A strongly polynomial algorithm for linear programs with at most two non-zero entries per row or column

Daniel Dadush[1], Zhuan Khye Koh[1], Bento Natura[2], Neil Olver[3], and László A. Végh[3]

[1]Centrum Wiskunde & Informatica, Amsterdam, The Netherlands
[2]University of California, Berkeley, USA and Georgia Tech, USA
[3]London School of Economics and Political Science, UK

November 13, 2023

Abstract

We give a strongly polynomial algorithm for minimum cost generalized flow, and hence for optimizing any linear program with at most two non-zero entries per row, or at most two non-zero entries per column. Primal and dual feasibility were shown by Megiddo (SICOMP '83) and Végh (MOR '17) respectively. Our result can be viewed as progress towards understanding whether all linear programs can be solved in strongly polynomial time, also referred to as Smale's 9th problem.

Our approach is based on the recent primal-dual interior point method (IPM) due to Allamigeon, Dadush, Loho, Natura and Végh (FOCS '22). The number of iterations needed by the IPM is bounded, up to a polynomial factor in the number of inequalities, by the *straight line complexity* of the central path. Roughly speaking, this is the minimum number of pieces of any piecewise linear curve that multiplicatively approximates the central path.

As our main contribution, we show that the straight line complexity of any minimum cost generalized flow instance is polynomial in the number of arcs and vertices. By applying a reduction of Hochbaum (ORL '04), the same bound applies to any linear program with at most two non-zeros per column or per row.

To be able to run the IPM, one requires a suitable initial point. For this purpose, we develop a novel multistage approach, where each stage can be solved in strongly polynomial time given the result of the previous stage. Beyond this, substantial work is needed to ensure that the bit complexity of each iterate remains bounded during the execution of the algorithm. For this purpose, we show that one can maintain a representation of the iterates as a low complexity convex combination of vertices. Our approach is black-box and can be applied to any log-barrier path following method.

Contents

1 Introduction

We consider linear programming (LP) in the following primal-dual form:

$$\begin{aligned} \min\ & \langle c, x\rangle \\ \mathbf{A}x &= b \\ x &\geq \mathbf{0}_n\,, \end{aligned} \qquad\qquad \begin{aligned} \max\ & \langle b, y\rangle \\ \mathbf{A}^\top y + s &= c \\ s &\geq \mathbf{0}_n\,, \end{aligned} \tag{LP}$$

where $\mathbf{A} \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$, and $\mathrm{rk}(\mathbf{A}) = m$. Our focus is on LP algorithms that find exact primal and dual optimal solutions, or conclude infeasibility or unboundedness. We say that the dual progam is a *two variable per inequality (2VPI) linear program* if every row of $\mathbf{A}^\top$ includes at most two nonzero entries. In such a case, we refer to the pair of LPs as a *2VPI primal-dual pair*.

The first polynomial-time algorithms were the ellipsoid method by Khachiyan in 1979 [Kha79] and interior point methods, introduced by Karmarkar in 1984 [Kar84]. However, it remains an outstanding open question to find a *strongly polynomial algorithm* for linear programming. The question was listed by the Fields medalist Smale as one of the most prominent mathematical challenges for the 21st century [Sma98]. In such an algorithm, only $\mathrm{poly}(n)$ basic arithmetic operations and comparisons are allowed, and the algorithm uses polynomial space.

The notion of strongly polynomial algorithms was first formally introduced by Megiddo [Meg83], under the term 'genuinely polynomial'. The same paper gave an algorithm for two variable per inequality feasibility systems, that is, for the dual feasibility problem in (LP) when all rows of $\mathbf{A}^\top$ have at most two nonzero entries. The corresponding primal feasibility problem can be reduced to the *maximum generalized flow problem*. For this, the first strongly polynomial algorithm was given by Végh [Vég17], followed by a faster and simpler algorithm by Olver and Végh [OV20]. The *minimum-cost generalized flow problem* is the dual of a 2VPI LP, where the two nonzero entries in each column of $\mathbf{A}$ are a -1 entry and a positive entry. As discussed below, this naturally corresponds to a network flow model with multipliers on the arcs. As shown in [Hoc04], all 2VPI LPs are reducible to a dual of a minimum-cost generalized flow problem. The existence of a strongly polynomial algorithm for this problem has been a longstanding open question, mentioned e.g. in [AC91, CM94a, CM94b, GPT91, HN94, NPT92, Way02, Vég17, OV20]. Our main result resolves this question.

Theorem 1.1. *There is a strongly polynomial algorithm for the minimum-cost generalized flow problem, and for two variable per inequality primal-dual pairs.*

1.1 Background and previous work

Strongly polynomial algorithms for well-conditioned LPs. In a seminal, Fulkerson-prize winning paper [Tar85], Tardos obtained the first strongly polynomial algorithm for minimum-cost circulations. A particularly important technique in this paper was *variable fixing*: by solving an approximate version of the LP with rounded costs, one can deduce that a certain variable is at the lower or upper capacity bound in an optimal solution.

Towards general LP, Tardos [Tar86] extended this approach to obtain a strongly polynomial algorithm for 'combinatorial LPs'. More precisely, for (LP) with an integer constraint matrix $\mathbf{A} \in \mathbb{Z}^{m\times n}$, this algorithm runs in $\mathrm{poly}(n, \log \Delta_{\mathbf{A}})$ iterations, where $\Delta_{\mathbf{A}}$ is the maximum subdeterminant of $\mathbf{A}$. The running time is independent of b and c. In particular, this bound is strongly polynomial if all entries of $\mathbf{A}$ are at most $\mathrm{poly}(n)$, such as for multicommodity flows and other combinatorial problems. Using an interior point approach discussed below, Vavasis and Ye [VY96] obtained an algorithm with $\mathrm{poly}(n, \log \bar{\chi}_{\mathbf{A}})$ arithmetic operations, where $\bar{\chi}_{\mathbf{A}}$ is the Dikin–Stewart–Todd condition number of the matrix $\mathbf{A}$. For integer matrices, $\bar{\chi}_{\mathbf{A}} = O(n\Delta_{\mathbf{A}})$, thus, this strengthens Tardos's result. A similar dependence, using a black-box approach extending Tardos's work [Tar86] was obtained by Dadush, Natura, and Végh [DNV20]. Further, Dadush, Huiberts, Natura, and Végh [DHNV23] strengthened this dependence to $\mathrm{poly}(n, \log \bar{\chi}^*_{\mathbf{A}})$, where $\bar{\chi}^*_{\mathbf{A}}$ is the optimized value of $\bar{\chi}_{\mathbf{A}}$ under column rescalings.

Prior results on 2VPI and generalized flows. 2VPI LPs are a natural class of LP that does not fall into the above 'well-conditioned' classes: even $\bar{\chi}^*_{\mathbf{A}}$ may be unbounded for the constraint matrix. At the same time, they form an interesting intermediate class, as it is easy to see that solving an arbitrary LP is reducible to solving one with at most three nonzero entries per row in $\mathbf{A}^\top$.

For finding a feasible solution to a 2VPI system, Megiddo's [Meg83] approach relied on parametric search. A faster parametric search algorithm was given by Cohen and Megiddo [CM94b]. Hochbaum and

Naor [HN94] used an efficient Fourier–Motzkin elimination to obtain what is still the fastest deterministic approach. Dadush, Koh, Natura, and Végh [DKNV22] used a variant of the discrete Newton method. Recently, Karczmarz [Kar22] gave an improved randomized strongly polynomial algorithm, also using parametric search.

Consider now *monotone* 2VPI (M2VPI) systems, where each inequality has at most one positive and at most one negative entry. If such an LP is bounded, then there exists unique pointwise minimal and a unique pointwise maximal solution. Already the algorithm in [Meg83] can be used to find these solutions. As noted by Adler and Cosares [AC91], an M2VPI linear program is strongly polynomially solvable if $b \geq \mathbf{0}$ or $b \leq \mathbf{0}$. Norton, Plotkin, and Tardos [NPT92] gave a strongly polynomial algorithm for a constant number of nonzero demands.

The generalized flow problem is (after normalization) the dual of the M2VPI problem. In this problem, we are given a directed graph $G = (V, E)$ with node demands b_i, $i \in V$ and arc costs c_e and gain factors $\gamma_e > 0$ for $e \in E$. While traversing the arc $e = (i, j)$, the flow value x_e gets multiplied to $\gamma_e x_e$. In the *minimum-cost generalized flow problem*, we need to exactly satisfy all node demands at a minimum cost. The *maximum generalized flow problem* is the special case when the objective is to maximize the net flow reaching a special sink node t.

This is a fundamental network optimization model that traces back to Kantorovich's 1939 paper [Kan39] introducing linear programming. Generalized flow networks can be used to model transportation of a commodity through a network with leakage, or conversions between various equities in financial networks, as well as generalized assignment problems. We refer the reader to [AMO93, Chapter 15] for further applications.

Goldberg, Plotkin, and Tardos [GPT91] gave the first weakly polynomial combinatorial algorithm for the maximum generalized flow problem. This was followed by a significant number of further such algorithms, such as [CM94a, GJO97, Rad04, RW09, TW98, Way02], see further references in [OV20]. In particular, [CM94a] gave a strongly polynomial approximation scheme, i.e., a strongly polynomial algorithm that achieves a fixed fraction of the optimum flow value in a capacitated generalized flow network. The strongly polynomial algorithms by Végh [Vég17] and Olver and Végh [OV20] rely on the variable fixing technique, however, in a new, 'continuous' scaling framework. While the original LP can be ill-conditioned, variable fixing is still possible, since the dual solutions can be used to 'relabel' the flow to make it 'locally' amenable to classical network flow arguments.

However, relabelling heavily relies on the special cost function of the flow maximization problem, and does not seem to be extendable to the minimum-cost version. For solving the minimum-cost generalized flow problem, the only known (weakly polynomial) combinatorial approach is the ratio-circuit cancelling algorithm by Wayne [Way02]. The fastest previous weakly polynomial algorithms can be obtained using interior point methods; an early such example is by Vaidya [Vai89]. Daitch and Spielman [DS08], and Lee and Sidford [LS14] gave fast algorithms for obtaining an additive ε-approximation; however, such approximation cannot be used to obtain exact optimal solutions. We also note that the latter results only apply for lossy flows, i.e., with gain factors $\gamma_e \leq 1$.

Interior Point Methods and their limitations Interior point methods (IPMs) give the fastest current weakly polynomial algorithms for general LP, see [CLS19, vdB20, vdBLSS20, JSWZ21] as well as for special classes such as minimum-cost circulations [CKL+22, BCK+23] and multicommodity flows [vdBZ23]. They are also potent approach in the context of strongly polynomial computability, and form the basis of our result.

The algorithms discussed next fall into the class of primal-dual path following algorithms. A key concept here is the *central path*, the algebraic curve formed by minimizers of $\langle c, x\rangle - \mu \sum_{i=1}^{n} \log(x_i)$ for $\mu > 0$. As $\mu \to 0$, the limit of the central path is an optimal solution. Path following methods maintain iterates in a certain neighborhood of the central path while geometrically decreasing μ, and thus, the optimality gap. The logarithmic barrier function above can be replaced by more general barrier functions. The affine scaling step is a standard way to find a movement direction. This can be interpreted as a least square computation in the local norm induced by the Hessian of the logarithmic barrier function.

Layered least squares (LLS) IPMs were introduced in the influential work of Vavasis and Ye [VY96]. The LLS step in the Vavasis–Ye algorithm decomposes the variables into different layers based on the values of the current iterate. The step direction is determined as a sequence of least squares computations that prioritizes decreasing variables at lower layers. Roughly speaking, such steps enable to traverse arbitrarily long but relatively straight segments of the central path in a single iteration. Combinatorial progress is measured by *crossover events*, where two variables get reordered consistently with their order in the limit optimal solution. This is very different from the variable fixing technique prevalent in the combinatorial approaches discussed above. In particular, while we can infer the occurrence of a new crossover event within a certain number of iterations, the argument only shows existence and we cannot identify the participating variables. The condition number $\bar{\chi}_{\mathbf{A}}$ appearing in the running time is a bound on the norms of oblique projections.

This led to a line of research on improved combinatorial IPMs [MMT98, MT03, MT05, LMT09]. The paper [DHNV23] revealed that $\bar{\chi}_{\mathbf{A}}$ is closely related to the *circuit imbalance measure* $\kappa_{\mathbf{A}}$ that bounds the maximum ratio of two nonzero entries of an elementary vector in the kernel of $\mathbf{A}$. Moreover, they obtained an LLS algorithm invariant under column rescaling, thus improving the dependence to the best $\kappa^*_{\mathbf{A}}$ value achievable under column rescalings.

The above results may raise hopes to finding a strongly polynomial IPM. However, the papers by Allamigeon, Benchimol, Gaubert, and Joswig [ABGJ18], Allamigeon, Gaubert, and Vandame [AGV22], and Zong, Lee, and Yue [ZLY23] yield a surprising negative answer. By analyzing the tropical limits of linear programs, these papers exhibit parametric families of LPs such that for suitably large parameter values, no path-following method can be strongly polynomial. This was first shown for the standard logarithmic barrier [ABGJ18], and later for arbitrary self-concordant barriers [AGV22].

1.2 The Subspace Layered Least Squares Interior Point Method and straight line complexity

The primal central path has a natural dual counterpart. The primal-dual central path point $(x^{\mathrm{cp}}(\mu), s^{\mathrm{cp}}(\mu))$ is the unique pair of primal and dual feasible solutions to (LP) such that $x^{\mathrm{cp}}_i(\mu)s^{\mathrm{cp}}_i(\mu) = \mu$ for all $1 \le i \le n$. Thus, the duality gap between $x^{\mathrm{cp}}(\mu)$ and $s^{\mathrm{cp}}(\mu)$ is $n\mu$.

The lower bounds in [ABGJ18, AGV22] are ultimately based on the following insight. The trajectory of any path following argument is a piecewise linear curve in the neighborhood of the central path; the number of pieces correspond to the number of iterations. Thus, a lower bound on the number of *any* piecewise linear curve in the neighborhood provides a lower bound on the number of iterations. For the examples in these papers, exponential lower bounds are shown.

The recent result by Allamigeon, Dadush, Loho, Natura, and Végh [ADL+23] complements these negative results by a positive algorithmic bound. Namely, they provide an IPM whose number of iterates matches such a lower bound within a strongly polynomial factor. Let us elaborate on the lower bound.

Assume (LP) is feasible and bounded with optimum value $v^\star$. Given $g \ge 0$, we denote by

$$\begin{aligned} \mathcal{P}_g &:= \{x \in \mathbb{R}^n : \mathbf{A}x = b\,,\ x \geq \mathbf{0}\,,\ \langle c, x\rangle \leq v^\star + g\}\,, \\ \mathcal{D}_g &:= \{s \in \mathbb{R}^n : \exists y \in \mathbb{R}^m\ \mathbf{A}^\top y + s = c\,,\ s \geq \mathbf{0}\,,\ \langle b, y\rangle \geq v^\star - g\} \end{aligned} \tag{1}$$

the feasible sublevel sets. They correspond to the sets of the primal and dual feasible points (x, s) with objective value within g from the optimum $v^\star$, respectively.

Assuming that (LP) has strictly feasible primal and dual solutions, the *max-central path* is defined as the parametric curve $g \mapsto z^{\mathrm{m}}(g) := (x^{\mathrm{m}}(g), s^{\mathrm{m}}(g)) \in \mathbb{R}^{2n}_+$, where

$$x^{\mathrm{m}}_i(g) := \max\{x_i : x \in \mathcal{P}_g\} \quad \text{and} \quad s^{\mathrm{m}}_i(g) := \max\{s_i : s \in \mathcal{D}_g\}\,, \qquad \forall i \in [n]\,. \tag{2}$$

The max-central path can be seen as a combinatorial proxy to the central path. In particular, for $g = n\mu$, $x^{\mathrm{cp}}(\mu) \in \mathcal{P}_g$ and $s^{\mathrm{cp}}(\mu) \in \mathcal{D}_g$, and it is easy to see that $x^{\mathrm{m}}(g)/n \le x^{\mathrm{cp}}(\mu) \le x^{\mathrm{m}}(g)$ and $s^{\mathrm{m}}(g)/n \le s^{\mathrm{cp}}(\mu) \le s^{\mathrm{m}}(g)$.

For each $1 \le i \le n$, the function $g \mapsto x^{\mathrm{m}}_i(g)$ is a piecewise linear concave function. It corresponds to trajectory of the shadow simplex algorithm that interpolates between the objective functions $\langle c, x\rangle$ and $-x_i$. The breakpoints correspond to basic feasible solutions, and therefore the number of linear pieces is at most the number of vertices, that is, at most 2^n. We will use the following definition.[1]

Definition 1.2. Let $f\colon \mathbb{R}_+ \to \mathbb{R}_+$ be a function and $\eta \in (0,1)$. The *straight line complexity* of f with respect to η, denoted $\mathrm{SLC}_\eta(f)$, is the infimum number of pieces of a continuous piecewise linear function $h\colon \mathbb{R}_+ \to \mathbb{R}_+$ where $\eta f \le h \le f$.

The number of iterations in the *Subspace Layered Least Squares (SLLS) IPM* in [ADL+23] can be bounded by the total straight line complexity of the (primal) max-central path curves.[2]

[1] We remark that our convention on the parameter η differs from [ADL+23], which uses $(1-\theta)f \le h \le f$ instead.
[2] The condition on the starting point asserts that it is near the central path; $x \circ s$ denotes the Hadamard product.

Theorem 1.3 ([ADL+23]). *There is an interior-point method that given an instance of* (LP) *and strictly feasible solutions* $x, s > \mathbf{0}$ *such that* $\|\frac{nx \circ s}{\langle x,s\rangle} - \mathbf{1}_n\| \le \beta$ *for a fixed* $0 < \beta < 1/6$*, finds a pair of primal and dual optimal solutions in*

$$O\left(\min_{\eta\in(0,1]} \sqrt{n}\log\left(\frac{n}{\eta}\right)\sum_{i=1}^{n}\mathrm{SLC}_\eta(x_i^{\mathrm{m}})\right) \tag{3}$$

many iterations.

The SLLS IPM requires at most a $\tilde{O}(n^{1.5})$ factor more iterations than *any* path-following IPM for any self-concordant barrier function. This is because it can be shown that each $\mathrm{SLC}_\eta(x_i^{\mathrm{m}})$ gives a lower bound on the number of piecewise linear segments traversing a corresponding wide neighborhood. Moreover, as noted above, $\mathrm{SLC}_1(x_i^{\mathrm{m}}) \le 2^n$, thus, the number of iterations is always at most singly exponential.

We note that the theorem could be equivalently written in terms of the dual straight line complexities. The equivalence of these can be shown using arguments from [ADL+23]. Further, we note that the neighborhood parameter η is not important for the overall bound. It is not difficult to show that for $0 < \eta < \eta' < 1$, $\mathrm{SLC}_{\eta'}(f) = O(\log(1/\eta)/\log(1/\eta'))\,\mathrm{SLC}_\eta(f)$; we do not include the proof.

According to Theorem 1.3, the number of iterations of SLLS boils down to estimating straight-line complexities of the variables. Note that this is a purely combinatorial question about understanding the structure of univariate piecewise linear functions $x_i^{\mathrm{m}}(g)$.

We also note that the SLLS algorithm does not have access to approximations of the max-central path curves attaining the minimum. Rather, the algorithm exploits the special 'polarized' structure of a piece of the central path with a straight line in the neighborhood.

1.3 Computational models

There are multiple related, yet distinct notions of a *strongly polynomial* computational model. Smale's question was posed in the *Blum–Shub–Smale (BSS) real model of computation* [BSS89]. In this model, the input can be given by arbitrary real numbers, and one step may compute a rational polynomial function of the previously computed quantities with real coefficients, or make comparisons between two quantities. In the more restrictive *real RAM* model, one can perform a sequence of elementary arithmetic operations $(+, -, \times, /)$ and comparisons $(\ge)$ on real numbers. In this paper, we say that an algorithm is *polynomial in the real RAM model* if the number of elementary arithmetic operations and comparisons is bounded polynomially in the dimension of the input; in the case of LP, this is $K = n \times m + n + m$.

We now turn to the Turing model. Consider a problem where the input is given by K integers; for LP, the input $(\mathbf{A}, b, c)$ is described by $K = 2(n \times m + n + m)$ integers representing the rational entires. An algorithm is *strongly polynomial in the Turing model* (see [GLS88]), if it only performs $\mathrm{poly}(K)$ (in the LP case, this means $\mathrm{poly}(m, n)$) elementary arithmetic operations and comparisons as in the real model. Additionally, the bit-complexity of the numbers during the computations must remain polynomially bounded in the encoding length of the input. Equivalently, the algorithm must be PSPACE. The model has an ambiguity regarding how divisions can be implemented, see discussion of variants in [GLS88, Section 1.3]. The results of this paper work with the most restricted setting: we maintain rational representations (p, q) of all numbers during the computation, and division $\frac{p}{q}/\frac{p'}{q'}$ corresponds to computing the representation $(pq', p'q)$.

While a strongly polynomial algorithm in the Turing model implies a polynomial algorithm in the real RAM model, the converse is not necessarily the case: enforcing PSPACE may be challenging. For example, Gaussian elimination needs to be done carefully to keep the sizes of numbers under control, see [Edm67] and [GLS88, Section 1.4]. The LLS interior point methods are (strongly) polynomial in the real RAM model whenever $\log(\bar{\chi}_{\mathbf{A}}) = \mathrm{poly}(n)$, yet we are not aware of any strongly polynomial implementation of such algorithms in the Turing model prior to this work. The situation is even worse regarding the SLLS IPM algorithm in [ADL+23]. The special SLLS step requires an (approximate) singular value decomposition, and the algorithm used in the paper also relies on squareroot computations. Hence, it is only strongly polynomial in the extended real model $(+, -, \times, /, \sqrt{\cdot})$

In the weakly polynomial model, i.e., when running time dependence on the total encoding length is allowed, the bit complexity of the algorithms can be controlled by approximately solving linear systems and roundings. In recent work, Ghadiri, Peng, and Vempala [GPV23] developed general tools that enable to keep the bit complexity of recent fast IPMs under control. However, these techniques are not applicable in the strongly polynomial model. In particular, it requires estimates on parameters such as the total bit length of the input or the condition number of the matrix. They also require rounding to a fixed number of bits depending on such numerical parameters; in the most stringent definition of strongly polynomial, this cannot be done.

We note that the algorithms presented in this paper are fully deterministic.

1.4 Our contributions

We prove Theorem 1.1, i.e., give a strongly polynomial algorithm for the minimum-cost generalized flow problem by showing that the total number of iterations of the SLLS IPM by [ADL+23] is strongly polynomially bounded, and that the SLLS IPM can be implemented in strongly polynomial time in the Turing model. Our result has three main ingredients:

(1) *Straight line complexity bound:* We establish in Theorem 4.1 a strongly polynomial bound $\mathrm{SLC}_\eta(x_e^{\mathrm{m}}) = O(nm)$ on the straight line complexities of the variables in the minimum-cost generalized flow problem, with $\eta = \Omega(1/(m^2 n))$, where m is the number of arcs and n is the number of nodes of the graph. This bound applies for the uncapacitated version of the problem as described above; if in addition arcs have capacities, the bound becomes $O(m^2)$.

(2) *Initialization:* IPMs require a strictly feasible and well centered starting point (x^0, s^0), even though a strictly feasible (or even a feasible) solution may not exist. We present a careful initialization scheme that solves Linear Programs in three stages, and preserves the straight line complexity bounds.

(3) *Implementation in the Turing model:* We show that the bit-length of the computations can be controlled in a model using only basic arithmetic operations and comparisons. Further, we show that the square-root computations in [ADL+23] can be avoided.

The straight line complexity is established via a combinatorial argument using structural properties of generalized flows. In contrast, the initialization and implementation tasks are applicable for general LP, and can be seen as a direct strengthening of the result in [ADL+23]. We now elaborate on each of these parts, and highlight the main technical ideas.

1.4.1 Straight-line complexity bound for generalized flows

Theorem 1.3 enables to bound the number of iterations in SLLS by bounding the straight-line complexities $\mathrm{SLC}_\eta(x_i^{\mathrm{m}})$ for a suitable $\eta > 0$. In the first step, we reduce this to an even more concrete combinatorial question of *circuit covers* as explained next.

Circuit covers. For the purposes of analyzing straight-line complexities, we can assume that a pair of primal and dual optimal solutions $(\bar{x}, \bar{s})$ to (LP) is provided.

For any vector $h \in \ker(\mathbf{A})$, with $\langle c, h\rangle \ge 0$ we can define the function $\bar{x}^h(g) : \mathbb{R}_+ \to \mathbb{R}_+^n$ by moving from $\bar{x}$ in the direction of h; this is called the *h-curve from $\bar{x}$*. Namely, we define $\bar{x}^h(g) = \bar{x} + \alpha(g)h$, where $\alpha(g)$ is chosen maximally so that $\bar{x}^h(g)$ is feasible, and has cost at most g larger than the cost of $\bar{x}$. For every $i \in [n]$, the i-th coordinate $\bar{x}_i^h(g)$ can easily be seen to be a piecewise linear concave function with two pieces, the first with slope $h_i/\langle c, h\rangle$ and the second constant.

Given $h, h' \in \ker(\mathbf{A})$, $\langle c, h\rangle, \langle c, h'\rangle \ge 0$, $\alpha > 0$ and $i \in [n]$, we say that *h α-dominates h' on $i \in [n]$* if $\bar{x}_i^h(g) \ge \alpha \bar{x}_i^{h'}(g)$ for all $g \ge 0$.

In the linear space $\ker(\mathbf{A})$, an *elementary vector* is a support minimal nonzero vector, and the support of an elementary vector is called a *circuit*. Note that the latter coincides with the notion of circuits of the linear matroid of $\mathbf{A}$; each circuit corresponds to a one-dimensional subspace of elementary vectors. We let $\mathcal{E}(\mathbf{A})$ denote the set of all elementary vectors.

Given an optimal solution $\bar{x}$, let us consider the augmentations from $\bar{x}$ by an elementary vector h. Noting that $\bar{x}^h(g)$ is invariant under rescaling h to αh for $\alpha > 0$, this gives one function per circuit. For any coordinate $i \in [n]$, let us now consider the pointwise maximum at the i-th coordinate $\hat{x}_i(g) = \max\{\bar{x}_i^h(g) : h \in \mathcal{E}(\mathbf{A})\}$. This is a piecewise linear function, but is not concave. Note also that the number of pieces can be exponential. Nevertheless, using standard circuit decomposition techniques, it is not difficult to show that $\hat{x}_i(g)$ approximates $x_i^{\mathrm{m}}(g)$ up to a factor n: $x_i^{\mathrm{m}}(g)/n \le \hat{x}_i(g) \le x_i^{\mathrm{m}}(g)$.

Our strategy to obtain SLC bounds is by constructing *circuit covers*. Given a primal optimal solution $\bar{x}$, an index $i \in [n]$ and $\alpha > 0$, we say that a set of vectors $S \subseteq \ker(\mathbf{A})$ is an *α-circuit cover of i with respect to $\bar{x}$* if for every $h' \in \mathcal{E}(\mathbf{A})$, there is a $h \in S$ that α-dominates h' on i. Then, the piecewise-linear function

$\bar{x}_i^S(g) = \max\{\bar{x}_i^h(g) : h \in S\}$ satisfies $\alpha x_i^{\mathrm{m}}(g)/n \le \bar{x}_i^S(g) \le x_i^{\mathrm{m}}(g)$. The upper convex envelope of this function has at most $|S|+1$ linear pieces, and consequently, $\mathrm{SLC}_{\alpha/n}(x_i^{\mathrm{m}}) \le |S|+1$.

As a simple demonstration of this technique, in Section 3 we give a simple straight-line complexity bound in terms of $\kappa_{\mathbf{A}}$, the *circuit imbalance* of $\mathbf{A}$. This is defined as the maximum absolute value of the ratio of two nonzero entries of an elementary vector in $\mathcal{E}(\mathbf{A})$. In particular, it is known that $\kappa_{\mathbf{A}} = 1$ if and only if there exists a totally unimodular matrix $\mathbf{B}$ such that $\ker(\mathbf{A}) = \ker(\mathbf{B})$. The LLS IPM algorithms such as [VY96] and [DHNV23] run in time $\mathrm{poly}(n, \log \kappa_{\mathbf{A}})$. We refer the reader to the survey [ENV22] for background and more applications of circuit imbalances.

Whereas the results in [ADL+23] together with [VY96] already imply a $\mathrm{poly}(n, \log \kappa_{\mathbf{A}})$ bound on the straight line complexity, we give simple direct bounds using the circuit cover approach. For each circuit, we identify a *'combinatorial signature'* comprising two critical indices. We select an 'undominated' family of signatures and pick a single circuit for each signature. These circuits will then $1/(n\kappa_{\mathbf{A}}^2)$-dominate all other circuits. In fact, using that the straight line complexities are invariant under column scaling (Lemma 2.2), we can replace the $\kappa_{\mathbf{A}}$ dependence by $\kappa_{\mathbf{A}}^\star$ dependence, the optimal circuit imbalance achievable under column scaling.

Circuit covers for generalized flows. We primarily work with minimum-cost generalized flow in its capacitated form, where arcs may have capacities, and all demands are zero; this reduction yields only n capacitated arcs when starting from the demand form. The circuits in this version of the generalized flow problem correspond to simple combinatorial structures: namely, a circuit either corresponds to a *conservative* directed cycle, where the product of the gain factors is one, or a *'bicycle'*, namely, a flow generating cycle connected by a path to a flow absorbing cycle. The latter are cycles where the product of the gain factors is greater and less than one, respectively. These structures played a fundamental role in all prior works on generalized flows, see e.g., [GPT91, Way02, Vég17, OV20], as well as for 2VPI algorithms, e.g., [Meg83, CM94b, DKNV22, Kar22].

We construct a circuit cover of size $O(mn)$ to bound the straight line complexity of the variables in the generalized flow problem. Similarly to the $\kappa_{\mathbf{A}}$ bound discussed above, we associate a combinatorial signature with each circuit that forms the basis of the cover. However, we need a much more careful construction to obtain this cover. In particular, the cover will not consist of circuits but more complicated objects.

The basis of our construction is *path domination.* Let us fix two nodes s and t in the graph. We demonstrate a small collection of s-t walks that "dominate" the collection of all s-t paths in a certain sense. The general cover will be constructed by combining such walks and extending this argument. We now highlight the main ideas of path domination.

Consider an s–t walk W; this induces an s–t flow $\bar{x}^W$ on the walk. We define the function $\vec{f}_W : \mathbb{R}_+^2 \to \mathbb{R}_+$ such that $\vec{f}_W(\lambda, r)$ denotes the maximum amount of flow that can be sent from s to t if there are r units available at s, each step of the walk satisfies the capacity bound, and the cost incurred in any step of the walk is at most λ. This corresponds to a certain maximal scaling of $\bar{x}^W$. (This scaling may have total cost larger than λ, and moreover since an arc of the graph can be used multiple times, it may also violate arc capacities; but as long as the walk is *n-recurrent*, meaning it uses each edge at most n times, scaling down by nm will yield such a feasible and cheap flow.) There are two possible bottleneck arcs that prevent a larger scaling of $\bar{x}^W$ from being used: a *cost bottleneck arc* e_{c} where $c_{e_{\mathrm{c}}}\bar{x}_{e_{\mathrm{c}}}^W$ is maximal, and a *flow bottleneck arc* e_{f} where $\bar{x}_{e_{\mathrm{f}}}^W/u_{e_{\mathrm{f}}}$ is maximal, where c_e and u_e denote the cost and capacity of arc e, respectively. We associate the combinatorial signature $(e_{\mathrm{c}}, e_{\mathrm{f}}, \preceq)$ or $(e_{\mathrm{c}}, e_{\mathrm{f}}, \succ)$ with W, where $\preceq$ means that e_{c} precedes or equals e_{f} on the walk W, and $\succ$ means that e_{f} precedes e_{c}.

We say that the s-t walk W' dominates W, if $\vec{f}_{W'} \ge \vec{f}_W$. Our goal in this context is to show that there is a small family of s-t walks that dominates all s-t paths. The key insight is the following path monotonicity property: consider a path P with signature $(e_{\mathrm{c}}, e_{\mathrm{f}}, \preceq)$, and take the subpath Q of P starting with e_{c} and ending in e_{f}. Let Q' be the highest gain path (i.e., one that maximizes the product of gain factors) starting with e_{c} and ending in e_{f} whose cost bottleneck is e_{c} and flow bottleneck is e_{f}. After replacing Q by Q' in P, the obtained walk P' will dominate P. Moreover, if the signature changes, then the flow bottleneck of P' remains $e'_{\mathrm{f}} = e_{\mathrm{f}}$, and the cost bottleneck e'_{c} will be on the part of P' after e_{f}. An analogous replacement can be made for the signature type $(e_{\mathrm{c}}, e_{\mathrm{f}}, \succ)$. Starting from any path P, using a sequence of such operations, we can reach a walk W whose subpath between the bottleneck arcs is already highest gain in the above sense.

Denoting the number of finite capacity arcs by $\bar{m}$, this argument enables to show the existence of a $O(m\bar{m})$ sized family of n-recurrent s–t walks that dominate all s–t paths. The family is constructed by fixing the signature, and from each signature, selecting the best walk for the three segments defined by s, t, and the two bottleneck arcs, such that each segment is highest gain subject to recurrence bounds and not having other bottlenecks.

Once we have this path domination result, we can use this to demonstrate domination of more complicated collections of objects with small dominating sets, and eventually all circuits. For instance: it easily follows that there is a small collection of s-s walks with the property that for any flow-generating cycle C containing s, there is a walk R in this collection that dominates C, in the sense that for every choice of cost bound $\lambda \in \mathbb{R}_+$, at least as much excess can be created using R than C. From this, by "composing" dominating sets for cycles and paths, we obtain a small dominating set for the collection of "simple flow-generating objects at t" consisting of a flow-generating cycles along with a path from this cycle to t.

1.4.2 Initialization

A strongly polynomial straight line complexity bound implies a strongly polynomial iteration bound of the SLLS IPM method; however, it requires an initial point $(x^0, s^0) \in \mathcal{P}_{++} \times \mathcal{D}_{++}$ near the central path. Such a point may not even exist; in fact, the primal or dual programs in (LP) may be infeasible. Whereas one could use the combinatorial algorithms to decide primal [OV20] and dual [HN94] feasibility, these algorithms do not directly yield strictly feasible solutions (which may again not exist).

The situation is analogous to Simplex, where Stage I can be used to find a feasible solution by solving an auxiliary LP. Various initialization methods have been developed for IPMs, but none of these is directly applicable for our purposes: only solving auxiliary systems with small straight line complexity, while remaining in the strongly polynomial model.

A common initialization technique is the self-dual homogenous formulation [YTM94]. However, writing the self-dual formulation of a generalized flow LP results in a more complicated problem and it is not clear if the straight line complexity admits a similar bound. (Note also that in the simpler case of (standard) network flows, the constraint matrix is totally unimodular, while the combined matrix does not have this property.)

We present two initialization methods. Our first approach in Section 5 uses a 'big-M' method, as in [VY96]. Let us create a negative copy of each variable with a large penalty cost. That is, one can replace the primal system using variables $(x, x', x'') \in \mathbb{R}^{3n}$ in the form

$$\min \langle c, x\rangle + M \langle \mathbf{1}_n, x'\rangle \quad \text{s.t.} \quad \mathbf{A}x - \mathbf{A}x' = b\,, \quad x + x'' = 2M\mathbf{1}_n\,, \quad x, x', x'' \geq \mathbf{0}\,. \tag{4}$$

Here, x' represents a negative copy of each variable and x'' corresponds to a slack variable for the box constraint $0 \leq x \leq 2M\mathbf{1}_n$. Such a system, along with its dual, is easy to initialize for sufficiently large M. Moreover, the constraint matrix remains 'nice', e.g., it can be still interpreted as a (capacitated) generalized flow problem, where the x' variables correspond to expensive reverse arcs. As long as there exists a pair of primal and dual optimal solutions $(x^\star, s^\star)$ to (LP) with $\|x^\star\|_\infty, \|s^\star\|_\infty < M\mathbf{1}_n$, then these will also be optimal solutions to the extended formulation.

However, finding a suitable large M becomes challenging. In [VY96], such a bound is derived based on $\bar{\chi}_{\mathbf{A}}$. This is hard to compute in general; one could use a repeated guessing of this condition number, but this would lead to a $\log\log \bar{\chi}_{\mathbf{A}}$ running time dependence. Bounds on the norms of optimal solutions are routinely derived using bit-complexity arguments, see e.g. [GLS88]; however, this is also not possible in the strongly polynomial model.

In Section 5, we use the deterministic combinatorial algorithms of e.g., [OV20] and [HN94] to solve up to n primal and dual feasibility problems to first obtain maximum support primal and dual solutions available. We then reduce the problem to a system with a pair of strictly positive primal and dual solutions. The reduction is achieved by deleting some variables and projecting out some others. In the generalized flow problem, these amount to graphical operations of deletions and contractions, and thus preserve the generalized flow structure. Given the strictly positive primal and dual solutions $(\hat{x}, \hat{s})$, choosing M larger than $\langle \hat{x}, \hat{s}\rangle$ divided by the smallest entry of $(\hat{x}, \hat{s})$ guarantees that $\|x^\star\|_\infty, \|s^\star\|_\infty < M\mathbf{1}_n$ for any pair of primal and dual optimal solutions.

Whereas the above approach can implement the big-M method, it is only applicable to the particular minimum-cost generalized flow setting as it requires feasibility solvers. Also, it needs to solve $2n$ systems as preprocessing. In Section 7, we develop a more principled, multistage initialization strategy that is applicable to general LP, preserves straight line complexity, and only requires solving four IPM problems. Since we will need to solve different LPs derived from (LP), one needs to clarify what 'preserving straight line complexity' means. We define $\mathrm{SLC}_\eta(\mathbf{A})$ as the maximum value of $\mathrm{SLC}(x_i^{\mathrm{m}})$ for any variable $i \in [n]$ in any LP of the form (LP) with constraint matrix $\mathbf{A}$, but taking any possible right hand side $b \in \mathbb{R}^m$ and cost $c \in \mathbb{R}^n$. All our auxiliary LPs will have SLC bounded by $\mathrm{SLC}_\eta(\mathbf{B})$, where $\mathbf{B} = \begin{pmatrix} \mathbf{A} & -\mathbf{A} & \mathbf{0}_{m\times n} \\ \mathbf{I}_n & \mathbf{0}_{n\times n} & \mathbf{I}_n \end{pmatrix}$ is the matrix also used in the big-M formulation (4).

Our strategy can be interpreted as a facial reduction strategy, where we carefully fix or project out variables that yield an equivalent LP to the original one, and where strictly feasible primal and dual solutions in fact exist. Throughout the process, the solutions from a previous stage provide a starting point near the central path.

1.4.3 Implementation in the Turing model

As discussed in Section 1.3, to obtain a strongly polynomial algorithm in the Turing model, one needs to avoid the square-root computations in [ADL+23], and also devise a new rounding approach, as the previous ones rely on bit-complexity information and rounding that are not implementable in the strongly polynomial model.

In Appendix B, we present a version of the SLLS IPM only using basic arithmetic operations and comparisons, and where every iteration can be carried out in strongly polynomial time. Theorem 6.10 presents the version of Theorem 1.3 proved here. For the sake of strong polynomiality and simplicity, the IPM is slower by a factor of n in terms of iteration count when compared to Theorem 1.3. This additional factor is immaterial for our general results, as we do not aim to optimize the running times. At a technical level, the slowdown in the above IPM is entirely due to the use of a strongly polynomial $2^{O(n)}$-approximate singular value decomposition (SVD) that we present in Appendix B.7. This SVD algorithm is very simple: it corresponds to a single Gram–Schmidt orthogonalization applied to a suitable permutation of the columns of the input matrix. By using more sophisticated strongly polynomial SVD algorithms, such as the randomized SVD by Diakonikolas, Tzamos and Kane [DTK23] or a recent deterministic "bicriteria" SVD of Dadush and Ramachandran [DR24], this additional factor of n can be removed. These algorithms are relatively complicated , and thus, for the sake of being self contained and easily verifiable, we prefer to rely on a weaker SVD subroutine in the present paper.

The second main ingredient is a general strongly polynomial technique to keep the bit-complexity of all iterations polynomially bounded in the input encoding length. This technique is not particular to the SLLS algorithm but can be used for any path-following method. The main subroutine takes an iterate (x, s) in the central path neighborhood, and computes $(\tilde{x}, \tilde{s})$ that is in a slightly larger neighborhood, may have slightly worse optimality gap, but its size is polynomially bounded in the input encoding length.

