to prove Lemma 14.

Proof of Lemma 14. We show that $\boldsymbol{r}^*$ is actually a Farkas' certificate of dual infeasibility, i.e., (i) $\boldsymbol{r}^* \geq \boldsymbol{0}$, (ii) $A\boldsymbol{r}^* = \boldsymbol{0}$, and (iii) $\boldsymbol{c}^T\boldsymbol{r}^* < 0$. Due to page limitation, we only prove (ii). For (ii), recall that for any $s+1 \leq k \leq t$ and any $u \in T(E^{(k)}) \setminus \{w_{k-1}\}$, we have $A\boldsymbol{r}_u^{(k-1)} = \boldsymbol{e}_u$ from Claims 13 and 17. Moreover, we have $A\boldsymbol{e}_{E^{(k)}} = \boldsymbol{e}_{h(E^{(k)})} - \sum_{u \in T(E^{(k)})} \gamma(E^{(k)}, u)\boldsymbol{e}_u$. Hence, for each $s+1 \leq k \leq t$,

$$\begin{aligned}
A\boldsymbol{r}_{w_k}^{(k)} &= A(e_{E^{(k)}} + \sum_{u \in T(E^{(k)})} \gamma(E^{(k)}, u)\boldsymbol{r}_u^{(k-1)}) \\
&= \boldsymbol{e}_{h(E^{(k)})} - \sum_{u \in T(E^{(k)})} \gamma(E^{(k)}, u)\boldsymbol{e}_u + A(\sum_{u \in T(E^{(k)})} \gamma(E^{(k)}, u)\boldsymbol{r}_u^{(k-1)}) \\
&= \boldsymbol{e}_{w_k} + \sum_{u \in T(E^{(k)})} \gamma(E^{(k)}, u)(A\boldsymbol{r}_u^{(k-1)} - \boldsymbol{e}_u) \\
&= \boldsymbol{e}_{w_k} + \sum_{u \in T(E^{(k)}) \setminus \{w_{k-1}\}} \gamma(E^{(k)}, u)(A\boldsymbol{r}_u^{(k-1)} - \boldsymbol{e}_u) \\
&\quad + \gamma(E^{(k)}, w_{k-1})(A\boldsymbol{r}_{k-1}^{(w_{k-1})} - \boldsymbol{e}_{w_k-1}) \\
&= \boldsymbol{e}_{w_k} + \gamma(E^{(k)}, w_{k-1})(A\boldsymbol{r}_{k-1}^{(w_{k-1})} - \boldsymbol{e}_{w_k-1}).
\end{aligned}$$

Namely, we have $A\boldsymbol{r}_{w_k}^{(k)} - \boldsymbol{e}_{w_k} = \gamma(E^{(k)}, w_{k-1})(A\boldsymbol{r}_{k-1}^{(w_{k-1})} - \boldsymbol{e}_{w_k-1})$. Therefore, we have

$$A\boldsymbol{r}_{w_t}^{(t)} - \boldsymbol{e}_{w_t} = \gamma(E^{(t)}, w_{t-1})(A\boldsymbol{r}_{w_{t-1}}^{(t-1)} - \boldsymbol{e}_{w_t-1}) = \cdots = \prod_{k=s+1}^{t} \gamma(E^{(k)}, w_{k-1})(A\boldsymbol{r}_{w_s}^{(s)} - \boldsymbol{e}_{w_s}).$$

Hence, we have

$$\begin{aligned}
A\boldsymbol{r}^* &= A(\boldsymbol{r}_{w_t}^{(t)} - \boldsymbol{r}_{w_s}^{(s)}) \\
&= \boldsymbol{e}_{w_t} + \prod_{k=s+1}^{t} \gamma(E^{(k)}, w_{k-1})(A\boldsymbol{r}_{w_s}^{(s)} - \boldsymbol{e}_{w_s}) - A\boldsymbol{r}_{w_s}^{(s)} \\
&= \boldsymbol{e}_{w_t} + \left(\prod_{k=s+1}^{t} \gamma(E^{(k)}, w_{k-1}) - 1\right) A\boldsymbol{r}_{w_s}^{(s)} - \prod_{k=s+1}^{t} \gamma(E^{(k)}, w_{k-1})\boldsymbol{e}_{w_s}.
\end{aligned}$$