To argue about the encoding length, we 'anchor' the point $(\tilde{x}, \tilde{s})$ to vertices of the primal and dual polytopes. In strongly polynomial time, we can write a Minkowski–Weyl decomposition of x and s using vertices and extreme rays. However, we cannot simply round the coefficients. In particular, it is possible that x is written as a 'highly unstable' convex combination of primal and dual vertices such that either $\langle v, u\rangle < \langle x, s\rangle / 2^n$ or $\langle v, u\rangle < 2^n \langle x, s\rangle$ for each pair of primal and dual vertices (v, u). We proceed in two stages. First, we try to find a value $\mu^\star \approx \langle x, s\rangle / n$ such that $\mu^\star$ has small encoding length. This is easy as long as the combination contains primal and dual vertices (v, u) with $\langle x, s\rangle / 2^n \le \langle v, u\rangle \le 2^n \langle x, s\rangle$. In the 'highly unstable' situation as above, it turns out that the direction from (x, s) pointing towards a pair of primal and dual vertices (v, u) with much better gap is a very good movement direction of the IPM. Hence, we can replace (x, s) during the rounding step by a much better iterate that is also numerically more stable. In the second stage, we add a cost bound to our feasible region according to $\mu^\star$. On this bounded polytope, we can now find a Minkowski–Weyl decomposition and simply round the coefficients. The guarantees of this rounding are based on the near-monotonocity property of the central path.

Organization. The rest of the paper is structured as follows. Section 2 introduces some necessary background, in particular, regarding straight line complexity and circuits. Section 2.1 proves the straight line complexity bound in terms of the circuit imbalance $\kappa_{\mathbf{A}}$. Section 4 analyzes the straight line complexity of minimum cost generalized flows. Section 5 gives the simpler initialization method for generalized flows, using existing feasibility solvers. Section 6 introduces some necessary concepts and results for interior point methods. Section 7 presents the general initialization procedure, starting with a high level overview. Section 8 describes the rounding procedure needed to control the bit-complexity. Appendix B describes a variant of the SLLS IPM needed for our purposes, with a simple new SVD procedure in Section B.7.

2 Preliminaries

Notation. We let $\mathbb{R}_{++}$ denote the set of positive reals, and $\mathbb{R}_{+}$ the set of nonnegative reals; similarly for $\mathbb{Z}_{++}, \mathbb{Z}_{+}, \mathbb{Q}_{++}$ and $\mathbb{Q}_{+}$. For $n \in \mathbb{N}$, we let $[n] := \{1, 2, \dots, n\}$. We let $\mathbf{0}_n, \mathbf{1}_n \in \mathbb{R}^n$ denote the all 0s and all 1s vectors, respectively. For $x \in \mathbb{R}^n$, we let $\mathrm{supp}(x) \subseteq [n]$ denote its support. For $\alpha \in \mathbb{R}$, we let $\alpha^+ = \max\{\alpha, 0\}$ and $\alpha^- = \max\{-\alpha, 0\}$; for a vector $x \in \mathbb{R}^n$, we use x^+ and x^- coordinatewise. We let $\mathrm{supp}^+(x) = \mathrm{supp}(x^+)$ and $\mathrm{supp}^-(x) = \mathrm{supp}(x^-)$. For functions $f_i : \mathbb{R}^n \to \mathbb{R}^m$, $i \in \mathcal{I}$, we let $\bigvee_{i \in \mathcal{I}} f_i$ denote the pointwise maximum.

We let $\ker(\mathbf{A})$ denote the kernel of the matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$. The standard inner product of the two vectors $x, y \in \mathbb{R}^n$ is denoted as $\langle x, y \rangle = x^\top y$. For $x, y \in \mathbb{R}^n$, we let $x \circ y = (x_1 y_1, \dots, x_n y_n)$ denote the Hadamard-product, and if $y \in \mathbb{R}^n_{++}$, we let $x/y = (x_1/y_1, \dots, x_n/y_n)$.

We let

$$\mathcal{P} := \{x \in \mathbb{R}^n \mid \mathbf{A}x = b, x \geq \mathbf{0}\}, \quad \mathcal{D} := \{s \in \mathbb{R}^n \mid \exists y : \mathbf{A}^\top y + s = c\,,\ s \geq \mathbf{0}\}$$

denote the primal and dual feasible regions of (LP). We let $\mathcal{P}_{++} := \mathcal{P} \cap \mathbb{R}^n_{++}$ and $\mathcal{D}_{++} := \mathcal{D} \cap \mathbb{R}^n_{++}$ denote the strictly feasible regions. Interior point methods require $\mathcal{P}_{++}, \mathcal{D}_{++} \neq \emptyset$. We do not make this assumption in general. In Section 7, we show how one can use a sequence of reductions to simpler IPM problems to first either find a suitable initial point (x^0, s^0), or conclude infeasibility or unboundedness of the input LP.

The sublevel sets. Assume $\mathcal{P}, \mathcal{D} \neq \emptyset$, in which case (LP) admits a pair of primal and dual optimal solutions $(\bar{x}, \bar{y}, \bar{s})$ with optimum value $\langle c, \bar{x} \rangle = \langle b, \bar{y} \rangle = v^\star$. Recall that this holds precisely if these two solutions are complementary: $\langle \bar{x}, \bar{s} \rangle = 0$; in particular, $\bar{x}_i \bar{s}_i = 0$ for all $i \in [n]$.

Recall the definitions of the sublevel sets $\mathcal{P}_g$, $\mathcal{D}_g$ in (1) as the set of primal and dual solutions with objective value within g from the optimum value $v^\star$.

The duality gap of any pair (x, y, s) of primal-dual feasible points of (LP) fulfills $\langle c, x \rangle - \langle b, y \rangle = \langle x, s \rangle$. In particular, we have $\langle x, \bar{s} \rangle = \langle c, x \rangle - v^\star$ and $\langle \bar{x}, s \rangle = v^\star - \langle b, y \rangle$. Thus, the two sets $\mathcal{P}_g$ and $\mathcal{D}_g$ are equivalently given by

$$\mathcal{P}_g = \{x \in \mathcal{P} : \langle x, \bar{s} \rangle \leq g\} \quad \text{and} \quad \mathcal{D}_g = \{s \in \mathcal{D} : \langle \bar{x}, s \rangle \leq g\}\,.$$

These expressions are in fact independent of the choice of optimal solutions $(\bar{x}, \bar{y}, \bar{s})$. The following is immediate.

Proposition 2.1. *Assume that $\mathcal{P}_{++}$ and $\mathcal{D}_{++}$ are nonempty. Then, for all $g \geq 0$, the sets $\mathcal{P}_g$ and $\mathcal{D}_g$ are bounded.*

In (2), we defined the primal and dual max central path for the strictly feasible case. We extend the definition of the max central path $x^{\mathrm{m}}_i(g)$ for the case when $\max\{x_i : x \in \mathcal{P}_g\}$ is unbounded. It is easy to see that this is equivalent to the existence of an unbounded direction $v \in \ker(\mathbf{A})$, $\langle c, v \rangle \leq 0$, $v_i > 0$. If such a direction exists, then the program is unbounded for all values, including the set of optimal solutions for $g = 0$. In this case, we define $x^{\mathrm{m}}_i(g) = \infty$ for all $g \geq 0$, and use the convention that $\mathrm{SLC}_\eta(x^{\mathrm{m}}_i) = 1$ for all $\eta \in (0,1)$. Note that whenever $x^{\mathrm{m}}_i(g) = \infty$, then either $\langle c, v \rangle < 0$ for the above vector v in which case the dual program is infeasible; or $\langle c, v \rangle = 0$ in which case $s_i = 0$ for every dual feasible $s \in \mathcal{D}$, and therefore $s^{\mathrm{m}}_i(g) = 0$ for all $g \geq 0$. We use a similar convention for defining $s^{\mathrm{m}}_i(g)$ and $\mathrm{SLC}_\eta(s^{\mathrm{m}}_i)$.

The following lemma asserts that column rescalings do not affect the straight line complexity bounds.

Lemma 2.2. *Given an instance of (LP) with data $(\mathbf{A}, b, c)$, let $\mathbf{D} \in \mathbb{R}^{n \times n}$ be a positive diagonal matrix, and consider the LP with data $(\mathbf{AD}, b, \mathbf{D}c)$. Let $(x^{\mathrm{m}}, s^{\mathrm{m}})$ and $(\bar{x}^{\mathrm{m}}, \bar{s}^{\mathrm{m}})$ denote the max central paths of the two LPs. Then, $\mathrm{SLC}_\eta(x^{\mathrm{m}}_i) = \mathrm{SLC}_\eta(\bar{x}^{\mathrm{m}}_i)$ and $\mathrm{SLC}_\eta(s^{\mathrm{m}}_i) = \mathrm{SLC}_\eta(\bar{s}^{\mathrm{m}}_i)$ for any $\eta \in (0,1)$ and $i \in [n]$.*

Proof. We only prove for the primal max central paths; the proof for the dual case is analogous. Note that there is a one-to-one mapping $x \to \mathbf{D}^{-1}x$ between the feasible solutions of the two systems, with $\langle c, x \rangle = \langle \mathbf{D}c, \mathbf{D}^{-1}x \rangle$. This implies that $\bar{x}^{\mathrm{m}}(g) = \mathbf{D}^{-1}x^{\mathrm{m}}(g)$ for all $g \geq 0$. It is now easy to see that the two functions x^{m}_i and $\bar{x}^{\mathrm{m}}_i$ have the same straight line complexities. □

Elementary vectors and conformal circuit decompositions.

Definition 2.3 (Elementary vectors and circuits)**.** Let $\mathbf{A} \in \mathbb{R}^{m \times n}$ and assume $\ker(\mathbf{A}) \neq \{\mathbf{0}_n\}$. A vector $z \in \ker(\mathbf{A})$ is an *elementary vector in* $\ker(\mathbf{A})$ if z is a support-minimal nonzero vector in $\ker(\mathbf{A})$. We let $\mathcal{E}(\mathbf{A})$ denote the set of all elementary vectors. A set $C \subseteq [n]$ is a *circuit of* $\mathbf{A}$ if it is the support of some elementary vector; we let $\mathcal{C}(\mathbf{A}) \subseteq 2^{[n]}$ denote the set of circuits.

We say that a vector $y \in \mathbb{R}^n$ *conforms to* $x \in \mathbb{R}^n$ if $x_i y_i > 0$ whenever $y_i \neq 0$. A *conformal circuit decomposition* of a vector $z \in \ker(\mathbf{A})$ is a decomposition of the form

$$z = \sum_{i=1}^{\ell} g^{(i)},$$

where $g^{(1)}, \dots, g^{(\ell)} \in \mathcal{E}(\mathbf{A})$, $\ell \le n$, and each $g^{(i)}$ conforms to z. This notion can be seen as a generalization of cycle decompositions of circulations of networks flows. The existence of such a decomposition is well-known, see e.g., [Ful68, Roc69].

Proposition 2.4. *For every* $\mathbf{A} \in \mathbb{R}^{m\times n}$*, every vector* $z \in \ker(\mathbf{A})$ *admits a conformal circuit decomposition.*

2.1 Straight line complexity and circuits

In this section, we establish an intimate connection between the SLC of an LP and its circuits. Recall the definition (2) of the max-central path $(x^{\mathrm{m}}, s^{\mathrm{m}})$ from the introduction.

Definition 2.5 (h-curve)**.** Let $\bar{x}$ be a primal optimal solution to (LP). Given a vector $h \in \ker(\mathbf{A})$ where $\langle c, h\rangle \ge 0$, the *$h$-curve from $\bar{x}$* is the function $\bar{x}^h : \mathbb{R}_+ \to (\mathbb{R}_+ \cup \{\infty\})^n$ that maps $\bar{x}^h(g)$ to $\bar{x} + \alpha h$, for $\alpha \in \mathbb{R}_+ \cup \{\infty\}$ chosen maximally such that $\bar{x} + \alpha h \ge \mathbf{0}$ and $\langle c, \alpha h\rangle \le g$.

Note that $\bar{x}^h = \bar{x}^{\beta h}$ for all $\beta > 0$. It is easy to see that

$$\bar{x}^h(g) = \bar{x} + \min\left(\frac{g}{\langle c, h\rangle}, \min_{j\in \mathrm{supp}^-(h)} \frac{\bar{x}_j}{|h_j|}\right) h\,, \tag{5}$$

with the convention that we omit the first term from the minimum if $\langle c, h\rangle = 0$ ($\bar{x}^h$ is a constant function in this case). We will call functions of this form — linear on some interval starting from 0, and then constant — 1-*simple functions*, a notion we will expand on in Section 4.2. The next lemma shows that for every $i \in [n]$ and $g \ge 0$, the ith coordinate of the max-central path at g is upper bounded by a circuit augmentation from an optimal solution, up to a factor n.

Lemma 2.6. *Let $\bar{x}$ be a primal optimal solution to* (LP) *and $i \in [n]$. For every $g \ge 0$ where $x_i^{\mathrm{m}}(g) > \bar{x}_i$, there exists an elementary vector $h \in \mathcal{E}(\mathbf{A})$ such that $\langle c, h\rangle \ge 0$, $h_i > 0$, $h_j \ge 0$ whenever $\bar{x}_j = 0$, and $\bar{x}_i^h(g) \ge x_i^{\mathrm{m}}(g)/n$.*

Proof. Let $\hat{x}$ be a primal feasible solution to (LP) such that $\hat{x}_i = x_i^{\mathrm{m}}(g)$. Consider a conformal circuit decomposition of $\hat{x} - \bar{x} = \sum_{j=1}^{\ell} h^{(j)}$ as in Proposition 2.4. Note that $\langle c, h^{(j)}\rangle \ge 0$ for all $j \in [\ell]$ because $\bar{x}$ is a primal optimal solution to (LP). Let $k \in \arg\max_{j\in[\ell]} h_i^{(j)}$. Then, $h_i^{(k)} > 0$ due to $\hat{x}_i > \bar{x}_i$. Since $\left\langle c, \bar{x} + h^{(k)}\right\rangle \le \langle c, \bar{x}\rangle + \langle c, \hat{x} - \bar{x}\rangle = \langle c, \hat{x}\rangle \le g$, we obtain

$$\bar{x}_i^{h^{(k)}}(g) \ge \bar{x}_i + h_i^{(k)} \ge \bar{x}_i + \frac{\sum_{j=1}^{\ell} h_i^{(j)}}{\ell} \ge \frac{\hat{x}_i}{\ell} \ge \frac{\hat{x}_i}{n} = \frac{x_i^{\mathrm{m}}(g)}{n}\,.$$

Note also that $h_j^{(k)} \ge 0$ whenever $\bar{x}_j = 0$ since $\bar{x} + h^{(k)} \ge \mathbf{0}$. □

Definition 2.7 (Dominance)**.** Let $\bar{x}$ be a primal optimal solution to (LP). Let $i \in [n]$ and $\alpha \ge 0$. Given vectors $h, h' \in \ker(\mathbf{A})$ where $\langle c, h\rangle, \langle c, h'\rangle \ge 0$, we say that *$h$ α-dominates h' on i with respect to $\bar{x}$* if $\bar{x}_i^h \ge \alpha \bar{x}_i^{h'}$. More generally, given sets $S, S' \subseteq W$, we say that *S α-dominates S' on i with respect to $\bar{x}$* if $\langle c, h\rangle \ge 0$ for all $h \in S$, and for every $h' \in S'$ with $\langle c, h'\rangle \ge 0$, there exists $h \in S$ such that h α-dominates h' on i with respect to $\bar{x}$.

Definition 2.8 (Circuit cover)**.** Let $\bar{x}$ be a primal optimal solution to (LP). Let $i \in [n]$ and $\alpha \ge 0$. An *α-primal circuit cover of i with respect to $\bar{x}$* is a set $S \subseteq \ker(\mathbf{A})$ which α-dominates $\mathcal{E}(\mathbf{A})$ on i with respect to $\bar{x}$.

The utility of a circuit cover is illustrated by the following lemma. Note that $x_i^{\mathrm{m}}(0)$ is the maximum of the i-th coordinate of an optimal solution. Assuming $x_i^{\mathrm{m}}(0) < \infty$, there exists a (basic) optimal solution $\bar{x}$ such that $x_i^{\mathrm{m}}(0) = \bar{x}_i$.

Lemma 2.9. *Fix $i \in [n]$ such that $x_i^{\mathrm{m}}(0) < \infty$, and let $\bar{x}$ be a primal optimal solution to* (LP) *such that $\bar{x}_i = x_i^{\mathrm{m}}(0)$. If S is a α-primal circuit cover of i with respect to $\bar{x}$, then* $\mathrm{SLC}_{\alpha/n}(x_i^{\mathrm{m}}) \le |S| + 1$.

Proof. We may assume that x_i^{m} is not constant, as otherwise $\mathrm{SLC}_{\alpha/n}(x_i^{\mathrm{m}}) = 1$. Consider the function $\bar{x}_i^S : \mathbb{R}_+ \to \mathbb{R}_+$ defined by $\bar{x}_i^S(g) := \max_{h\in S} \bar{x}_i^h(g)$. It is piecewise-linear with at most $2|S|$ pieces, and its upper convex envelope has at most $|S|+1$ pieces. So, it is left to show that $\alpha x_i^{\mathrm{m}}/n \le \bar{x}_i^S \le x_i^{\mathrm{m}}$. The upper bound is immediate by Definition 2.5. For the lower bound, let $g > 0$. By Lemma 2.6, there exists an elementary vector $h' \in \mathcal{E}(\mathbf{A})$ such that $\langle c, h'\rangle \ge 0$ and $\bar{x}_i^{h'}(g) \ge x_i^{\mathrm{m}}(g)/n$. Since S is an α-primal circuit cover of i with respect to $\bar{x}$, there exists a vector $h \in S$ such that $\langle c, h\rangle \ge 0$ and

$$\bar{x}_i^S(g) \ge \bar{x}_i^h(g) \ge \alpha \bar{x}_i^{h'}(g) \ge \frac{\alpha}{n} x_i^{\mathrm{m}}(g). \qquad \square$$

2.2 Reducing 2VPI LPs to generalized flows

As already noted, solving 2VPI linear programs strongly polynomially reduces to solving minimum-cost generalized flow problems. We observe that the reduction by Hochbaum [Hoc04] also preserves the straight line complexities up to a constant factor. Hence, our straight line complexity bound for generalized flows is directly applicable to 2VPI linear programs. We first state Hochbaum's reduction.

Theorem 2.10 ([Hoc04])**.** *Given an instance of* (LP) *with data* $(\mathbf{A}, b, c)$, $\mathbf{A} \in \mathbb{R}^{m\times n}$ *such that every column of* $\mathbf{A}$ *contains at most two nonzero entries, one can in strongly polynomial time construct another instance of* (LP) *with data* $(\bar{\mathbf{A}}, \bar{b}, \bar{c})$ *such that* $\bar{\mathbf{A}} \in \mathbb{R}^{(2m)\times(2n)}$, $\bar{c} = (c, c)$, *and every column of* $\bar{\mathbf{A}}$ *contains at most one positive and at most one negative entry such that the following hold: For every feasible solution* $x \in \mathbb{R}^n$ *to the original system,* $\bar{x} = (x, x) \in \mathbb{R}^{2n}$ *is a feasible solution to the new system with* $\langle c, x\rangle = \langle \bar{c}, \bar{x}\rangle$*; and for every feasible solution* $\bar{x} = (\bar{x}', \bar{x}'')$ *to the modified system,* $(\bar{x}'', \bar{x}')$ *is also a feasible solution to the modified system, while* $x = (\bar{x}' + \bar{x}'')/2$ *is a feasible solution to the original system with* $\langle c, x\rangle = \langle \bar{c}, \bar{x}\rangle$.

After rescaling the columns in the new system $(\bar{\mathbf{A}}, \bar{b}, \bar{c})$, we obtain a minimum-cost generalized flow problem (with possible loops). According to Lemma 2.2, rescaling columns does not affect the $\mathrm{SLC}_\eta(x_i^{\mathrm{m}})$ values. The reduction immediately shows that the SLC values of the two systems are within a factor two. Thus, we obtain:

Lemma 2.11. *Given* $(\mathbf{A}, b, c)$ *and* $(\bar{\mathbf{A}}, \bar{b}, \bar{c})$ *as in Theorem 2.10, let* x_i^{m} *and* $\bar{x}_i^{\mathrm{m}}$ *denote the respective max central path curves. Then,*

$$x_i^{\mathrm{m}} \le \bar{x}_i^{\mathrm{m}} = \bar{x}_{n+i}^{\mathrm{m}} \le 2x_i^{\mathrm{m}} \quad \forall i \in [n].$$

Consequently, $\mathrm{SLC}_{\eta/2}(x_i^{\mathrm{m}}) \le \mathrm{SLC}_\eta(\bar{x}_i^{\mathrm{m}})$, *for any* $\eta \in (0,1)$ *and* $i \in [n]$.

3 Straight line complexity in terms of the circuit imbalance measure

In this section, we show how the straight-line complexities can be bounded for (LP) in terms of the circuit imbalance $\kappa_{\mathbf{A}}$. While this is not needed for our main result on generalized flows, where $\kappa_{\mathbf{A}}$ may be arbitrarily large, it gives a simple demonstration of our approach using circuit covers. Let us start with the definition of $\kappa_{\mathbf{A}}$.

Definition 3.1 (Circuit imbalances)**.** Let $\mathbf{A} \in \mathbb{R}^{m\times n}$. If $\ker(\mathbf{A}) = \{\mathbf{0}_n\}$ we define the *circuit imbalance of* $\mathbf{A}$ as $\kappa_{\mathbf{A}} = 1$. Otherwise, we let

$$\kappa_{\mathbf{A}} := \max\left\{\left|\frac{h_i}{h_j}\right| : h \in \mathcal{E}(\mathbf{A}),\, i, j \in \mathrm{supp}(h)\right\}.$$

Theorem 3.2. *Assume* $\mathcal{P}, \mathcal{D} \neq \emptyset$ *for an instance of* (LP) *given by* $(\mathbf{A}, b, c)$. *For each* $i \in [n]$, *we have* $\mathrm{SLC}_\eta(x_i^{\mathrm{m}}) \le \min\{m, n-m\} + 1$ *for* $\eta = 1/(n^2\kappa_{\mathbf{A}}^2)$.

Using Lemma 2.9, the theorem follows from the following lemma.

Lemma 3.3. *Fix* $k \in [n]$ *with* $x_k^{\mathrm{m}}(0) < \infty$ *and let* $\bar{x}$ *be a primal optimal solution to* (LP) *with* $\bar{x}_k = x_k^{\mathrm{m}}(0)$. *Then, there exists an* $1/(n\kappa_{\mathbf{A}}^2)$*-primal circuit cover* S *of* k *with respect to* $\bar{x}$ *with* $|S| \le \min\{m, n-m\} + 1$.

Proof. Let us denote $\kappa = \kappa_{\mathbf{A}}$. We can clearly pick the primal optimal solution $\bar{x}$ as in the statement to be a basic solution.let $\bar{s}$ be any dual basic optimal solution. Thus, $\mathrm{supp}(\bar{x}) \le m$ and $\mathrm{supp}(\bar{s}) \le n-m$. By the complementarity of $\bar{x}$ and $\bar{s}$, we can reorder the index set such that $\bar{x}_1 \ge \bar{x}_2 \ge \cdots \ge \bar{x}_n$ and $\bar{s}_1 \le \bar{s}_2 \le \cdots \le \bar{s}_n$. Let $p := \max\{i \in [n] : \bar{x}_i > 0\}$ and $d := \min\{i \in [n] : \bar{s}_i > 0\}$. Clearly, $p < d$. By the basic choice of $\bar{x}$ and $\bar{s}$, $p \le m$ and $d \ge n-m+1$.

We now construct the circuit cover S of k. Note that for any circuit $C \in \mathcal{C}(\mathbf{A})$ and any elementary vector $h \in \mathcal{E}(\mathbf{A})$ such that $\langle c, h\rangle \geq 0$ and $\mathrm{supp}(h) = C$, the h-curve $\bar{x}^h$ is the same. Given a circuit $C \in \mathcal{C}'$, we define $h^C \in \mathcal{E}(\mathbf{A})$ as a fixed elementary vector with $\mathrm{supp}(h^C) = C$ normalized such that $h_k \in \{0,1\}$. Further, let

$$\mathcal{C}' := \{C \in \mathcal{C}(\mathbf{A}) \,:\, \left\langle c, h^C\right\rangle \geq 0\,,\, h^C_k = 1\,, h^C_j \geq 0\ \forall p < j \leq n\}\,.$$

Note that if $\mathcal{C}' = \emptyset$ then Lemma 2.6 implies $x^{\mathrm{m}}_k(g) = x^{\mathrm{m}}_k(0)$ for every $g \geq 0$ and hence $\mathrm{SLC}_1(x^{\mathrm{m}}_k) = 1$. For the rest, we assume $\mathcal{C}' \neq \emptyset$.

We will define the cover S in terms of the support circuits. As the first step, let us define a 'combinatorial signature' of circuits. Given a circuit $C \in \mathcal{C}'$, let

$$I^p(C) := \max\{1 \leq i \leq p \,:\, h^C_i < 0\} \quad \text{and} \quad I^d(C) := \max\{d \leq j \leq n \,:\, h^C_j > 0\}.$$

We define $I^p(C) = 0$ if the first set is empty and $I^d(C) = 0$ if the second set is empty. We say that a circuit $C \in \mathcal{C}'$ is *dominated* if there exists a circuit $C' \in \mathcal{C}'$ such that

$$I^p(C) \geq I^p(C') \quad \text{and} \quad I^d(C) \geq I^d(C')\,,$$

and at least one of the two inequalities above is strict. Let $\mathcal{D} \subseteq \mathcal{C}'$ be a maximal collection of undominated circuits with distinct $(I^p(C), I^d(C))$, and define

$$S := \{h^C \,:\, C \in \mathcal{D}\}\,.$$

Clearly, $|S| = |\mathcal{D}| \leq \min(p+1, n-d+2) \leq \min\{m, n-m\}+1$. We show that S is an $1/(n\kappa^2)$-circuit cover of i with respect to $\bar{x}$.

To see this, consider any $h \in \mathcal{E}(\mathbf{A})$ with $\langle c, h\rangle \geq 0$. After normalization, either $h = h^C$ or $h = -h^C$ for a circuit $C \in \mathcal{C}(\mathbf{A})$. We can assume the first case, as in the latter case $\bar{x}^h_k(g) \leq \bar{x}_k$ for $g \geq 0$. Hence, $\bar{x}^h$ would be 1-dominated by h^C on k for any $C \in \mathcal{C}'$. Moreover, we can assume $C \in \mathcal{C}'$, as otherwise $\bar{x}^h_k(g) = \bar{x}_k$ for all $g \geq 0$.

By definition, there is a $C' \in \mathcal{D}$ such that $I^p(C) \geq I^p(C')$ and $I^d(C) \geq I^d(C')$. For $h = h^C$ and $h' = h^{C'}$, our goal is to show that

$$\bar{x}^h_k(g) \leq n\kappa^2 \bar{x}^{h'}_k(g) \quad \forall g \geq 0\,. \tag{6}$$

Let $i' = I^p(C') \leq I^p(C) = i$ and $j' = I^d(C') \leq I^d(C) = j$. By the definition of $\kappa = \kappa_{\mathbf{A}}$, we have

$$\frac{1}{\kappa} \leq |h_\ell|, |h'_\ell| \leq \kappa\,, \quad \forall \ell \in [n]\,. \tag{7}$$

Noting that $\langle c, h\rangle = \langle \bar{s}, h\rangle$ and $\langle c, h'\rangle = \langle \bar{s}, h'\rangle$ because of $h, h' \in \ker(\mathbf{A})$ and $c - \bar{s} \in \ker(\mathbf{A})^\perp$. Using $h_k = h'_k = 1$, we can write

$$\bar{x}^h_k(g) = \bar{x}_k + \min\left(\frac{g}{\langle \bar{s}, h\rangle}, \min_{\ell \in \mathrm{supp}^-(h)} \frac{\bar{x}_\ell}{|h_\ell|}\right) \quad \text{and} \quad \bar{x}^{h'}_k(g) = \bar{x}_k + \min\left(\frac{g}{\langle \bar{s}, h'\rangle}, \min_{\ell \in \mathrm{supp}^-(h')} \frac{\bar{x}_\ell}{|h'_\ell|}\right)\,.$$

We show that

$$\langle \bar{s}, h'\rangle \leq n\kappa^2 \langle \bar{s}, h\rangle \quad \text{and} \quad \min_{\ell \in \mathrm{supp}^-(h)} \frac{\bar{x}_\ell}{|h_\ell|} \leq n\kappa^2 \min_{\ell' \in \mathrm{supp}^-(h')} \frac{\bar{x}_{\ell'}}{|h'_{\ell'}|}\,. \tag{8}$$

Let us start by showing the first inequality. If $j' = 0$, then $\langle \bar{s}, h'\rangle = 0$ and this trivially holds.Otherwise, by the definition of $j = I^d(C) \geq j' = I^d(C') \geq d$ and using (7), we get

$$\langle \bar{s}, h'\rangle \leq \sum_{\ell=d}^{j'} \bar{s}_\ell h'_\ell \leq n\kappa \bar{s}_{j'} \leq n\kappa^2 \bar{s}_j h_j \leq n\kappa^2 \langle \bar{s}, h\rangle$$

whenever $j' \geq d$; the last inequality follows since $h_\ell \geq 0$ for all $\ell > p$ since $C \in \mathcal{C}'$.

Let us now verify the second inequality in (8). Let ℓ and ℓ' denote the minimizers, respectively. Since $C \in \mathcal{C}$, it follows that either $i = 0$, that is, $\mathrm{supp}^-(h) = \emptyset$ or $1 \leq \ell \leq i = I^p(C)$; similarly for h'. If $i' = 0$ then the expression for h' is ∞. Hence, we can assume $1 \leq \ell' \leq i' \leq i$. We get

$$\frac{x_{\ell'}}{|h_{\ell'}|} \geq \frac{x_{\ell'}}{\kappa} \geq \frac{x_i}{\kappa} \geq \frac{1}{\kappa^2} \cdot \frac{x_i}{|h_i|} \geq \frac{1}{\kappa^2} \cdot \frac{\bar{x}_\ell}{|h_\ell|}\,,$$

where the first inequality uses (7); the second inequality uses $\ell' \leq i$ and the ordering of the indices; the third inequality uses again (7); and the last inequality uses the choice of ℓ as the minimizer, noting that $h_i < 0$. $\square$

Remark 3.4. We note that, together with Theorem 1.3, we obtain a bound $O(\min\{m, n-m\}n^{1.5}\log(\kappa_{\mathbf{A}}+n))$ on the number of iterations of the SLLS Algorithm in [ADL+23]. In comparison, the crossover analysis in [VY96] yields an $O(n^{3.5}\log(\kappa_{\mathbf{A}}+n))$ iteration bound. In [DHNV23], an amortized version of the crossover analysis yields an $O(n^{2.5}\log(n)\log(\kappa_{\mathbf{A}}+n))$ bound; this is also applicable to earlier LLS algorithms such as the one in [VY96]. Thus, our iteration bound improves on the state of the art, even if the time complexity per iteration of the SLLS steps is higher than for the LLS steps. Further, using the scale-invariance of straight line complexities (Lemma 2.2), the bound immediately becomes $O(\min\{m, n-m\}n^{1.5}\log(\kappa^\star_{\mathbf{A}}+n))$, where $\kappa^\star_{\mathbf{A}}$ is the smallest value of $\kappa_{\mathbf{AD}}$ taken over positive diagonal matrices $\mathbf{D} \in \mathbb{R}^{n\times n}$.

4 Minimum-cost generalized flow

Let $G = (V, E)$ be a directed multigraph with arc capacities $u \in (\mathbb{R}_{++} \cup \{\infty\})^E$ and gain factors $\gamma \in \mathbb{R}^E_{++}$. A *flow* in G is any nonnegative vector $x \in \mathbb{R}^E_+$. Note that a flow is allowed to violate arc capacities. For a node $i \in V$, we denote $\delta^{\text{in}}(i)$ and $\delta^{\text{out}}(i)$ as the set of incoming and outgoing arcs of i respectively. The *net flow* of x at node i is defined as

$$\nabla_i x := \sum_{e\in\delta^{\text{in}}(i)} \gamma_e x_e - \sum_{e\in\delta^{\text{out}}(i)} x_e \,.$$

A flow x is a *circulation* if $\nabla_i x = 0$ for all $i \in V$. For $i, j \in V$, we denote by $E_{i,j} \subseteq E$ the subset of arcs with tail i and head j.

An instance of the *minimum-cost generalized flow* problem is given by a directed multigraph $G = (V, E)$ with node demands $b \in \mathbb{R}^V$, arc costs $c \in \mathbb{R}^E$, capacities $u \in (\mathbb{R}_{++} \cup \{\infty\})^E$ and gain factors $\gamma \in \mathbb{R}^E_{++}$. It can be formulated as the following LP:

$$\begin{aligned} \min\ & \langle c, x\rangle \\ \nabla_i x &= b_i \qquad \forall i \in V \\ \mathbf{0} \le x &\le u\,. \end{aligned} \tag{MGF}$$

Throughout this section, we will use n for the number of nodes of G and m for the number of arcs; *note that applied to* (MGF)*, this is the reverse of the convention used for general LPs*. Let $E_c \subseteq E$ denote the subset of arcs with finite capacities. We define $m_c := |E_c|$ for the number of finite capacity arcs.

We assume that (MGF) has a finite optimum, since otherwise the max central path does not exist. Let r and s be the dual slack variables corresponding to the upper bound and nonnegativity constraints in (MGF) respectively. For every $\lambda \ge 0$, the primal max central path is given by

$$\begin{aligned} \forall e \in E, \qquad x^{\text{m}}_e(\lambda) := \max\ & x_e \\ \nabla_i x &= b_i \qquad \forall i \in V \\ \mathbf{0} \le x &\le u \\ \langle r^*, u_{E_c} - x_{E_c}\rangle + \langle s^*, x\rangle &\le \lambda \\ \forall e \in E_c, \qquad x^{\text{m}}_{\overleftarrow{e}}(\lambda) := \max\ & u_e - x_e \\ \nabla_i x &= b_i \qquad \forall i \in V \\ \mathbf{0} \le x &\le u \\ \langle r^*, u_{E_c} - x_{E_c}\rangle + \langle s^*, x\rangle &\le \lambda\,, \end{aligned} \tag{MCP-Flow}$$

where $(r^*, s^*) \ge \mathbf{0}$ is any dual optimal solution to (MGF). Our goal in this section is to prove the following bound on the SLC of each coordinate of the primal max-central path.

Theorem 4.1. *Given an instance of minimum-cost generalized flow with a finite optimum,*

$$\mathrm{SLC}_\eta(x^{\text{m}}_e) = O(m(m_c+n)) \qquad \textit{for every arc } e \in E$$

and

$$\mathrm{SLC}_\eta(x^{\text{m}}_{\overleftarrow{e}}) = O(m(m_c+n)) \qquad \textit{for every arc } e \in E_c$$

for some $\eta = \Omega(1/(m^2 n))$.

We can reduce to the case where $x^*_e = u_e$ for all $\mathrm{supp}^-(c)$ and $x^*_e = 0$ for all $e \in \mathrm{supp}^+(c)$, for every primal optimal solution x^* to (MGF). This is achieved by replacing c with $c_e = s^*_e - r^*_e$ for all $e \in E_c$, and $c_e = s^*_e$ for all $e \in E \setminus E_c$, where (r^*, s^*) is any dual optimal solution to (MGF). Note that $\langle r^*, s^*_{E_c} \rangle = 0$. This has no impact on x^{m} because (r^*, s^*) remains a dual optimal solution. Summarizing, without loss of generality we assume the following going forward.

Assumption 4.2. *Every primal optimal solution x^* to (MGF) satisfies $x^*_e = u_e$ for all $e \in \mathrm{supp}^-(c)$ and $x^*_e = 0$ for all $e \in \mathrm{supp}^+(c)$.*

4.1 Reduction to the generalized circulation problem

It will be convenient for our purposes to work not with the original minimum cost generalized flow problem, but a derived instance of *minimum-cost generalized circulation*. The latter problem is a special case of the former problem in which all the node demands are zero. The goal is to find a minimum-cost *circulation*, which has net flow zero everywhere, satisfying the capacities.

$$\begin{aligned} \min\ &\langle c, x\rangle \\ \nabla_i x &= 0 \qquad \forall i \in V \\ \mathbf{0} \le x &\le u\,. \end{aligned} \tag{MGC}$$

For our purposes, we will convert to a trivial instance of the minimum-cost generalized circulation problem where all costs are nonnegative, meaning that the optimal solution is $x^* = \mathbf{0}$. The straight line complexity of the max-central path cannot decrease, however. We will make use of the residual graph:

Definition 4.3 (Residual graph)**.** Let x^* be any primal feasible solution to (MGF). The *residual graph with respect to* x^* is the multigraph G_{x^*} with vertex set V and arc set $\{e \in E : x^*_e < u_e\} \cup \{\overleftarrow{e} : e \in \mathrm{supp}(x^*)\}$, where $\overleftarrow{e}$ is an arc from j to i whenever $e \in E_{i,j}$. Every arc $e \in E(G_{x^*}) \cap E$ retains the same gain factor γ_e and cost c_e, but has capacity $u_e - x^*_e$. For every $e \in \mathrm{supp}(x^*)$, its *reverse arc* $\overleftarrow{e}$ receives gain factor $\gamma_{\overleftarrow{e}} := 1/\gamma_e$, cost $c_{\overleftarrow{e}} := -c_e/\gamma_e$, and capacity $u_{\overleftarrow{e}} := \gamma_e x^*_e$.

Note that given an instance of (MGF) satisfying Assumption 4.2, and choosing x^* to be a primal optimal solution, the resulting instance of (MGC) on G_{x^*} has the property that all costs are nonnegative, and $\mathbf{0}$ is an optimal solution. The proof is deferred to Appendix A.

Lemma 4.4. *Let $(G = (V, E), \gamma, c, u, b)$ be an instance of minimum-cost generalized flow as given by (MGF) satisfying Assumption 4.2, and let x^* be any primal optimal solution. Let x^{m} be the corresponding primal max-central path as given by (MCP-Flow). Consider the instance of minimum-cost generalized circulation on the residual graph G_{x^*}, and let $\hat{x}^{\mathrm{m}}$ be the corresponding primal max-central path. For any $e \in E$ and $\eta \in (0, 1)$,*

(i) If $x^*_e = u_e$, then $\mathrm{SLC}_\eta(x^{\mathrm{m}}_e) = 1$.

(ii) If $x^*_e < u_e$, then $\mathrm{SLC}_{\eta/2}(x^{\mathrm{m}}_e) \le \mathrm{SLC}_\eta(\hat{x}^{\mathrm{m}}_e)$.

(iii) If $e \in E_c$ and $x^*_e > 0$, then $\mathrm{SLC}_{\eta/2}(x^{\mathrm{m}}_{\overleftarrow{e}}) \le \mathrm{SLC}_\eta(\hat{x}^{\mathrm{m}}_{\overleftarrow{e}})$.

(iv) If $e \in E_c$ and $x^*_e = 0$, then $\mathrm{SLC}_\eta(x^{\mathrm{m}}_{\overleftarrow{e}}) = 1$.

We consider the generalized circulation problem obtained from the described reduction from now on. As such, we use $(G = (V, E), \gamma, c, u)$ to refer to this instance, which has nonnegative costs and where $\mathbf{0}$ is an optimal solution. As already mentioned, we write $n = |V|$ and $m = |E|$. Let F denote the set of finite capacity arcs in this instance, and define $\bar{m} := |F|$. Note that if we begin with an instance of (MGF) with m_c finite capacity arcs, and apply the reduction using a *basic* optimal solution x^* to (MGF), then $|\mathrm{supp}(x^*)| \le |V|$, and so $\bar{m} \le m_c + n$.

4.2 Simple functions

Recalling the definition of an h-curve, and particularly (5), we see that these curves have a particularly simple form: they are linear for some interval starting at 0, and then constant for the remainder. The following definition is a multivariate extension of this simple functional form.

Definition 4.5. Fix some $n \geq 1$. We say that a function $f : \mathbb{R}_+^n \to \mathbb{R}_+$ is 1-*simple* if it can be written as

$$f(x) = \min(v_0, v_1 x_1, v_2 x_2, \dots, v_n x_n)$$

for some $v_0, v_1, \dots, v_n \in \mathbb{R}_+ \cup \{\infty\}$. Say that f is k-*simple* if it is the maximum of at most k 1-simple functions (and just "simple" if it is k-simple for some k).

Let us connect univariate k-simple functions to the general approach discussed in Section 2.1. Given an optimal primal solution $\bar{x}$, coordinate i, and an α-primal circuit cover S of i with respect to $\bar{x}$, consider the function $\bar{x}^S := \bigvee_{h \in S} \bar{x}^h$; this is a $|S|$-simple function by construction. Lemma 2.9 shows that $(\alpha/n) x_i^{\mathrm{m}} \leq \bar{x}^S \leq x_i^{\mathrm{m}}$. The bound on $\mathrm{SLC}_{\alpha/n}(x_i^{\mathrm{m}})$ that results is abstractly a consequence of $\bar{x}^S$ being $|S|$-simple.

The primary reason we introduce the notion of simple functions here is that we will make use of the following "composition" lemma. It shows that a bivariate simple function composed with a univariate simple function in a certain way yields another univariate simple function. We will later apply this with simple functions obtained from dominating collections of walks and other objects.

Lemma 4.6. *Suppose $f : \mathbb{R}_+^2 \to \mathbb{R}_+$ is k_1-simple and $g : \mathbb{R}_+ \to \mathbb{R}_+$ is k_2-simple. Then $h : \mathbb{R}_+ \to \mathbb{R}_+$ defined by $h(x) = f(x, g(x))$ is $O(k_1 + k_2)$-simple.*

We postpone the proof of this statement, and the further basic statements about simple functions that follow, to Appendix A.

Lemma 4.7. *Given $f_1, \dots, f_r : \mathbb{R}_+^n \to \mathbb{R}_+$, with f_i being k_i-simple for each i, then $\bigvee_i f_i$ is $(k_1 + k_2 + \dots + k_r)$-simple.*

Proof. Immediate. □

Lemma 4.8. *Given $f_1, \dots, f_r : \mathbb{R}_+ \to \mathbb{R}_+$, with f_i being k_i-simple for each i, then $\bigwedge_i f_i$ is $(k_1 + k_2 + \dots + k_r)$-simple.*

The following essentially says that there is a *unique* minimal description of any simple function as the maximum of 1-simple functions.

Lemma 4.9. *Suppose $f : \mathbb{R}_+^n \to \mathbb{R}_+$ is k-simple but not $(k-1)$-simple, with $f = \bigvee_{i \leq k} f^{(i)}$ for 1-simple functions $f^{(i)}$. Then for any description of f as $f = \bigvee_{i \leq \ell} g^{(i)}$ for 1-simple functions $g^{(i)}$, there is a set $S \subseteq \{1, 2, \dots, \ell\}$ with $|S| = k$ so that $\{f^{(i)} : i \leq k\} = \{g^{(i)} : i \in S\}$.*

4.3 Some flow-related definitions

Definition 4.10 (Walk, trail, path and cycle)**.** A *walk* is a sequence $W = (v_0, e_1, \dots, e_\ell, v_\ell)$ where e_i is an arc from v_{i-1} to v_i for all $i \in [\ell]$. It is *closed* if $v_0 = v_\ell$, and *open* otherwise. If $e_i \neq e_j$ for all $i \neq j$, then it is called a *trail*. If $v_0 = v_\ell$ and $v_i \neq v_j$ for all $0 \leq i < j \leq \ell$, then it is called a *cycle at* v_0.

The *gain* of W is

$$\gamma(W) := \prod_{i=1}^{\ell} \gamma_{e_i},$$

with the convention $\gamma(W) := 1$ if $\ell = 0$. We call W *flow-generating* if $\gamma(W) > 1$, *conservative* if $\gamma(W) = 1$, and *flow-absorbing* if $\gamma(W) < 1$.

For an r-s walk Q and an s-t walk W, we denote $Q \oplus W$ as the concatenated r-t walk. If W' is a prefix or suffix of W, we denote $W \setminus W'$ as the subwalk obtained by removing W' from W.

Definition 4.11 (Objects)**.** A *flow-generating object at $t \in V$* is a pair (C, W) where C is a flow-generating s-s walk for some $s \in V$ and W is an s-t walk. It is *simple* if C is a cycle, W is a path, and $V(C) \cap V(W) = \{s\}$.

A *flow-absorbing object at $s \in V$* is a pair (W, D) where W is an s-t walk for some $t \in V$ and D is a flow-absorbing t-t walk. It is *simple* if W is a path, D is a cycle, and $V(W) \cap V(D) = \{t\}$.

A *conservative object* is a triple (C, W, D) where either

(i) for some $s, t \in V$, C is a flow-generating s-s walk, W is an s-t walk, and D is a flow-absorbing t-t walk; or

(ii) C is a conservative closed s-s-walk for some s, W is the trivial path at s, and $D = C$.

The object is *simple* in case (ii) if C is a cycle, and in case (i) if (C, W) and (W, D) are simple, and

- $V(C) \cap V(D) = \emptyset$ in the case that $E(W) \neq \emptyset$; or
- the intersection of C and D is a path, in the case that $E(W) = \emptyset$.

(Case (ii) for a conservative object may look somewhat strange, but it essentially views a conservative cycle as a degenerate bicycle; this will be convenient in covering all cases with a single argument.)

We extend the concatentation operator $\oplus$ to objects in the natural way. For a flow-generating object $U_1 = (C_1, W_1)$ at s and an s-t walk Q, we use $U_1 \oplus Q$ to denote the flow-generating object at t obtained by combining U_1 and Q, i.e., the object $(C_1, W_1 \oplus Q)$. Similarly, for a flow-absorbing object $U_2 = (W_2, C_2)$ at t, $Q \oplus U_2 := (Q \oplus W_2, C_2)$. And if $U_1 = (C_1, W_1)$ and $U_2 = (C_2, W_2)$ are respectively flow-generating and flow-absorbing objects at the same node s, then $U_1 \oplus U_2$ is the conservative object $(C_1, W_1 \oplus Q \oplus W_2, C_2)$.

Definition 4.12 (Recurrence)**.** A walk W is called *k-recurrent* if every arc appears at most k times as a step in W. Similarly, an object U is *k-recurrent* if every arc appears at most k times in total as a step in some constituent walk of U.

Note that k-recurrent only upper bounds the number of repetitions of an arc; for $k \leq \ell$, any k-recurrent walk is also ℓ-recurrent. When considering flows supported on the arc set of a walk, it will be important to be able distinguish between flow on different "steps" of the walk that involve the same arc of the graph. We do this formally by defining the "splitting" of a walk, which simply makes parallel copies of arcs to turn the walk into a corresponding trail.

Definition 4.13 (Splitting)**.** Let $\tilde{G} = (V, \tilde{E})$ be the directed multigraph with the same node set as G, but with $10n$ parallel copies of each arc, each with the same gain factor, cost and capacity as the corresponding arc in G. (The choice of $10n$ is just to be sufficiently large for our purposes.) For each $e \in E$, we use $e^1, e^2, \ldots, e^{10n}$ to index the corresponding copies in $\tilde{E}$.

Given a $10n$-recurrent walk $W = (v_0, e_1, v_1, e_2, \ldots, e_r, v_r)$ in G, we define a *splitting* of W to be a trail $\tilde{W} = (v_0, e_1^{\sigma_1}, v_2, e_2^{\sigma_2}, \ldots, e_r^{\sigma_r}, v_r)$ in $\tilde{G}$, where for each $i \neq j$ with $e_i = e_j$, $\sigma_i \neq \sigma_j$.

Given an object U, a splitting of U is a tuple of trails in $\tilde{G}$, each being a splitting of the corresponding walk in U, and where in addition the trails are arc-disjoint.

Note that up to trivial relabelling of copies of arcs, the splitting of a walk or object is unique.

For an object or walk U, we use $V(U)$ and $E(U)$ to denote its node set and arc set respectively, and also use $E(\tilde{U}) \subseteq \tilde{E}$ to denote the arc set of a splitting $\tilde{U}$.

Definition 4.14 (Induced flows)**.** Given an s-t-walk W with splitting $\tilde{W}$, we say that $\tilde{x} \in \mathbb{R}_+^{\tilde{E}}$ is a *flow induced by* $\tilde{W}$ if $\tilde{x}$ is nonzero, supported on $E(\tilde{W})$, and $\gamma_e \tilde{x}_e = \tilde{x}_f$ for any pair of consecutive arcs $e, f \in E(\tilde{W})$ where e comes before f in $\tilde{W}$. We say that $\bar{x} \in \mathbb{R}_+^{E}$ is a *flow induced by* W if $\bar{x}$ is the projection onto G of a flow $\tilde{x}$ induced by a splitting of W, that is, $\bar{x}_e = \sum_j \tilde{x}_{e^j}$.

Given a flow-generating object $U = (C, W)$ at t, a flow induced by a splitting $\tilde{U} = (\tilde{C}, \tilde{W})$ is a vector $\tilde{x} \in \mathbb{R}_+^{\tilde{E}}$ that can be written as a sum of a flow induced by $\tilde{C}$ and a flow induced by $\tilde{W}$, and where in addition $\nabla_v \tilde{x} = 0$ for all $v \neq t$. The definition for flow-absorbing objects is completely analogous; and for a conservative object $U = (C, W, D)$, $\tilde{x}$ should satisfy $\nabla \tilde{x} = 0$, and be a sum of flows induced by the components of a splitting of U.

We remark that all flows induced by an object are the same up to scaling[3].

Induced flows have no particular scaling. The following will be a crucial notion: it is the largest possible flow induced by a walk (or object), with the property that on each step of the walk or object, the flow does not exceed the capacity of the arc, and the cost of that step (flow times arc cost) does not exceed a given bound λ.

Definition 4.15. Let W be a walk, with $\tilde{W}$ a splitting of W and $\tilde{x}$ a flow induced by $\tilde{W}$. Define $x^{\tilde{W}} : \mathbb{R}_+ \to \mathbb{R}_+^{\tilde{E}}$ to be the function that maps $x^{\tilde{W}}(\lambda)$ to the largest scaling of $\tilde{x}$ so that $\tilde{x}_a \leq u_a$ and $c_a \tilde{x}_a \leq \lambda$ for each $a \in E(\tilde{W})$. Then let $x^W : \mathbb{R}_+ \to \mathbb{R}_+^E$ be the projection of $x^{\tilde{W}}$ onto G, i.e., $x_e^W(\lambda) = \sum_j x_{e^j}^{\tilde{W}}(\lambda)$.

We define $x^{\tilde{U}}(\lambda)$ and $x^U(\lambda)$ for an object U with splitting $\tilde{U}$ in identical fashion.

Note that $x^W(\lambda)$ and $x^U(\lambda)$ do not depend on the choice of splitting, and so are well-defined.

[3] Specifically for the case of a conservative object $U = (C, \{s\}, C)$ for a conservative s-s-walk C, a flow $\tilde{x}$ induced by a splitting $\tilde{U}$ is exceptionally not unique up to scaling, in that flow can be arbitrarily shifted between the two copies of C. But this is unimportant, and moreover the flow induced by U is still unique up to scaling.

Remark 4.16. This definition is closely related to the definition of h-curves for general LPs provided in Definition 2.5. It is more general, in that we define x^U for objects that are not conservative, and hence which do not lie in the kernel. If we consider a conservative object U, and take h to be a flow induced by U, then x^U and the h-curve $\mathbf{0}^h$ are "close": if U is k-recurrent, then $\frac{1}{km}x^U(\lambda) \leq \mathbf{0}^h(\lambda) \leq x^U(\lambda)$. The reason that they are not identical, only within a factor km, is simply because of the per-step nature of the capacity bounds (meaning $x^U(\lambda)$ might overload an arc by a factor k) and cost bounds (meaning $x^U(\lambda)$ could have total cost $km\lambda$, given each arc could in principle contribute a cost of $k\lambda$).

4.4 SLC bounds via domination

We will follow exactly the general plan discussed in Section 2.1: we demonstrate the existence of a small primal circuit cover. Recall the notion of an elementary vector from Definition 2.3. The following is precisely this same notion, in the context of generalized circulations.

Definition 4.17 (Elementary circulation)**.** A nonzero circulation f in G is *elementary* if $\mathrm{supp}(f)$ is inclusion-wise minimal, i.e., there is no nonzero circulation f' in G with $\mathrm{supp}(f') \subsetneq \mathrm{supp}(f)$.

The following motivates our definitions of simple objects, and in particular simple conservative objects. We postpone the proof to the appendix.

Lemma 4.18. *A flow is an elementary circulation if and only if it is induced by a simple conservative object.*

Let $\mathcal{E}$ denote the collection of simple conservative objects. For any $e \in E$ and collection $\mathcal{U}$ of conservative objects, we define $x_e^{\mathcal{U}} := \bigvee_{U\in\mathcal{U}} x_e^U$. The following is essentially Lemma 2.9 for this setting, taking into account the scaling necessary to make $x^U(\lambda)$ feasible for cost bound λ.

Lemma 4.19. *Fix any edge $e \in E$. Suppose that $\mathcal{D}$ is a collection of k-recurrent conservative objects that dominate $\mathcal{E}$ at e, in that $x_e^{\mathcal{D}} \geq \alpha x_e^{\mathcal{E}}$ for some constant α. Then $\mathrm{SLC}_{\alpha/(m^2k)}(x_e^{\mathrm{m}}) \leq |\mathcal{D}|$.*

Proof. Consider the collection $\mathcal{D}$ guaranteed by the previous theorem. For any λ, $x^{\mathrm{m}}(\lambda)$ can be decomposed into at most m elementary circulations: $x^{\mathrm{m}}(\lambda) = \sum_{i=1}^r x^{(i)}$, for some $r \leq m$. Each elementary circulation $x^{(i)}$ is a flow induced by a simple object $C^{(i)} \in \mathcal{E}$, and $x^{(i)} \leq x^{C^{(i)}}(\lambda)$ (it is an inequality rather than an equality due to the ℓ_∞ cost bound in the definition of $x^{C^{(i)}}(\lambda)$). Hence $x_e^{\mathcal{E}} \geq x_e^{\mathrm{m}}/m$. Since all circuits in $\mathcal{D}$ are k-recurrent, $x^C(\lambda)/(mk)$ is a feasible generalized circulation of cost at most λ, for every $C \in \mathcal{D}$. Thus $\frac{1}{mk}x_e^{\mathcal{D}}(\lambda) \leq x_e^{\mathrm{m}}(\lambda)$ for all λ.

So

$$\frac{\alpha}{m^2k}x_e^{\mathrm{m}} \leq \frac{\alpha}{mk}x_e^{\mathcal{E}} \leq \frac{1}{mk}x_e^{\mathcal{D}} \leq x_e^{\mathrm{m}},$$

showing that $\mathrm{SLC}_{\alpha/(m^2k)}(x_e^{\mathrm{m}}) \leq |\mathcal{D}| + 1$. □

As such, our goal is now to demonstrate such a dominating collection $\mathcal{D}$; we will do this with $|\mathcal{D}| = O(m\bar{m})$ and $k = O(n)$.

4.5 Path domination

The results in this section are the core of our analysis. They concern a strong "bivariate" notion of domination.

Definition 4.20. Given two distinct nodes s and t, and an s-t-walk W, let

$$\vec{f}_W(\lambda, r) := \min\big(\nabla_t x^W(\lambda), \gamma(W)r\big). \tag{9}$$

In other words, $\vec{f}_W(\lambda, r)$ is the maximum amount of flow that can be sent to t with a flow induced by W, given that each step respects the cost and capacity bounds, and that there are only r units available at s to be sent.

Similarly, let

$$\overleftarrow{f}_W(\lambda, r) := \min\big(-\nabla_s x^W(\lambda), \gamma(W)^{-1}r\big);$$

this is the maximum amount of flow we can send from s subject to the cost and capacity bounds, given that at most r units arrive at t.

The function $\vec{f}_W$ for a given s-t-walk W is a 1-simple function. We can write

$$\vec{f}_W(\lambda, r) = \min\{\lambda/\mathrm{cost}(W), \gamma(W)r, \mathrm{limit}(W)\},$$

where cost(W) is the largest cost of a step of the walk per unit of flow measured at t; $\gamma(W)$ is the gain of the walk; and limit(W) is the maximum amount of flow that can arrive at t given that each step respects the capacity. Similarly, $\overleftarrow{f}_W$ is 1-simple.

Definition 4.21. We say that an s-t-walk W' *dominates* an s-t-walk W if $\gamma(W') \geq \gamma(W)$ and $\nabla_t x^{W'} \geq \nabla_t x^W$. We say that W' *strongly dominates* W if in addition, $-\nabla_s x^{W'} \leq -\nabla_s x^W$.

If W' dominates W, then by (9), clearly $\vec{f}_{W'} \geq \vec{f}_W$; indeed this is equivalent.

Write $\mathcal{P}(s, t)$ for the set of all s-t-paths for any distinct $s, t \in V$. Given a collection $\mathcal{W}$ of s-t-walks, define

$$\vec{f}_{\mathcal{W}} := \bigvee_{W \in \mathcal{W}} \vec{f}_W,$$

and similarly for $\overleftarrow{f}_{\mathcal{W}}$. The main theorem of this section is the following. It shows that the collection of s-t-paths can be dominated by a small collection of n-recurrent s-t-walks: for any s-t-path P, any cost bound λ, and any amount of flow r available at s, there is a walk in the collection that does a better job at sending flow to t under the same cost and flow restrictions.

Theorem 4.22. *Fix distinct nodes s and t. Then there is an $O(m\bar{m})$-sized collection $\mathcal{W}$ of n-recurrent s-t-walks such that for every $P \in \mathcal{P}(s, t)$, there is a walk $W \in \mathcal{W}$ that strongly dominates P. Hence*

$$\vec{f}_{\mathcal{W}} \geq \vec{f}_{\mathcal{P}(s,t)}.$$

The additional strength of strong domination over domination is not needed to deduce $\vec{f}_{\mathcal{W}} \geq \vec{f}_{\mathcal{P}(s,t)}$ in the above, but will be of utility in Section 4.8.

We also have the following complementary statement; this will follow easily from Theorem 4.22.

Theorem 4.23. *Fix distinct nodes s and t. Then there is an $O(m\bar{m})$-sized collection $\mathcal{W}$ of n-recurrent s-t-walks such that*

$$\overleftarrow{f}_{\mathcal{W}} \geq \overleftarrow{f}_{\mathcal{P}(s,t)}.$$

The remainder of this subsection will be devoted to proving Theorem 4.22 and Theorem 4.23.

Definition 4.24 (Bottlenecks, signature and backbone of a walk)**.** Consider an s-t walk W with at least one step, and let $\tilde{x}$ be a flow induced by a splitting $\tilde{W}$.

Define the *cost bottleneck step* of $\tilde{W}$ to be the arc $a_c \in E(\tilde{W})$ for which $c_{a_c}\tilde{x}_{a_c}$ is maximal, breaking ties towards steps closer to t. The *cost bottleneck* of W is then the arc $e_c \in E$ that corresponds to a_c. Similarly, define the *flow bottleneck step* of $\tilde{W}$ to be the arc $a_f \in E(\tilde{W})$ for which $\tilde{x}_{a_f}/u_{a_f}$ is maximal, breaking ties towards arcs closer to t; *exceptionally*, if all arcs of W have infinite capacity, set $a_f = a_c$. Again the *flow bottleneck* of W is the arc $e_f \in E$ that corresponds to a_f.

The *signature* of W is

$$\sigma(W) := \begin{cases} (e_c, e_f, \leq), & \text{if } a_c \text{ is earlier in the walk than } a_f, \text{ or } a_c = a_f \\ (e_c, e_f, >), & \text{otherwise.} \end{cases}$$

The *backbone* of W, denoted $\beta(W)$, is the subwalk of W that starts and ends with the bottleneck steps (including the bottleneck steps). We also write $\tau(W)$ for the subwalk of W before $\beta(W)$, and $\eta(W)$ for the subwalk after $\beta(W)$; that is,

$$W = \tau(W) \oplus \beta(W) \oplus \eta(W).$$

If $\tilde{W}$ is a splitting of W, we also define the precisely corresponding partition into subtrails,

$$\tilde{W} = \tau(\tilde{W}) \oplus \beta(\tilde{W}) \oplus \eta(\tilde{W}).$$

For any walk W, it is easy to verify that $\sigma(\beta(W)) = \sigma(W)$.

Definition 4.25. We say that a path P is *σ-capped* if $\sigma(P) = \sigma$ and $\beta(P) = P$.

For each signature σ, let $S(\sigma)$ be any highest gain path amongst all σ-capped paths.

Given an s-t-walk W with signature σ, define patch(W) to be the s-t-walk obtained from W by replacing $\beta(W)$ with $S(\sigma)$. (Note that we do not care about computing patch(W); all of this is purely existential.)

Lemma 4.26 (Patching a walk)**.** *Suppose W is an s-t walk whose backbone $\beta(W)$ is a path, and let $W' =$* patch(W)*. Then W' strongly dominates W. Furthermore, if $\sigma(W') \neq \sigma(W)$, then $\eta(W')$ is a* strict *suffix of $\eta(W)$.*

Proof. Let $\sigma = \sigma(W)$, and suppose that $\sigma = (e_c, e_f, \leq)$; the case $\sigma = (e_c, e_f, >)$ will be completely analogous, swapping the roles of the cost and flow bottlenecks. We may assume that $e_c \neq e_f$, since otherwise there is a unique path with first arc e_c and final arc e_f, ensuring that $S(\sigma) = \beta(W)$ and hence $W' = W$, meaning that there is nothing to prove. Recalling that $e_f = e_c$ whenever there is no finite capacity arc, we in particular can assume that W contains a finite capacity arc going forward.

Let $\tilde{W}$ be a splitting of W, and $\tilde{W}'$ a corresponding splitting of W', in the sense that each step of W not on the backbone is represented by the same arc in $\tilde{W}$ and $\tilde{W}'$. Let $\tilde{W} = \tilde{\tau} \oplus \tilde{\beta} \oplus \tilde{\eta}$ and $\tilde{W}' = \tilde{\tau} \oplus \tilde{S} \oplus \tilde{\eta}$, where $\tilde{\eta}$, $\tilde{\tau}$, $\tilde{\beta}$ and $\tilde{S}$ are splittings of $\eta(W)$, $\tau(W)$, $\beta(W)$ and $S(\sigma)$ respectively. Write a_c and a_f for the cost and flow bottleneck steps of $\tilde{W}$ (so a_c is a copy of e_c in $\tilde{G}$, and a_f a copy of e_f). Similarly, let a_c' and a_f' be the cost and flow bottleneck steps of $\tilde{W}'$. Since $S(\sigma)$ has signature σ, $a_c' \notin E(\tilde{S}) \setminus \{a_c\}$ and $a_f' \notin E(\tilde{S}) \setminus \{a_f\}$. As $\beta(W)$ is a σ-capped path, the definition of $S(\sigma)$ implies that $\gamma(S(\sigma)) \geq \gamma(\beta(W))$, and hence $\gamma(W') \geq \gamma(W)$.

Consider some fixed $\lambda \geq 0$. Let $\tilde{x} := x^{\tilde{W}}(\lambda)$ and $\tilde{y} := x^{\tilde{W}'}(\lambda)$. We wish to show that $\nabla_t \tilde{y} \geq \nabla_t \tilde{x}$ and $-\nabla_s \tilde{y} \leq -\nabla_s \tilde{x}$. Since $\tilde{W}$ and $\tilde{W}'$ have identical trails from s to a_c, and from a_f to t, it suffices to show that

$$\tilde{y}_{a_f} \geq \tilde{x}_{a_f} \quad \text{and} \quad \tilde{y}_{a_c} \leq \tilde{x}_{a_c}. \tag{10}$$

Define $\tilde{z}$ to be the flow induced by $\tilde{W}'$ scaled so that $\tilde{z}_{a_f} = \tilde{x}_{a_f}$. Since $\gamma(\tilde{S}) \geq \gamma(\tilde{\beta})$, $\tilde{z}_{a_c} \leq \tilde{x}_{a_c}$. This means that in addition $\tilde{z}_a \leq \tilde{x}_a$ for all $a \in E(\tilde{W}') \setminus E(\tilde{S})$. In particular this holds on a_c' and a_f', and so $c_{a_c'} \tilde{z}_{a_c'} \leq c_{a_c'} \tilde{x} \leq \lambda$ and $\tilde{z}_{a_f'} \leq \tilde{x}_{a_f'} \leq u_{a_f'}$. Hence $\tilde{z} \leq \tilde{y}$; $\tilde{z}$ satisfies the capacity and cost bounds, so $\tilde{y}$ can only be an induced flow with larger scaling.

We consider two cases:

- Suppose $\tilde{x}$ is constrained by the flow bottleneck a_f, i.e., $\tilde{x}_{a_f} = u_{a_f}$. Then since $\tilde{y} \geq \tilde{z}$ and $\tilde{z}_{a_f} = u_{a_f}$, $\tilde{y} = \tilde{z}$. So (10) holds: $\tilde{y}_{a_f} = \tilde{x}_{a_f}$ and $\tilde{y}_{a_c} \leq \tilde{x}_{a_c}$. Further, a_f must be the flow bottleneck step of $\tilde{W}'$; it is at capacity, and $\tilde{y}_a = \tilde{x}_a < u_a$ for all arcs $a \in E(\tilde{\eta})$.
- Suppose $\tilde{x}$ is constrained by the cost bottleneck a_c, i.e., $c_{a_c} \tilde{x}_{a_c} = \lambda$. Since $c_{a_c} \tilde{y}_{a_c} \leq \lambda$, $\tilde{y}_{a_c} \leq \tilde{x}_{a_c}$. We also have $\tilde{y}_{a_f} \geq \tilde{z}_{a_f} = \tilde{x}_{a_f}$. So (10) holds. Further, a_c' cannot be an arc of $\tilde{\tau}$, since $c_a \tilde{y}_a \leq c_a \tilde{x}_a \leq \lambda$ for all $a \in E(\tilde{\tau})$, with a strict inequality if $c_{a_c} \tilde{y}_{a_c} < \lambda$ (since then $\tilde{y}_a < \tilde{x}_a$ for all $a \in \tilde{\tau}$).

The first of the above cases must occur for sufficiently large λ, since W has a finite capacity arc. Thus $a_f' = a_f$. If at least one arc of W has strictly positive cost, then for sufficiently small positive λ, the cost bottleneck constrains the flow, and the second case occurs. Then we have the claim that $a_c' \in E(\tilde{\eta}) \cup \{a_c\}$. But if all arcs of W have zero cost, then all arcs of W' also have zero cost, and so e_c' and e_c are both the last arcs in W and W', and hence the same. Either way, we deduce that if $\sigma(W') \neq \sigma(W)$, then $\sigma(W') = (a_c', a_f, >)$, and hence $\eta(W')$ is a strict suffix of $\eta(W)$. □

Proof of Theorem 4.22. For any signature σ, say that a walk $\bar{L}$ is a *left σ-extension* if it starts from s, contains $S(\sigma)$ as a suffix, has signature σ, and $\bar{L} = \tau(\bar{L}) \oplus S(\sigma)$. In other words, if we consider a splitting $\tilde{L}$ of $\bar{L}$, it contains a splitting $\tilde{S}$ of $S(\sigma)$ as its suffix, and the cost and flow bottleneck steps are the first and last arcs of $\tilde{S}$ (in the order specified by the signature). We similarly define a walk $\bar{R}$ to be a *right σ-extension* if it ends at t, contains $S(\sigma)$ as a prefix, has signature σ, and $\bar{R} = S(\sigma) \oplus \eta(\bar{R})$.

Now define, for any signature σ, $L(\sigma)$ to be a highest-gain $(n-2)$-recurrent walk such that $L(\sigma) \oplus S(\sigma)$ is a left σ-extension, as long as at least one such walk exists; if not, $L(\sigma)$ is undefined. Similarly, let $R(\sigma)$ be a highest-gain *path* such that $S(\sigma) \oplus R(\sigma)$ is a right σ-extension, or undefined if there are none such. Let Σ be the collection of all signatures for which $L(\sigma)$ and $R(\sigma)$ are both defined; note that

$$\Sigma \subseteq (E \times F \times \{\leq, >\}) \cup \{(e, e, \leq) : e \in E \setminus F\},$$

taking into account the possibility of walks consisting only of infinite capacity arcs.

Now define

$$W(\sigma) := L(\sigma) \oplus S(\sigma) \oplus R(\sigma) \quad \text{for each } \sigma \in \Sigma, \tag{11}$$

and

$$\mathcal{W} := \{W(\sigma) : \sigma \in \Sigma\}.$$

It is easy to see that $\sigma(W(\sigma)) = \sigma$. Clearly, $|\mathcal{W}| \leq |\Sigma| \leq m(2\bar{m} + 1)$.

Call an s-t-walk W *stable* if $\eta(W)$ is a path and $\sigma(\text{patch}(W)) = \sigma(W)$.

Claim 4.27. *Let W be a stable s-t walk, with $\tau(W)$ being $(n-3)$-recurrent. Then there exists a walk $W' \in \mathcal{W}$ which strongly dominates W.*

Proof. Let $\sigma = \sigma(W)$. Since $\tau(W) \oplus \beta(W) = \tau(W) \oplus S(\sigma)$ is itself a left σ-extension, and similarly $\beta(W) \oplus \eta(W)$ a right σ-extension, $\sigma \in \Sigma$. So choose $W' = W(\sigma)$; note that $\sigma(W') = \sigma$. Let $Q = \text{patch}(W)$; since Q strongly dominates W by Lemma 4.26, it suffices to show that W' strongly dominates Q. Since $\tau(W)$ is $(n-3)$-recurrent, $\tau(Q)$ is $(n-2)$-recurrent, and since W is stable, $\sigma(Q) = \sigma$.

Comparing Q to W', we observe that $\gamma(\tau(W')) \geq \gamma(\tau(Q))$, $\gamma(\eta(W')) \geq \gamma(\eta(Q))$, and they share $S(\sigma)$. This all follows by the definition of $W(\sigma)$, the definition of patching, and the fact that $\tau(Q)$ is $(n-2)$-recurrent and $\eta(Q)$ is a path. Let $\tilde{Q}$ and $\tilde{W}'$ be splittings of Q and W' respectively, chosen so that their common part $S(\sigma)$ has the same splitting $\tilde{S}$.

Fix any λ, and consider the flows $\tilde{x} := x^{\tilde{Q}}(\lambda)$ and $\tilde{y} := x^{\tilde{W}'}(\lambda)$. Then $\tilde{x}_a = \tilde{y}_a$ for all $a \in E(\tilde{S})$, given that the bottleneck steps are identical and form the endpoints of the common path $\tilde{S}$. Since $\gamma(\eta(\tilde{W}')) \geq \gamma(\eta(\tilde{Q}))$, $\nabla_t \tilde{y} \geq \nabla_t \tilde{x}$. Since $\gamma(\tau(\tilde{W}')) \geq \gamma(\tau(\tilde{Q}))$, $-\nabla_s \tilde{y} \leq -\nabla_s \tilde{x}$. So W' strongly dominates Q, and hence W. □

Now consider any path $W^{(1)} \in \mathcal{P}(s,t)$. If $\beta(W^{(1)})$ has at most 2 arcs, then necessarily $S(\sigma(W^{(1)})) = \beta(W^{(1)})$, since there is a unique $\sigma(W^{(1)})$-capped path. So in this case, $W^{(1)}$ is stable. Otherwise, $|E(\eta(W^{(1)}))| \leq n-4$. Construct the sequence $W^{(1)}, W^{(2)}, \dots, W^{(\ell)}$ by setting $W^{(i+1)} = \text{patch}(W^{(i)})$, stopping once we reach a stable walk. If $W^{(i)}$ is not stable, then $\eta(W^{(i+1)})$ is a strict subwalk of $\eta(W^{(i)})$; this means that we maintain the property that $\eta(W^{(i)})$ is a path, and furthermore, $|E(\eta(W^{(i)}))| \leq n - i - 3$. So we must reach a stable walk, and in fact $\ell \leq n-3$.

If $\tau(W^{(i)})$ is k-recurrent, then $\tau(W^{(i+1)})$ is $(k+1)$-recurrent, since it is obtained by appending some part of $\beta(W^{(i)})$ to $\tau(W^{(i)})$. Since $\tau(W^{(1)})$ is 1-recurrent and $\ell \leq n-3$, $\tau(W^{(\ell)})$ is $(n-3)$-recurrent.

By Claim 4.27 applied to $W^{(\ell)}$, the walk $W' = W(\sigma(W^{(\ell)})) \in \mathcal{W}$ strongly dominates $W^{(\ell)}$. By Lemma 4.26, $W^{(i+1)}$ strongly dominates $W^{(i)}$ for each $i < \ell$. Strong domination is clearly transitive, and so W' does indeed strongly dominate $W^{(1)}$.

For each $P \in \mathcal{P}(s,t)$, there is a $W \in \mathcal{W}$ which dominates P, meaning $\vec{f}_W \geq \vec{f}_P$. Hence $\vec{f}_{\mathcal{W}} \geq \vec{f}_{\mathcal{P}(s,t)}$. □

Finally, we deduce the complementary version.

Proof of Theorem 4.23. Let $\overleftarrow{G}$ be the instance obtained from G by replacing each arc e with the reverse arc $\overleftarrow{e}$, where $\gamma_{\overleftarrow{e}} = 1/\gamma_e$, $u_{\overleftarrow{e}} = \gamma_e u_e$ and $c_{\overleftarrow{e}} = c_e/\gamma_e$; note that we do *not* flip the sign of the costs, which remain nonnegative. Now apply Theorem 4.22 to $\overleftarrow{G}$ and the collection $\overleftarrow{\mathcal{P}}$ of paths from t to s in $\overleftarrow{G}$, obtaining a dominating collection $\overleftarrow{\mathcal{W}}$; let $\mathcal{W}$ consist of the reversals of all walks in $\overleftarrow{\mathcal{W}}$.

Then for every $P \in \mathcal{P}(s,t)$, its reversal $\overleftarrow{P} \in \overleftarrow{\mathcal{P}}$ is dominated by some $\overleftarrow{W} \in \overleftarrow{\mathcal{W}}$. Let W be the reversal of $\overleftarrow{W}$. Then

$$\gamma(W) = \gamma(\overleftarrow{W})^{-1} \leq \gamma(\overleftarrow{P})^{-1} = \gamma(P).$$

Further, it is easy to see that $x^{\overleftarrow{W}}_{\overleftarrow{e}}(\lambda) = \gamma_e x^W_e(\lambda)$ for every e and λ. Thus

$$\nabla_s x^W(\lambda) = -\nabla_s x^{\overleftarrow{W}}(\lambda) \leq -\nabla_s x^{\overleftarrow{P}}(\lambda) = \nabla_s x^P(\lambda)$$

for every λ. So $\overleftarrow{f}_W \geq \overleftarrow{f}_P$, from the definition of $\overleftarrow{f}$, and hence $\overleftarrow{f}_{\mathcal{W}} \geq \overleftarrow{f}_{\mathcal{P}(s,t)}$. □

4.6 Weak domination bounds for non-conservative objects

While our goal is to dominate simple conservative objects, we build up to this in stages. Our next step will be building small collections that dominate flow-generating objects and flow-absorbing objects; these will become building blocks in the next section.

Given U, a flow-generating object at t, we will be interested in $\nabla x_t^U(\lambda)$, the maximum amount of flow that can be generated at t using U, subject to the cost and capacity bounds. So define

$$f_U^+(\lambda) := \nabla x_t^U(\lambda) \quad \text{for all } \lambda \geq 0,$$

and as usual, $f_{\mathcal{U}}^+ := \bigvee_{U \in \mathcal{U}} f_U^+$ for a collection of flow-generating objects at t. Given two flow-generating objects at t, U and U', we will say that U' dominates U if $f_{U'}^+ \geq f_U^+$; and similarly for collections of objects.

For U a flow-absorbing object at s and $\mathcal{U}$ a collection of such, we define the analogous notion

$$f_U^-(\lambda) := -\nabla x_s^U(\lambda) \qquad \text{and} \qquad f_{\mathcal{U}}^- := \bigvee_{U \in \mathcal{U}} f_U^-.$$

Flow-generating cycles through s. Given an s-s-walk W, define $f_W^+(\lambda) := \nabla x_s^W(\lambda)$; that is, $f_W^+ = f_{(W,\{s\})}^+$. Define $f_{\mathcal{W}}^+$ for a collection of s-s-walks $\mathcal{W}$ in the usual way as $f_{\mathcal{W}}^+ := \bigvee_{W \in \mathcal{W}} f_W^+$. Let $C^+(s)$ denote the collection of flow-generating cycles at s.

Theorem 4.28. *There is an $O(m\bar{m})$-sized collection $\mathcal{W}$ of $(n+1)$-recurrent flow-generating s-s-walks such that*

$$f_{\mathcal{W}}^+ \geq f_{C^+(s)}^+.$$

Proof. Let G' be the instance obtained from G by adding a new node s' and redirecting all the incoming arcs of s to s'; so $V(G') = V(G) \cup \{s\}$, and we can identify $E(G')$ with $E(G)$. The latter provides a natural correspondence between s-s walks in G and s-s' walks in G'.

By Theorem 4.22 applied to G' and the collection $\mathcal{P}'$ of s-s' paths, there exists an $O(m\bar{m})$-sized collection $\mathcal{W}'$ of $(n+1)$-recurrent s-s' walks that strongly dominate $\mathcal{P}'$ in G'. Let $\mathcal{W}$ be the collection of $(n+1)$-recurrent s-s walks in G corresponding to $\mathcal{W}'$.

Given any $C \in C^+(s)$, consider the path $P \in \mathcal{P}'$ corresponding to C. Choosing $W' \in \mathcal{W}'$ that strongly dominates P, we have that $\nabla_{s'} x^{W'} \geq \nabla_{s'} x^P$ and $-\nabla_s x^{W'} \leq -\nabla_s x^P$ (all flows viewed in G'). With $W \in \mathcal{W}$ corresponding to W', it follows that

$$\nabla_s x^W = \nabla_{s'} x^{W'} + \nabla_s x^{W'} \geq \nabla_{s'} x^P + \nabla_s x^P = \nabla_s x^C,$$

and so $f_W^+ \geq f_C^+$. $\square$

Flow-generating objects at t. For given nodes s and t, let $\mathcal{G}(s,t)$ denote the collection of simple flow-generating objects at t consisting of a flow-generating cycle through s, followed by an s-t-path. Let $\mathcal{G}(t) := \bigcup_{s \in V} \mathcal{G}(s,t)$, be the collection of all simple flow-generating objects at t.

Lemma 4.29. *For any $s, t \in V$, there is an $O(m\bar{m})$-sized collection $\mathcal{H}$ of $O(n)$-recurrent objects, each consisting of a flow-generating cycle at s along with an s-t walk, such that*

$$f_{\mathcal{H}}^+ \geq f_{\mathcal{G}(s,t)}^+.$$

Proof. Assume $s \neq t$, since otherwise the claim is immediate from Theorem 4.28.

Let $\mathcal{R}$ be the collection of flow-generating s-s walks guaranteed by Theorem 4.28. Let $\mathcal{W}$ be the collection of s-t-walks guaranteed by Theorem 4.22, applied to $\mathcal{P}(s,t)$. Let $\mathcal{U} := \{(R, W) : R \in \mathcal{R}, W \in \mathcal{W}\}$ be the collection of flow-generating objects obtained by taking all possible combinations; so $|\mathcal{U}| = O(m^2\bar{m}^2)$.

We now observe that for any flow-generating s-s-walk R and any s-t-walk W, if we define the flow-generating object $U = (R, W)$ we have

$$f_U^+(\lambda) = \vec{f}_W(\lambda, f_R^+(\lambda)).$$

As a consequence,

$$f_{\mathcal{G}(s,t)}^+(\lambda) = \vec{f}_{\mathcal{P}(s,t)}\big(\lambda, f_{C^+(s)}^+(\lambda)\big) \qquad \text{and} \qquad f_{\mathcal{U}}^+(\lambda) = \vec{f}_{\mathcal{W}}\big(\lambda, f_{\mathcal{R}}^+(\lambda)\big).$$

It follows that $f_{\mathcal{U}}^+ \geq f_{\mathcal{G}(s,t)}^+$.

This shows a $|\mathcal{U}| = O(m^2\bar{m}^2)$-sized dominating set. We now improve this to $O(m\bar{m})$. By Lemma 4.6, using that $\vec{f}_{\mathcal{W}}$ and $f_{\mathcal{R}}^+$ are both $O(m\bar{m})$-simple, $f_{\mathcal{U}}^+$ is $O(m\bar{m})$-simple. The existence of the desired $\mathcal{H} \subseteq \mathcal{U}$ follows by Lemma 4.9. Note that since the objects in $\mathcal{R}$ and $\mathcal{W}$ are $(n+1)$-recurrent and n-recurrent respectively, the objects in $\mathcal{H}$ are $(2n+1)$-recurrent. $\square$

This immediately gives us a dominating set of size $O(nm\bar{m})$ for $\mathcal{G}(t)$.