Now, if $\mathrm{nontriv}^{(t)}(w_t) = \mathrm{true}$, we can show that $\mathrm{nontriv}^{(s)}(w_s) = \mathrm{true}$. Hence, $A\boldsymbol{r}_{w_s}^{(s)} = \boldsymbol{e}_{w_s}$ by Claim 13. Therefore, we have

$$\begin{aligned}
&\boldsymbol{e}_{w_t} + \left(\prod_{k=s+1}^{t} \gamma(E^{(k)}, w_{k-1}) - 1\right) A\boldsymbol{r}_{w_s}^{(s)} - \prod_{k=s+1}^{t} \gamma(E^{(k)}, w_{k-1})\boldsymbol{e}_{w_s} \\
&= \boldsymbol{e}_{w_t} + (\prod_{k=s+1}^{t} \gamma(E^{(k)}, w_{k-1}) - 1)\boldsymbol{e}_{w_s} - \prod_{k=s+1}^{t} \gamma(E^{(k)}, w_{k-1})\boldsymbol{e}_{w_s} \\
&= \boldsymbol{e}_{w_t} - \boldsymbol{e}_{w_s} = \boldsymbol{0},
\end{aligned}$$

where the last equality holds since $w_t = w_s$. If $\mathrm{nontriv}^{(t)}(w_t) = \mathrm{false}$, we can show that $\prod_{k=s+1}^{t} \gamma(E^{(k)}, w_{k-1}) = 1$. Therefore, we have

$$\begin{aligned}
&\boldsymbol{e}_{w_t} + \left(\prod_{k=s+1}^{t} \gamma(E^{(k)}, w_{k-1}) - 1\right) A\boldsymbol{r}_{w_s}^{(s)} - \prod_{k=s+1}^{t} \gamma(E^{(k)}, w_{k-1})\boldsymbol{e}_{w_s} \\
&= \boldsymbol{e}_{w_t} - \boldsymbol{e}_{w_s} = \boldsymbol{0}.
\end{aligned}$$

In either case, we have $A\boldsymbol{r}^* = \boldsymbol{0}$. ◀

Combining Lemma 11 and Lemma 14, we obtain Lemma 10.

Now, we are ready to show Proposition 8.

Proof of Proposition 8. Note that subroutines DualFeasibility, DualSolution, and FarkasCertificateOfDualInfeasibility constitute a certifying algorithm for the feasibility for the dual LP problem (4) (Algorithm 1). The correctness of this algorithm follows from Lemmas 9, 11, and 14.

Now, we analyze the running time of the algorithm. The most time-consuming part of the algorithm is the for-loop from line 2 to 19 in DualFeasibility. This for-loop has m iterations, and $\mathrm{O}(mn)$ operations for computing $\boldsymbol{r}_v^{(k)}$ each $v \in V$ in each iteration. Hence, it takes $\mathrm{O}(m^3 n)$ time. Moreover, since in each of the m iterations the numbers glow $\mathrm{O}(\max(\max_{i,j}(A_{ij}), \max_i(b_i), \max_j(c_j)) \cdot n)$ time, the bit-lengths of the numbers appearing during the algorithm can be bounded by a polynomial in the size of the input. Hence, the algorithm is a strongly polynomial time one. ◀