Theorem 4.30. *For any $t \in V$, there is an $O(nm\bar{m})$-sized collection $\mathcal{H}$ of $O(n)$-recurrent flow-generating objects at t such that*

$$f^+_{\mathcal{H}} \geq f^+_{\mathcal{G}(t)}.$$

Proof. Immediate from the previous theorem; simply take the union of the dominating collections for $\mathcal{G}(s,t)$ for each s (or in other words, apply Lemma 4.7). □

The above, along with the next section, suffices to obtain a polynomial SLC bound for x^{m}_e, but not for the claimed $O(m\bar{m})$ bound. For this we need the following stronger result, the proof of which we delay until Section 4.8.

Theorem 4.31. *For any $t \in V$, there is an $O(m\bar{m})$-sized collection $\mathcal{H}$ of $O(n)$-recurrent flow-generating objects at t such that*

$$f^+_{\mathcal{H}} \geq f^+_{\mathcal{G}(t)}.$$

Absorbing versions. A completely symmetric version of the above concerns, instead of the maximum excess we can generate at t, the maximum deficit we can create at t. Let $\mathcal{A}(t)$ the collection of all simple flow-absorbing objects at t.

Theorem 4.32. *For any $t \in V$, there is an $O(nm\bar{m})$-sized collection $\mathcal{B}$ of $O(n)$-recurrent flow-absorbing objects at t such that*

$$f^-_{\mathcal{B}} \geq f^-_{\mathcal{A}(t)}.$$

Proof. Just as with the proof of Theorem 4.23, the claim can be obtained simply by applying Theorem 4.30 to the reversed instance $\overleftarrow{G}$. □

4.7 Dominating simple conservative objects

We are now ready to prove the main domination theorem for an arc.

Theorem 4.33. *Suppose that K is such that for every node s, there exist dominating collections of $O(n)$-recurrent flow-generating objects at s, and $O(n)$-recurrent flow-absorbing objects at s, each of size at most K.*

Then for any arc e, there is an $O(m\bar{m} + K)$-sized collection $\mathcal{D}$ of $O(n)$-recurrent conservative objects such that $x^{\mathcal{D}}_e \geq x^{\mathcal{E}}_e/8$.

Combined with Theorem 4.30 and Theorem 4.32 this gives a bound of $O(nm\bar{m})$ on the SLC. The final ingredient for the stronger $O(m\bar{m})$ follows in the next section.

Before giving the full proof, let us give a high-level overview, only discussing what is needed to get a strongly polynomial bound.

Fix an arc e, with tail s and head t. Consider first the collection $\mathcal{E}_p \subseteq \mathcal{E}$ of simple conservative objects $C = (C_g, C_p, C_a)$ in which $e \in E(C_p)$. Then C can be viewed as the composition of a flow-generating object $U_g = (C_g, P_1)$ at s, the arc e, and a flow-absorbing object $U_a = (P_2, C_a)$ at t (so $C_p = P_1 \oplus e \oplus P_2$). But this suggests a straightforward choice of a dominating collection: take $\mathcal{H}$ to be a small collection of flow-generating objects at s dominating all simple flow-generating objects at s; similarly take $\mathcal{B}$ to be a small collection of flow-absorbing objects at t dominating (in the sense of Theorem 4.32) all simple flow-absorbing objects at t; and define a collection $\mathcal{Q}^{(1)}$ of conservative objects by $\mathcal{Q}^{(1)} = \{U'_g \oplus e \oplus U'_a : U'_g \in \mathcal{H}, U'_a \in \mathcal{B}\}$. This does the job; for any cost bound λ, there is some $U'_g \in \mathcal{H}$ that can create at least as much excess at s as U_g; similarly some $U'_a \in \mathcal{B}$ can get rid of at least as much flow at t as U_a; and so $U'_g \oplus e \oplus U'_a \in \mathcal{Q}^{(1)}$ can send at least as much flow through e as C.

The case where e lies on the flow-generating or flow-absorbing cycle is more involved. In general, the flow-generating and flow-absorbing cycles of a circuit may overlap (recall Definition 4.11), but let us ignore this complication here. Further, the place where the flow-generating cycle and connecting path of a conservative object meet turns out not to be too important. So just for concreteness, let $\mathcal{E}_g \subseteq \mathcal{E}$ be the collection of simple conservative objects $C = (C_g, C_p, C_a)$ in which $e \in E(C_g)$, $e \notin E(C_a)$, and C_p and C_g overlap at s. We consider how to dominate $\mathcal{E}_g$.

The dominating set $\mathcal{Q}^{(1)}$ described above does not suffice, for the following reason. Suppose $C = (C_g, C_p, C_a) \in \mathcal{E}_g$ has the property that the flow-generating cycle C_g has gain very close to 1. Then, for some value of λ, it may be possible to send a very large amount of flow through e using C_g, while creating only a very small amount of excess at s. Perhaps getting rid of excess is very expensive: we can only get rid of some small amount of flow at s, far less than the amount of flow C_g sends through e. So we cannot hope to dominate C by an object that generates lots of excess at s, sends it through e, and then gets rid of it.

Instead, let us look more carefully at cycles through e. If we remove e from any cycle through e, what remains is a t-s path P. Theorem 4.22 provides us with a small dominating collection $\mathcal{W}_1$ of t-s walks — dominating in the sense that $\vec{f}_{\mathcal{W}_1} \geq \vec{f}_{\mathcal{P}(t,s)}$. The complementary version Theorem 4.23 provides us with a second collection $\mathcal{W}_2$ of t-s walks that dominate $\mathcal{P}(t,s)$ in the sense that $\overleftarrow{f}_{\mathcal{W}_2} \geq \overleftarrow{f}_{\mathcal{P}(t,s)}$. We can construct a collection of closed walks through e by adding e to each of the walks in $\mathcal{W}_1$ and $\mathcal{W}_2$. Some of these may be flow-generating walks; we can collect all these together into a small collection $\mathcal{H}_2$ of flow-generating objects at s. We can also collect together those that are flow-absorbing walks, and hence construct a small collection $\mathcal{B}_2$ of flow-absorbing objects at t. Some could also yield conservative cycles; these turn out to be even better for us, so assume there are none.

Now for our fixed $C = (C_g, C_p, C_a) \in \mathcal{E}_g$, with $C_g = e \oplus P$, two things can happen. If we are lucky, the walk $W_2 \in \mathcal{W}_2$ with $\overleftarrow{f}_{W_2} \geq \overleftarrow{f}_P$ has the property that $R_2 := e \oplus W_2$ is a flow-generating cycle. Complementary domination works perfectly for us; it tells us that on W_2, can always send flow such that the same amount of flow arrives at the head s of the path as with a flow on P, but *the same or more* leaves from t compared to the flow on P. This yields the same amount of flow on e, but with less excess at t to dispose of. This excess we can dispose of using one of our flow-absorbing objects in $\mathcal{B}_2$.

The more difficult case is that R_2 is a flow-absorbing cycle. In this case, we look also at the walk $W_1 \in \mathcal{W}_1$ with $\vec{f}_{W_1} \geq \vec{f}_P$; then $R_1 := e \oplus W_1$ is certainly a flow-generating cycle, since $\gamma(W_1) \geq \gamma(P)$. It turns out that the conservative object $(R_1, \{s\}, R_2)$ does the job.

We now proceed with the detailed proof.

Proof of Theorem 4.33. Let $e \in E$, with tail s and head t, and $\mathcal{E}' \subseteq \mathcal{E}$ the collection of simple conservative objects containing e. We will proceed in two stages. In the first stage, we will construct an $O((K + m\bar{m})^2)$-sized collection $\mathcal{Q}$ of $O(n)$-recurrent conservative objects that dominates $\mathcal{E}$, in that $x_e^{\mathcal{Q}} \geq x_e^{\mathcal{E}}/4$. In the second stage, we will choose a smaller $O(m\bar{m})$-sized collection $\mathcal{D} \subseteq \mathcal{Q}$, and show that $x_e^{\mathcal{D}} \geq x_e^{\mathcal{Q}}/2$.

We first describe $\mathcal{Q}$. We will construct an $O(K + m\bar{m})$-sized collection $\mathcal{H}$ of flow-generating objects at s, an $O(K + m\bar{m})$-sized collection $\mathcal{B}$ of flow-absorbing objects at t, and an $O(K + m\bar{m})$-sized collection $\mathcal{Q}_2$ of conservative cycles through e. The collection $\mathcal{Q}$ is constructed from $\mathcal{Q}_2$, along with all possible ways of combining an object in $\mathcal{H}$ with arc e and an object in $\mathcal{B}$ to obtain a conservative object.

The collections $\mathcal{H}$ and $\mathcal{B}$ each consist of two parts. For $\mathcal{H}$, the first part $\mathcal{H}_1$ is a K-sized collection of $O(n)$−recurrent flow-generating objects dominating $\mathcal{G}(s)$, as we assume to exist. Similarly, $\mathcal{B}_1$ is a K-sized collection of $O(n)$−recurrent flow-absorbing objects dominating $\mathcal{A}(t)$.

To describe the second parts of these collections, we proceed as follows. Let $\mathcal{W}_1$ be a collection of n-recurrent walks of size $O(m\bar{m})$ that dominate $\mathcal{P}(t,s)$, in that $\vec{f}_{\mathcal{W}_1} \geq \vec{f}_{\mathcal{P}(t,s)}$, as guaranteed by Theorem 4.22. (Note that we consider t-s-paths, not s-t-paths.) Similarly, let $\mathcal{W}_2$ be a collection of n-recurrent walks of size $O(m\bar{m})$ for which $\overleftarrow{f}_{\mathcal{W}_2} \geq \overleftarrow{f}_{\mathcal{P}(t,s)}$. Now define

$$\begin{aligned} \mathcal{W}^+ &:= \{W \in \mathcal{W}_1 \cup \mathcal{W}_2 : \gamma(e \oplus W) > 1\}, \\ \mathcal{W}^- &:= \{W \in \mathcal{W}_1 \cup \mathcal{W}_2 : \gamma(e \oplus W) < 1\}, \quad \text{and} \\ \mathcal{W}^= &:= \{W \in \mathcal{W}_1 \cup \mathcal{W}_2 : \gamma(e \oplus W) = 1\}. \end{aligned}$$

Then

$$\mathcal{H}_2 := \{(e \oplus W^+, \{s\}) : W^+ \in \mathcal{W}^+\} \qquad \text{and} \qquad \mathcal{B}_2 = \{(\{t\}, W^- \oplus e) : W^- \in \mathcal{W}^-\},$$

with $\mathcal{H} := \mathcal{H}_1 \cup \mathcal{H}_2$ and $\mathcal{B} := \mathcal{B}_1 \cup \mathcal{B}_2$.

Define

$$\mathcal{Q}_1 := \{U_g \oplus e \oplus U_a : U_g \in \mathcal{H}, U_a \in \mathcal{B}\}.$$

Also define

$$\mathcal{Q}_2 := \{(e \oplus W, \{s\}, e \oplus W) : W \in \mathcal{W}^=\},$$

and finally, $\mathcal{Q} := \mathcal{Q}_1 \cup \mathcal{Q}_2$.

We now show that $x_e^{\mathcal{Q}} \geq x_e^{\mathcal{E}'}/4$, already providing a polynomial-sized collection dominating $\mathcal{E}$. Fix any $C \in \mathcal{E}'$ and $\lambda \geq 0$. Our goal is to find some $U \in \mathcal{Q}$ for which $x_e^U(\lambda) \geq x_e^C(\lambda)/4$.

Write $C = (C_g, C_p, C_a)$, where C_g is a flow-generating cycle, C_a a flow-absorbing cycle, and C_p a path (or possibly $C_g = C_a$ is a conservative cycle), with C_g intersecting C_p in a single node, and C_p intersecting C_a in a single node. Let $\bar{x} = x^C(\lambda)$. We can split $\bar{x}$ as

$$\bar{x} = \bar{x}^g + \bar{x}^p + \bar{x}^a,$$

where $\bar{x}^g$, $\bar{x}^p$ and $\bar{x}^a$ are flows induced by C_g, C_p and C_a respectively, all appropriately scaled.

Either $e \in E(C_p)$, or e lies on one or both of C_g and C_a. We consider three cases: $e \in E(P)$, $\bar{x}^g_e \geq \frac{1}{2}\bar{x}_e$, and $\bar{x}^a_e > \frac{1}{2}\bar{x}_e$.

Case 1: $e \in E(C_p)$**.** Write $C_p = P_1 \oplus e \oplus P_2$. Then $U_g := (C_g, P_1) \in \mathcal{G}(s)$, and $U_a := (P_2, C_a) \in \mathcal{A}(t)$. Choose $U'_g \in \mathcal{H}$ with $f^+_{U'_g}(\lambda) \geq f^+_{U_g}(\lambda)$, and $U'_a \in \mathcal{B}$ with $f^-_{U'_a}(\lambda) \geq f^-_{U_a}(\lambda)$. Then let $Z := U'_g \oplus e \oplus U'_a$; this is in $\mathcal{Q}$. Observe that

$$\bar{x}_e = x^C_e(\lambda) = \min\{f^+_{U_g}(\lambda), f^-_{U_a}(\lambda)/\gamma_e, u_e, \lambda/c_e\}; \tag{12}$$

the largest a flow induced by C can be scaled, while satisfying the capacity and cost bounds, is either determined by the step over e itself, or by the amount we can generate at s, or the amount we can destroy at t. Similarly,

$$x^Z_e(\lambda) = \min\{f^+_{U'_g}(\lambda), f^-_{U'_a}(\lambda)/\gamma_e, u_e, \lambda/c_e\}. \tag{13}$$

Since each term in (13) is at least as large as its corresponding term in (12), $x^Z_e(\lambda) \geq \bar{x}_e$, as needed.

Case 2: $\bar{x}^g_e \geq \frac{1}{2}\bar{x}_e$**.** In this case, $e \in E(C_g)$ (e might lie on C_a as well). Let i be the common node of C_g and C_p., from the definition of Let $C_g = P_1 \oplus e \oplus P_2$, where P_1 is the path in C_g from i to s, and P_2 the path in C_g from t to i.

Also let $P := P_2 \oplus P_1$, i.e., the t-s-path obtained by removing e from C_g. Let $W_1 \in \mathcal{W}_1$ dominate P, i.e., $\vec{f}_{W_1} \geq \vec{f}_P$. Similarly let $W_2 \in \mathcal{W}_2$ with $\overleftarrow{f}_{W_2} \geq \overleftarrow{f}_P$. Let $R_1 := e \oplus W_1$ and $R_2 := e \oplus W_2$.

Since W_1 dominates P, $\gamma(W_1) \geq \gamma(P)$, and so $\gamma(R_1) \geq \gamma(C_g) \geq 1$. Similarly, $\gamma(R_2) \leq \gamma(C_g)$. We distinguish a few subcases.

- $\gamma(R_1) = 1$ or $\gamma(R_2) = 1$. Suppose $\gamma(R_1) = 1$. If $x^{R_1}_e(\lambda) = u_e$ or $c_e x^{R_1}(\lambda) = \lambda$, then of course $x^{R_1}_e(\lambda) \geq \bar{x}^g_e$. Otherwise, $x^{R_1}_a(\lambda) = x^{W_1}_a(\lambda)$ for all $a \in E(W_1)$. Domination tells us that $\nabla_s x^{W_1}(\lambda) \geq \nabla_s x^P(\lambda)$. All flow in $x^{R_1}(\lambda)$ arriving at s through W_1 enters e, and so $x^{R_1}_e(\lambda) \geq \bar{x}^g_e$.

  Similarly, if $\gamma(R_2) = 1$, using that $\nabla_t x^{W_2}(\lambda) \geq \nabla_t x^P(\lambda)$, one obtains $x^{R_2}_e(\lambda) \geq \bar{x}^g_e$. So choosing j such that $\gamma(R_j) = 1$ and defining the conservative object $Z := (R_j, \{s\}, R_j) \in \mathcal{Q}$, we have that

  $$x^Z_e(\lambda) = x^{R_j}_e(\lambda) \geq \bar{x}^g_e \geq \bar{x}_e/2.$$

- $\gamma(R_1) \geq \gamma(R_2) > 1$. First, if C_g and C_a overlap, so that C_g and C_a are both cycles at i, we may choose i to be *any* common vertex of C_g and C_i; this has no effect on the flows induced by C. So if e lies on both C_g and C_a, we may assume that $i = s$; and otherwise, that i is the last node on the path on which C_g and C_a overlap.

  Consider the flow-absorbing object $U_a := (P_2 \oplus C_p, C_a)$. We observe that this object is simple. This is clear if C_g and C_a are disjoint, given that C is simple. If on the other hand they overlap, then P_2 only shared the vertex i with C_a, by our choice of i.

  Since $U_a \in \mathcal{A}(t)$, we can choose $U'_a = (P'_a, C'_a) \in \mathcal{B}$ dominating U_a. Then $Z := (R_2, e \oplus P'_a, C'_a) \in \mathcal{Q}$. We claim that $x^Z_e(\lambda) \geq \bar{x}^g_e$.

  First, we split $\bar{x}^g$ further into two parts: $\bar{x}^g = \bar{x}^1 + \bar{x}^2$, where $\bar{x}^1$ is a flow induced by $P \oplus e$, and $\bar{x}^2$ is a flow induced by P_2. (Note that $P \oplus e$ is the same cycle as C_g, but at t rather than at i.) The flow $\bar{x}^1$ has positive excess at t, and zero excess elsewhere; also $\bar{x}^1_e = \bar{x}^g_e$, since e is not in the support of $\bar{x}^2$.

  Consider the restriction $\bar{y}$ of $\bar{x}^1$ to P; this has excess $\bar{x}^1_e = \bar{x}^g_e$ at s, and some deficit $-\nabla_t \bar{y} \leq \gamma_e \bar{x}^g_e$ at t (since $P \oplus e$ is a flow-generating cycle). Since $\overleftarrow{f}_{W_2}(\lambda, \bar{x}^g_e) \geq \overleftarrow{f}_P(\lambda, \bar{x}^g_e)$, there is a flow $\bar{z}^g$ induced by W_2 where $\nabla_s \bar{z}^g = \bar{x}^g_e$, but $-\nabla_t \bar{z}^g \geq -\nabla_t \bar{y}$. Extending $\bar{z}^g$ to R_2 by setting $\bar{z}^g_e = \bar{x}^g_e$, $\bar{z}^g$ is a flow induced by R_2, $\bar{z}^g_e = \bar{x}^1_e$, and $\nabla_t \bar{z}^g \leq \nabla_t \bar{x}^1$.

Now we consider the flow-absorbing side. $\bar{x}^2 + \bar{x}^p + \bar{x}^a$ is a flow induced by U_a. But $f^-_{U'_a}(\lambda) \geq f^-_{U_a}(\lambda)$; thus U'_a can absorb at least as much flow at t as U_a can. So we can extend $\bar{z}^g$ to a flow $\bar{z}$ induced by Z and with $\bar{z} \leq x^Z(\lambda)$.

In conclusion, $x^Z_e(\lambda) \geq \bar{z}^g_e \geq \bar{x}^g_e \geq \frac{1}{2}\bar{x}_e$.

- $\gamma(R_1) > 1 > \gamma(R_2)$. Then $W_1 \in \mathcal{W}^+$ and $W_2 \in \mathcal{W}^-$. Define the conservative object

$$Z := (R_1, \{s\}, R_2) \in \mathcal{Q}.$$

We will show that $x^Z_e(\lambda) \geq \bar{x}_e/2$.

Let $\hat{x}^{R_1}$ be the flow induced by the flow-generating object R_1, scaled so that the amount of flow entering s is $\bar{x}^g_e$. Note that $\hat{x}^{R_1} \leq x^{R_1}(\lambda)$: because $\vec{f}_{W_1}(\lambda) \geq \vec{f}_P(\lambda)$, we can send flow on W_1 so that the same amount of flow arrives at s, with the same or less flow entering t, and still satisfying the capacity and cost bounds. The amount of flow entering s in $\bar{x}^g$ is at least $\bar{x}^g_e$ just from flow conservation at s. Similarly, let $\hat{x}^{R_2}$ be the flow induced by the flow-absorbing object R_2, scaled to that $\hat{x}^{R_2} = \bar{x}^g_e$ (equivalently, so that $\hat{x}^{R_2}$ has $\gamma_e \bar{x}^g_e$ units of flow leaving t). Again, since $\overleftarrow{f}_{W_2}(\lambda) \geq \overleftarrow{f}_P(\lambda)$, $\hat{x}^{R_2} \leq x^{R_2}(\lambda)$.

Now let $\hat{x} = \mu\hat{x}^{R_1} + (1-\mu)\hat{x}^{R_2}$ be the convex combination of $\hat{x}^{R_1}$ and $\hat{x}^{R_2}$ for which $\nabla\hat{x} = 0$. Since $\hat{x}^{R_1}$ has zero net flow everywhere except for a positive net flow at s (since $\gamma(R_1) > 1$), and $\hat{x}^{R_2}$ the same with a negative net flow at s (since $\gamma(R_2) < 1$), such a convex combination exists. Further, $x^Z(\lambda) = \alpha\hat{x}$ for some $\alpha \geq 1$, since $\hat{x}$ is a flow induced by Z, and it does not violate any capacity or cost bounds (since it is a convex combination of flows that satisfy these bounds). So it suffices to show that $\hat{x}_e \geq \frac{1}{4}\bar{x}_e$.

If $\mu \geq 1/2$, then observe that the amount of flow in $\hat{x}$ arriving at s is at least $\frac{1}{2}\bar{x}^g_e$, by the construction of $\hat{x}^{R_1}$. All of this flow leaves along e, so $\hat{x}_e \geq \frac{1}{2}\bar{x}^g$. On the other hand if $\mu < 1/2$, then $\hat{x}_e \geq \frac{1}{2}\hat{x}^{R_2}_e \geq \frac{1}{2}\bar{x}^g_e$. So in either case, $\hat{x}_e \geq \frac{1}{2}\bar{x}^g_e \geq \frac{1}{4}\bar{x}_e$.

Case 3: $\bar{x}^a_e > \frac{1}{2}\bar{x}_e$. This is essentially identical to the case $\bar{x}^g_e \geq \frac{1}{2}\bar{x}_e$, with the roles of flow-generating objects and flow-absorbing objects exchanged. Formally, one can obtain the result by considering the reversed graph $\overleftarrow{G}$, which flips the role of flow-generating and flow-absorbing cycles in a conservative object, and turns this case into the previous one.

Reducing the size of the dominating collection to $O(K + m\bar{m})$. It remains to define an $O(K+m\bar{m})$-sized collection $\mathcal{D}$ of $O(n)$-recurrent conservative objects for which $x^{\mathcal{D}}_e \geq x^{\mathcal{Q}}_e/2$. We will in fact choose $\mathcal{D} \subseteq \mathcal{Q}$.

The collection $\mathcal{Q}_2$ already has size $O(K + m\bar{m})$, so we will include all of these objects. Consider any $C \in \mathcal{Q}_1$, which we can write as $U_g \oplus e \oplus U_a$ for some $U_g \in \mathcal{H}$ and $U_a \in \mathcal{B}$. Let $x^C(\lambda) = x^{C,g}(\lambda) + x^{C,a}(\lambda)$ be the splitting of x^C into the appropriately scaled flows induced by U_g and U_a. Define, for any collection $\mathcal{C} \subseteq \mathcal{Q}_1$, $x^{\mathcal{C},g}_e := \bigvee_{C\in\mathcal{C}} x^{C,g}_e$, and $x^{\mathcal{C},a}_e := \bigvee_{C\in\mathcal{C}} x^{C,a}_e$.

Claim 4.34. *$x^{\mathcal{Q}_1,g}_e$ and $x^{\mathcal{Q}_1,a}_e$ are both $O(K + m\bar{m})$-simple.*

Given this, by Lemma 4.9 there are collections $\mathcal{D}^+$ and $\mathcal{D}^-$, both subsets of $\mathcal{Q}_1$ and both of size $O(m\bar{m})$, so that $x^{\mathcal{D}^+,g}_e = x^{\mathcal{Q}_1,g}$ and $x^{\mathcal{D}^-,a}_e = x^{\mathcal{Q}_1,a}$. Then

$$x^{\mathcal{D}^+\cup\mathcal{D}^-}_e \geq \max(x^{\mathcal{D}^+,g}_e, x^{\mathcal{D}^-,a}_e) \geq \max(x^{\mathcal{Q}_1,a}_e, x^{\mathcal{Q}_1,g}_e) \geq \tfrac{1}{2}x^{\mathcal{Q}_1}_e.$$

Hence $\mathcal{D} := \mathcal{D}^+ \cup \mathcal{D}^- \cup \mathcal{Q}_2$ is our desired collection. It remains only to prove the claim.

Proof of Claim 4.34. For $C = U_g \oplus e \oplus U_a \in \mathcal{Q}_1$, with $U_g \in \mathcal{H}$ and $U_a \in \mathcal{B}$, define the following two functions:

- $h_C(\lambda, r)$ is the maximum flow on e in a flow induced by $U_g \oplus e$ that satisfies the capacity and cost constraints, *and* has an excess of at most r at t. Then h_C is a 1-simple function:

$$h_C(\lambda, r) = \min(x^{U_g\oplus e}_e(\lambda), K_C r)$$

for some constant K_C that measures the ratio between the flow through e and the excess at t in an induced flow.

- $g_C(\lambda) := f^{-}_{U_a}(\lambda)$. This is again 1-simple (it grows linearly with λ until possibly reaching some threshold where it remains constant).

The functions $h := \bigvee_{C\in Q_1} h_C$ and $g := \bigvee_{C\in Q_1} g_C$ are then both $O(K + m\bar{m})$-simple. Now observe that we can write

$$x_e^{Q_1,g}(\lambda) = h(\lambda, g(\lambda));$$

we pick the flow-absorbing part of C to accept the maximum amount of flow at t, and the flow-generating part that has as much flow through e as possible, given the usual capacity and cost constraints, as well as the constraint on how much flow can arrive at t. It follows by Lemma 4.6 that $x_e^{Q_1,g}$ is $O(K+m\bar{m})$-simple.

The proof for $x_e^{Q_1,a}$ is analogous, exchanging the roles of the flow-generating and flow-absorbing objects in the above argument. □

□

4.8 Strong domination bounds for non-conservative objects

In this section, we prove the following theorem, strengthening the previous Theorem 4.30 by a factor of n.

Theorem 4.35. *For any $t \in V$, there is an $O(m\bar{m})$-sized collection $\mathcal{H}$ of $O(n)$-recurrent flow-generating objects at t such that*

$$f^+_{\mathcal{H}} \geq f^+_{\mathcal{G}(t)}.$$

The corresponding absorbing version follows in precisely the same way as Theorem 4.32 follows from Theorem 4.30. Combined with Theorem 4.33 and Lemma 4.19, this completes the proof of Theorem 4.1.

At a high level, the proof is based on the same patching approach used for path domination. We define a dominating collection of flow-generating objects based on a small set of possible signatures; the signatures are now more complicated, as they include information about where in the flow-generating object $U = (C, W)$ each bottleneck should reside; in the "cycle" part C, or the "path" part P. In order to show that the dominating collection does the job, we consider an arbitrary simple flow-generating object at t, and replace some part of the object in a way that can only increase the amount of flow that can be generated at t, for any cost bound λ. We do this at most three times; each time, either the signature "stabilizes", and in this case we can show domination from an object in our collection, or if not, the signature changes in a way that increases the number of bottlenecks in the path part.

We now define analogous notions of various concepts that we defined previously for walks to flow-generating objects.

Definition 4.36 (Bottlenecks, signature and backbone of a flow-generating object)**.** Consider a flow-generating object $U = (D, W)$ at t, and let $\tilde{x}$ be a flow induced by a splitting $\tilde{U} = (\tilde{D}, \tilde{W})$.

Define the *cost bottleneck step* of $\tilde{U}$ to be the arc $a_c \in E(\tilde{U})$ for which $c_{a_c}\tilde{x}_{a_c}$ is maximal, breaking ties towards arcs closer to t according to the order in the trail $\tilde{D} \oplus \tilde{W}$. The *cost bottleneck* of U is then the arc $e_c \in E$ that corresponds to a_c. Similarly, define the *flow bottleneck step* of $\tilde{U}$ to be the arc $a_f \in E(\tilde{U})$ for which $\tilde{x}_{a_f}/u_{a_f}$ is maximal, breaking ties towards arcs closer to t; *exceptionally*, if all arcs of U have infinite capacity, set $a_f = a_c$. Again the *flow bottleneck* of U is the arc $e_f \in E$ that corresponds to a_f. We define the *cost* of U as $c_{a_c}\tilde{x}_{a_c}/\nabla_t\tilde{x}$, and the *congestion* of U as $\tilde{x}_{a_f}/(u_{a_f}\nabla_t\tilde{x})$.

The *signature* of U is $\sigma(U) := (e_c, e_f, \square, q)$, where e_c and e_f are the cost and flow bottlenecks, $\square \in \{\preceq, \succ\}$ describes the relative ordering of a_c and a_f on the trail $\tilde{D} \oplus \tilde{W}$, and $q \in \{0, 1, 2\}$ is the number of bottleneck steps that lie on $\tilde{W}$. We say that $\sigma(U)$ is of *cycle type* if $q = 0$, of *mixed type* if $q = 1$, and of *path type* if $q = 2$. We also say that an object is of cycle, mixed or path type if its signature is.

The *backbone* of U, denoted $\beta(U)$, is the subwalk of $D \oplus W$ that starts and ends with the bottleneck steps (including the bottleneck steps). We write $\tau(U)$ for the subwalk of $D \oplus W$ before $\beta(U)$, and $\eta(U)$ for the subwalk after $\beta(U)$; that is

$$D \oplus W = \tau(U) \oplus \beta(U) \oplus \eta(U).$$

We also define the corresponding partition of the trail $\tilde{D} \oplus \tilde{W}$ into subtrails $\tau(\tilde{U})$, $\beta(\tilde{U})$ and $\eta(\tilde{U})$.

Let Σ^g be the collection of all possible signatures for flow-generating objects, with Σ^g_0, Σ^g_1 and Σ^g_2 being the signatures of cycle, mixed and path type respectively. We will construct a set $\mathcal{D}$ of $O(m\bar{m})$ flow-generating objects at t which dominates the set of simple flow-generating objects at t. It consists of

3 parts $\mathcal{D} = \mathcal{D}_0 \cup \mathcal{D}_1 \cup \mathcal{D}_2$, such that $\mathcal{D}_2$ dominates those with signature from Σ_2^g, $\mathcal{D}_1 \cup \mathcal{D}_2$ dominates those with signature from Σ_1^g, and $\mathcal{D}_0 \cup \mathcal{D}_1 \cup \mathcal{D}_2$ dominates those with signature from Σ_0^g.

For a signature $\sigma = (e_1, e_2, \Box, q) \in \Sigma^g$, let $\sigma_* := (e_1, e_2, \Box)$. Let $S(\sigma) := S(\sigma_*)$, that is, a highest gain σ_*-capped path.

4.8.1 Dominating objects of path type

For any signature $\sigma \in \Sigma_2^g$, let $Q_2(\sigma)$ be any $(2n+3)$-recurrent flow-generating object at the first node of $S(\sigma)$, such that $Q_2(\sigma) \oplus S(\sigma)$ has signature σ. Let $R(\sigma) := R(\sigma_*)$; recall that it is a highest gain path from the last node of $S(\sigma_*)$ to t, such that $S(\sigma_*) \oplus R(\sigma_*)$ has signature σ_*. We define

$$U_2(\sigma) := Q_2(\sigma) \oplus S(\sigma) \oplus R(\sigma) \quad \text{for each } \sigma \in \Sigma_2^g,$$

and

$$\mathcal{D}_2 := \{U_2(\sigma) : \sigma \in \Sigma_2^g\}.$$

We claim that $\mathcal{D}_2$ dominates all simple flow-generating objects at t of cycle type. In fact, we will dominate a larger collection; this will be useful later in order to dominate objects of mixed and cycle type as well.

Lemma 4.37. *Let U be a flow-generating object at t with signature $\sigma \in \Sigma_2^g$. If U is $(n+3)$-recurrent and $\beta(U) \oplus \eta(U)$ is a path, then there exists an object $\bar{U} \in \mathcal{D}_2$ which dominates U, in that $f_{\bar{U}}^+ \geq f_U^+$.*

Proof. Let $P = \beta(U) \oplus \eta(U)$ and let s be the starting node of P. Let Z be the flow-generating object at s such that $U = Z \oplus P$. By Theorem 4.22, there exists an n-recurrent s-t walk W which strongly dominates the path P. From the proof of Theorem 4.22 (see (11)), we know that $\beta(W) = S(\sigma')$ and $\eta(W) = R(\sigma')$ for some signature $\sigma' \in \Sigma_2^g$.

Let $U' = Z \oplus W$. Then U' dominates U: for any λ, and given whatever excess the object Z can create at s, W manages to send at least as much to t as P. Formally,

$$f_{U'}^+(\lambda) = \vec{f}_W(\lambda, f_Z^+(\lambda)) \geq \vec{f}_P(\lambda, f_Z^+(\lambda)) = f_U^+(\lambda) \quad \text{for any } \lambda \in \mathbb{R}_+.$$

Now observe that for any $\lambda \in \mathbb{R}_+$,

$$-\nabla_s x^W(\lambda) \leq -\nabla_s x^P(\lambda) \leq \nabla_s x^Z(\lambda).$$

The first inequality follows from the strong dominance of W over P, while the second inequality is due to $\sigma \in \Sigma_2^g$. This shows that Z can generate at least as much excess at s as W manages to absorb. Hence, the cost and flow bottlenecks of U' are realized in W, implying that $\sigma(U') = \sigma'$.

Finally, we show that $\bar{U} := U_2(\sigma') \in \mathcal{D}_2$ dominates U'. Note that $U_2(\sigma')$ exists because Z is one such candidate — it has recurrence $2n+3$. The point is that since $U_2(\sigma')$ and U' have the same signature σ', and are identical on the part that contains both bottleneck arcs, the maximum flow that can be sent to t is the same in both objects:

$$\nabla_t x^{U_2(\sigma')}(\lambda) = \gamma(R(\sigma'))\nabla_s x^{S(\sigma')}(\lambda) = \nabla_t x^{U'}(\lambda) \qquad \text{for all } \lambda. \qquad \Box$$

4.8.2 Dominating objects of mixed type

Before constructing the dominating collection $\mathcal{D}_1$, we first need a few definitions.

For a flow-generating object $U = (D, W)$ of mixed type, the *core* of U is $(D, W \setminus \eta(U))$. Given a signature σ, we say that U is a *σ-core* if $\sigma(U) = \sigma$ and $E(\eta(U)) = \emptyset$.

For any signature σ of mixed type, let $Y(\sigma)$ be a $(n+2)$-recurrent σ-core chosen to have smallest possible *cost* in the case that σ is of the form $(e_c, e_f, \preceq, 1)$, or of smallest possible *congestion* if σ is of the form $(e_c, e_f, \succ, 1)$.

For any signature $\sigma \in \Sigma_1^g$, let $R_1(\sigma)$ be a highest-gain path from the last node of $Y(\sigma)$ to t, such that $Y(\sigma) \oplus R_1(\sigma)$ has signature σ. We define

$$U(\sigma) := Y(\sigma) \oplus R_1(\sigma) \quad \text{for each } \sigma \in \Sigma_1^g$$

and

$$\mathcal{D}_1 := \{U(\sigma) : \sigma \in \Sigma_1^g.\}$$

Lemma 4.38. *Let U be a flow-generating object at t with signature $\sigma \in \Sigma_1^g$. Let e be the bottleneck arc of $\beta(U)$ that is closer to t. If U is $(n+2)$-recurrent and $e \oplus \eta(U)$ is a path, then there exists an object in $\mathcal{D}_1 \cup \mathcal{D}_2$ which dominates U.*

Proof. Let s be the first node of $\eta(U)$ (so e has head s). Define $U' := Y(\sigma) \oplus \eta(U)$; we replace everything up until s with $Y(\sigma)$.

Claim 4.39. $f^+_{U'} \geq f^+_U$. *Furthermore, if $\sigma(U') \neq \sigma(U)$, then U' is of path type, and $\beta(U') \oplus \eta(U')$ is a path.*

Proof. Suppose that $\sigma := \sigma(U) = (e_c, e_f, \preceq, 1)$; the case $\sigma = (e_c, e_f, \succ, 1)$ will be completely analogous, swapping the roles of the cost and flow bottlenecks. Note that U contains a finite capacity arc because σ is of mixed type. Let j be the head of e_f, and let R be the core of U. Write $\tilde{R}$ and $\tilde{Y}$ for splittings of R and $Y(\sigma)$ respectively, chosen so that an arc $a_c \in \tilde{E}$ is the cost bottleneck step for both, and an arc $a_f \in \tilde{E}$ the flow bottleneck step for both.

To show that $f^+_{U'} \geq f^+_U$, it clearly suffices to show that $x^{\tilde{Y}}_{a_f}(\lambda) \geq x^{\tilde{R}}_{a_f}(\lambda)$ (or equivalently, $\nabla_j x^{\tilde{Y}}(\lambda) \geq \nabla_j x^{\tilde{R}}(\lambda)$) for all λ, given that U' and U share $\eta(U)$. So fix any $\lambda \geq 0$, and let $\tilde{y} := x^{\tilde{Y}}(\lambda)$, $\tilde{x} := x^{\tilde{R}}(\lambda)$. If $\tilde{y}_{a_f} = u_{e_f}$, then clearly the claim holds. If not, then $c_{e_c}\tilde{y}_{a_c} = \lambda \geq c_{e_c}\tilde{x}_{a_c}$. Since $Y(\sigma)$ has minimum cost amongst $(n+2)$-recurrent σ-cores, it has cost no larger than R; that is, $c_{e_c}\tilde{y}_{a_c}/\nabla_j \tilde{y} \leq c_{e_c}\tilde{x}_{a_c}/\nabla_j \tilde{x}$. Combining, $\nabla_j \tilde{y} \geq \nabla_j \tilde{x}$ as required.

Finally, we observe that if the cost bottleneck step of U' is not a_c, then it cannot be any other arc of $\tilde{Y}$, since a_c is the cost bottleneck step of $\tilde{Y}$; so in this case, the cost bottleneck must be an arc of $\eta(U)$. On the other hand, the flow bottleneck step of U' remains a_f — it cannot be any other arc of $\tilde{Y}$ and $a_f \oplus \eta(\tilde{U})$ because a_f is the flow bottleneck step of $\tilde{Y}$ and $a_f \oplus \eta(\tilde{U})$. So if $\sigma(U') \neq \sigma(U)$, then $\beta(U') \oplus \eta(U') = e_f \oplus \eta(U)$, which is a path. □

Let us first consider the case where $\sigma(U') = \sigma(U)$. We claim that $U(\sigma) \in \mathcal{D}_1$ dominates U'. Fix any $\lambda \in \mathbb{R}_+$. Since $\sigma(U(\sigma)) = \sigma(U') = \sigma \in \Sigma_1^g$, we obtain

$$\nabla_t x^{U(\sigma)}(\lambda) = \gamma(R_1(\sigma))\nabla_s x^{Y(\sigma)}(\lambda) \geq \gamma(\eta(U))\nabla_s x^{Y(\sigma)}(\lambda) = \nabla_t x^{U'}(\lambda),$$

where the inequality is due to our choice of $R_1(\sigma)$ and the fact that $\eta(U') = \eta(U)$ is a path.

Next, consider the case where $\sigma(U') \neq \sigma(U)$. According to Claim 4.39, U' is of path type and $\beta(U') \oplus \eta(U')$ is a path. Since U' is $(n+3)$-recurrent, by Lemma 4.37, there exists an object in $\mathcal{D}_2$ which dominates U'.

So, there is an object in $\mathcal{D}_1 \cup \mathcal{D}_2$ which dominates U', and U' dominates U, completing the proof. □

4.8.3 Dominating objects of cycle type

For any signature $\sigma \in \Sigma_0^g$, let $L_0(\sigma)$ be an $(n+1)$-recurrent walk from the last node of $S(\sigma)$ to the first node of $S(\sigma)$, and let $R_0(\sigma)$ be a 2-recurrent walk from the last node of $S(\sigma)$ to t, such that $U(\sigma) := (L_0(\sigma) \oplus S(\sigma), R_0(\sigma))$ is a flow-generating object with signature σ, $\tau(U) = L_0(\sigma)$ and maximizes the net flow at t. Formally, it maximizes

$$\left(1 - \frac{1}{\gamma(L_0(\sigma))\gamma(S(\sigma))}\right)\gamma(R_0(\sigma)).$$

The dominating collection is defined as

$$\mathcal{D}_0 := \{U(\sigma) : \sigma \in \Sigma_0^g\}.$$

Lemma 4.40. *Let U be a flow-generating object at t with signature $\sigma \in \Sigma_0^g$. If U is simple, then there exists an object in $\mathcal{D}_0 \cup \mathcal{D}_1 \cup \mathcal{D}_2$ which dominates U.*

Proof. Denote $U = (C, P)$, where C is a flow-generating cycle and P is a path. Let s be the unique common node of C and P. By Theorem 4.28, there exists an $(n+1)$-recurrent flow-generating s-s walk C' which dominates C. From the proof, we know that $\beta(C') = S(\sigma')$ for some signature $\sigma' \in \Sigma_0^g$, and $\eta(C')$ is a path. In the exceptional case where $\beta(C') = C'$, it is a highest gain cycle with signature σ'. Note that the first and last arcs of C' are its bottlenecks. For such a signature σ', in this proof we redefine $S(\sigma')$ as a highest gain cycle with signature σ'. Then, $\beta(C') = S(\sigma')$ for this case as well.

Now, consider the $(n+2)$-recurrent flow-generating object $U' = (C', P)$ at t. Since C' dominates C, for any $\lambda \in \mathbb{R}_+$

$$\nabla_t x^{U'}(\lambda) = \vec{f}_W(\lambda, f^+_{C'}(\lambda)) \geq \vec{f}_W(\lambda, f^+_C(\lambda)) = \nabla_t x^U(\lambda).$$

This shows that U' dominates U.

We first consider the case where $\sigma(U') = \sigma'$. Let r be the last node of $\beta(C')$. Let C'' be the cycle obtained by rerouting C' to start and end at r, and define the flow-generating object $U'' = (C'', \eta(C') \oplus P)$. Clearly, U'' is $(n+3)$-recurrent. Let $\tilde{U}' = (\tilde{C}', \tilde{P})$ be a splitting of U', and let $\tilde{U}'' = (\tilde{C}'', \tilde{\eta} \oplus \tilde{P})$ be a corresponding splitting of U'', in the sense that $\tilde{C}'$ is a rerouting of $\tilde{C}''$ to start and end at r. Write a_c and a_f as the cost bottleneck step and flow bottleneck step of $\tilde{U}'$ respectively. Let $\tilde{x}$ be a flow induced by $\tilde{U}'$, and let $\tilde{y}$ be a flow induced by $\tilde{U}''$, scaled so that $\nabla_t \tilde{x} = \nabla_t \tilde{y}$. Then, $\tilde{x}_e = \tilde{y}_e$ for all $e \in E(\tau(\tilde{C}')) \cup E(\beta(\tilde{C}')) \cup E(\tilde{P})$. On $\eta(\tilde{C}')$, every arc e satisfies $\tilde{x}_e = \tilde{y}_e + \tilde{y}_{\bar{e}}$, where $\bar{e}$ is its corresponding arc in $\tilde{\eta}$. Since $a_c, a_f \in E(\beta(\tilde{C}'))$, it follows that $\sigma(U'') = \sigma'$ and $\nabla_t x^{U''} = \nabla_t x^{U'}$.

We claim that $U(\sigma')$ dominates U''. For any $\lambda \in \mathbb{R}_+$,

$$\begin{aligned}
\nabla_t x^{U(\sigma')}(\lambda) &= \left(1 - \frac{1}{\gamma(L_0(\sigma'))\gamma(S(\sigma'))}\right) \gamma(R_0(\sigma')) \nabla_s x^{S(\sigma')}(\lambda) \\
&\geq \left(1 - \frac{1}{\gamma(\tau(U''))\gamma(S(\sigma'))}\right) \gamma(\eta(C'))\gamma(P) \nabla_s x^{S(\sigma')}(\lambda) = \nabla_t x^{U''}(\lambda),
\end{aligned}$$

where the inequality is due to our choice of $L_0(\sigma), R_0(\sigma)$ and the fact that $\tau(U'')$ is $(n+1)$-recurrent and $\eta(U'') = \eta(C') \oplus P$ is 2-recurrent.

Next, we consider the case where $\sigma(U') \neq \sigma'$. Then, $\sigma(U') \in \Sigma_1^g \cup \Sigma_2^g$. Since U' is $(n+2)$-recurrent, we are done by Lemma 4.38 or Lemma 4.37, depending on whether $\sigma(U')$ is in Σ_1^g or Σ_2^g. □

Clearly, $\mathcal{D} = \mathcal{D}_0 \cup \mathcal{D}_1 \cup \mathcal{D}_2$ has size at most $|\Sigma^g| = O(m\bar{m})$. By Lemmas 4.37, 4.38 and 4.40, Theorem 4.35 is proved.

5 Initialization for generalized flows

In this section, we show that (LP) can be solved using the SLLS IPM after a preprocessing relying on primal and dual feasibility oracles for the generalized flow problem. Our input problem is

$$\min \langle c, x \rangle \quad \text{s.t.} \quad \mathbf{A}x = b\,, \quad x \geq \mathbf{0}\,, \tag{14}$$

where the variables correspond to arcs and the rows to nodes of a minimum-cost generalized flow problem as in (MGF). In the preprocessing and initialization stage, we will solve feasibility systems for any $b' \in \mathbb{R}^m$, $I \subseteq [n]$, and $c' \in \mathbb{R}^I$, $\ell, r \in (\mathbb{R} \cup \{-\infty\})^I$ of the forms

$$\mathbf{A}_I x = b'\,, \quad x \geq \ell \quad \text{and} \quad [\mathbf{A}_I]^\top y + s = c'\,, \quad s \geq r\,, \tag{15}$$

Here, $\mathbf{A}_I$ is the submatrix correspond to the columns in I; in these systems, we set all variables outside I to 0. Strongly polynomial subroutines are available, as they easily reduce to generalized flow maximization [Vég17, OV20], and to 2VPI feasibility, e.g., [Meg83, HN94], respectively.

Throughout this section, let us assume both the primal and dual programs in (LP) are feasible. We can check both primal and dual feasibilities using the oracles above. E.g., if the dual is infeasible, then the dual feasibility algorithm returns an unbounded direction for the primal.

Finding maximum support solutions. Given an instance of (LP), we first find primal and dual feasible solutions of maximal support. Namely, we find sets $P^\star, Q^\star \subseteq [n]$ and primal and dual feasible solutions $(\bar{x}, \bar{s}) \in \mathcal{P} \times \mathcal{D}$ satisfying the following *(i)* $\mathrm{supp}(\bar{x}) = P^\star$, $\mathrm{supp}(\bar{s}) = Q^\star$; *(ii)* for any $i \in [n] \setminus P^\star$, $x_i = 0$ for every $x \in \mathcal{P}$, and for $j \in [n] \setminus Q^\star$, $s_i = 0$ for every $s \in \mathcal{D}$. We note that duality theory implies $P^\star \cup Q^\star = [n]$, see also Lemma 7.2 later.

These can be found by solving at most n primal and n dual instances of (15) as follows. First, we find any primal feasible solution $\hat{x} \in \mathcal{P}$. Then, for any $j \in [n] \setminus \mathrm{supp}(\hat{x})$, we solve the primal LP in (15) of the form $\mathbf{A}z = 0$, $z_j \geq 1$, $z_i \geq 0$ for $i \in [n] \setminus (\mathrm{supp}(\hat{x}) \cup \{j\})$. If this auxiliary LP has a feasible solution z, then $x^{(j)} := \hat{x} + \alpha z \in \mathcal{P}$ for a sufficiently small $\alpha > 0$ with $x^{(j)}_j > 0$; and if it is infeasible then clearly $x_j = 0$ must hold for every $x \in \mathcal{P}$. We obtain the desired $P^\star$ as the set of indices in $\mathrm{supp}(\hat{x})$ and the indices where the auxiliary LP is feasible, and the desired $\bar{x}$ as the average of all such solutions. The set $Q^\star$ and vector $\bar{s}$ can be found analogously.

Reducing the LP. We obtain an equivalent problem instance by replacing the costs by $\bar{s}$. Let us define the index sets

$$R := P^\star \cap Q^\star\,,\qquad S := P^\star \setminus Q^\star\,,\qquad T := [n]\setminus P^\star\,.$$

Let us first delete all variables $e \in T$, leading to the system

$$\min \langle \bar{s}_R, x_R\rangle + \langle \bar{s}_S, x_S\rangle \quad \text{s.t.} \quad \mathbf{A}_R x_R + \mathbf{A}_S x_S = b\,,\quad x_R, x_S \geq \mathbf{0}\,, \tag{16}$$

This clearly has the same optimum value as the original system. In the next step, let us remove the nonnegativity constraints from all variables in S. Note that $\bar{s}_e > 0$ for $e \in R$ and $\bar{s}_e = 0$ for S. Thus, we obtain the system

$$\min \langle \bar{s}_R, x_R\rangle \quad \text{s.t.} \quad \mathbf{A}_R x_R + \mathbf{A}_S x_S = b\,,\quad x_R \geq \mathbf{0}\,, \tag{17}$$

with the guarantee that its optimum value equals the optimum value of (16), and therefore of the original system (14) (with respect to the modified cost function $\bar{s}$.)

Consider a variable $e \in P^\star \setminus Q^\star$ that represents an (i, j) arc $e \in E_{i,j}$. We can obtain an equivalent minimum-cost generalized flow instance by contracting this arc. Namely, we replace every arc $e' \in E_{k,i}$ of gain factor $\gamma_{e'}$ by a (k, j)-arc of gain factor $\gamma_{e'}\gamma_e$, and replace every arc $e' \in E_{i,k}$ by a (j, k) arc of gain factor $\gamma_{e'}/\gamma_e$. In both cases we keep the cost $c_{e'}$ for the new arc. Further, we replace b_j by $b_i + \gamma_{ij} b_i$, and delete the node i. Note that in the matrix form this corresponds to column operations that change the row of i so that it contains -1 in the column corresponding to e and 0 in all other entries, followed by a row operation which adds γ_{ij} times the row of i to the row of j. Further, this mapping preserves the primal solution $\hat{x}_{R\cup S}$ on the variables in R with the same values.

We contract every arc in S this way to obtain a minimum-cost generalized flow instance

$$\min \langle c', x'\rangle \quad \text{s.t.} \quad \mathbf{A}'x' = b',\quad x' \geq \mathbf{0}\,. \tag{18}$$

Moreover, a strictly positive feasible solution is known. Assume we can find primal and dual optimal solutions (x', y', s') to this system. This can be naturally mapped back to primal and dual optimal solutions $(\tilde{x}, \tilde{y}, \tilde{s})$ to the system (17), with $\tilde{s}_i = 0$ for $i \in S$. If $\tilde{x}_S \geq \mathbf{0}$, then after appending the variables $i \in T$ with $\tilde{x}_i = 0$, we get an optimal primal solution to the original input instance (14), noting that (14) and (17) have the same optimum value.

Assume now $\tilde{x}_S$ has some negative coordinates. Observe that $(\tilde{y}, \tilde{s})$ is also an optimal dual solution to (16), because it is feasible to the dual, and because (16) and (17) have the same optimum values. By complementary slackness, the following system is feasible, and any solution x yields an optimal solution to (16):

$$\mathbf{A}_R x_R + \mathbf{A}_S x_S = b\,,\quad x_R, x_S \geq \mathbf{0}\,,\quad x_i = 0 \quad \forall i \in \operatorname{supp}(\tilde{s})\,.$$

We solve this system using the primal oracle in (15). After adding coordinates $x_i = 0$ for $i \in T$, we obtain a primal optimal solution $x^\star$ to the original problem (14). Given $x^\star$, we can obtain a dual optimal solution $s^\star$ by solving a dual feasibility problem

$$\mathbf{A}^\top y + s = c\,,\quad s_i = 0 \quad \forall i \in \operatorname{supp}(x^\star).$$

Initializing the IPM. According to the above reduction, we can focus our attention on solving the min-cost generalized flow problem (14) under the assumption that $c > \mathbf{0}$ and a strictly positive primal feasible solution $\hat{x} > \mathbf{0}$ is provided. Thus, $(\bar{x}, \mathbf{0}, c)$ form a pair of primal and dual feasible solutions with gap $\gamma = \langle c, \bar{x}\rangle$. We select M such that

$$M \geq \max\left\{\gamma \max_{i\in[n]} \max\left\{\frac{1}{\bar{x}_i}, \frac{1}{c_i}\right\}, \frac{n^2}{\beta}\left(\|\bar{x}\|_\infty + \|c\|_\infty\right)\right\}.$$

The first term in the bound guarantees that for any primal and dual optimal solutions $(x^\star, y^\star, s^\star)$ to (18) we have $\|x^\star\|_\infty, \|s^\star\|_\infty < M$, noting that $\langle c, x^\star\rangle \leq \langle c, \bar{x}\rangle = \gamma$ and $\langle \bar{x}, s^\star\rangle \leq \langle \bar{x}, c\rangle = \gamma$. The second term will be used to argue for the centrality of the initial solution of the extended system below.

We now write the 'big-M system' (4) of [VY96]. For simplicity, in the following argument we denote the input to the system (18) $(\mathbf{A}, b, c)$ as for the original system, and we let $\hat{x} > 0$ denote the primal feasible solution. As noted already, this is still a min-cost generalized flow problem.

$$\begin{aligned} \min\ & \langle c, x'\rangle + M\langle \mathbf{1}_n, x'\rangle \\ \mathbf{A}x - \mathbf{A}x' &= b \\ x + x'' &= 2M\mathbf{1}_n \\ x, x', x'' &\geq \mathbf{0}_n\,, \end{aligned} \qquad \begin{aligned} \max\ & \langle b, y\rangle + 2M\langle \mathbf{1}_n, y'\rangle \\ \mathbf{A}^\top y + y' + s &= c \\ y' + s'' &= \mathbf{0}_n \\ -\mathbf{A}^\top y + s' &= M\mathbf{1}_n \\ s, s', s'' &\geq \mathbf{0}_n\,, \end{aligned} \tag{19}$$

We can initialize with $(x,x',x'') = (M\mathbf{1}_n, M\mathbf{1}_n, M\mathbf{1}_n - \bar{x})$ and $(s,s',s'') = (M\mathbf{1}_n, M\mathbf{1}_n - c, M\mathbf{1}_n)$. By the choice of $M \geq \frac{n^2}{\beta}(\|\bar{x}_R\|_\infty + \|\bar{s}_R\|_\infty)$, it is easy to check that this point satisfies the initialization condition in Theorem 1.3.

We claim that any optimal solution to this system is optimal to the original system; that is, all the auxiliary variables are $\mathbf{0}$. To see this, note that by the choice of M, for any primal and dual optimal solutions $(x^\star, y^\star, s^\star)$ to (18), $(x^\star, \mathbf{0}_n, M\mathbf{1}_n - x^\star)$ and $(s^\star, \mathbf{0}_n, M\mathbf{1}_n - c - s^\star)$ are optimal to (19) by complementary slackness. Also by complementary slackness, it follows that the auxiliary variables x' and s'' must be $\mathbf{0}_n$ for *any* primal and dual optimal solutions.

Further, we note that (19) is a min-cost generalized flow problem with capacity bounds. This can be reduced to an uncapacitated min-cost generalized flow problem by adding a new node for each arc. Hence, the straight line complexity bounds are applicable.

Finally, for the sake of the strongly polynomial model, we need to guarantee that the bit-complexity of M is bounded in terms of the encoding length of the input; the definition of the encoding length is given in Section 8. Still, we note that it suffices to guarantee that the maximum support solutions $\hat{x}$ and $\hat{s}$ found in the first step have bounded encoding length. This can be ensured by requiring a basic solution from each of the feasibility oracle calls, and choosing the maximal possible values of each coefficient α if it is bounded, and $\alpha = 1$ if unbounded. Given arbitrary feasible solutions, a basic solution can be obtained by a simple support reduction procedure in strongly polynomial time.

6 Background on interior point methods

Projections and lifts. Let $W \subseteq \mathbb{R}^n$ be a linear subspace. We define two important operations: projection to a coordinate subspace, and fixing coordinates to 0. Namely, for $I \subseteq [n]$, let

$$\pi_I(W) := \{w_I : w \in W\}, \qquad \varphi_I(W) := \{w_I : w \in W, w_{[n]\setminus I} = \mathbf{0}\}. \tag{20}$$

It is easy to verify that the coordinate projection of $W^\perp$ is the orthogonal complement of $\varphi_I(W)$.

Lemma 6.1. *For any subspace $W \subseteq \mathbb{R}^n$ and $I \subseteq [n]$, $I \neq \emptyset$, $\pi_I(W^\perp) = [\varphi_I(W)]^\perp$*

The *lifting map* can be used to map back elements of $\pi_I(W)$ to W.

Definition 6.2. Given a subset $I \subseteq [n]$, $I \notin \{\emptyset, [n]\}$, and a subspace $W \subseteq \mathbb{R}^n$, we define the *lifting map* $L_I^W : \pi_I(W) \to W \subseteq \mathbb{R}^n$ as follows:

$$L_I^W(x) := \arg\min\{\|w\| : w \in W, w_I = x\}. \tag{21}$$

We further define $\ell_I^W : \pi_I(W) \to \pi_{[n]\setminus I}(W)$ by $\ell_I^W(x) := [L_I^W(x)]_{[n]\setminus I}$, i.e., the restriction of the output of the lifting map to the coordinates in $[n] \setminus I$.

Note that the lifting map is well-defined and is a linear map that can be computed in strongly polynomial time given a basis of W. This map plays a key role in LLS IPMs, see e.g. [DHNV23].

Subspace formulation for LP. We now reformulate (LP) in the language of subspaces, a more convenient language for the initialization techniques in Section 7. Give an instance of (LP) with $(\mathbf{A}, b, c)$, let $W = \ker(\mathbf{A})$ and $d \in \mathbb{R}^n$ such that $\mathbf{A}d = b$. Note that such a d can be found in strongly polynomial time if it exists; if it does not exists, then the LP is infeasible. Thus, we can write the LP as follows:

$$\begin{aligned} \min\ & \langle c, x\rangle & \qquad \min\ & \langle d, s\rangle \\ & x \in W + d & & s \in W^\perp + c \\ & x \geq \mathbf{0}, & & s \geq \mathbf{0}, \end{aligned} \tag{LP-subspace}$$

We repeat the definitions of $\mathcal{P}$, $\mathcal{D}$, $\mathcal{P}_{++}$, and $\mathcal{D}_{++}$, also indexing them as we will consider different linear programs.

$$\mathcal{P}(W,d) := \{x \in \mathbb{R}^n \mid x \in W + d, x \geq \mathbf{0}\}, \quad \mathcal{D}(W,c) := \{s \in \mathbb{R}^n \mid s \in W^\perp + c, , \ s \geq \mathbf{0}\}$$

and the strictly feasible sets $\mathcal{P}_{++}(W,d) = \mathcal{P}(W,d) \cap \mathbb{R}^n_{++}$ and $\mathcal{D}_{++}(W,d) = \mathcal{D}(W,c) \cap \mathbb{R}^n_{++}$. We simply use $\mathcal{P}, \mathcal{D}, \mathcal{P}_{++}, \mathcal{D}_{++}$ when clear from the context. For $(x,s) \in \mathcal{P}_{++} \times \mathcal{D}_{++}$, the *normalized duality gap* is

$$\overline{\mu}(x,s) := \frac{\langle x, s\rangle}{n}.$$

We will use the following simple but useful property.

Proposition 6.3. *Given $x, x' \in W + d$, $s, s' \in W^\perp + c$, we have that*

$$\langle x, s\rangle + \langle x', s'\rangle = \langle x, s'\rangle + \langle x', s\rangle .$$

In particular, if $\langle x', s'\rangle = 0$, then

$$\langle x, s\rangle = \langle x, s'\rangle + \langle x', s\rangle .$$

Proof. This follows since $x - x' \in W$ and $s - s' \in W^\perp$ live in orthogonal subspaces. □

We formulate a simple linearity property of the the duality gap; see e.g. [ADL+23].

Proposition 6.4 (Linearity of duality gap). *Given $x^{(1)}, \dots, x^{(k)} \in W + d$, $s^{(1)}, \dots, s^{(k)} \in W^\perp + c$ forming the sequence $z^{(1)} = (x^{(1)}, s^{(1)}), \dots, z^{(k)} = (x^{(k)}, s^{(k)})$ and $\lambda \in \mathbb{R}^k$ such that $\sum_{i=1}^k \lambda_i = 1$, we have that*

$$\overline{\mu}\left(\sum_{i=1}^k \lambda_i z^{(i)}\right) = \sum_{i=1}^k \lambda_i \overline{\mu}(z^{(i)}) .$$

The central path and the analytic center. Given (LP-subspace), if $\mathcal{P}_{++}, \mathcal{D}_{++} \neq \emptyset$, then for every $\mu > 0$ there exists a unique point $(x(\mu), s(\mu)) \in \mathcal{P} \times \mathcal{D}$ with $x_i(\mu)s_i(\mu) = \mu$ for every $i \in [n]$. The *central path* is the algebraic curve formed by the points $(x(\mu), s(\mu))$ for $\mu \in (0, \infty)$. The limit of the central path is a primal-dual optimal solution pair $(x^\star, s^\star) \in \mathcal{P} \times \mathcal{D}$, such that $B^\star = \mathrm{supp}(x^\star)$ and $N^\star = \mathrm{supp}(s^\star)$ form a partition of $[n]$.

The SLLS algorithm maintains iterates in the *ℓ_2-neighborhood* of the central path. For the rest of the paper, we use a fixed value $\beta \in (0, 1/6]$.

$$\mathcal{N}^2(\beta) := \left\{(x, s) \in \mathcal{P}_{++} \times \mathcal{D}_{++} : \left\| \frac{x \circ s}{n^{-1}\langle x, s\rangle} - \mathbf{1} \right\|_2 \leq \beta \right\} , \tag{22}$$

We let $\overline{\mathcal{N}}^2(\beta)$ denote the closure of the ℓ_2 neighborhood. This will correspond to the part of neighborhood containing the limit optimal solutions on the central path.

We use the following proximity properties. For the first one, see e.g., [Gon92, Lemma 5.4] and [MT03, Proposition 2.1].

Proposition 6.5. *Let $z = (x, s) \in \mathcal{N}^2(\beta)$ for $\beta \in (0, 1/4]$ and $\mu = \overline{\mu}(z)$, and consider the central path point $z^{\mathrm{cp}}(\mu) = (x^{\mathrm{cp}}(\mu), s^{\mathrm{cp}}(\mu))$. For each $i \in [n]$,*

$$\frac{x_i}{1+2\beta} \leq \frac{1-2\beta}{1-\beta}\, x_i \leq x_i^{\mathrm{cp}}(\mu) \leq \frac{x_i}{1-\beta} , \quad \textit{and}$$
$$\frac{s_i}{1+2\beta} \leq \frac{1-2\beta}{1-\beta}\, s_i \leq s_i^{\mathrm{cp}}(\mu) \leq \frac{s_i}{1-\beta} .$$

We will often use the following proposition which is immediate from the definition of $\mathcal{N}^2(\beta)$.

Proposition 6.6. *Let $z = (x, s) \in \mathcal{N}^2(\beta)$ for $\beta \in (0, 1/4]$, and $\mu = \overline{\mu}(z)$. Then for each $i \in [n]$*

$$(1-\beta)\mu \leq s_i x_i \leq (1+\beta)\mu .$$

The following lemma shows that the choice of parameter $\mu = \langle x, s\rangle / n$ is essentially optimal for the purpose of minimizing centrality error at (x, s).

Lemma 6.7 ([MT03, Lemma 4.4]). *For $\beta \in (0, 1/4]$, let $(x, s) \in \mathcal{P}_{++} \times \mathcal{D}_{++}$ and $\mu' > 0$ satisfy $\|xs/\mu' - \mathbf{1}\|_2 \leq \beta$. Then, for $\mu := \langle x, s\rangle / n$, we have that*

$$\left\| \frac{(x \circ s)}{\mu} - \mathbf{1} \right\|_2 \leq \frac{\beta}{1-\beta} , \quad \textit{and} \quad (1 - \beta/\sqrt{n})\mu' \leq \mu \leq (1 + \beta/\sqrt{n})\mu' . \tag{23}$$

A key property of the central path is *'near monotonicity'*, formulated in the following lemma, see [VY96, Lemma 16].

Lemma 6.8. *For the central path point $z^{\mathrm{cp}}(\mu) = (x^{\mathrm{cp}}(\mu), s^{\mathrm{cp}}(\mu))$ and any $z = (x, s) \in \mathcal{P} \times \mathcal{D}$, it holds that*

$$z \le n\left(1 + \frac{\overline{\mu}(z)}{\mu}\right) z^{\mathrm{cp}}(\mu)\,.$$

Moreover, if $\overline{\mu}(z) \le \mu$, then $z \le n z^{\mathrm{cp}}(\mu)$.

Proof. We only prove the first part; for the proof of the second statement see [VY96, Lemma 16]. By Proposition 6.3,

$$\langle x, s^{\mathrm{cp}}(\mu)\rangle + \langle x^{\mathrm{cp}}(\mu), s\rangle = \langle x, s\rangle + \langle x^{\mathrm{cp}}(\mu), s^{\mathrm{cp}}(\mu)\rangle = n\overline{\mu}(z) + n\mu\,.$$

Dividing by μ, and using that $x_i^{\mathrm{cp}}(\mu) s_i^{\mathrm{cp}}(\mu) = \mu$, we get

$$\sum_{i=1}^n \left(\frac{x_i}{x_i^{\mathrm{cp}}(\mu)} + \frac{s_i}{s_i^{\mathrm{cp}}(\mu)}\right) \le n\left(1 + \frac{\overline{\mu}(z)}{\mu}\right)\,.$$

The first part follows since all terms are nonnegative. □

Assuming $\mathcal{P}_{++}(W, d) \neq \emptyset$ and $\mathcal{P}$ is bounded, its *analytic center* is

$$\tilde{x} := \operatorname*{arg\,max}_{x \in \mathcal{P}_{++}(W,d)} \sum_{i=1}^n \log x_i\,.$$

By Lagrangian duality, there exists a unique $\tilde{v} \in W^\perp$ such that $\tilde{x}_i \tilde{v}_i = 1$ for all $i \in [n]$. We define the analytic center of $\mathcal{D}_{++}(W, c)$ analogously as the maximizer of $\sum_{i=1}^n \log x_i$ over $\mathcal{D}_{++}(W, c)$. This is equivalent to the existence of $w \in W$ with $\tilde{s}_i \tilde{v}_i = 1$ for all $i \in [n]$. We note that one of $\mathcal{P}_{++}(W, d)$ and $\mathcal{D}_{++}(W, d)$ is always unbounded; hence, the analytic center may only exist for either of them.

It is well-known that the central path limit point $(x^\star, s^\star)$ coincide with the analytic centers of the primal and dual optimal facets, see e.g., [RTV05, Theorem I.30]. Namely, $x^\star$ is the analytic center of $\pi_{B^\star}(W)$, and $s^\star$ is the analytic center of $\pi_{N^\star}(W^\perp)$.

We also use the following step-length estimate. In Appendix B, we show a stronger version Proposition B.16.

Lemma 6.9 (Step-length estimate for general directions, [ADL+23])**.** *Let $z = (x, s) \in \mathcal{N}^2(\beta)$, $\beta \in (0, 1/6]$. Consider directions $\Delta x \in W$, $\Delta s \in W^\perp$ that satisfy $\|\Delta x \circ \Delta s\| \le \beta\mu/4$. Let*

$$\gamma := \frac{\|(x + \Delta x) \circ (s + \Delta s)\|}{\mu}\,.$$

Then $(x + \alpha\Delta x, s + \alpha\Delta s) \in \overline{\mathcal{N}}^2(2\beta)$ and $\overline{\mu}(x + \alpha\Delta x, s + \alpha\Delta s) \le (1 + \frac{3}{2}\beta/\sqrt{n})(1-\alpha)\mu$, for all $0 \le \alpha \le 1 - \frac{4\gamma}{\beta}$.

6.1 A stronger version of the SLLS IPM

We will require following adapted version of the IPM in Theorem 1.3, which we prove in Appendix B. The two main differences are that the iterations are implementable in strongly polynomial time, and the second is that the output guarantees are stronger, namely, it outputs certificates that the returned optimal solutions are close the analytic centers of the corresponding optimal faces.

Theorem 6.10. *Let $\beta \in (0, 1/6]$ be a constant, $\eta \in (0, 1)$. Given a point $(x, s) \in \mathcal{N}^2(\beta)$, for (LP-subspace) with data (W, d, c), there exists an IPM, that terminates in*

$$O\left(n^{1.5} \log\left(\frac{n}{\eta}\right) \sum_{i=1}^n \mathrm{SLC}_\eta(x_i^{\mathrm{m}})\right)$$

many iterations and returns a quadruple $(x^\star, s^\star, v^\star, w^\star) \in \mathbb{R}^{4n}$ that for $B^\star := \mathrm{supp}(x^\star)$ and $N^\star := \mathrm{supp}(s^\star)$ has the following properties:

1. *$B^\star$ and $N^\star$ form a partition of $[n]$,*
2. *$x^\star \in \mathcal{P}(W, d)$, $s^\star \in \mathcal{D}(W, c)$,*
3. *$v^\star \in W^\perp$, $v_B^\star > \mathbf{0}_B$ and $w^\star \in W$, $w_N^* > \mathbf{0}_N$.*
4. *$\|(x_{B^\star}^\star \circ v_{B^\star}^*, s_{N^\star}^\star \circ w_{N^\star}^\star) - \mathbf{1}_n\|_2 \le \beta$.*

Furthermore, each iteration can be implemented by a strongly polynomial algorithm in the Turing model.

6.2 Straight-line complexity of a subspace

For a subspace $W \subseteq \mathbb{R}^n$ we define, in a slight overload of notation, the following worst-case straight line complexity.

Definition 6.11. For a subspace $W \subseteq \mathbb{R}^n$ and $\eta \in (0,1)$ we let

$$\mathrm{SLC}_\eta(W) := \sup_{c,d \geq \mathbf{0}} \left\{ \max_{i\in[n]} \mathrm{SLC}_\eta(x_i^{\mathrm{m}}) \mid x^{\mathrm{m}} \text{ is MCP for (LP-subspace) with data } (W,d,c) \right\}. \tag{24}$$

Thus, we define the subspace SLC over all possible primal-dual feasible LPs corresponding to the subspace W, i.e., $W = \ker(\mathbf{A})$ in the matrix formulation (LP). As noted in the introduction, we would obtain essentially the same bound if using the dual $\mathrm{SLC}_\eta(s_i^{\mathrm{m}})$ in the definition. The following Lemma, shown in Section 7.3, asserts that $\mathrm{SLC}(W)$ may only go up by one when deleting or projecting out variables.

Lemma 6.12. *Let $W \subseteq \mathbb{R}^n$ be a subspace and $\eta \in (0,1)$. Then, for any index set $I \subseteq [n]$, $\mathrm{SLC}_\eta(\pi_I(W)) \leq \mathrm{SLC}_\eta(W) + 1$ and $\mathrm{SLC}_\eta(\varphi_I(W)) \leq \mathrm{SLC}_\eta(W) + 1$.*

7 An initialization framework for general Linear Programs

Section 5 gave an initialization technique for generalized flows. This requires primal and dual feasibility subroutines that are available for this problem but not in general. Also, it requires solving $2n$ feasibility problems as preprocessing before a single call to the SLLS IPM.

In this section, we present a new, general initialization technique applicable for general instances of (LP-subspace) with data (W,d,c). It only requires four calls to SLLS IPM. Moreover, the running time of all these IPMs will be upper bounded by the subspace straight-line complexity bound $\mathrm{SLC}_\eta(\bar{W})$, where $\bar{W} := \{(w,-w,\bar{w}) : w - \bar{w} \in W\}$. This subspace also corresponds to the system used in the 'big-M' initialization (4); however, we will use a different system. For a generalized flow problem, this still corresponds to a (capacitated) generalized flow problem. The main result of this section is following theorem.

Theorem 7.1 (Initialization for LP)**.** *Given a subspace $W \subseteq \mathbb{R}^n$ and $d,c \in \mathbb{R}^n$, there exists an algorithm (Algorithm 1) that either finds primal-dual optimal solutions to (LP-subspace) with data (W,d,c) or finds a certificate of primal or dual infeasibility. The running time of the algorithm is*

$$O\left(\mathrm{poly}(n) \cdot \min_{\eta\in(0,1]} \log\left(\frac{1}{\eta}\right) \mathrm{SLC}_\eta(\bar{W})\right) \tag{25}$$

and it requires space strongly polynomial in $\mathrm{size}(\mathbf{A},d,c)$, where $\mathbf{A}$ is a representation of W as $\ker(\mathbf{A}) = W$. Here, $\bar{W} := \{(w,-w,\bar{w}) : w - \bar{w} \in W\}$.

The main vehicle of this result is the SLLS IPM in [ADL+23], which requires that primal and dual are strictly feasible and furthermore requires as input an initial point near the central path.

We present a initialization technique with three stages, where the first stage corresponds to a conic feasibility problem, the second stage corresponds to primal and dual feasibility and the third stage corresponds to an optimization problem. Compared to the initialization in Section 5, the first two stages correspond to finding maximum support primal and dual solutions. In the third stage, we similarly remove variables outside the primal support and project out variables outside the dual support. Using the stronger output guarantees in Theorem 6.10, the first two stages will provide well-centered starting solutions to the subsequent stages.

Farkas lemma. To prepare for our initialization technique, we will review the Farkas lemma and a conic version of it.

Lemma 7.2 (Farkas lemma)**.** *Let $W \subseteq \mathbb{R}^n$ and $d,c \in \mathbb{R}^n$. Then, exactly one of the following is feasible:*

(P1) $x \in W + d, x \geq \mathbf{0}$, *or*

(P2) $y \in W^\perp, y \geq \mathbf{0}, \langle d,y\rangle < 0$.

Analogously, exactly one of the following is feasible:

(D1) $s \in W^\perp + c, s \geq \mathbf{0}$, *or*

(D2) $z \in W, z \geq \mathbf{0}, \langle c, z\rangle < 0$.

Lemma 7.3 (Conic Farkas lemma)**.** *Let $W \subseteq \mathbb{R}^n$ be a subspace and $i \in [n]$. Then exactly one of the following is feasible:*

(C1) $x \in W, x \geq \mathbf{0}, x_i > 0$, *or*

(C2) $x \in W^\perp, x \geq \mathbf{0}, x_i > 0$.

Proof. Clearly, both cannot be feasible: if y is feasible for (C1) and z is feasible for (C2), then $0 = \langle y, z\rangle \geq y_i z_i > 0$, a contradiction.

Now assume that (C1) is infeasible. Then, by scaling the system $x \in W, x \geq \mathbf{0}, x_i \geq 1$ is also infeasible, which is equivalent to the system $\bar{x} \in W - e_i, \bar{x} \geq \mathbf{0}$. By Lemma 7.2 there exists a certificate of infeasibility $y \in W^\perp, y \geq 0, \langle \bar{y}, -e_i\rangle < 0$, which is equivalent to $y \in W^\perp, y \geq 0, y_i > 0$, giving a feasible solution to (C2). □

7.1 High level description

Stage I: Conic feasibility. In the first stage, our goal is to find the strictly complementary partition (B, N) for the subspace W guaranteed by Lemma 7.3. Thus, there exist $x \in W \cap \mathbb{R}^n_+$ and $s \in W^\perp \cap \mathbb{R}^n_+$ such that $x_i > 0$ for $i \in B$ and $s_i > 0$ for $i \in N$. We aim for the stronger requirement that these solutions be the analytic centers of the following (feasible and bounded) regions:

$$x_B \in \varphi_B(W), \mathbf{0}_B \leq x_B \leq \mathbf{1}_B, \qquad s_N \in \varphi_N(W^\perp), \mathbf{0}_N \leq s_N \leq \mathbf{1}_N\,, \tag{26}$$

where we recall the definition of φ from (20).

We do so by running the interior point method on the following primal-dual system.

$$\begin{aligned} &\min\ \langle \mathbf{1}_n, \hat{x}\rangle & &\min\ \langle \mathbf{1}_n, \check{s}\rangle \\ &\bar{x} - \hat{x} \in W & &\bar{s} - \check{s} \in W^\perp \\ &\bar{x} + \check{x} = \mathbf{1}_n & &\bar{s} - \check{s} + \hat{s} = \mathbf{1}_n \\ &\bar{x}, \hat{x}, \check{x} \geq \mathbf{0}_n\,, & &\bar{s}, \hat{s}, \check{s} \geq \mathbf{0}_n\,, \end{aligned} \tag{27}$$

The variables $\bar{x}$ and $\bar{s}$ correspond to the *true* variables in (LP-subspace). The variable $\hat{x}$ is auxiliary, and describes the *subspace error* in W in the primal of (27). The variable $\check{x}$ is auxiliary and corresponds to the slack variables $\hat{x} \leq \mathbf{1}_n$ in the upper bounds constraints on $\bar{x}$. The variables dual to $\check{x}$ are the variables $\check{s}$, which correspond to subspace error in $W^\perp$. Analogously, $\hat{s}$ functions as an upper bound on the term $\bar{s} - \check{s}$ and is the dual variable to the primal subspace error variables $\hat{x}$.

We will show that (27) is easy to initialize with the primal and dual solutions $(\frac{1}{2}\mathbf{1}_{3n}, \mathbf{1}_{3n})$, both systems have optimum objective values 0, and the supports of the optimal solutions $\bar{x}^c$ and $\bar{s}^c$ form the partition (B, N) as in Lemma 7.3.

In particular, the relative interior of the optimal region of the primal in (27), restricted to the coordinates in B, corresponds exactly to the relative interior of the primal feasible region in (26). The analogous statement holds for $\bar{s}_N$ on the dual side. We can use the result from Theorem 6.10 to find feasible vectors $\bar{x}^c_B$ and $\bar{s}^c_N$ that satisfy (26), as well as conic vectors that certify the centrality of these vectors. This information will be useful for initializing the feasibility systems in Stage II.

Stage II: Primal and dual feasibility. Having obtained the conic maximum support partition (B, N) of the subspace W, our next goal is to find maximum support primal and dual feasible solutions with some additional guarantees. First, we note that given any vector $x' \in W + d$ with $x'_N \geq \mathbf{0}_N$, we can find a feasible $x \in W + d, x \geq \mathbf{0}_n$ as $x = x' + \alpha\bar{x}^c_B$ for a sufficiently large $\alpha > 0$; moreover, $\mathrm{supp}(x) = \mathrm{supp}(x') \cup B$. Hence, we can focus on finding a maximum support primal feasible solution on the subspace $\pi_N(W)$ projected to the coordinates in N. We can analogously argue about the dual coordinates in B.

Thus, we aim to find vectors near the analytic center of the (relative interior of the) optimal facets of the following region.

$$s \in \pi_B(W^\perp) + c_B, s \geq \mathbf{0}_B\,. \tag{28}$$

The above correspond to the coordinate-projection of the dual feasible region $\mathcal{D}(W, c)$ onto the coordinates in B. The analogous primal feasibility problem is

$$x \in \pi_N(W) + d_N, x \geq \mathbf{0}_N. \tag{29}$$

Here, the feasible region corresponds to the coordinate-projection of the primal feasible region $\mathcal{P}(W, d)$ onto the coordinates in N; as noted above, feasibility on coordinates in B can be obtained using the Stage I solution.

Furthermore, note that the feasible regions in (28) and (29) are bounded, as unboundedness of (28) would imply the existence of a ray $v \in \pi_B(W^\perp), v \geq \mathbf{0}_B, v \neq \mathbf{0}_B$, which can not exist, as we already found a strictly positive vector $\bar{x}^{\mathrm{c}}_B > \mathbf{0}_B$ in the orthogonal space i.e., $\bar{x}^{\mathrm{c}}_B \in \pi_B(W^\perp)^\perp = \varphi_B(W)$ and so we arrive at the contradiction $0 = \left\langle \bar{x}^{\mathrm{c}}_B, v \right\rangle > 0$.