3.2 A certifying algorithm for the feasibility for the primal LP problem

In this subsection, we provide a certifying algorithm for the feasibility for the primal LP problem (2) with a gainfree Leontief substitution system, using the data computed in DualFeasibility. More precisely, we show that subroutines PrimalFeasibility (Algorithm 6), PrimalSolution (Algorithm 7), and FarkasCertificateOfPrimalInfeasibility (Algorithm 8), together with DualFeasibility, constitute a certifying algorithm for the feasibility for the primal LP problem (2) (Algorithm 5). PrimalFeasibility determines feasibility of the primal LP problem (2) using the same criterion as in (ii) of Theorem 3.6 in [16]. PrimalSolution is similar to PrimalRetrieval in [16]; however, PrimalSolution also computes a primal feasible solution when the dual LP problem is infeasible. FarkasCertificateOfPrimalInfeasibility returns a Farkas' certificate of the primal infeasibility, where the gainfree property is again crucial for the correctness.

The following example shows how these algorithms work.

▶ **Example 18.** Recall Example 7 in Subsection 3.1. In this example, we have $\mathrm{nontriv}^{(m)}(v) = \mathrm{true}$ for each $v \in V$ and $\mathrm{PrimalFeasibility}(\boldsymbol{b}, \mathrm{nontriv}^{(m)}) = \mathrm{true}$ for any $\boldsymbol{b}(\geq \boldsymbol{0})$. Hence, PrimalSolution is called in Algorithm 5. As the dual LP problem is infeasible, DualFeasibility$(A, \boldsymbol{0})$ is called in PrimalSolution and in particular $\boldsymbol{q} = (1, 1, 1, 1)^T$ is obtained.

Algorithm 5 Combinatorial certifying algorithm for the feasibility for the primal LP problems with gainfree Leontief substitution systems.

Input: A matrix A and vectors $\boldsymbol{b}$ and $\boldsymbol{c}$ for the primal LP problem (2).
$(\boldsymbol{y}^{(m)}, \boldsymbol{r}^{(k)}(k = 0, ..., m), \text{change}^{(k)}(k = 0, ..., m), p^{(k)}(k = 0, ..., m), \text{nontriv}^{(m)}, \boldsymbol{q}, \text{VALUE}) \leftarrow$ DualFeasibility$(A, \boldsymbol{c})$.
if *PrimalFeasibility*$(\boldsymbol{b}, \text{nontriv}^{(m)}) = \text{true}$ **then**
  $\boldsymbol{x}^* \leftarrow$ PrimalSolution$(A, \boldsymbol{b}, \text{nontriv}^{(m)}, p^{(k)}(k = 0, ..., m), \boldsymbol{q}, \text{VALUE})$.
  print "primal-feasible" and **return** $\boldsymbol{x}^*$.
else
  $\boldsymbol{z}^* \leftarrow$ FarkasCertificateOfPrimalInfeasibility$(\boldsymbol{y}^{(m)}, \text{nontriv}^{(m)})$.
  print "primal-infeasible" and **return** $\boldsymbol{z}^*$.
end

Then in the while-loop in PrimalSolution variables $\boldsymbol{x}^*$ and $\boldsymbol{f}$ are updated as follows. Initially, $\boldsymbol{x}^* = \boldsymbol{0}$ and $\boldsymbol{f} = (b_1, b_2, b_3, b_4)^T$. First, we may choose v_1 according to $\boldsymbol{q}$ and since $p^{(1)}(v_1) = E_4$, $x^*(E_4) = b_1$ and $\boldsymbol{f}$ remains unchanged. Then we may choose v_2 and since $p^{(1)}(v_2) = E_5$, $x^*(E_5) = b_2$ and $\boldsymbol{f}$ remains unchanged. Then we may choose v_3 and since $p^{(1)}(v_3) = E_6$, $x^*(E_6) = b_3$ and $\boldsymbol{f}$ remains unchanged. Finally, we choose v_4 and since $p^{(1)}(v_4) = E_7$, $x^*(E_7) = b_4$. Then we obtain a feasible solution $\boldsymbol{x}^* = (0, 0, 0, b_1, b_2, b_3, b_4)^T$ of the primal LP problem (2).

Algorithm 6 PrimalFeasibility.