We solve dual feasibility (28) by running the interior point in Theorem 6.10 on the following auxiliary primal-dual system, again noting that $\pi_B(W^\perp)^\perp = \varphi_B(W)$.

$$\begin{aligned} \min \ & \langle c_B, \bar{x}_B \rangle \\ & \bar{x}_B \in \varphi_B(W) \\ \bar{x}_B + \check{x}_B &= \mathbf{1}_B, \\ \bar{x}_B, \check{x}_B &\geq \mathbf{0}_B. \end{aligned} \qquad \begin{aligned} \min \ & \langle \mathbf{1}_B, \check{s}_B \rangle \\ \bar{s}_B - \check{s}_B &\in \pi_B(W^\perp) + c_B, \\ \bar{s}_B, \check{s}_B &\geq \mathbf{0}_B. \end{aligned} \tag{30}$$

It will turn out, that we can initialize these systems directly, using the output of Theorem 6.10 under the Stage I program. Depending on the objective value of this primal-dual system we will be able to either decide dual infeasibility or find a point near the analytic center of the optimal facet of (28).

An analogous program exists for the variables $(\bar{x}_N, \hat{x}_N)$ and $(\bar{s}_N, \hat{s}_N)$ for primal feasibility (29). In case that both (29) and (28) are feasible, let $\bar{x}^{\mathrm{p}}_N$ and $\bar{s}^{\mathrm{d}}_B$ be the corresponding vectors returned by the algorithm.

Stage III: Optimization Finally, we aim to solve the optimization problem. Here, we will need to restrict the set of variables we are operating on even further. To initialize the IPM for optimization, we require strictly feasible primal-dual solutions. However, Stage II may have returned feasible solutions $\bar{x}^{\mathrm{p}}_N$ and $\bar{s}^{\mathrm{d}}_B$ that are not fully supported on N and B. By the property that these vector are near the analytic center of the relative interior of the optimal facet, this allows us to conclude that all feasible primal solutions $x \in \mathcal{P}(W, d)$ fulfill that $x_{\bar{N}} = \mathbf{0}_{\bar{N}}$ for $\bar{N} := N \setminus \mathrm{supp}(\bar{x}^{\mathrm{p}}_N)$. Similarly, all feasible dual solutions $s \in \mathcal{D}(W, c)$ fulfill that $s_{\bar{B}} = \mathbf{0}_{\bar{B}}$ for $\bar{B} := B \setminus \mathrm{supp}(\bar{s}^{\mathrm{d}}_B)$. Hence, we are able to delete these variable sets and only consider the variables $B^* := \mathrm{supp}(\bar{s}^{\mathrm{d}}_B)$ and $N^* := \mathrm{supp}(\bar{x}^{\mathrm{p}}_N)$ for the optimization problem. The corresponding optimization problem now operates on a subspace $W^* \subseteq \mathbb{R}^{B^* \cup N^*}$ of W on which the coordinates in $\bar{N}$ are fixed to 0 and the coordinates in $\bar{B}$ are projected out. That is,

$$W^* := \pi_{[n]\setminus(\bar{B}\cup\bar{N})}\left(\varphi_{[n]\setminus\bar{N}}(W)\right) = \varphi_{B^*\cup N^*}\left(\pi_{[n]\setminus\bar{B}}(W)\right). \tag{31}$$

Recall from Section 5 that the deletion and projection operations can be interpreted as arc deletions and contractions for generalized flows, resulting in another generalized flow instance. Again, we can initialize this new optimization system with the vectors obtained from the algorithm in Theorem 6.10 under the Stage II program.

The output of this last call to an IPM, gives us an optimal solution on the coordinates $B^* \cup N^*$. Some extra combinatorial work is necessary to lift these primal and dual optimization vectors to a primal-dual optimal solution of the original LP (LP-subspace) on the full coordinate set $B \cup N$.

7.2 The algorithm and analysis

The full algorithm is presented in Algorithm 1. We now show that it works correctly, by analyzing the three stages separately.

7.2.1 Stage I, Strict conic feasibility

In this section we show how to solve the conic feasibility problem. Let data (W, d, c) for (LP-subspace) be given. Our aim is to find the partition $B \cup N = [n]$ and corresponding vectors $x \in W, x \geq \mathbf{0}, x_B > \mathbf{0}$ and $s \in W^\perp, s \geq \mathbf{0}, s_N > \mathbf{0}$, which exists by Lemma 7.3. Note that for generalized flow problems we are

Algorithm 1: LP-SOLVE

Input : An instance of (LP-subspace) with data (W, d, c).
Output: One of the following:

- A pair of primal-dual optimal solutions: (x^*, s^*) feasible to (LP-subspace) such that $\langle x^*, s^* \rangle = 0$.
- A certificate of primal infeasibility: $y \in W^\perp \cap \mathbb{R}^n_+$ such that $\langle d, y \rangle < 0$.
- A certificate of dual infeasibility: $y \in W \cap \mathbb{R}^n_+$ such that $\langle c, y \rangle < 0$.

$\beta^{\mathrm{c}} \leftarrow 2^{-16}, \beta^{\mathrm{p}} \leftarrow \beta^{\mathrm{d}} \leftarrow 2^{-8}, \beta^* \leftarrow 2^{-4}$
`// Stage I: Conic Feasibility`
$W^{\mathrm{c}} \leftarrow \{(\bar{w}, \hat{w}, \check{w}) \in \mathbb{R}^{3n} : \bar{w} - \hat{w} \in W, \bar{w} = -\check{w}\}$
$(x^{\mathrm{c}}, s^{\mathrm{c}}, v^{\mathrm{c}}, w^{\mathrm{c}}) \leftarrow \mathrm{IPM}(W^{\mathrm{c}}, \frac{1}{2}\mathbf{1}_{3n}, \mathbf{1}_{3n}, \beta^{\mathrm{c}})$
$(\bar{x}^{\mathrm{c}}, \hat{x}^{\mathrm{c}}, \check{x}^{\mathrm{c}}) \leftarrow x^{\mathrm{c}}, (\bar{s}^{\mathrm{c}}, \hat{s}^{\mathrm{c}}, \check{s}^{\mathrm{c}}) \leftarrow s^{\mathrm{c}}$
$(\bar{w}^{\mathrm{c}}, \hat{w}^{\mathrm{c}}, \check{w}^{\mathrm{c}}) \leftarrow w^{\mathrm{c}}, (\bar{v}^{\mathrm{c}}, \hat{v}^{\mathrm{c}}, \check{v}^{\mathrm{c}}) \leftarrow v^{\mathrm{c}}$
$B \leftarrow \mathrm{supp}(\bar{x}^{\mathrm{c}}), N \leftarrow \mathrm{supp}(\bar{s}^{\mathrm{c}})$
`// Stage II: Feasibility`
`// Primal Feasibility`
$W^{\mathrm{p}} \leftarrow \{(\bar{w}, \hat{w}) : \bar{w} \in \varphi_N(W^\perp), \bar{w} + \hat{w} = 0\}$
$d^{\mathrm{p}} \leftarrow (d_N, \mathbf{0}_N) + (\beta^{\mathrm{c}})^{-1}(\|d_N\|_1 + 1)\|(\bar{s}^{\mathrm{c}}_N, \hat{s}^{\mathrm{c}}_N)\|_1(\bar{w}^{\mathrm{c}}_N, \hat{w}^{\mathrm{c}}_N)$
$(s^{\mathrm{p}}, x^{\mathrm{p}}, w^{\mathrm{p}}, v^{\mathrm{p}}) \leftarrow \mathrm{IPM}(W^{\mathrm{p}}, (\bar{s}^{\mathrm{c}}_N, \hat{s}^{\mathrm{c}}_N), d^{\mathrm{p}}, \beta^{\mathrm{p}})$
$(\bar{s}^{\mathrm{p}}, \hat{s}^{\mathrm{p}}) \leftarrow s^{\mathrm{p}}, (\bar{x}^{\mathrm{p}}, \hat{x}^{\mathrm{p}}) \leftarrow x^{\mathrm{p}}$
$(\bar{w}^{\mathrm{p}}, \hat{w}^{\mathrm{p}}) \leftarrow w^{\mathrm{p}}, (\bar{v}^{\mathrm{p}}, \hat{v}^{\mathrm{p}}) \leftarrow v^{\mathrm{p}}$
if $\hat{x}^{\mathrm{p}} \neq \mathbf{0}_N$ **then**
  return (FARKAS, $(\mathbf{0}_B, \bar{s}^{\mathrm{p}})$)
`// Dual Feasibility`
$W^{\mathrm{d}} \leftarrow \{(\bar{w}, \check{w}) : \bar{w} \in \varphi_B(W), \bar{w} + \check{w} = 0\}$
$c^{\mathrm{d}} \leftarrow (c_B, \mathbf{0}_B) + (\beta^{\mathrm{c}})^{-1}(\|c_B\|_1 + 1)\|(\bar{x}^{\mathrm{c}}_B, \check{x}^{\mathrm{c}}_B)\|_1(\bar{v}^{\mathrm{c}}_B, \check{v}^{\mathrm{c}}_B)$
$(x^{\mathrm{d}}, s^{\mathrm{d}}, v^{\mathrm{d}}, w^{\mathrm{d}}) \leftarrow \mathrm{IPM}(W^{\mathrm{d}}, (\bar{x}^{\mathrm{c}}_B, \check{x}^{\mathrm{c}}_B), c^{\mathrm{d}}, \beta^{\mathrm{d}})$
$(\bar{x}^{\mathrm{d}}, \check{x}^{\mathrm{d}}) \leftarrow x^{\mathrm{d}}, (\bar{s}^{\mathrm{d}}, \check{s}^{\mathrm{d}}) \leftarrow s^{\mathrm{d}}$
$(\bar{v}^{\mathrm{d}}, \check{v}^{\mathrm{d}}) \leftarrow v^{\mathrm{d}}, (\bar{w}^{\mathrm{d}}, \check{w}^{\mathrm{d}}) \leftarrow w^{\mathrm{d}}$
if $\check{s}^{\mathrm{d}} \neq \mathbf{0}_B$ **then**
  return (FARKAS, $(\bar{x}^{\mathrm{d}}, \mathbf{0}_N)$)
`// Stage III: Optimization`
$B^* \leftarrow \mathrm{supp}(s^{\mathrm{d}}), \bar{B} \leftarrow B \setminus B^*$
$N^* \leftarrow \mathrm{supp}(x^{\mathrm{p}}), \bar{N} \leftarrow N \setminus N^*$
$W^* \leftarrow \pi_{[n]\setminus(\bar{B}\cup\bar{N})}\left(\varphi_{[n]\setminus\bar{N}}(W)\right)$
$x' \leftarrow d_{B^*\cup N^*} + L^{W^*}_{N^*}(x^{\mathrm{p}}_{N^*} - d_{N^*})$
$s' \leftarrow c_{B^*\cup N^*} + L^{(W^*)^\perp}_{B^*}(s^{\mathrm{d}}_{B^*} - c_{B^*})$
$\mu^0 \leftarrow (\beta^{\mathrm{c}})^{-1}\|x' \circ s'\|_1$
$x^0 \leftarrow x' + \mu^0(w^{\mathrm{d}}_{B^*}, \mathbf{0}_{N^*})$
$s^0 \leftarrow s' + \mu^0(\mathbf{0}_{B^*}, v^{\mathrm{p}}_{N^*})$
$(x^*, s^*, \bullet, \bullet) \leftarrow \mathrm{IPM}(W^*, x^0, s^0, \beta^*)$
$\tilde{x} \leftarrow L^{\varphi_{[n]\setminus\bar{N}}(W)}_{B^*\cup N^*}(x^*)$
$\tilde{s} \leftarrow L^{\varphi_{[n]\setminus\bar{B}}(W^\perp)}_{B^*\cup N^*}(s^*)$
$\alpha \leftarrow \min\{\alpha \in \mathbb{R} : \tilde{x}_{\bar{B}} + \alpha\bar{w}^{\mathrm{d}}_{\bar{B}} \geq \mathbf{0}_{\bar{B}}\}$
$x'' \leftarrow \tilde{x}_{\bar{B}} + \alpha\bar{w}^{\mathrm{d}}_{\bar{B}}$
$\alpha \leftarrow \min\{\alpha \in \mathbb{R} : \tilde{s}_{\bar{N}} + \alpha\bar{v}^{\mathrm{p}}_{\bar{N}} \geq \mathbf{0}_{\bar{N}}\}$
$s'' \leftarrow \tilde{s}_{\bar{N}} + \alpha\bar{v}^{\mathrm{p}}_{\bar{N}}$
return $\left((x^*, \mathbf{0}_{\bar{N}}, x''), (s^*, \mathbf{0}_{\bar{B}}, s'')\right)$

procedure *IPM***:**
  Input : data $(W \subseteq \mathbb{R}^n, x > \mathbf{0}_n, s > \mathbf{0}_n)$ for (LP-subspace), $\beta \in (0, 1/6)$ with $\|\frac{nxs}{\langle x,s\rangle} - \mathbf{1}\|_2 \leq \beta$.
  Output: (x^*, s^*, v^*, w^*) as in Theorem 6.10

able to solve this problem with combinatorial algorithms (e.g., Bellman-Ford). However, using an IPM we can solve the conic feasibility problem for arbitrary LP with strongly polynomially bounded SLC in strongly polynomial time. An additional benefit of using an IPM instead of combinatorial methods is that we are able to find specific conic solutions which will be useful to directly implement later stages of the initialization procedure.

We consider following the LP of the form (LP-subspace) with data $(W^{\mathrm{c}}, d^{\mathrm{c}}, c^{\mathrm{c}})$, where $d^{\mathrm{c}} = \frac{1}{2}\mathbf{1}_{3n}$, $c^{\mathrm{c}} = \mathbf{1}_{3n}$ and

$$\begin{aligned} W^{\mathrm{c}} &:= \{(\bar{w}, \hat{w}, \check{w}) \in \mathbb{R}^{3n} : \bar{w} - \hat{w} \in W, \bar{w} = -\check{w}\}, \\ (W^{\mathrm{c}})^\perp &:= \{(\bar{v}, \hat{v}, \check{v}) \in \mathbb{R}^{3n} : \bar{v} - \check{v} \in W^\perp, \check{v} = \bar{v} + \hat{v}\}. \end{aligned} \tag{32}$$

The orthogonality of the two subspaces above can be verified, by noting that $\dim(W^{\mathrm{c}}) = n + \dim(W)$ and $\dim((W^{\mathrm{c}})^\perp) = n + \dim(W^\perp) = 2n - \dim(W)$, hence $\dim(W^{\mathrm{c}}) + \dim((W^{\mathrm{c}})^\perp) = 3n$ and furthermore, for vectors $(\bar{w}, \hat{w}, \check{w})$ and $(\bar{v}, \hat{v}, \check{v})$ fulfilling the constraints on the right hand side of (32) we have that

$$\begin{aligned} \langle(\bar{w}, \hat{w}, \check{w}), (\bar{v}, \hat{v}, \check{v})\rangle &= \langle\bar{w}, \bar{v}\rangle + \langle\hat{w}, \hat{v}\rangle + \langle\check{w}, \check{v}\rangle = \langle\bar{w}, \bar{v}\rangle + \langle\hat{w}, \hat{v}\rangle - \langle\bar{w}, \check{v}\rangle \\ &= \langle\bar{w}, \bar{v} - \check{v}\rangle + \langle\hat{w}, \hat{v}\rangle = \langle\bar{w}, \bar{v} - \check{v}\rangle + \langle\hat{w}, \check{v} - \bar{v}\rangle = \langle\bar{w} - \hat{w}, \check{v} - \bar{v}\rangle = 0. \end{aligned} \tag{33}$$

The corresponding LP for data $(W^{\mathrm{c}}, d^{\mathrm{c}}, c^{\mathrm{c}})$ is therefore given as

$$\begin{array}{ll} \min\ \langle c^{\mathrm{c}}, x\rangle & \min\ \langle d^{\mathrm{c}}, s\rangle \\ \quad x \in W^{\mathrm{c}} + d^{\mathrm{c}} & \quad s \in (W^{\mathrm{c}})^\perp + c^{\mathrm{c}} \\ \quad x \geq \mathbf{0}_{3n}, & \quad s \geq \mathbf{0}_{3n}, \end{array} \tag{Init-Conic-LP}$$

Note that $d^{\mathrm{c}} \circ c^{\mathrm{c}} = \frac{1}{2}\mathbf{1}_{3n}$ and therefore, $(d^{\mathrm{c}}, c^{\mathrm{c}})$ is on the central path of (Init-Conic-LP) with parameter $\mu = \frac{1}{2}$. In particular, we can call the IPM on this system and obtain a solution $(x^{\mathrm{c}}, s^{\mathrm{c}}, v^{\mathrm{c}}, w^{\mathrm{c}})$ such that the output guarantees in Theorem 6.10 hold. Let us decompose the output as $x^{\mathrm{c}} = (\bar{x}^{\mathrm{c}}, \hat{x}^{\mathrm{c}}, \check{x}^{\mathrm{c}})$, $s^{\mathrm{c}} = (\bar{s}^{\mathrm{c}}, \hat{s}^{\mathrm{c}}, \check{s}^{\mathrm{c}})$, $v^{\mathrm{c}} = (\bar{v}^{\mathrm{c}}, \hat{v}^{\mathrm{c}}, \check{v}^{\mathrm{c}})$ and $w^{\mathrm{c}} = (\bar{w}^{\mathrm{c}}, \hat{w}^{\mathrm{c}}, \check{w}^{\mathrm{c}})$. Note that for the vector $(\bar{w}, \hat{w}, \check{w}) := \frac{1}{2}(-\mathbf{1}_n, -\mathbf{1}_n, \mathbf{1}_n) \in W^{\mathrm{c}}$, we obtain that $x' := d^{\mathrm{c}} + (\bar{w}, \hat{w}, \check{w}) = (\mathbf{0}_n, \mathbf{0}_n, \mathbf{1}_n)$ is feasible to (LP-subspace) and analogously for $(\bar{v}, \hat{v}, \check{v}) := (-\mathbf{1}_n, \mathbf{0}_n, -\mathbf{1}_n) \in (W^{\mathrm{c}})^\perp$ we have that $s' := c^{\mathrm{c}} + (\bar{v}, \hat{v}, \check{v}) = (\mathbf{0}_n, \mathbf{1}_n, \mathbf{0}_n)$ is feasible to the dual. As $\langle x', s' \rangle = 0$, they are complementary and so primal-dual optimal solutions to (Init-Conic-LP). The primal optimal region is therefore given by $\{(\bar{x}, \hat{x}, \check{x}) \geq \mathbf{0}_{3n} : \bar{x} \in W, \hat{x} = \mathbf{0}_n, \check{x} = \mathbf{1}_n - \bar{x}\} \subseteq \mathcal{P}(W^{\mathrm{c}}, d^{\mathrm{c}})$ and the dual optimal region is given by $\{(\bar{s}, \hat{s}, \check{s}) \geq \mathbf{0}_{3n} : \bar{s} \in W^\perp, \check{s} = \mathbf{1}_n - \bar{s}\} \subseteq \mathcal{D}(W^{\mathrm{c}}, c^{\mathrm{c}})$. By the output guarantee of Theorem 6.10 we have that $B := \operatorname{supp}(x^{\mathrm{c}}) \cap [n]$ and $N := \operatorname{supp}(s^{\mathrm{c}}) \cap [n]$ form a partition of $[n]$, which we extract in Line 7 of Algorithm 1. In particular, we have that the relative interior of the optimal region of the primal of (Init-Conic-LP) is given by $\{(\bar{x}, \hat{x}, \check{x}) \geq \mathbf{0}_{3n} : \bar{x} \in W, \hat{x} = \mathbf{0}_n, \check{x} = \mathbf{1}_n - \bar{x}, \check{x} > \mathbf{0}_n, \bar{x}_B > \mathbf{0}_B\} \subseteq \mathcal{P}(W^{\mathrm{c}}, d^{\mathrm{c}})$ and the relative interior of the optimal dual region is given by $\{(\bar{s}, \hat{s}, \check{s}) \geq \mathbf{0}_{3n} : \bar{s} \in W^\perp, \hat{s} = \mathbf{1}_n - \bar{s}, \hat{s} > \mathbf{0}_n, \bar{s}_N > \mathbf{0}_N\}$.

The conic vectors v^c and w^c fulfill by Theorem 6.10 furthermore the property that

$$\|(\bar{x}^{\mathrm{c}}_B \circ \bar{v}^{\mathrm{c}}_B, \check{x}^{\mathrm{c}}_B \circ \check{v}^{\mathrm{c}}_B) - \mathbf{1}_{2|B|}\|_2 \leq \beta^{\mathrm{c}} \quad \text{and} \quad \|(\bar{s}^{\mathrm{c}}_N \circ \bar{w}^{\mathrm{c}}_N, \hat{s}^{\mathrm{c}}_N \circ \hat{w}^{\mathrm{c}}_N) - \mathbf{1}_{2|N|}\|_2 \leq \beta^{\mathrm{c}} . \tag{34}$$

The properties in (34) are required for the initialization of the feasibility systems. Intuitively, the property above means that the conic vectors are close to the analytic center of the optimal face of the conic feasibility problem.

We further have by Theorem 6.10 the properties that $v^{\mathrm{c}} \in (W^{\mathrm{c}})^\perp$ and $v^{\mathrm{c}}_{\operatorname{supp}(x^{\mathrm{c}})} > \mathbf{0}$. So, in particular,

$$(\bar{v}^{\mathrm{c}}_B, \check{v}^{\mathrm{c}}_B) \in \pi_{(B,B)}((W^{\mathrm{c}})^\perp) \cap \mathbb{R}^{2|B|}_{++} = \{(\bar{v}, \check{v}) : \bar{v} - \check{v} \in \pi_B(W^\perp), \bar{v} > \mathbf{0}_B, \check{v} > \mathbf{0}_B\} . \tag{35}$$

Analogously, we have that $w^{\mathrm{c}} \in W^{\mathrm{c}}$ and $w^{\mathrm{c}}_{\operatorname{supp}(s^{\mathrm{c}})} > \mathbf{0}$. So, in particular,

$$(\bar{w}^{\mathrm{c}}_N, \hat{w}^{\mathrm{c}}_N) \in \pi_{(N,N)}(W^{\mathrm{c}}) \cap \mathbb{R}^{2|N|}_{++} = \{(\bar{w}, \hat{w}) : \bar{w} - \hat{w} \in \pi_N(W), \bar{w} > \mathbf{0}_N, \hat{w} > \mathbf{0}_N\} . \tag{36}$$

7.2.2 Stage II, From strict conic feasibility to analytic centers

In the call to Line 2 we have identified the partition $B \cup N = [n]$ such that the property (C1) in Lemma 7.3 holds for all $i \in B$ and analogously the property (C2) holds for all $i \in N$. In this section we describe how Algorithm 1 solves dual feasibility in lines 17 to 25. The primal feasibility problem is solved analogously in lines 9 to 17. The original dual feasibility problem for data (W, d, c) in form (LP-subspace) is given by $s \in W^\perp + c, s \geq \mathbf{0}$. The subproblems we are considering now are (28) and (29).

Solving the dual feasibility problem on the coordinates in B is the challenging part, as for feasibility on the coordinates in N we can make use of the vector $\bar{s}^c \in W^\perp$ that was found in the previous section and that has the properties that $\bar{s}^c_B = \mathbf{0}_B$ and $\bar{s}^c_N > \mathbf{0}_N$. We are not yet able to decide feasibility of the system (28). However, we can already conclude that the feasible region is bounded, which is certified by the primal conic vector $\bar{x}^{\mathrm{c}}_B \in \varphi_B(W) \cap \mathbb{R}^B_{++} = \pi_B(W^\perp)^\perp \cap \mathbb{R}^B_{++}$. This is crucial, as our goal in this section is to find a vector that approximates the analytic center of the relative interior of the feasible region of (28). We do so by considering the following subspace W^{d} in Line 18 of Algorithm 1. Here,

$$\begin{aligned} W^{\mathrm{d}} &:= \{(\bar{w}, \check{w}) : \bar{w} \in \varphi_B(W), \bar{w} + \check{w} = 0\} , \\ (W^{\mathrm{d}})^\perp &:= \{(\bar{v}, \check{v}) : \bar{v} - \check{v} \in \varphi_B(W)^\perp\} \end{aligned}$$

and the corresponding dual feasibility system (Init-Dual-Feas-LP) is in form (LP-subspace) with data $(W^{\mathrm{d}}, (\bar{x}^{\mathrm{c}}_B, \check{x}^{\mathrm{c}}_B), c^{\mathrm{d}})$, where $c^{\mathrm{d}} = (c_B, \mathbf{0}_B) + (\beta^{\mathrm{c}})^{-1}(\|c_B\|_1 + 1)\|(\bar{x}^{\mathrm{c}}_B, \check{x}^{\mathrm{c}}_B)\|_1(\bar{v}^{\mathrm{c}}_B, \check{v}^{\mathrm{c}}_B)$.

$$\begin{array}{ll} \min \ \left\langle c^{\mathrm{d}}, (\bar{x}_B, \check{x}_B)\right\rangle & \min \ \left\langle (\bar{x}^{\mathrm{c}}_B, \check{x}^{\mathrm{c}}_B), (\bar{s}_B, \check{s}_B)\right\rangle \\ (\bar{x}_B, \check{x}_B) \in W^{\mathrm{d}} + (\bar{x}^{\mathrm{c}}_B, \check{x}^{\mathrm{c}}_B) & (\bar{s}_B, \check{s}_B) \in (W^{\mathrm{d}})^\perp + c^{\mathrm{d}} \\ (\bar{x}_B, \check{x}_B) \geq \mathbf{0}_{2|B|} , & (\bar{s}_B, \check{s}_B) \geq \mathbf{0}_{2|B|} , \end{array} \tag{Init-Dual-Feas-LP}$$

The vector c^{d} is chosen such that the term $(c_B, \mathbf{0}_B)$ gets dominated by the multiple of the conic term $(\bar{v}^{\mathrm{c}}_B, \check{v}^{\mathrm{c}}_B)$. The conic term is not affecting the optimal solution as $(\bar{v}^{\mathrm{c}}_B, \check{v}^{\mathrm{c}}_B) \in (W^{\mathrm{d}})^\perp$ according to (35) and is just needed for centrality of the initial vectors as we will see next. For

$$\mu^{\mathrm{d}} := (\beta^{\mathrm{c}})^{-1}(\|c_B\|_1 + 1)\|(\bar{x}^{\mathrm{c}}_B, \check{x}^{\mathrm{c}}_B)\|_1 \tag{37}$$

we get with (34) that

$$\begin{aligned}\left\|\frac{(\bar{x}^{\mathrm{c}}_B,\check{x}^{\mathrm{c}}_B)\circ c^{\mathrm{d}}}{\mu^{\mathrm{d}}}-\mathbf{1}_{2|B|}\right\|_2 &= \left\|(\bar{x}^{\mathrm{c}}_B,\check{x}^{\mathrm{c}}_B)\circ(\bar{v}^{\mathrm{c}}_B,\check{v}^{\mathrm{c}}_B)-\mathbf{1}_{2|B|}+(\mu^{\mathrm{d}})^{-1}(\bar{x}^{\mathrm{c}}_B,\check{x}^{\mathrm{c}}_B)\circ(c_B,\mathbf{0}_B)\right\|_2\\ &\le \left\|(\bar{x}^{\mathrm{c}}_B,\check{x}^{\mathrm{c}}_B)\circ(\bar{v}^{\mathrm{c}}_B,\check{v}^{\mathrm{c}}_B)-\mathbf{1}_{2|B|}\right\|_2+\left\|(\mu^{\mathrm{d}})^{-1}(\bar{x}^{\mathrm{c}}_B,\check{x}^{\mathrm{c}}_B)\circ(c_B,\mathbf{0}_B)\right\|_2\\ &\le \beta^{\mathrm{c}}+\beta^{\mathrm{c}}=2\beta^{\mathrm{c}}\,,\end{aligned}\tag{38}$$

and therefore using Lemma 6.7 we get that

$$\left((\bar{x}^{\mathrm{c}}_B,\check{x}^{\mathrm{c}}_B),c^{\mathrm{d}}\right)\in\mathcal{N}^2\left(\frac{2\beta^{\mathrm{c}}}{1-2\beta^{\mathrm{c}}}\right)\subseteq\mathcal{N}^2(\beta^{\mathrm{d}})\,.\tag{39}$$

Unlike in the previous section, we are not able to describe the optimal region yet. Now, consider the output $(x^{\mathrm{d}},s^{\mathrm{d}},v^{\mathrm{d}},w^{\mathrm{d}})$ under the algorithm in Theorem 6.10. Again, let us write $x^{\mathrm{d}}=(\bar{x}^{\mathrm{d}},\check{x}^{\mathrm{d}})$, $s^{\mathrm{d}}=(\bar{s}^{\mathrm{d}},\check{s}^{\mathrm{d}})$, $v^{\mathrm{d}}=(\bar{v}^{\mathrm{d}},\check{v}^{\mathrm{d}})$ and $w^{\mathrm{d}}=(\bar{w}^{\mathrm{d}},\check{w}^{\mathrm{d}})$.

If $\check{s}^{\mathrm{d}}\neq\mathbf{0}_B$ as checked in Line 23, then the corresponding primal vector is going to be a Farkas certificate of dual infeasibility. Namely, consider the primal solution corresponding to the update $(\bar{w},\check{w}):=(-\bar{x}^{\mathrm{c}}_B,\check{x}^{\mathrm{c}}_B)\in W^{\mathrm{d}}$ which results in the feasible vector $x':=(\bar{x}^{\mathrm{c}}_B,\check{x}^{\mathrm{c}}_B)+(\bar{w},\check{w})=(\mathbf{0}_B,\bar{x}^{\mathrm{c}}_B+\check{x}^{\mathrm{c}}_B)=(\mathbf{0}_B,\mathbf{1}_B)$. As $\check{s}^{\mathrm{d}}\neq\mathbf{0}_B$, we have that $\left\langle x',s^{\mathrm{d}}\right\rangle=\left\langle\mathbf{1}_B,\check{s}^{\mathrm{d}}\right\rangle>0$. Therefore, x' is not an optimal primal solution. In particular, $x^{\mathrm{d}}_B=x^{\mathrm{d}}_B-x'_B\in\pi_B(W^\perp)$ is a Farkas certificate of infeasibility of System (28) as

$$\left\langle x^{\mathrm{d}}_B,c_B\right\rangle=\left\langle x^{\mathrm{d}}_B-x'_B,c_B\right\rangle=\left\langle x^{\mathrm{d}}-x',(c_B,\mathbf{0}_B)\right\rangle=\left\langle x^{\mathrm{d}}-x',c^{\mathrm{d}}\right\rangle=\left\langle x^{\mathrm{d}}-x',s^{\mathrm{d}}\right\rangle<0\,,\tag{40}$$

where we used that $x'-x^{\mathrm{d}}\in W^{\mathrm{d}}$ and $(\bar{v}^{\mathrm{c}}_B,\check{v}^{\mathrm{c}}_B)\in(W^{\mathrm{d}})^\perp$. The certificate x^{d}_B can be extended to a certificate of dual infeasibility of (LP-subspace) by padding it with zeros on the coordinates in N.

Otherwise, we have that $\check{s}^{\mathrm{d}}=\mathbf{0}_B$ and therefore, $\bar{s}^{\mathrm{d}}\in\varphi_B(W)^\perp=\pi_B(W^\perp)$ and $\bar{s}^{\mathrm{d}}\geq\mathbf{0}_B$ by feasibility. In particular, we have obtained a feasible solution to (28). Furthermore, we obtain conic vectors v^{d} and w^{d} which will become important in the third stage program, where we use them for initialization. Recall that the IPM returns a support-maximal solution. In particular, if for some coordinate $i\in B$ we have that $\bar{s}^{\mathrm{d}}_i=0$, then $s_i=0$ for all feasible solutions to (28). Let $\bar{B}:=B\setminus\operatorname{supp}(s^{\mathrm{d}})$ and $B^*:=\operatorname{supp}(s^{\mathrm{d}})$.

By Theorem 6.10 we obtain that

$$\|\bar{s}^{\mathrm{d}}_{B^*}\circ w^{\mathrm{d}}_{B^*}-\mathbf{1}_{B^*}\|_2\leq\beta^{\mathrm{d}}\,.\tag{41}$$

A completely analogous analysis can be applied for primal feasibility in Algorithm 1 on lines 9 to 17. We either terminate with a certificate of primal infeasibility, or we obtain a feasible solution to the system (29) and conic vectors v^{p} and w^{p} which will be used in the third stage program.

We also find a subset $N^*=\operatorname{supp}(\bar{x}^{\mathrm{p}})$ and a set $\bar{N}:=N\setminus N^*$. Furthermore, we have that

$$\|\bar{x}^{\mathrm{p}}_{N^*}\circ v^{\mathrm{p}}_{N^*}-\mathbf{1}_{N^*}\|_2\leq\beta^{\mathrm{p}}\,.\tag{42}$$

The corresponding orthogonal subspaces for primal feasibility are

$$\begin{aligned}W^{\mathrm{p}}&:=\{(\bar{w},\hat{w}):\bar{w}\in\varphi_N(W^\perp),\bar{w}+\hat{w}=0\}\\(W^{\mathrm{p}})^\perp&:=\{(\bar{v},\hat{v}):\bar{v}-\hat{v}\in\pi_N(W)\}\end{aligned}\tag{43}$$

and the corresponding optimization problem is

$$\begin{aligned}&\min\ \langle d^{\mathrm{p}},(\bar{s}_N,\hat{s}_N)\rangle &&\min\ \left\langle(\bar{s}^{\mathrm{c}}_N,\hat{s}^{\mathrm{c}}_N),(\bar{x}_N,\hat{x}_N)\right\rangle\\&(\bar{s}_N,\hat{s}_N)\in W^{\mathrm{p}}+(\bar{s}^{\mathrm{c}}_N,\hat{s}^{\mathrm{c}}_N) &&(\bar{x}_N,\hat{x}_N)\in(W^{\mathrm{p}})^\perp+d^{\mathrm{p}}\\&(\bar{s}_N,\hat{s}_N)\geq\mathbf{0}_{2|N|}\,, &&(\bar{x}_N,\hat{x}_N)\geq\mathbf{0}_{2|N|}\,,\end{aligned}\tag{Init-Primal-Feas-LP}$$

We are now ready to solve the optimization problem.

7.2.3 Stage III, From analytic centers to optimization

The procedures in Sections 7.2.1 and 7.2.2 allowed us determine the set $B^*\cup N^*\subseteq[n]$ of variables for which strictly feasible solution exist (unless we have decided infeasibility). Accordingly, for $\bar{B}:=B\setminus B^*$ and $\bar{N}:=N\setminus N^*$ we define

$$W^*:=\pi_{[n]\setminus(\bar{B}\cup\bar{N})}\left(\varphi_{[n]\setminus\bar{B}}(W)\right)=\varphi_{B^*\cup N^*}\left(\pi_{[n]\setminus\bar{N}}(W)\right)\,.\tag{44}$$

in Line 28 of Algorithm 1. The subspace W^* is the subspace on which our final call to the interior point method in Theorem 1.3 will be applied. Using the output of the algorithms described in the previous section, we are given $x^p_{N^*}$ and $s^d_{B^*}$ near the analytic center of the following two feasibility systems:

$$x \in \pi_{N^*}(W^*) + x^f_{N^*}, x \geq \mathbf{0}, \qquad \text{and} \qquad s \in \pi_{B^*}((W^*)^\perp) + s^f_{B^*}, s \geq \mathbf{0}. \tag{45}$$

which is certified by $w^{\mathrm{d}}_{B^*}$ and $v^{\mathrm{p}}_{N^*}$, respectively in (41) and (42).

These vectors suffice for initialization of the corresponding subsystem of the original system on the coordinates B^* and N^*, as we will see next.

Consider the vector $x' := d_{B^*\cup N^*} + L^{W^*}_{N^*}(x^{\mathrm{p}}_{N^*} - d_{N^*})$ and analogously on the dual side $s' := c_{B^*\cup N^*} + L^{(W^*)^\perp}_{B^*}(s^{\mathrm{d}}_{B^*} - c_{B^*})$. We require these shifts of d and c as we have to set the coordinates in $\bar{N}$ to zero on the primal side and the coordinates $\bar{B}$ to 0 on the dual side, as we have concluded that all feasible solutions to the original system have these coordinates set to 0.

Using (41) and (42) we can now find vectors near the central path by choosing μ^0 large enough. More precisely, we choose

$$\mu^0 = (\beta^{\mathrm{c}})^{-1}\|x' \circ s'\|_1 \tag{46}$$

in Line 31 of Algorithm 1 and set our initial vectors to be $x^0 := x' + \mu^0(w^{\mathrm{d}}_{B^*}, \mathbf{0}_{N^*})$ an $s^0 := s' + \mu^0(\mathbf{0}_{B^*}, v^{\mathrm{p}}_{N^*})$. These are central as

$$\begin{aligned}\left\|\frac{x^0 \circ s^0}{\mu^0} - \mathbf{1}\right\|_2 &\leq \frac{1}{\mu_0}\|x' \circ s'\|_2 + \|w^{\mathrm{d}}_{B^*} \circ s'_{B^*} - \mathbf{1}_{B^*}\|_2 + \|v^{\mathrm{p}}_{N^*} \circ x'_{N^*} - \mathbf{1}_{N^*}\|_2 \\ &= \frac{1}{\mu_0}\|x' \circ s'\|_2 + \|w^{\mathrm{d}}_{B^*} \circ s^{\mathrm{d}}_{B^*} - \mathbf{1}_{B^*}\|_2 + \|v^{\mathrm{p}}_{N^*} \circ x^{\mathrm{p}}_{N^*} - \mathbf{1}_{N^*}\|_2 \\ &\leq \beta^{\mathrm{c}} + \beta^{\mathrm{d}} + \beta^{\mathrm{p}} \leq \frac{1}{2}\beta^*\end{aligned} \tag{47}$$

This implies by Lemma 6.7 that

$$\left\|\frac{x^0 \circ s^0}{(|B^*| + |N^*|)^{-1}\langle x^0, s^0\rangle} - \mathbf{1}\right\|_2 \leq \frac{\frac{1}{2}\beta^*}{1 - \frac{1}{2}\beta^*} \leq \beta^*, \tag{48}$$

hence (x^0, s^0) is sufficient to apply the algorithm in Theorem 6.10 in Line 34 with the appropriate choice of β^* in Line 1 in Algorithm 1.

This results in primal-dual optimal solutions (x^*, s^*) to the programs

$$\min\langle s^0, x\rangle : x \in W^* + x^0, x \geq \mathbf{0}_{B^*\cup N^*}, \qquad \min\langle x^0, s\rangle : s \in (W^*)^\perp + s^0, s \geq \mathbf{0}_{B^*\cup N^*}. \tag{49}$$

This time, we do not require to extract the conic centrality certificates optimal face of the system, as we are able to lift the solution to the original system directly.

It remains to show that we are able to lift this solution to an optimal solution in the variables $[n]$. Extending the solution on the primal side on $\bar{N}$ is easy, as we set all the values to 0. Analogously, we can set the values on the dual side to 0 on the coordinates in $\bar{B}$. This leaves us with the problem of extending the solution on the primal side to $\bar{B}$ and on the dual side to $\bar{N}$. However, notice that we can utilize the vectors $w^{\mathrm{d}}_{\bar{B}}$ and $v^{\mathrm{p}}_{\bar{N}}$ that were returned in the call to the feasibility solver in the previous section (Section 7.2.2). First, let us lift the solution x^* to a vector on all coordinates. That is, let $\tilde{x} = L^{\varphi_{[n]\setminus\bar{N}}(W)}_{B^*\cup N^*}(x^*)$ and analogously on the dual side let $\tilde{s} = L^{\varphi_{[n]\setminus\bar{B}}(W^\perp)}_{B^*\cup N^*}(s^*)$, see Lines 35 and 36 in Algorithm 1.

The solutions obtained from the optimization algorithm satisfy all the required properties, except that $\tilde{x}_{\bar{B}}$ or $\tilde{s}_{\bar{N}}$ may not be non-negative. However, since $\tilde{s}_{\bar{B}} = \mathbf{0}_{\bar{B}}$, we only need to ensure non-negativity on the primal side of $\bar{B}$, and complementarity holds automatically. To achieve this, we can use the vector $\bar{w}^{\mathrm{d}}$, which satisfies $\bar{w}^{\mathrm{d}} \in \varphi_B(W)$, $\bar{w}^{\mathrm{d}} \geq \mathbf{0}_B$, and $\mathrm{supp}(\bar{w}^{\mathrm{d}}) = \bar{B}$ due to strict complementarity of $\bar{w}^{\mathrm{d}}$ and $\bar{s}^{\mathrm{d}}$. We can augment $\tilde{x}$ by $(\alpha\bar{w}^{\mathrm{d}}, \mathbf{0}_N)$ for sufficiently large $\alpha > 0$ to ensure that all variables in $\bar{B}$ become non-negative (38). Similarly, we use the vector $\bar{v}^{\mathrm{p}}$ to augment $\tilde{s}$ on the dual side (40).