Input: A vector $\boldsymbol{b}$ and $\text{nontriv}^{(m)}$.
if $b(v) = 0$ *for all* v *with* $\text{nontriv}^{(m)}(v) = \text{false}$ **then**
  return true.
else
  return false.
end

Now, we show correctness of subroutines PrimalFeasibility, PrimalSolution, and FarkasCertificateOfPrimalInfeasibility, and show the following proposition.

▶ **Proposition 19.** *Algorithm 5 is a combinatorial strongly polynomial time certifying algorithm that runs in* $\mathrm{O}(m^3 n)$ *time for the feasibility for the primal LP problem* (2) *with a gainfree Leontief substitution system.*

To show Proposition 19, we use the following lemmas.

▶ **Lemma 20.** *If* PrimalFeasibility *returns* true*, then the primal LP problem* (2) *is feasible and* PrimalSolution *returns a feasible solution of* (2)*.*

▶ **Lemma 21.** *If* PrimalFeasibility *returns* false*, then the primal LP problem* (2) *is infeasible and* FarkasCertificateOfPrimalInfeasibility *returns a Farkas' certificate of the primal infeasibility.*

Proof of Proposition 19. Note that subroutines PrimalFeasibility, PrimalSolution, and FarkasCertificateOfPrimalInfeasibility, together with DualFeasibility, constitute a certifying algorithm for the feasibility for the primal LP problem (2) (Algorithm 5). Correctness of this algorithm follows from Lemmas 20 and 21.

Algorithm 7 PrimalSolution.

Input: A matrix A and a vector $\boldsymbol{b}$ for the constraint of the primal LP problem (2), and $\text{nontriv}^{(m)}, p^{(k)}(k = 0, ..., m), \boldsymbol{q}, \text{VALUE}$.

if $\text{VALUE} = \text{false}$ **then**
  $(\boldsymbol{y}^{(m)}, \boldsymbol{r}^{(k)}(k = 0, ..., m), \text{change}^{(k)}(k = 0, ..., m), p^{(k)}(k = 0, ..., m), \boldsymbol{q}, \text{VALUE}) \leftarrow \text{DualFeasibility}(A, \boldsymbol{0})$.
end
For each $E \in \mathcal{E}$, $x^*(E) \leftarrow 0$, $\tilde{V} \leftarrow \{i \in V \mid \text{nontriv}^{(m)}(i) = \text{true}\}$, and for each $v \in V$, $f(v) \leftarrow b(v)$.
while $\tilde{V} \neq \emptyset$ **do**
  Choose an arbitrary $v \in \tilde{V}$ with maximum $q(v)$.
  $E \leftarrow p^{(q(v))}(v)$.
  $x^*(E) \leftarrow f(v)$.
  $f(u) \leftarrow f(u) + \gamma(E, u)x(E)$ for each $u \in T(E)$.
  $\tilde{V} \leftarrow \tilde{V} \setminus \{v\}$.
end
return $\boldsymbol{x}^*$.

Algorithm 8 FarkasCertificateOfPrimalInfeasibility.

Input: A vector $\boldsymbol{y}^{(m)}$ and $\text{nontriv}^{(m)}$.

for *each* $v \in V$ **do**
  if $\text{nontriv}^{(m)}(v) = \text{true}$ **then**
    $z^*(v) \leftarrow 0$.
  else
    $z^*(v) \leftarrow$ the coefficient of M in $y^{(m)}(v)$.
  end
end
return $\boldsymbol{z}^*$.

Now, we analyze the running time of the above algorithm. The most time-consuming part of is DualFeasibility, which runs in $\mathrm{O}(m^3 n)$ time as shown in the proof of Proposition 8. This completes the proof. ◀

3.3 Proof of the main theorem (Theorem 6)

Combining the results in Subsections 3.1 and 3.2, we can show our main theorem.

Proof of Theorem 6. From Theorem 4, Algorithms 1 and 5 constitute a certifying algorithm for solving the LP problem. Correctness and the running time of the algorithm follow from Propositions 8 and 19. ◀

Finally, the following example shows that gainfreeness is necessary for convergence of the for-loop from line 2 to 19 in DualFeasibility for a feasible dual LP problem of an LP problem with a Leontief substitution system.

▶ **Example 22.** For the following matrix A (which is Leontief but not gainfree) and vector $\boldsymbol{c}$

$$A = \begin{pmatrix} 1 & -1 \\ -(1/2) & 1 \end{pmatrix} \text{ and } \boldsymbol{c} = \begin{pmatrix} 0 \\ 0 \end{pmatrix},$$

it is clear that $(0,0)^T$ is a feasible solution of the dual LP problem (4). However, one can see that $\boldsymbol{y}^{(0)} = (M, M)^T$, $\boldsymbol{y}^{(1)} = ((1/2)M, M)^T$, $\boldsymbol{y}^{(2)} = ((1/2)M, (1/2)M)^T$, $\boldsymbol{y}^{(3)} = ((1/4)M, (1/2)M)^T$, $\boldsymbol{y}^{(3)} = ((1/4)M, (1/4)M)^T$, and so on, and the for-loop from line 2 to 19 in DualFeasibility does not converge in a finite number of iterations.

4 Conclusion

We proposed a combinatorial strongly polynomial time certifying algorithm for the LP problems with gainfree Leontief substitution systems. Since the dual LP problems with gainfree Leontief substitution systems contains the feasibility for unit-positive Horn systems, we resolved the open questions raised in [15].

An interesting future direction would be to make other non-certifying algorithms certifying. A candidate would be to extend our result on unit Horn systems to unit *q-Horn* systems, introduced in [18]. Unit q-Horn systems include not only unit Horn systems but also unit-two-variable-per-inequality (UTVPI) systems, and the feasibility for unit q-Horn systems is solvable in polynomial time [18]. Furthermore, a certifying algorithm for the feasibility for UTVPI systems is known [25]. Therefore, giving a certifying algorithm for the feasibility for unit q-Horn systems would be an interesting future work.

References

1. Ilan Adler and Steven Cosares. A strongly polynomial algorithm for a special class of linear programs. *Operations Research*, 39:955–960, 1991.
2. Richard Bellman. On a routing problem. *Quarterly of applied mathematics*, 16(1):87–90, 1958.
3. Bart Bogaerts, Stephan Gocht, Ciaran McCreesh, and Jakob Nordström. Certified symmetry and dominance breaking for combinatorial optimisation. In *Proceedings of the 36th AAAI Conference on Artificial Intelligence (AAAI'22)*, 2022.
4. Riccardo Cambini, Giorgio Gallo, and Maria Grazia Scutellà. Flows on hypergraphs. *Mathematical Programming*, 78(2):195–217, 1997.
5. R. Chandrasekaran and K. Subramani. A combinatorial algorithm for horn programs. *Discrete Optimization*, 10:85–101, 2013.
6. Maria Chudnovsky, Jan Goedgebeur, Oliver Schaudt, and Mingxian Zhong. Obstructions for three-coloring graphs with one forbidden induced subgraph. In *Proceedings of the twenty-seventh annual ACM-SIAM symposium on Discrete algorithms*, pages 1774–1783. SIAM, 2016.
7. Derek G Corneil, Barnaby Dalton, and Michel Habib. Ldfs-based certifying algorithm for the minimum path cover problem on cocomparability graphs. *SIAM Journal on Computing*, 42(3):792–807, 2013.
8. Richard W. Cottle and Arthur F. Veinott, Jr. Polyhedral sets having a least element. *Mathematical Programming*, 3:238–249, 1972.
9. George B. Dantzig. Optimal solution of a dynamic leontief model with substitution. *Econometrica*, 23(3):295–302, 1955.
10. Marcel Dhiflaoui, Stefan Funke, Carsten Kwappik, Kurt Mehlhorn, Michael Seel, Elmar Schömer, Ralph Schulte, and Dennis Weber. Certifying and repairing solutions to large lps how good are lp-solvers? In *Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 255–256, 2003.