Proof of Theorem 7.1. Let us first show that the operations in Algorithm 1 before and after the calls to the IPM can be computed in strongly polynomial time. The operations for which this is not straightforward are coordinate fixing φ, coordinate projection π, and applying the lifting map L. Given a subspace $W = \ker(\mathbf{A}) \subseteq \mathbb{R}^n$ for some $\mathbf{A} \in \mathbb{R}^{m\times n}$ and a subset $I \subseteq [n]$, we can compute the coordinate fixing

$\varphi_I(W)$ as $\varphi_I(W) = \ker(\mathbf{A}_I)$. In particular, we can compute coordinate fixes in linear time. The coordinate projection $\pi_I(W)$ can be computed as follows. Let $J = [n] \setminus I$ and reorder the columns of $\mathbf{A}$ such that the columns J appear before the columns in I. Let $r = \operatorname{rank}(\mathbf{A}_J)$. Now, let $\mathbf{M}$ be the matrix that arises from $\mathbf{A}$ by performing Gaussian elimination. Then $\mathbf{M}_{[r+1,m],J} = \mathbf{0}_{(m-r)\times|J|}$ and $\pi_I(W) = \ker(\mathbf{M}_{[r+1,m],J})$. For $r = m$, we have $\pi_I(W) = \mathbb{R}^I$. Note that analogous computations can be performed for the dual subspace $W^\perp$, given as $W^\perp = \operatorname{im}(\mathbf{A}^\top)$, to obtain $\varphi_I(W^\perp)$ and $\pi_I(W^\perp)$. Finally, we need to show that the lifting map L can be computed in strongly polynomial time. It is a well-known fact that L is a projection, which can be represented using basic matrix multiplications and inversions, see Proposition B.17.

Further, note that the calls to the IPM uses subspaces, for which the SLC is bounded as the SLC of $\bar{W} := \{(w, -w, \bar{w}) : w - \bar{w} \in W\}$. In Stage I of Algorithm 1 for conic feasibility, the subspace $\bar{W}$ is used. The subsequent calls to the IPM are performed on subspaces of $\bar{W}$ arising from projecting and fixing some coordinates. For every such subspace W', Lemma 6.12 implies $\mathrm{SLC}_\eta(W') \le \mathrm{SLC}_\eta(W) + 2$.

It remains to show that the number of iterations of the IPMs as well as the space used are as claimed. Here, Theorem 6.10 gives us the bound on the number of iterations. Following up each iteration of the algorithm in Theorem 6.10 with the algorithm in Lemma 8.2 and subsequently applying a constant number of standard corrector steps (Proposition B.1) gives us the desired space bound. □

7.3 SLC preserving subspace operations

We now prove Lemma 6.12. We first need an auxiliary statement characterizing the max central path curve; the proof follows using standard arguments in polyhedral combinatorics.

Lemma 7.4. *Given an instance of* (LP-subspace) *with data (W, d, c), $c, d \ge \mathbf{0}_n$, let x^{m} denote the max central path and let $i \in [n]$. Let $g^\star \ge 0$ denote the largest breakpoint of x^{m}_i, that is, $x^{\mathrm{m}}_i(g)$ is an affine function on $[g^\star, \infty)$. Then, for each $g \in [0, g^\star]$, $x^{\mathrm{m}}_i(g) = \lambda v^{(1)}_i + (1-\lambda)v^{(2)}_i$, such that $v^{(1)}$ and $v^{(2)}$ are basic feasible solutions to the primal problem $x \in W + d$, $x \ge \mathbf{0}$, and $\lambda \in [0, 1]$. Moreover, if $g < g^\star$ then $\langle c, x\rangle = g$ for any optimal solution x to $\max\{x_i : x \in \mathcal{P}_g\}$.*

Lemma 6.12. *Let $W \subseteq \mathbb{R}^n$ be a subspace and $\eta \in (0, 1)$. Then, for any index set $I \subseteq [n]$, $\mathrm{SLC}_\eta(\pi_I(W)) \le \mathrm{SLC}_\eta(W) + 1$ and $\mathrm{SLC}_\eta(\varphi_I(W)) \le \mathrm{SLC}_\eta(W) + 1$.*

Proof. Let $J := [n] \setminus I$. Let us start with the first statement: let $\bar{W} = \pi_I(W) = \{w_I : w \in W\}$. Consider vectors $\bar{c}, \bar{d} \in \mathbb{R}^I_+$, and let $\bar{x}^{\mathrm{m}}$ denote the max central path for the LP with data $(\bar{W}, \bar{d}, \bar{c})$. Note that replacing $\bar{d}$ by any $d' \in \bar{W} + \bar{d}$ and $\bar{c}$ by any $c' \in \bar{W}^\perp + c$ does not change the straight line complexities. Thus, we can assume that $\bar{d} \ge 0$ is a primal and $\bar{c}$ is a dual optimal solution. In particular, the optimum value is $\langle \bar{c}, \bar{d}\rangle = 0$.

We construct vectors $c, d \in \mathbb{R}^n_+$ such that for the max central path of the LP with data (W, d, c) and max central path x^{m}, we have $\mathrm{SLC}(\bar{x}^{\mathrm{m}}_i) \le \mathrm{SLC}(x^{\mathrm{m}}_i) + 1$. Let $\bar{\mathcal{V}} \subseteq \mathbb{R}^I$ denote the finite set of all basic feasible solutions to the LP for $(\bar{W}, \bar{d}, \bar{c})$, and

$$M := \max\{\|L^W_I(\bar{v} - \bar{d})\|_\infty : \bar{v} \in \bar{\mathcal{V}}\}\,.$$

Let us define

$$c_i := \begin{cases} \bar{c}_i & \text{if } i \in I\,, \\ 0 & \text{if } i \in J\,. \end{cases} \quad \text{and} \quad d_i := \begin{cases} \bar{d}_i & \text{if } i \in I\,, \\ M & \text{if } i \in J\,. \end{cases}$$

Note that for every $\bar{v} \in \bar{\mathcal{V}}$ and $v = L^W_I(\bar{v} - \bar{d}) + d$, we have $v \in W + d$, $v \ge \mathbf{0}$, that is, v is feasible for the LP with (W, d, c).

Let us select the breakpoint $\bar{g}^\star$ for the function $\bar{x}^{\mathrm{m}}_i$ as in Lemma 7.4. We claim that $\bar{x}^{\mathrm{m}}_i(g) = x^{\mathrm{m}}_i(g)$ for all $i \in I$ and $g \in [0, \bar{g}^\star]$. This immediately yields the desired bound $\mathrm{SLC}(\bar{x}^{\mathrm{m}}_i) \le \mathrm{SLC}(x^{\mathrm{m}}_i) + 1$.

First, we show $\bar{x}^{\mathrm{m}}_i(g) \ge x^{\mathrm{m}}_i(g)$. This follows because for every $x \in W + d$, $x \ge \mathbf{0}$, we have $x_I \in \bar{W} + \bar{d}$, $x_I \ge \mathbf{0}$ and further $\langle \bar{c}, x_I\rangle = \langle c, x\rangle$ since $c_J = \mathbf{0}_J$. For the converse direction $\bar{x}^{\mathrm{m}}_i(g) \le x^{\mathrm{m}}_i(g)$, consider a point $\bar{x} \in \bar{W} + \bar{d}$, $\bar{x} \ge \mathbf{0}$, $\langle \bar{c}, \bar{x}\rangle \le g$ with $\bar{x}^{\mathrm{m}}_i(g) = \bar{x}_i$. By Lemma 7.4, $\bar{x} = \lambda v^{(1)} + (1-\lambda)v^{(2)}$ for $v^{(1)}, v^{(2)} \in \bar{\mathcal{V}}$, $\lambda \in [0, 1]$. Since L^W_I is a linear operator,

$$\|L^W_I(\bar{x} - \bar{d})\|_\infty \le \lambda\|L^W_I(v^{(1)} - \bar{d})\|_\infty + (1-\lambda)\|L^W_I(v^{(2)} - \bar{d})\|_\infty \le M\,,$$

by the definition of M, and therefore $x = L^W_I(\bar{x} - \bar{d}) + d \in W$, $x \ge \mathbf{0}$ is a feasible solution to the larger LP with $\langle c, x\rangle = \langle \bar{c}, \bar{x}\rangle \le g$.

Let us now show the second statement on coordinate deletion. Let $\bar{W} = \varphi_I(W) = \{w_I : w \in W, w_J = \mathbf{0}_J\}$. Consider vectors $\bar{c}, \bar{d} \in \mathbb{R}^I_+$, and let $\bar{x}^{\mathrm{m}}$ denote the max central path for the LP with data $(\bar{W}, \bar{d}, \bar{c})$. As in the first case, we can assume $\bar{d}$ and $\bar{c}$ are primal and dual optimal solutions with $\langle \bar{d}, \bar{c} \rangle = 0$. Recall the circuit imbalance $\kappa_{\mathbf{A}}$ from Definition 3.1, and note that it only depends on the subspace $W = \ker(\mathbf{A})$. Thus, we can use κ_W for the circuit imbalance corresponding to the subspace. Let us define

$$c_i := \begin{cases} \bar{c}_i & \text{if } i \in I\,, \\ \kappa_W \|\bar{c}\|_1 + 1 & \text{if } i \in J\,. \end{cases} \quad \text{and} \quad d_i := \begin{cases} \bar{d}_i & \text{if } i \in I\,, \\ 0 & \text{if } i \in J\,, \end{cases}$$

Let x^{m} denote the max central path for the LP with data (W, d, c). Again, our goal is to show $\mathrm{SLC}_\eta(\bar{x}^{\mathrm{m}}_i) \le \mathrm{SLC}_\eta(x^{\mathrm{m}}_i) + 1$. We select $\bar{g}^\star$ for the function $\bar{x}^{\mathrm{m}}_i$ as in Lemma 7.4. The proof is complete by showing $\bar{x}^{\mathrm{m}}_i(g) = x^{\mathrm{m}}_i(g)$ for all $i \in I$ and $g \in [0, \bar{g}^\star]$. Let us assume $g < \bar{g}^\star$; the case $g = \bar{g}^\star$ follows by continuity.

First, note that $x^{\mathrm{m}}_i(g) \ge \bar{x}^{\mathrm{m}}_i(g)$ for every value $g \ge 0$, since every solution $\bar{x} \in \bar{W} + \bar{d}$, $\bar{x} \ge \mathbf{0}$ maps to $x = (\bar{x}, \mathbf{0}_J)$ such that $x \in W + d$, $x \ge \mathbf{0}$ and $\langle c, x \rangle = \langle \bar{c}, \bar{x} \rangle$. We now show the converse direction $x^{\mathrm{m}}_i(g) \le \bar{x}^{\mathrm{m}}_i(g)$.

For a contradiction, assume that $x^{\mathrm{m}}_i(g) > \bar{x}^{\mathrm{m}}_i(g)$. Let $x \in W + d$, $x \ge \mathbf{0}$ and $\bar{x} \in W + d$, $\bar{x} \ge \mathbf{0}$ be such that $x^{\mathrm{m}}_i(g) = x_i > \bar{x}_i = \bar{x}^{\mathrm{m}}_i(g) = \bar{x}_i$ and $\langle \bar{c}, \bar{x} \rangle, \langle c, x \rangle \le g$. Consider a conformal circuit decomposition of $z = x - (\bar{x}, \mathbf{0}_J) \in W$ as $x - (\bar{x}, \mathbf{0}_J) = \sum_{\ell=1}^k h^{(\ell)}$ as in Proposition 2.4.

Let $K = \{\ell \in [k] : \mathrm{supp}(h^{(\ell)}) \cap J \ne \emptyset\}$. If $K = \emptyset$, then $\mathrm{supp}(x) \subseteq I$ holds, and x_I gives a better solution than $\bar{x}$ for the program defining $\bar{x}^{\mathrm{m}}_i(g)$, a contradiction. Thus, $K \ne \emptyset$. Take any $\ell \in K$; let $j \in \mathrm{supp}(h^{(\ell)}) \cap J$. From the definition of the circuit imbalance, it follows that

$$\left\langle c, h^{(\ell)} \right\rangle \ge c_j h^{(\ell)}_j - \|c_I\|_1 \cdot \|h^{(\ell)}_I\|_\infty > \|\bar{c}\|_1 (\kappa_W h^{(\ell)}_j - \|h^{(\ell)}_I\|_\infty) > 0\,.$$

Consequently, for $y = x - \sum_{\ell \in K} h^{(\ell)}$, we must have $\langle c, y \rangle < \langle c, x \rangle \le g$. By the conformity of the decomposition, we also have $y_i \ge \bar{x}_i$. Further, $\mathrm{supp}(y) \subseteq I$. To summarize, $y_i \ge \bar{x}_i = \bar{x}^{\mathrm{m}}_i$ while $\langle \bar{c}, y_I \rangle < g$, contradicting the last part of Lemma 7.4. □

8 A strongly polynomial rounding procedure

In this section, we develop tools to bound the bit-complexity of the iterates of the SLLS IPM, in order to obtain a strongly polynomial algorithm in the Turing model. We start by defining the sizes of numbers, as in e.g., [GLS88].

Definition 8.1 (Sizes of numbers)**.** For an integer $n \in \mathbb{Z}$, we define the *encoding length of* n as $\mathrm{size}(z) = 1 + \lceil \log_2(|n| + 1) \rceil$. For a rational number $r \in \mathbb{Q}$ represented as $r = p/q$ for integers $p, q \in \mathbb{Z}$, we let $\mathrm{size}(z) = \mathrm{size}(p) + \mathrm{size}(q)$. Further, for a rational vector or matrix, the size is defined as the sum of the sizes of the entries. For matrices and vectors $a_1, a_2, \ldots, a_k$ of possibly different dimensions, we let $\mathrm{size}(a_1, a_2, \ldots, a_k) = \sum_{i=1}^k \mathrm{size}(a_i)$.

Our strongly polynomial implementation of the SLLS IPM algorithm [ADL+23] has two main ingredients. In Appendix B, we present a variant of this algorithm that proves Theorem 6.10. This implementation avoids the square root computations in [ADL+23], even if it is slower by a factor n. The algorithm uses $\mathrm{poly}(n)$ basic arithmetic operations and comparisons.

Moreover, as stated in Theorem 6.10, every iteration can be implemented by a strongly polynomial algorithm in the Turing model. That is, given an iterate $(x, s) \in \mathbb{Q}^{2n}$, implementing an iteration takes $\mathrm{poly}(n)$ time and uses $\mathrm{poly}(n)\,\mathrm{size}(\mathbf{A}, b, c, x, s)$ space. In particular, the next iterate $(x^+, s^+) \in \mathbb{Q}^{2n}$ has $\mathrm{size}(x^+, s^+) = \mathrm{poly}(n)\,\mathrm{size}(\mathbf{A}, b, c, x, s)$. We emphasize that this guarantee in each iteration does not suffice to obtain an strongly polynomial overall IPM algorithm in the Turing model. The standard affine scaling step direction (also used in many other IPMs) is the solution to a linear system where the matrix is determined by the current iterate (x, s). Thus, the encoding length may increase super-polynomially in super-polylogarithmically many iterations.

The main contribution of this section is the following rounding lemma.

Lemma 8.2. *Given* $(x, s) \in \mathcal{N}^2(\beta) \cap \mathbb{Q}^{2n}$, *there exists a strongly polynomial algorithm that finds a point* $(\tilde{x}, \tilde{s}) \in \mathcal{N}^2(3\beta) \cap \mathbb{Q}^{2n}$ *with* $\overline{\mu}(\tilde{x}, \tilde{s}) \le (1 + \beta/n^2)\overline{\mu}(x, s)$ *and* $\mathrm{size}(\tilde{x}, \tilde{s}) \le O(n^7)\,\mathrm{size}(\mathbf{A}, b, c)$.

Thus, after each iterate (x, s) of the SLLS IPM (or after every polylog(n) iterates), we can round it to another iterate $(\hat{x}, \hat{s})$ with possibly slightly worse gap bound in a slightly wider neighborhood. Note that every iteration decreases the gap at least by a factor $1-\beta/O(\sqrt{n})$; thus, we only lose a small fraction of the progress due to the rounding. After each such rounding, we can move back to the smaller neighborhood $\mathcal{N}^2(\beta)$ by at most two corrector steps (see Proposition B.1).

Lemma 8.2 is applicable much more broadly than the SLLS IPM. Whenever a path-following IPM method computes every iteration in strongly polynomial time, this additional rounding is applicable. Thus, if the total number of elementary arithmetic operations and comparisons in the original algorithm is strongly polynomial, then we can modify it using this rounding step to obtain a strongly polynomial algorithm in the Turing model. This is applicable to all LLS IPMs such as [VY96, MT03, DHNV23]. The rounding steps could also be used in weakly polynomial algorithms; however, in case our running time is allowed to depend on the total encoding length $L = \mathrm{size}(\mathbf{A}, b, c)$, it may be more efficient to use other methods such as [GPV23].

It is interesting to note that the new iterate $(\tilde{x}, \tilde{s})$ could have much better duality gap than (x, s): the rounding algorithm itself may take long steps down the central path. In particular, using the rounding it turns out that the number of iterates is at most $O(\sqrt{n}\log(n))$ times the number of vertices of the polytope: adding this rounding step to any (possibly weakly polynomial) path following IPM already achieves the property highlighted in the title *"Interior point methods are not worse than Simplex"* of [ADL+23].

We use the following simple bounds from [GLS88].

Lemma 8.3. *The following bounds hold:*

(i) For rational numbers $r_1, \ldots, r_k$, $\mathrm{size}(\sum_{i=1}^k r_i) \le 2\sum_{i=1}^k \mathrm{size}(r_i)$, and $\mathrm{size}(\prod_{i=1}^k r_i) \le \sum_{i=1}^k \mathrm{size}(r_i)$.

(ii) For an invertible matrix $\mathbf{M} \in \mathbb{Q}^{n\times n}$, $\mathrm{size}(\mathbf{M}^{-1}) \le 4n^2\,\mathrm{size}(\mathbf{M})$.

(iii) For $\mathbf{A} \in \mathbb{Q}^{m\times n}$, $b \in \mathbb{Q}^m$, $c \in \mathbb{Q}^n$, let $\mathcal{P} = \{x \in \mathbb{R}^n \mid \mathbf{A}x = b, x \ge \mathbf{0}_n\}$ and $\mathcal{D} = \{s \in \mathbb{R}^n \mid \exists y \in \mathbb{R}^m, \mathbf{A}^\top y + s = c, s \ge \mathbf{0}_n\}$ denote the primal and dual feasible regions. Then every vertex and extreme ray of $\mathcal{P}$ has size $O(n^3\,\mathrm{size}(\mathbf{A}, b))$, and every vertex and extreme ray of $\mathcal{D}$ has size $O(n^3\,\mathrm{size}(\mathbf{A}, c))$.

We use two further auxiliary tools.

Lemma 8.4 (Anchoring numbers)**.** *Let $\alpha, r \in \mathbb{Q}_{++}$ such that $r/2^n \le \alpha \le 2^n r$, and let $K \in \mathbb{Z}$. Then, there is an $O(\log^2(n+K))$ time strongly polynomial algorithm that returns a number $\hat{\alpha} \in \mathbb{Q}$ with $\alpha \le \hat{\alpha} \le (1+1/K)\alpha$ and $\mathrm{size}(\hat{\alpha}) \le \mathrm{size}(r) + O(n+K)$.*

Proof. Using binary search, in $O(\log n)$ iterations we can find a value $k \in [-n, n]$ such that $2^k r \le \alpha \le 2^{k+1} r$; each iteration takes $O(\log n)$ arithmetic operations. Next, we let $\delta = (1+1/K)$, and with a second binary search in $O(\log K)$ iterations we find a value $j \in [0, K]$ such that $2^k r\delta^{j-1} \le \alpha \le 2^k r\delta^j$; again, each iteration takes $O(\log K)$ arithmetic operations. Then, $\hat{\alpha} = 2^k r\delta^j$ satisfies the claimed properties. □

Lemma 8.5 (Minkowski–Weyl decomposition)**.** *For $\mathbf{A} \in \mathbb{Q}^{m\times n}$, $b \in \mathbb{Q}^m$, let $\mathcal{P} = \{x \in \mathbb{R}^n \mid \mathbf{A}x = b, x \ge \mathbf{0}_n\}$. Assume we are given a point $x \in \mathcal{P}$. Then, there is a strongly polynomial algorithm that returns extreme points $v^{(1)}, \ldots, v^{(k)}$, extreme rays $r^{(1)}, \ldots, r^{(\ell)}$, and multipliers $\lambda_1, \ldots, \lambda_k \ge 0$, $\eta_1, \ldots, \eta_\ell \ge 0$ such that $\sum_{i=1}^k \lambda_i = 1$, $k + \ell \le n+1$, and*

$$x = \sum_{i=1}^k \lambda_i v^{(i)} + \sum_{j=1}^\ell \eta_j r^{(j)}\,.$$

We prove Lemma 8.2 in two steps. First, we show that every iterate (x, s) can be replaced by another feasible pair $(\hat{x}, \hat{s})$ whose normalized gap has bounded bit-complexity. This may involve an unbounded decrease in the gap value, as the algorithm identifies an improving direction along the central path. In the second step, we replace such a point by another one nearby that has approximately the same normalized gap and also bounded bit-complexity.

Lemma 8.6. *Let $(x, s) \in \mathcal{N}^2(\beta)$. There is a strongly polynomial algorithm that returns a solution $(\hat{x}, \hat{s}) \in \mathcal{N}^2(2\beta)$ with $\overline{\mu}(\hat{x}, \hat{s}) \le \overline{\mu}(x, s)$ and a value $\mu^\star \in \mathbb{Q}$ such that $\overline{\mu}(\hat{x}, \hat{s}) \le \mu^\star \le (1+\beta/(4n^2))\overline{\mu}(\hat{x}, \hat{s})$, and $\mathrm{size}(\mu^\star) = O(n^3)\,\mathrm{size}(\mathbf{A}, b, c)$.*

Proof. Let $\mu = \overline{\mu}(x, s)$; thus, $\langle x, s\rangle = n\mu$. Let us construct a Minkowski–Weyl decomposition of $(x, s) \in \mathcal{P} \times \mathcal{D}$ into vertices and extreme rays of $\mathcal{P} \times \mathcal{D}$. Note that every vertex is of the form $(v^{(i)}, u^{(i)})$, where $v^{(i)}$ is a vertex of $\mathcal{P}$ and $u^{(i)}$ is a vertex of $\mathcal{D}$. Similarly, for every extreme ray $(r^{(j)}, t^{(j)})$, the vector $r^{(j)}$ is an extreme ray of $\mathcal{P}$ and $t^{(j)}$ is an extreme ray of $\mathcal{D}$. The number of vertices and extreme rays is at most $2n+1$. We write the decomposition in the form

$$(x,s) = \sum_{i=1}^{k} \lambda_i (v^{(i)}, u^{(i)}) + (r,t)\,,$$

where $\lambda \geq 0$, $\sum_{i=1}^k \lambda_i = 1$, and (r,t) is a ray of $\mathcal{P}\times\mathcal{D}$ (i.e., the sum of all extreme ray terms). Let us partition the vertices according to their objective values:

$$I^- = \{i \in [k] : \overline{\mu}(v^{(i)}, u^{(i)}) \leq \mu\}\,, \quad I^+ = \{i \in [k] : \overline{\mu}(v^{(i)}, u^{(i)}) > \mu\}\,,$$

Let us define

$$(x^-, s^-) = \frac{\sum_{i\in I^-} \lambda_i (v^{(i)}, u^{(i)})}{\sum_{i\in I^-}\lambda_i}\,, \quad (x^+, s^+) = \frac{\sum_{i\in I^+} \lambda_i (v^{(i)}, u^{(i)})}{\sum_{i\in I^+}\lambda_i}\,.$$

Thus, we can write

$$(x,s) = (\bar{x}, \bar{s}) + (r,t), \quad \text{where} \quad (\bar{x}, \bar{s}) = (1-\alpha)(x^-, s^-) + \alpha(x^+, s^+)$$

for some $\alpha \in [0,1]$. That is, $(\bar{x}, \bar{s}) \in \mathcal{P}\times\mathcal{D}$ is the part of (x,s) in the convex hull of the vertices. We also let $\mu^- = \overline{\mu}(x^-, s^-)$ and $\mu^+ = \overline{\mu}(x^+, s^+)$. We note that $I^- \neq \emptyset$ but I^+ might be empty. In that case, we let $(x^+, s^+) = (0,0)$ and set $\alpha = 0$. Let us also define the accuracy parameter

$$\delta := \frac{\beta}{16n^2}\,.$$

Consider the index $i \in I^-$ with $\overline{\mu}(v^{(i)}, u^{(i)})$ maximal. By definition, $\overline{\mu}(v^{(i)}, u^{(i)}) \leq \mu$. Since $v^{(i)}$ and $u^{(i)}$ are primal and dual vertices, $\mathrm{size}(\overline{\mu}(v^{(i)}, u^{(i)})) = O(n^3\,\mathrm{size}(\mathbf{A}, b, c))$ by Lemma 8.3(iii). If $\overline{\mu}(v^{(i)}, u^{(i)}) \geq \mu\delta^2/(4n)$, then using Lemma 8.4, we can find in strongly polynomial time a value $\mu^\star \in \mathbb{Q}$ with $\mu \leq \mu^\star \leq (1+\delta)\mu$ and $\mathrm{size}(\mu^\star) = O(n^3\,\mathrm{size}(\mathbf{A}, b, c))$.

Similarly, whenever $I^+ \neq \emptyset$, we can pick $j \in I^+$ with $\overline{\mu}(v^{(j)}, u^{(j)})$ minimal. If $\overline{\mu}(v^{(j)}, u^{(j)}) \leq 4n\mu/\delta^2$, then we can again find $\mu^\star \in \mathbb{Q}$ with small encoding length satisfying $\mu \leq \mu^\star \leq (1+\delta)\mu$ in strongly polynomial time. In both cases, we output $(\hat{x}, \hat{s}) = (x,s)$ with the value $\mu^\star$. For the rest of the proof, we can assume that

$$\overline{\mu}(v^{(i)}, u^{(i)}) < \frac{\delta^2\mu}{4n} \quad \forall i \in I^- \quad \text{and} \quad \overline{\mu}(v^{(j)}, u^{(j)}) > \frac{4n\mu}{\delta^2} \quad \forall j \in I^+\,. \tag{50}$$

In particular, $\mu^- < \mu\delta^2/(4n)$, and $\mu^+ > 4n\mu/\delta^2$ if $I^+ \neq \emptyset$. We now consider two cases.

Case I: (50) holds and $I^+ \neq \emptyset$. Let us define

$$(x', s') := (x^+, s^+) + \frac{(r,t)}{\alpha} \quad \text{and} \quad \mu' = \overline{\mu}(x', s')\,. \tag{51}$$

Thus, $(x,s) = (1-\alpha)(x^-, s^-) + \alpha(x', s')$. By the linearity of the duality gap (Proposition 6.4), we have $\mu = (1-\alpha)\mu^- + \alpha\mu'$, implying

$$\alpha \leq \frac{\mu}{\mu'}\,. \tag{52}$$

We show the following strong property.

Claim 8.7. *$(x^-, s^-) \leq (1+\delta)(x,s)$, and for every $i \in [n]$, either $x_i^- \geq (1-\delta)x_i$ or $s_i^- \geq (1-\delta)s_i$.*

Proof. First, assume for a contradiction that $x_i^- \leq (1-\delta)x_i$ and $s_i^- \leq (1-\delta)s_i$ for some $i \in [n]$. We claim that $\mu^+ \leq 2n\mu/\delta^2$ must hold, leading to a contradiction with (50).

Since $(1-\alpha)x_i^- + \alpha x_i' = x_i$ and $(1-\alpha)s_i^- + \alpha s_i' = s_i$, the assumptions on x_i^- and s_i^- yield $\alpha x_i' \geq \delta x_i$ and $\alpha s_i' \geq \delta s_i$. Together with (52), we get

$$\delta x_i \leq \frac{\mu}{\mu'} x_i' \quad \text{and} \quad \delta s_i \leq \frac{\mu}{\mu'} s_i'$$

Multiplying these inequalities, we have

$$\delta^2 x_i s_i \leq \left(\frac{\mu}{\mu'}\right)^2 x_i' s_i' \leq n\frac{\mu^2}{\mu'}\,,$$

where the last inequality follows by the definition $\mu' = \langle x', s'\rangle / n$. By $(x, s) \in \mathcal{N}^2(\beta)$, we get $x_i s_i \geq (1-\beta)\mu$, and therefore $(1-\beta)\delta^2\mu' \leq n\mu$. Thus, $\mu' \leq 2n\mu/\delta^2$. Finally, note that $\mu' = \langle x', s'\rangle / n \geq \langle x^+, s^+\rangle / n = \mu^+$, and therefore $\mu^+ \leq 2n\mu/\delta^2$, in contradiction with (50).

Next, assume that $x_i^- \geq (1+\delta)x_i$ for some $i \in [n]$; the case $s_i^- \geq (1+\delta)x_i$ follows analogously. Again, we use $(1-\alpha)x_i^- + \alpha x_i' = x_i$. The lower bound on x_i^- and $x_i' \geq 0$ imply $1-\alpha \leq 1/(1+\delta)$, that is, $\alpha \geq \delta/(1+\delta)$. Together with (52), we get $\mu' \leq (1+\delta)\mu/\delta$. Thus, $\mu^+ \leq 2\mu/\delta$, again a contradiction with (50). ◻

In light of the above claim, let

$$B := \{i \in [n] : x_i^- \geq (1-\delta)x_i\} \quad \text{and} \quad N := \{i \in [n] : s_i^- \geq (1-\delta)s_i\}\,.$$

By the above claim, $B \cup N = [n]$. Further, $B \cap N = \emptyset$ since $\mu^- < \mu\delta^2/(4n)$. Now, we can find a possibly much better solution $(\hat{x}, \hat{s})$ by following the direction $\Delta x = x^- - x$ and $\Delta s = s^- - s$. For each $i \in [n]$ we have $\Delta x_i \Delta s_i \leq n\delta\mu$ in both cases $i \in B$ or $i \in N$, using the near-monotonicity property (Lemma 6.8). Thus, $\|\Delta x \circ \Delta s\| \leq n^{1.5}\delta/\mu < \beta\mu/4$, and therefore Lemma 6.9 on determining the step-size is applicable. We have

$$\gamma := \frac{\|(x+\Delta x)\circ(s+\Delta s)\|}{\mu} = \frac{\|x^- \circ s^-\|}{\mu} \leq \frac{\langle x^-, s^-\rangle}{\mu} = \frac{n\mu^-}{\mu} \leq \delta^2\,.$$

Lemma 6.9 guarantees that for any step-length ϱ such that $(1-\varrho) \geq 4\gamma/\beta$, the iterate

$$(\hat{x}, \hat{s}) := (x, s) + \varrho(\Delta x, \Delta s) = \varrho(x^-, s^-) + (1-\varrho)(x, s)\,.$$

will be in $\mathcal{N}^2(2\beta)$, with $\hat{\mu} = \overline{\mu}(\hat{x}, \hat{s}) \leq 1.5(1-\varrho)\mu$. Hence, we can use any value ϱ such that $(1-\varrho)\mu \geq 4n\mu^-/\beta$. Note also that by the linearity of the gap we can write $\hat{\mu} = \varrho\mu^- + (1-\varrho)\mu$.

It is left to specify a suitable value of ϱ. Again, taking $i \in I^-$ with $\overline{\mu}(v^{(i)}, u^{(i)})$ maximal, we have $\mu^- \leq \overline{\mu}(v^{(i)}, u^{(i)}) < \delta^2\mu/(4n)$ by the assumption (50). Let us pick ϱ such that $\hat{\mu} = \frac{6n}{\beta}\overline{\mu}(v^{(i)}, u^{(i)})$; note that $\hat{\mu} < \mu$. Then, $(\hat{x}, \hat{s})$ and $\mu^\star = \hat{\mu}$ satisfy the requirements of the lemma.

Case II: (50) holds and $I^+ = \emptyset$. We can write $(x, s) = (x^- + r, s^- + t)$ for the rays r and t. Since r and t live in orthogonal linear spaces, $\langle r, t\rangle = 0$. Since both are nonnegative, we must have $r_i = 0$ or $t_i = 0$ for each $i \in [n]$. On the other hand, we must have that

$$x_i s_i = (x_i^- + r_i)(s_i^- + t_i) \geq (1-\beta)\mu > \frac{4n(1-\beta)}{\delta^2}\mu^- \geq \frac{4(1-\beta)}{\delta^2}x_i^- s_i^-\,.$$

Let us define $B = \{i \in [n] \mid r_i = 0\}$ and $N = \{i \in [n] \mid t_i = 0\}$. By the above inequality, B and N form a partition of $[n]$. Using this partition, we can move to a new $(\hat{x}, \hat{s})$ the same way as in the previous case, noting that we have the even stronger $\|\Delta x \circ \Delta s\| = 0$ for $\Delta x = x^- - x$ and $\Delta s = s^- - s$. ◻

Remark 8.8. In both cases of the above proof where we 'shoot down' along the central path, it exhibits a *'polarized'* structure: primal variables in B stay roughly the same and scale down linearly with μ in N; for dual variables, the role of N and B is reversed. We note that this is the crucial structural property exploited in the SLLS steps of [ADL+23]; see also Appendix B.

The following lemma will be the ingredient to complete the proof of Lemma 8.2.

Lemma 8.9. *Let $\mathcal{P} = \{x : \mathbf{A}x = b, x \geq \mathbf{0}_n\} \subset \mathbb{R}^n$ be a polytope with nonempty interior and let y be a vector in the interior of $\mathcal{P}$. For an index set $J \subseteq [n]$, let $R \geq \max_{x\in P,} \|x_J/y_J\|_\infty$ be an integer, and $K > 0$ any integer. There is a strongly polynomial algorithm that computes a vector $z \in P$ such that $\|(z_J - y_J)/y_J\|_2 \leq 1/K$ with* $\text{size}(z) \leq \tilde{O}(n^3\,\text{size}(\mathbf{A}, b) + n\log(K+R))$.

Proof. First, we obtain a Minkowski–Weyl decomposition in strongly polynomial time as in Lemma 8.5 as $y = \sum_{i=1}^k \lambda_i v^i$, $\sum_{i=1}^k \lambda_i = 1$, $\lambda \geq 0$, such that $k \leq n+1$ and each $v^{(i)}$ is a vertex of $\mathcal{P}$. Note that the decomposition does not include extreme rays, since the polytope is bounded. Let $\delta = 1/(2n^2RK)$, and let us round the λ_i's to $\tilde{\lambda}_i$ such that each $\tilde{\lambda}_i$ is an integer multiple of δ, $\sum_{i=1}^k \tilde{\lambda}_i = 1$, $\tilde{\lambda} \geq 0$, and $\|\tilde{\lambda} - \lambda\|_\infty \leq \delta$. To achieve this, we first round down each $\tilde{\lambda}_i$ to the nearest integer multiple of δ. This can be done by $O(n\log(1/\delta)) = O(n\log(nRk))$ comparisons in total by doing a binary search on each coordinate. Let λ_i' denote these coordinates. At this point, $1 - \sum_{i=1}^k \lambda_i' = \ell\delta$ for an integer $\ell \leq n$. We now pick arbitrary ℓ coordinates and set $\tilde{\lambda}_i = \lambda_i' + 1$ for them, and set $\tilde{\lambda}_i = \lambda_i'$ for all other coordinates.

We claim that $z = \sum_{i=1}^k \tilde\lambda_i v_i$ satisfies the requirements. First, note that

$$\left\|\frac{z-y}{y}\right\|_2 \le \sum_{i=1}^k |\lambda_i - \tilde\lambda_i| \cdot \left\|\frac{v^{(i)}}{y}\right\|_2 \le 2Rn^{1.5}\delta < \frac{1}{K},$$

since each $|\lambda_i - \tilde\lambda_i| \le \delta$ and $\|v^{(i)}/y\|_2 \le \sqrt{n}\|v^{(i)}/y\|_\infty \le \sqrt{n}R$ by the definition of R. The bound on the encoding length of z follows since each vertex $v^{(i)}$ have size bounded as $O(n^3\,\mathrm{size}(\mathbf{A}, b))$ (Lemma 8.3(iii)), and each coefficient $\tilde\lambda_i$ is a ratio of two integers $\le n^2RK$. □

We are ready to prove Lemma 8.2.

Proof of Lemma 8.2. Given $(x, s) \in \mathcal{N}^2(\beta) \cap \mathbb{Q}^{n+n}$, we first use Lemma 8.6 to obtain $(\hat x, \hat s) \in \mathcal{N}^2(2\beta)$ and a value $\mu^\star$ with $\hat\mu \le \mu^\star \le (1+\beta/(4n^2))\hat\mu$ for $\hat\mu = \overline{\mu}(\hat x, \hat s)$, satisfying $\mathrm{size}(\mu^\star) = O(n^3\,\mathrm{size}(\mathbf{A}, b, c))$. Let us also fix any $d \in \mathbb{Q}^n$ such that $\mathbf{A}d = b$; by choosing a nonsingular $m \times m$ submatrix $\mathbf{B}$ of $\mathbf{A}$, for $d = \mathbf{B}^{-1}b$ we get $\mathrm{size}(d) = O(n^3\,\mathrm{size}(\mathbf{A}, b))$; this d can be fixed throughout the algorithm.

Let us now apply Lemma 8.9 with $K = \lceil 8n/\beta \rceil$ on the primal-dual polyhedron, and the index set J corresponding to the x' and s' variables.

$$Q = \left\{(x', y', s') \in \mathbb{R}^{n+m+n} \mid \mathbf{A}x' = b\,,\ \mathbf{A}^\top y' + s' = c, x', s' \ge \mathbf{0}_n\,, \langle c, x'\rangle + \langle d, s'\rangle \le \langle c, d\rangle + n\mu^\star \right\}$$

and the input point $(\hat x, \hat s)$. We note that Q is in fact a polytope. For the x' and s' variables, we have both lower and upper bounds. Hence, if there is an infinite ray in Q, it is supported on the y' variables. But since $\mathbf{A}$ is assumed to have full row rank, for any $y' \ne \mathbf{0}$ we have $\mathbf{A}^\top y' \ne \mathbf{0}$. Hence, no such ray may exist.