11 Lester R Ford Jr. Network flow theory. Technical report, Rand Corp Santa Monica Ca, 1956.
12 Loukas Georgiadis and Robert E Tarjan. Dominator tree certification and divergent spanning trees. *ACM Transactions on Algorithms (TALG)*, 12(1):1–42, 2015.
13 Fred Glover. A bound escalation method for the solution of integer linear programs. *Cahiers du Centre d'Etudes de Recherche Operationelle*, 6(3):131–168, 1964.
14 Andrew V Goldberg. Scaling algorithms for the shortest paths problem. *SIAM Journal on Computing*, 24(3):494–504, 1995.
15 Pratik Bijaiprakash Gupta. A certifying algorithm for Horn constraint systems. Master's thesis, The University of Texas at Dallas, 2014.
16 Robert G Jeroslow, Kipp Martin, Ronald L Rardin, and Jinchang Wang. Gainfree leontief substitution flow problems. *Mathematical Programming*, 57(1):375–414, 1992.
17 Haim Kaplan and Yahav Nussbaum. Certifying algorithms for recognizing proper circular-arc graphs and unit circular-arc graphs. *Discrete Applied Mathematics*, 157(15):3216–3230, 2009.
18 Kei Kimura and Kazuhisa Makino. Trichotomy for integer linear systems based on their sign patterns. *Discrete Applied Mathematics*, 200:67–78, 2016. `doi:10.1016/j.dam.2015.07.004`.
19 Kei Kimura and Kazuhisa Makino. A combinatorial certifying algorithm for linear programming problems with gainfree leontief substitution systems. *arXiv*, 2023. `arXiv:2306.03368`.
20 Dieter Kratsch, Ross M McConnell, Kurt Mehlhorn, and Jeremy P Spinrad. Certifying algorithms for recognizing interval graphs and permutation graphs. *SIAM Journal on Computing*, 36(2):326–353, 2006.
21 Wassily W Leontief. *The structure of American economy, 1919-1939*. Oxford University Press, second edition, 1951.
22 Ross M McConnell, Kurt Mehlhorn, Stefan Näher, and Pascal Schweitzer. Certifying algorithms. *Computer Science Review*, 5(2):119–161, 2011.
23 Kurt Mehlhorn, Stefan Naher, and Stefan Näher. *LEDA: A platform for combinatorial and geometric computing*. Cambridge university press, 1999.
24 Kurt Mehlhorn, Adrian Neumann, and Jens M Schmidt. Certifying 3-edge-connectivity. *Algorithmica*, 77(2):309–335, 2017.
25 Antoine Miné. The octagon abstract domain. *Higher-order and symbolic computation*, 19(1):31–100, 2006.
26 Edward F Moore. The shortest path through a maze. In *Proc. Int. Symp. Switching Theory, 1959*, pages 285–292, 1959.
27 Jens M Schmidt. Contractions, removals, and certifying 3-connectivity in linear time. *SIAM Journal on Computing*, 42(2):494–535, 2013.
28 Alexander Schrijver. *Theory of linear and integer programming*. John Wiley & Sons, 1998.
29 K Subramani and Piotr Wojciechowski. A combinatorial certifying algorithm for linear feasibility in utvpi constraints. *Algorithmica*, 78(1):166–208, 2017.
30 K Subramani and James Worthington. A new algorithm for linear and integer feasibility in horn constraints. In *International Conference on AI and OR Techniques in Constriant Programming for Combinatorial Optimization Problems*, pages 215–229. Springer, 2011.
31 J.D. Ullman and A. Van Gelder. Efficient test for top-down termination of logical rules. *Journal of the Association for Computing Machinery*, 35:345–373, 1988.
32 Hans Van Maaren and Chuangyin Dang. Simplicial pivoting algorithms for a tractable class of integer programs. *Journal of Combinatorial Optimization*, 6(2):133–142, 2002.