Note also that $\langle c, x'\rangle + \langle d, s'\rangle \le \langle c, d\rangle + n\mu^\star$ is equivalent to $\langle x', s'\rangle \le n\mu^\star$ by Proposition 6.3. In particular, $(\hat x, \hat y, \hat s) \in Q$, where $\hat y$ is such that $\mathbf{A}^\top \hat y + \hat s = c$. By the centrality of $(\hat x, \hat s)$, using Lemma 6.8 with $\hat\mu/\mu \le 1 + \delta$ and Proposition 6.5, one can check that $R = 2n$ is a valid bound for J. Further, the size of the description of Q is $\mathrm{size}(\mathbf{A}, b, c, d, \mu^\star) = O(n^3\,\mathrm{size}(\mathbf{A}, b, c))$. Therefore, the algorithm returns a point (x', s') with $\mathrm{size}(x', s') = \tilde O(n^6\,\mathrm{size}(\mathbf{A}, b, c))$ and

$$\left\|\frac{x'}{\hat x} - \mathbf{1}_n\right\|_2 + \left\|\frac{s'}{\hat s} - \mathbf{1}_n\right\|_2 \le \sqrt{2}\left(\left\|\frac{x'}{\hat x} - \mathbf{1}_n\right\|_2^2 + \left\|\frac{s'}{\hat s} - \mathbf{1}_n\right\|_2^2\right)^{1/2} \le \frac{\sqrt{2}}{K} \le \frac{\beta}{4n}. \tag{53}$$

We can bound the ratio $\mu'/\hat\mu$ by

$$\left(1 + \frac{\beta}{4n^2}\right)\hat\mu \ge \mu^\star \ge \mu' = \frac{1}{n}\sum_{i=1}^n x_i's_i' \ge \frac{(1-1/K)^2}{n}\sum_{i=1}^n \hat x_i \hat s_i = \left(1 - \frac{\beta}{4n}\right)^2 \hat\mu. \tag{54}$$

We are ready to show that $(x', s') \in \mathcal{N}^2(3\beta)$.

$$\begin{aligned}\left\|\frac{x' \circ s'}{\mu'} - \mathbf{1}_n\right\|_2 &= \left\|\left(\frac{x'}{\hat x} - \mathbf{1}_n + \mathbf{1}_n\right) \circ \left(\frac{s'}{\hat s} - \mathbf{1}_n + \mathbf{1}_n\right) \circ \frac{\hat x \circ \hat s}{\mu'} - \mathbf{1}_n\right\|_2 \\ &\le \left(\left\|\frac{x'}{\hat x} - \mathbf{1}_n\right\|_2 + \left\|\frac{s'}{\hat s} - \mathbf{1}_n\right\|_2 + \left\|\frac{x'}{\hat x} - \mathbf{1}_n\right\|_2 \left\|\frac{s'}{\hat s} - \mathbf{1}_n\right\|_2\right)\left\|\frac{\hat x \circ \hat s}{\mu'}\right\|_\infty + \left\|\frac{\hat x \circ \hat s}{\mu'} - \mathbf{1}_n\right\|_2\end{aligned} \tag{55}$$

Using (53), and bringing out a factor $\hat\mu/\mu'$ in both terms, we get

$$\begin{aligned}\left\|\frac{x' \circ s'}{\mu'} - \mathbf{1}_n\right\|_2 &\le \frac{\hat\mu}{\mu'}\left(\frac{\beta}{4n} + \left(\frac{\beta}{4n}\right)^2\right)\left\|\frac{\hat x \circ \hat s}{\hat\mu}\right\|_\infty + \frac{\hat\mu}{\mu'}\left\|\frac{\hat x \circ \hat s}{\hat\mu} - \frac{\mu'}{\hat\mu}\mathbf{1}_n\right\|_2 \\ &\le \frac{\hat\mu}{\mu'}\left(\frac{\beta}{2n}(1 + 2\beta) + \left\|\frac{\hat x \circ \hat s}{\hat\mu} - \mathbf{1}_n\right\|_2 + \left\|\frac{\mu'}{\hat\mu}\mathbf{1}_n - \mathbf{1}_n\right\|_2\right) \\ &\le \left(1 + \frac{\beta}{2n}\right)\left(\frac{\beta}{n} + 2\beta + \sqrt{n}\frac{\beta}{4n}\right) \le 3\beta.\end{aligned} \tag{56}$$

Here, the first inequality used (53), the second used that $(\hat x, \hat s) \in \mathcal{N}^2(\beta)$ and the triangle inequality, the third used again $(\hat x, \hat s) \in \mathcal{N}^2(\beta)$ and the two sided bound (54) on $\mu'/\hat\mu$. The final inequality uses $\beta \le 1/6$ and $n \ge 2$. Therefore, $(x', s') \in \mathcal{N}^2(3\beta)$, as required. □

References

[ABGJ18] X. Allamigeon, P. Benchimol, S. Gaubert, and M. Joswig. Log-barrier interior point methods are not strongly polynomial. *SIAM Journal on Applied Algebra and Geometry*, 2(1):140–178, 2018. 3

[AC91] I. Adler and S. Cosares. A strongly polynomial algorithm for a special class of linear programs. *Operations Research*, 39(6):955–960, 1991. 1, 2

[ADL+23] X. Allamigeon, D. Dadush, G. Loho, B. Natura, and L. A. Végh. Interior point methods are not worse than Simplex, 2023. 3, 4, 5, 6, 8, 13, 32, 33, 34, 42, 43, 45, 53, 54, 57, 60

[AGV22] X. Allamigeon, S. Gaubert, and N. Vandame. No self-concordant barrier interior point method is strongly polynomial. In *Proceedings of the 54th Annual ACM Symposium on Theory of Computing (STOC)*, pages 515–528, 2022. 3

[AMO93] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice-Hall, Inc., feb 1993. 2

[BCK+23] J. v. d. Brand, L. Chen, R. Kyng, Y. P. Liu, R. Peng, M. P. Gutenberg, S. Sachdeva, and A. Sidford. A deterministic almost-linear time algorithm for minimum-cost flow. In *2023 IEEE 64th Annual Symposium on Foundations of Computer Science (FOCS)*, 2023. 2

[BSS89] L. Blum, M. Shub, and S. Smale. On a theory of computation and complexity over the real numbers: NP-completeness, recursive functions and universal machines. *Bulletin of the American Mathematical Society*, 21(1):1–46, 1989. 4

[CKL+22] L. Chen, R. Kyng, Y. P. Liu, R. Peng, M. P. Gutenberg, and S. Sachdeva. Maximum flow and minimum-cost flow in almost-linear time. In *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 612–623. IEEE, 2022. 2

[CLS19] M. B. Cohen, Y. T. Lee, and Z. Song. Solving linear programs in the current matrix multiplication time. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 938–942, 2019. 2

[CM94a] E. Cohen and N. Megiddo. New algorithms for generalized network flows. *Mathematical Programming*, 64(1):325–336, 1994. 1, 2

[CM94b] E. Cohen and N. Megiddo. Improved algorithms for linear inequalities with two variables per inequality. *SIAM Journal on Computing*, 23(6):1313–1347, 1994. 1, 6

[DHNV23] D. Dadush, S. Huiberts, B. Natura, and L. A. Végh. A scaling-invariant algorithm for linear programming whose running time depends only on the constraint matrix. *Mathematical Programming*, 2023. (in press). 1, 3, 6, 13, 31, 43

[DKNV22] D. Dadush, Z. K. Koh, B. Natura, and L. A. Végh. An accelerated newton–dinkelbach method and its application to two variables per inequality systems. *Mathematics of Operations Research*, 2022. 2, 6

[DNV20] D. Dadush, B. Natura, and L. A. Végh. Revisiting Tardos's framework for linear programming: Faster exact solutions using approximate solvers. In *Proceedings of the 61st Annual IEEE Symposium on Foundations of Computer Science*, 2020. 1

[DR24] D. Dadush and A. Ramachandran. Strongly polynomial frame scaling to high precision. In *Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2024. to appear. 8, 63

[DS08] S. I. Daitch and D. A. Spielman. Faster approximate lossy generalized flow via interior point algorithms. In *Proceedings of the fortieth annual ACM Symposium on Theory of Computing (STOC)*, pages 451–460. ACM, 2008. 2

[DTK23] I. Diakonikolas, C. Tzamos, and D. M. Kane. A strongly polynomial algorithm for approximate forster transforms and its application to halfspace learning. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, pages 1741–1754, 2023. 8, 63

[Edm67] J. Edmonds. Systems of distinct representatives and linear algebra. *J. Res. Nat. Bur. Standards Sect. B*, 71(4):241–245, 1967. 4

[ENV22] F. Ekbatani, B. Natura, and A. L. Végh. Circuit imbalance measures and linear programming. In *Surveys in Combinatorics 2022*, London Mathematical Society Lecture Note Series, page 64–114. Cambridge University Press, 2022. 6

[Ful68] D. Fulkerson. Networks, frames, blocking systems. *Mathematics of the Decision Sciences, Part I, Lectures in Applied Mathematics*, 2:303–334, 1968. 10

[GJO97] D. Goldfarb, Z. Jin, and J. B. Orlin. Polynomial-time highest-gain augmenting path algorithms for the generalized circulation problem. *Mathematics of Operations Research*, 22(4):793–802, 1997. 2

[GLS88] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric algorithms and combinatorial optimization*, volume 2. Springer Science & Business Media, 1988. 4, 7, 42, 43, 66

[Gon92] C. C. Gonzaga. Path-following methods for linear programming. *SIAM review*, 34(2):167–224, 1992. 32

[GPT91] A. V. Goldberg, S. A. Plotkin, and É. Tardos. Combinatorial algorithms for the generalized circulation problem. *Math. Oper. Res.*, 16(2):351–381, 1991. 1, 2, 6, 52

[GPV23] M. Ghadiri, R. Peng, and S. S. Vempala. The bit complexity of efficient continuous optimization. In *2023 IEEE 64th Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE, 2023. 4, 43

[HN94] D. S. Hochbaum and J. Naor. Simple and fast algorithms for linear and integer programs with two variables per inequality. *SIAM Journal on Computing*, 23(6):1179–1192, 1994. 1, 2, 7, 29

[Hoc04] D. S. Hochbaum. Monotonizing linear programs with up to two nonzeroes per column. *Operations Research Letters*, 32(1):49–58, January 2004. 1, 11

[JSWZ21] S. Jiang, Z. Song, O. Weinstein, and H. Zhang. A faster algorithm for solving general LPs. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*. ACM, June 2021. 2

[Kan39] L. V. Kantorovich. Mathematical methods of organizing and planning production. *Publication House of the Leningrad State University*, page 68, 1939. English translation in *Management Science* 6(4):366-422, 1960. 2

[Kar84] N. Karmarkar. A new polynomial-time algorithm for linear programming. In *Proceedings of the 16th Annual ACM Symposium on Theory of Computing (STOC)*, pages 302–311, 1984. 1

[Kar22] A. Karczmarz. Improved strongly polynomial algorithms for deterministic mdps, 2vpi feasibility, and discounted all-pairs shortest paths. In *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 154–172. SIAM, 2022. 2, 6

[Kha79] L. G. Khachiyan. A polynomial algorithm in linear programming. In *Doklady Academii Nauk SSSR*, volume 244, pages 1093–1096, 1979. 1

[LMT09] G. Lan, R. D. Monteiro, and T. Tsuchiya. A polynomial predictor-corrector trust-region algorithm for linear programming. *SIAM Journal on Optimization*, 19(4):1918–1946, 2009. 3, 55

[LS14] Y. T. Lee and A. Sidford. Path finding methods for linear programming: Solving linear programs in $\tilde{O}(\sqrt{\mathrm{rank}})$ iterations and faster algorithms for maximum flow, part ii. In *55th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 424–433. IEEE, 2014. 2

[Meg83] N. Megiddo. Towards a genuinely polynomial algorithm for linear programming. *SIAM Journal on Computing*, 12(2):347–353, 1983. 1, 2, 6, 29

[MMT98] N. Megiddo, S. Mizuno, and T. Tsuchiya. A modified layered-step interior-point algorithm for linear programming. *Mathematical Programming*, 82(3):339–355, 1998. 3

[MT03] R. D. C. Monteiro and T. Tsuchiya. A variant of the Vavasis-Ye layered-step interior-point algorithm for linear programming. *SIAM Journal on Optimization*, 13(4):1054–1079, 2003. 3, 32, 43

[MT05] R. D. C. Monteiro and T. Tsuchiya. A new iteration-complexity bound for the MTY predictor-corrector algorithm. *SIAM Journal on Optimization*, 15(2):319–347, 2005. 3

[MTY93] S. Mizuno, M. Todd, and Y. Ye. On adaptive-step primal-dual interior-point algorithms for linear programming. *Mathematics of Operations Research*, 18:964–981, 11 1993. 53

[NPT92] C. H. Norton, S. A. Plotkin, and É. Tardos. Using separation algorithms in fixed dimension. *Journal of Algorithms*, 13(1):79–98, 1992. 1, 2

[OV20] N. Olver and L. A. Végh. A simpler and faster strongly polynomial algorithm for generalized flow maximization. *Journal of the ACM (JACM)*, 67(2):1–26, 2020. 1, 2, 6, 7, 29

[Rad04] T. Radzik. Improving time bounds on maximum generalised flow computations by contracting the network. *Theoretical Computer Science*, 312(1):75–97, 2004. 2

[Roc69] R. T. Rockafellar. The elementary vectors of a subspace of R^N. In *Combinatorial Mathematics and Its Applications: Proceedings North Carolina Conference, Chapel Hill, 1967*, pages 104–127. The University of North Carolina Press, 1969. 10

[RTV05] C. Roos, T. Terlaky, and J.-P. Vial. *Interior point methods for linear optimization*. Springer Science & Business Media, 2005. 33

[RW09] M. Restrepo and D. P. Williamson. A simple GAP-canceling algorithm for the generalized maximum flow problem. *Mathematical Programming*, 118(1):47–74, 2009. 2

[Sma98] S. Smale. Mathematical problems for the next century. *The Mathematical Intelligencer*, 20(2):7–15, 1998. 1

[Tar85] É. Tardos. A strongly polynomial minimum cost circulation algorithm. *Combinatorica*, 5(3):247–255, Sep 1985. 1

[Tar86] É. Tardos. A strongly polynomial algorithm to solve combinatorial linear programs. *Operations Research*, pages 250–256, 1986. 1

[TW98] É. Tardos and K. D. Wayne. Simple maximum flow algorithms in lossy networks. In *Proceedings of IPCO, Lecture Notes in Computer Science*, volume 1412, pages 310–324, 1998. 2

[Vai89] P. M. Vaidya. Speeding-up linear programming using fast matrix multiplication. In *30th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 332–337. IEEE, 1989. 2

[vdB20] J. van den Brand. A deterministic linear program solver in current matrix multiplication time. In *Proceedings of the Symposium on Discrete Algorithms (SODA)*, pages 259–278. SIAM, 2020. 2

[vdBLSS20] J. van den Brand, Y. T. Lee, A. Sidford, and Z. Song. Solving tall dense linear programs in nearly linear time, 2020. 2

[vdBZ23] J. van den Brand and D. Zhang. Faster high accuracy multi-commodity flow from single-commodity techniques, 2023. 2

[Vég17] L. A. Végh. A strongly polynomial algorithm for generalized flow maximization. *Mathematics of Operations Research*, 42(2):179–211, 2017. 1, 2, 6, 29

[VY96] S. A. Vavasis and Y. Ye. A primal-dual interior point method whose running time depends only on the constraint matrix. *Mathematical Programming, Series B*, 74(1):79–120, 1996. 1, 2, 6, 7, 13, 30, 32, 33, 43

[Way02] K. D. Wayne. A polynomial combinatorial algorithm for generalized minimum cost flow. *Mathematics of Operations Research*, pages 445–459, 2002. 1, 2, 6

[Ye97] Y. Ye. *Interior-Point Algorithms: Theory and Analysis*. John Wiley and Sons, New York, 1997. 53

[YTM94] Y. Ye, M. J. Todd, and S. Mizuno. An $o(\sqrt{nL})$-iteration homogeneous and self-dual linear programming algorithm. *Mathematics of Operations Research*, 19(1):53–67, February 1994. 7

[ZLY23] M. Zong, Y. T. Lee, and M.-C. Yue. Short-step methods are not strongly polynomial-time. *Mathematical Programming*, July 2023. 3

A Omitted proofs

Lemma 4.4. *Let $(G = (V, E), \gamma, c, u, b)$ be an instance of minimum-cost generalized flow as given by (MGF) satisfying Assumption 4.2, and let x^* be any primal optimal solution. Let x^{m} be the corresponding primal max-central path as given by (MCP-Flow). Consider the instance of minimum-cost generalized circulation on the residual graph G_{x^*}, and let $\hat{x}^{\mathrm{m}}$ be the corresponding primal max-central path. For any $e \in E$ and $\eta \in (0,1)$,*

(i) If $x^*_e = u_e$, then $\mathrm{SLC}_\eta(x^{\mathrm{m}}_e) = 1$.

(ii) If $x^*_e < u_e$, then $\mathrm{SLC}_{\eta/2}(x^{\mathrm{m}}_e) \leq \mathrm{SLC}_\eta(\hat{x}^{\mathrm{m}}_e)$.

(iii) If $e \in E_c$ and $x^*_e > 0$, then $\mathrm{SLC}_{\eta/2}(x^{\mathrm{m}}_{\overleftarrow{e}}) \leq \mathrm{SLC}_\eta(\hat{x}^{\mathrm{m}}_{\overleftarrow{e}})$.

(iv) If $e \in E_c$ and $x^*_e = 0$, then $\mathrm{SLC}_\eta(x^{\mathrm{m}}_{\overleftarrow{e}}) = 1$.

Proof. Every feasible solution x of (MGF) can be mapped to a feasible solution $\hat{x}$ of (MGC), given by $\hat{x}_e := \min(x_e, u_e - x^*_e)$ for all $e \in E(G_{x^*}) \cap E$, and $\hat{x}_{\overleftarrow{e}} := \gamma_e \min(u_e - x_e, x^*_e)$ for all $e \in \mathrm{supp}(x^*)$. This is obtained by first mapping x to a feasible solution $\bar{x}$ of (MGC), defined by $\bar{x}_e := \max(x_e - x^*_e, 0)$ for all $e \in E(G_{x^*}) \cap E$, and $\bar{x}_{\overleftarrow{e}} := \gamma_e \max(x^*_e - x_e, 0)$ for all $e \in \mathrm{supp}(x^*)$. Then, augment $\bar{x}$ along every cycle of the form $(e, \overleftarrow{e})$ to obtain $\hat{x}$. Note that $\langle c, \hat{x}\rangle = \langle c, \bar{x}\rangle = \langle c, x\rangle - \langle c, x^*\rangle$. With the mapped solution $\hat{x}$, we deduce that $\hat{x}^{\mathrm{m}}_e(\lambda) = \min(x^{\mathrm{m}}_e(\lambda), u_e - x^*_e)$ for all $e \in E(G_{x^*}) \cap E$ and $\lambda \geq 0$. Similarly, $\hat{x}^{\mathrm{m}}_{\overleftarrow{e}}(\lambda) = \gamma_e \min(x^{\mathrm{m}}_{\overleftarrow{e}}(\lambda), x^*_e)$ for all $e \in \mathrm{supp}(x^*)$ and $\lambda \geq 0$.

Let $e \in E$ and $\eta \in (0, 1)$. Let $f : \mathbb{R}_+ \to \mathbb{R}_+$ be a continuous piecewise-affine function such that $\eta \hat{x}^{\mathrm{m}}_e \leq f \leq \hat{x}^{\mathrm{m}}_e$. If $x^*_e \leq u_e/2$, then for any $\lambda \geq 0$ where $\hat{x}^{\mathrm{m}}_e(\lambda) = u_e - x^*_e$, we have $f(\lambda) \geq \eta(u_e - x^*_e) \geq \eta u_e/2 \geq \eta x^{\mathrm{m}}_e(\lambda)/2$. Since $f \leq \hat{x}^{\mathrm{m}}_e \leq x^{\mathrm{m}}_e$, it follows that $\eta x^{\mathrm{m}}_e/2 \leq f \leq x^{\mathrm{m}}_e$. Thus, $\mathrm{SLC}_{\eta/2}(x^{\mathrm{m}}_e) \leq \mathrm{SLC}_\eta(\hat{x}^{\mathrm{m}}_e)$. On the other hand, if $x^*_e \geq u_e/2$, then $x^{\mathrm{m}}_e \geq x^*_e \geq u_e/2 \geq x^{\mathrm{m}}_e/2$. In this case, $\mathrm{SLC}_{1/2}(x^{\mathrm{m}}_e) = 1 \leq \mathrm{SLC}_\eta(\hat{x}^{\mathrm{m}}_e)$.

If $e \in F$ and $x^*_e > 0$, then the argument for $x^{\mathrm{m}}_{\overleftarrow{e}}$ is analogous but opposite. Let $h : \mathbb{R}_+ \to \mathbb{R}_+$ be a continuous piecewise-affine function such that $\eta \hat{x}^{\mathrm{m}}_{\overleftarrow{e}} \leq h \leq \hat{x}^{\mathrm{m}}_{\overleftarrow{e}}$. If $x^*_e \leq u_e/2$, then $x^{\mathrm{m}}_{\overleftarrow{e}} \geq u_e - x^*_e \geq u_e/2 \geq x^{\mathrm{m}}_{\overleftarrow{e}}/2$. So, $\mathrm{SLC}_{1/2}(x^{\mathrm{m}}_{\overleftarrow{e}}) = 1 \leq \mathrm{SLC}_\eta(\hat{x}^{\mathrm{m}}_{\overleftarrow{e}})$. On the other hand, if $x^*_e \geq u_e/2$, then for any $\lambda \geq 0$ where $\hat{x}^{\mathrm{m}}_{\overleftarrow{e}}(\lambda) = \gamma x^*_e$, we have $h(\lambda) \geq \eta\gamma_e x^*_e \geq \eta\gamma_e u_e/2 \geq \eta\gamma_e x^{\mathrm{m}}_{\overleftarrow{e}}(\lambda)/2$. As $h \leq \hat{x}^{\mathrm{m}}_{\overleftarrow{e}} \leq \gamma_e x^{\mathrm{m}}_{\overleftarrow{e}}$, it follows that $\eta x^{\mathrm{m}}_{\overleftarrow{e}}/2 \leq h/\gamma_e \leq x^{\mathrm{m}}_{\overleftarrow{e}}$. Hence, $\mathrm{SLC}_{\eta/2}(x^{\mathrm{m}}_{\overleftarrow{e}}) \leq \mathrm{SLC}_\eta(\hat{x}^{\mathrm{m}}_{\overleftarrow{e}})$.

Finally, it is easy to see that $\mathrm{SLC}_\eta(x^{\mathrm{m}}_e) = 1$ for all $e \in E_c$ with $x^*_e = u_e$, and $\mathrm{SLC}_\eta(x^{\mathrm{m}}_{\overleftarrow{e}}) = 1$ for all $e \in E_c$ with $x^*_e = 0$, as required. □

Lemma 4.6. *Suppose $f : \mathbb{R}^2_+ \to \mathbb{R}_+$ is k_1-simple and $g : \mathbb{R}_+ \to \mathbb{R}_+$ is k_2-simple. Then $h : \mathbb{R}_+ \to \mathbb{R}_+$ defined by $h(x) = f(x, g(x))$ is $O(k_1 + k_2)$-simple.*

Proof. Write $f(z) = \bigvee_{i \leq k_1} f^{(i)}$, where $f^{(i)}(z) = \min(v_{i0}, v_{i1}z_1, v_{i2}z_2)$ for $v_{ij} \in \mathbb{R}_+ \cup \{\infty\}$ for all i, j. Then $h = \bigvee_{i \leq k_1} h^{(i)}(x)$ where

$$h^{(i)}(x) := f^{(i)}(x, g(x)) = \min(v_{i0}, v_{i1}x, v_{i2}g(x)).$$

Consider any interval $[x_1, x_2]$ where $g(x)$ is constant; then $h^{(i)}(x)$ is the minimum of a constant and a linear function over this interval. The same holds if $g(x)$ is linear on this interval. So $h^{(i)}$ is a simple function, and hence h is as well.

Given a k-simple function $r : \mathbb{R}_+ \to \mathbb{R}_+$, say that $x \in \mathbb{R}_{++}$ is a *local apex* of r if for some $\epsilon > 0$, r is linear on $(x - \epsilon, x)$ and constant on $(x, x + \epsilon)$. A k-simple function has at most k local apexes, so it suffices to show that h has at most $2k_1 + k_2$ local apexes. Note that a local apex of h must be a local apex of $h^{(i)}$ for some i.

Fix some i. We claim that $h^{(i)}$ has at most 2 local apexes that are not local apexes of g. To see this, first observe that $\frac{v_{i2}g(x)}{v_{i0}}$ and $\frac{v_{i1}x}{v_{i2}g(x)}$ are both nondecreasing functions. The former is clear from monotonicity of g; the latter is immediate after writing $g(x) = \max_{i \leq k_2} \min(w_{i0}, w_{i1}x)$, so that $g(x)/x = \max_i \min(w_{i0}/x, w_{i1})$. This means that there will be values $x_{i1} \leq x_{i2} \in \mathbb{R}_+ \cup \{\infty\}$ so that

$$h^{(i)}(x) = \begin{cases} v_{i1}x & \text{for } x \leq x_{i1} \\ v_{i2}g(x) & \text{for } x_{i1} \leq x \leq x_{i2} \\ v_{i0} & \text{for } x_{i2} \leq x \end{cases}.$$

Then x_{i1} and x_{i2} are the only possible local apexes of $h^{(i)}$ aside from the k_2 (or fewer) local apexes of g. Thus h has at most $2k_1 + k_2$ local apexes in total, as required. □

Lemma 4.8. *Given $f_1, \ldots, f_r : \mathbb{R}_+ \to \mathbb{R}_+$, with f_i being k_i-simple for each i, then $\bigwedge_i f_i$ is $(k_1+k_2+\ldots+k_r)$-simple.*

Proof. It suffices to show this for $r = 2$. Let $f := f_1 \wedge f_2$.

Given a simple function g, say that y is a "flat value" of g if there is an interval for which $g(x) = y$ for all x in the interval. Let $H(g)$ denote the set of flat values of g. Then the g is $|H(g)|$-simple if g is bounded, and $(|H(g)| + 1)$-simple if g is unbounded.

It suffices then to bound $|H(f)|$ by $|H(f_1)| + |H(f_2)|$; if f is unbounded, then f_1 and f_2 are as well. But clearly $H(f) \subseteq H(f_1) \cup H(f_2)$; the claim holds. □

To prove Lemma 4.9, we first show the following claim.

Claim A.1. *Suppose $f = \bigvee_{i\le k} f^{(i)}$ and $g = \bigvee_{j\le \ell} g^{(j)}$, where each $f^{(i)}$ and $g^{(j)}$ are 1-simple, and $f \ge g$. Then for each $j \le \ell$, there exists an $i \le k$ so that $f^{(i)} \ge g^{(j)}$.*

Proof. Fix any $j \le \ell$, and let us express $g^{(j)}(x) = \min(w_0, w_1x_1, \ldots, w_nx_n)$. We can assume $w_a > 0$ for all $0 \le a \le n$, as otherwise $g^{(j)} \equiv 0$ and the claim is trivial. First, we show the lemma under the assumption that $w_a < \infty$ for each a. Defining $y = (w_0/w_1, w_0/w_2, \ldots, w_0/w_n)$, we have $g^{(j)}(y) = w_0$, and the minimum is achieved simultaneously on all terms. Since $f \ge g$, there must be an $i \le k$ with $f^{(i)}(y) \ge g^{(j)}(y)$. Let us denote $f^{(i)}(x) = \min(v_0, v_1x_1, \ldots, v_nx_n)$. We must have $v_0 \ge g^{(j)}(y) = w_0$ and $v_a y_a \ge g^{(j)}(y) = w_0$ for each $1 \le a \le n$. That is, $v_a w_0/w_a \ge w_0$, implying $v_a \ge w_a$ since $0 < w_0 < \infty$. Thus, $f^{(i)} \ge g^{(j)}$ as claimed.

Next, we drop the restriction that every w_a is finite. For any positive integer M, let $g^{(j,M)}$ denote the function $g^{(j,M)}(x) = \min(w_0^M, w_1^M x_1, \ldots, w_n^M x_n)$, where $w_i^M := \min(w_i, M)$. Then $g^{(j,M)} \le g^{(j,N)}$ for $M \le N$, and $g^{(j,M)} \to g^{(j)}$ pointwise as $M \to \infty$. For each M, there is a corresponding i_M so that $f^{(i_M)} \ge g^{(j,M)}$, since we have proved the lemma already in the case that all coefficients are finite. Thus there is an index i so that $f^{(i)} \ge g^{(j,M)}$ for an infinite sequence of choices of M; necessarily $f^{(i)} \ge g^{(j)}$ as required. □

Lemma 4.9. *Suppose $f : \mathbb{R}_+^n \to \mathbb{R}_+$ is k-simple but not $(k-1)$-simple, with $f = \bigvee_{i\le k} f^{(i)}$ for 1-simple functions $f^{(i)}$. Then for any description of f as $f = \bigvee_{i\le \ell} g^{(i)}$ for 1-simple functions $g^{(i)}$, there is a set $S \subseteq \{1, 2, \ldots, \ell\}$ with $|S| = k$ so that $\{f^{(i)} : i \le k\} = \{g^{(i)} : i \in S\}$.*

Proof. By the above claim (applied with both functions equal to f), for each $i \le k$ there is some $\sigma(i)$ so that $g^{(\sigma(i))} \ge f^{(i)}$. Let $g' := \bigvee_{i\le k} g^{(\sigma(i))}$. Then

$$f \ge g' \ge \bigvee_{i\le k} f^{(i)} = f.$$

Applying the claim again, this time with both functions equal to g', we determine that for each $g^{(\sigma(i))}$ there exists an $f^{(j)}$ with $f^{(j)} \ge g^{(\sigma(i))}$. Since $f^{(j)} \ge f^{(i)}$, we must have $i = j$ by the minimal choice of the representation of f. Thus, $g^{(\sigma(i))} = f^{(i)}$ for each i, proving the claim with $S = \{\sigma(i) : i \le k\}$. □

Lemma 4.18. *A flow is an elementary circulation if and only if it is induced by a simple conservative object.*

Proof. For the forward direction, let x be an elementary circulation. By Theorem 2.6 in [GPT91], x is either supported on a conservative cycle, or induced by a conservative object $U = (C, P, D)$ where C is a flow-generating cycle, P is a path, and D is a flow-absorbing cycle. We are clearly done in the former case, so let us focus on the latter; it remains to show that U is simple. Let s be the starting node and t be the final node in P. By shortcutting, we may assume that $V(C) \cap V(P) = \{s\}$ and $V(D) \cap V(P) = \{t\}$. Consider the following two cases:

Case 1: $s \ne t$, i.e., $E(P) \ne \emptyset$. For a contradiction, suppose that there exists a node $r \in V(C) \cap V(D)$. Let C_r and D_r be cycles at r with the same support as C and D respectively. Then, $(C_r, \{r\}, D_r)$ is a conservative object. Any flow induced by this object is supported on a proper subset of $\operatorname{supp}(x)$, a contradiction.

Case 2: $s = t$, i.e., $E(P) = \emptyset$. For a contradiction, suppose that the intersection of C and D is not a path, i.e., this intersection has multiple connected components. Let C_1 be a minimal subpath of C whose endpoints lie in different components; its intermediate nodes do not lie in $V(D)$ by minimality. Let Q be a path with $E(Q) \subset E(D)$ and where $R := C_1 \oplus Q$ is a directed cycle. Let r be the starting node of C_1, and again let C_r and D_r be cycles at r with the same support as C and D respectively.

If $\gamma(R) > 1$, then $(R, \{r\}, D_r)$ is a conservative object. If $\gamma(R) = 1$, then $(R, \{r\}, R)$ is a conservative object. If $\gamma(R) < 1$, then $(C_r, \{r\}, R)$ is a conservative object. In each scenario, the induced flow is supported on a proper subset of $\operatorname{supp}(x)$, a contradiction.

For the converse direction, we observe that removing any single arc from a simple conservative object yields something that is not strongly connected, and hence cannot have a nonzero circulation. □

B An Implementation of the SLLS IPM with Strongly Polynomial Iterations

In this section, we prove Theorem 6.10, which is the adapted IPM. This provides a slower version of the original subspace layered least squares (SLLS) IPM, that satisfies two additional properties. Firstly, each iteration can be implemented in strongly polynomial time. In particular, the bit-complexity of the iterates increases at most polynomially with each iteration. Secondly, we provide certificates that the outputted optimal solutions are close to the analytic centers of the respective optimal faces.

In terms of analysis, we will mainly focus on correctness and ensuring strongly polynomial iterations. For the iteration complexity, the analysis is very similar to that of [ADL+23], so we only provide a sketch of how the arguments differ.

The section is organized as follows. We begin in Appendix B.1 with a primer on standard predictor-corrector methods. The subspace LLS IPM will follow this framework, while occasionally using more aggressive predictor steps. The corresponding subspace LLS step is presented in Appendix B.2. The pseudocode for the IPM is presented in Appendix B.3. In Appendix B.4, we argue the correctness of the IPM, and in Appendix B.5 we quickly sketch how the iteration complexity analysis differs from [ADL+23]. In Appendix B.6, we show how to compute the required steps and step sizes in strongly polynomial time. We note that the IPM will rely on an approximate SVD that is given in Appendix B.7.

For the sake of notational simplicity, throughout this section we use the notation

$$x^\alpha y^\beta := (x_1^\alpha y_1^\beta, \dots, x_n^\alpha y_n^\beta),$$

for $x, y \in \mathbb{R}^n$, $\alpha, \beta > 0$, where we will always restrict to choices of x, y, α, β where the individual coordinates on the right hand side are well-defined. Similarly, for $x \in \mathbb{R}^n_{++}$ and $\alpha > 0$ and $W \subseteq \mathbb{R}^n$ a linear subspace, we use $x^\alpha W := \{x^\alpha w : w \in W\}$.

B.1 Predictor-Corrector Methods

Given $z = (x, s) \in \mathcal{P}_{++} \times \mathcal{D}_{++}$, the search directions commonly used in interior-point methods are obtained as the solution $(\Delta x, \Delta s)$ to the following linear system for some $\nu \in [0, 1]$.

$$\Delta x \in W \tag{57}$$
$$\Delta s \in W^\perp \tag{58}$$
$$s\Delta x + x\Delta s = \nu\mu\mathbf{1} - xs \tag{59}$$

Predictor-corrector methods, such as the Mizuno–Todd–Ye Predictor-Corrector (MTY P-C) algorithm [MTY93], alternate between two types of steps. In *corrector steps*, we use $\nu = 1$. This gives the *centrality direction*, denoted as $\Delta z^{\mathrm{c}} = (\Delta x^{\mathrm{c}}, \Delta s^{\mathrm{c}})$. In *predictor steps*, we use $\nu = 0$. This direction is also called the *affine scaling direction*, and will be denoted as $\Delta z^{\mathrm{a}} = (\Delta x^{\mathrm{a}}, \Delta s^{\mathrm{a}})$ throughout.

Let $z := (x, s) \in \mathcal{N}^2(\beta)$ be our current iterate. In our algorithm, we will first apply a corrector step to get $z^{\mathrm{c}} := z + \Delta z^{\mathrm{c}}$, which will reduce our centrality error by a factor 2, that is, $z^{\mathrm{c}} \in \mathcal{N}^2(\beta/2)$, without changing the gap $\overline{\mu}(z)$. Following this, we apply a predictor step to get $z^+ := z^{\mathrm{c}} + \alpha^{\mathrm{a}}\Delta z^{\mathrm{a}}$, for $\alpha^{\mathrm{a}} \in (0, 1]$, which will make progress along the central path while maintaining that $z^+ \in \overline{\mathcal{N}}^2(\beta)$. Here we slightly abuse notation, by letting $\Delta z^{\mathrm{a}} := (\Delta z^{\mathrm{c}})^{\mathrm{a}}$, that is the predictor direction computed from the recentered iterate z^{c}. The step length $\alpha^{\mathrm{a}} > 0$ will be chosen such that

$$\alpha^{\mathrm{a}} \le \sup\{\alpha \in [0, 1] : \forall \alpha' \in [0, \alpha] : z + \alpha'\Delta z^{\mathrm{a}} \in \mathcal{N}^2(\beta)\}.$$

Thus, we conclude $z^+ = z^{\mathrm{c}} + \alpha^{\mathrm{a}}\Delta z^{\mathrm{a}} \in \overline{\mathcal{N}}^2(\beta)$. We remark that the closure allows us to take a step that goes all the way to an optimal solution. If $z^+ \in \mathcal{N}^2(\beta)$, i.e., if we have not arrived at an optimal solution, then z^+ remains a valid iterate for the next step.

The next proposition summarizes well-known properties, see e.g. [Ye97, Section 4.5.1].

Proposition B.1. *Let $z = (x, s) \in \mathcal{N}^2(\beta)$ for $\beta \in (0, 1/6]$.*

(i) For $z \in \mathcal{N}^2(\beta)$, let Δz^{c} be the corrector direction at z. Then for $z^{\mathrm{c}} = z + \Delta z^{\mathrm{c}}$, we have $\overline{\mu}(z^{\mathrm{c}}) = \overline{\mu}(z)$ and $z^{\mathrm{c}} \in \mathcal{N}^2(\beta/2)$.

(ii) For the affine scaling step, we have $\overline{\mu}(z^+) = (1 - \alpha^{\mathrm{a}})\overline{\mu}(z)$ and $z^+ \in \overline{\mathcal{N}}^2(\beta)$.

(iii) *The affine scaling step-length* α^{a} *can be chosen in the range*

$$0 \le \alpha^{\mathrm{a}} \le \max\left\{\frac{\beta}{2\sqrt{n}}, 1 - \frac{2\|\Delta x^{\mathrm{a}}\Delta s^{\mathrm{a}}\|_2}{\beta\overline{\mu}(z)}\right\}.$$

(iv) *After a sequence of* $O(\sqrt{n}t)$ *corrector and predictor steps, we obtain an iterate* $z' = (x', s') \in \mathcal{N}^2(\beta)$ *such that* $\overline{\mu}(z') \le \overline{\mu}(z)/2^t$.

B.2 Subspace Layered Least Squares Steps

In this section, we describe the main step directions used by [ADL+23] to accelerate the IPM over long straight parts of the central path. On these segments, the central path will be *polarized* according to partition $B \cup N = [n]$ that will be guessed by the algorithm. On such polarized segments, the primal variables indexed by N will scale down linearly with respect to parameter, while the primal variables in B will remain roughly constant. For the dual, the situation is reversed, the variables in B scale down while the variables in N remain roughly constant. We describe these properties more formally below.

Definition B.2 (Polarized Central Path Segment)**.** A segment $\mathrm{CP}[\mu_1, \mu_0] := \{(x^{\mathrm{cp}}(\mu), s^{\mathrm{cp}}(\mu)) : \mu \in [\mu_1, \mu_0]\}$, $0 \le \mu_1 \le \mu_0$, is γ-polarized, $\gamma \in (0, 1]$, if there exists a partition $B \cup N = [n]$ such that

$$x_B^{\mathrm{cp}}(\mu_1) \ge \gamma x_B^{\mathrm{cp}}(\mu_0) \quad \text{and} \quad s_N^{\mathrm{cp}}(\mu_1) \ge \gamma s_N^{\mathrm{cp}}(\mu_0),$$

where the inequalities are to be interpreted coordinate by coordinate.

From the standard monotonocity properties of the central path, one derives for any $\mu \in [\mu_0, \mu_1]$ for the primal side that

1. $\frac{\gamma}{n} x_B^{\mathrm{cp}}(\mu_0) \le x_B^{\mathrm{cp}}(\mu) \le n x_B^{\mathrm{cp}}(\mu_0)$,
2. $\frac{\mu}{\mu_0 n} x_N^{\mathrm{cp}}(\mu_0) \le x_N^{\mathrm{cp}}(\mu) \le \frac{n\mu}{\gamma\mu_0} x_N^{\mathrm{cp}}(\mu_0)$.

The situation for s^{cp} is symmetric with the role of B and N reversed.

The main property of the subspace IPM of [ADL+23] is that it traverses any γ-polarized segment as above in $\mathrm{poly}(n)\log(n/\gamma)$ iterations.

For this purpose, for any iterate $z = (x, s)$ on such a polarized segment, we will search for directions $(\Delta x, \Delta s)$ for which $(x_N + \alpha\Delta x_N) \approx (1 - \alpha)x_N$ and $(x_B + \alpha\Delta x_B) \approx x_B$ for $\alpha \in [0, 1]$ as close to 1 as possible, and vice versa for $(s, \Delta s)$ with B and N reversed. We note that the desired notion of approximation here is multiplicative. This goal helps motivates the subspace layered least squares step below.

Definition B.3 (Subspace LLS direction)**.** Let $z := (x, s) \in \mathcal{N}^2(\beta)$, $\mu = \overline{\mu}(z)$, $B \cup N = [n]$, $B, N \ne \emptyset$, be a partition. Let $V \subseteq W$, $U \subseteq W^\perp$ be linear subspaces satisfying $\dim(\pi_N(V)) = \dim(V)$, and $\dim(\pi_B(U)) = \dim(U)$. The *Subspace LLS* update direction $(\Delta x^\ell, \Delta s^\ell) \in W \times W^\perp$ at z with respect to (B, N, U, V) is defined as

$$\Delta x^\ell := \arg\min_{\delta \in V} \left\|\frac{x_N + \delta_N}{\hat{x}_N}\right\|_2^2$$
$$\Delta s^\ell := \arg\min_{\delta \in U} \left\|\frac{s_B + \delta_B}{\hat{s}_B}\right\|_2^2,$$

where $\hat{x} := \sqrt{x\mu/s} \approx x$ and $\hat{s} := \sqrt{s\mu/x} \approx s$. Note that $\Delta x^\ell, \Delta s^\ell$ are indeed well-defined, as our assumptions that $\dim(\pi_N(V)) = \dim(V)$, $\dim(\pi_B(U)) = \dim(U)$ allows us to uniquely determine $\Delta x^\ell, \Delta s^\ell$ from their coordinates in N and B respectively.

We note that rescaled norms $\|\hat{x}^{-1} \cdot\|_2$ and $\|\hat{s}^{-1} \cdot\|_2$ used to measure error are multiplicatively close to $\|x^{-1} \cdot\|_2$ and $\|s^{-1} \cdot\|_2$, since $\sqrt{1 - \beta}\hat{x} \le x \le \sqrt{1 + \beta}\hat{x}$ and similarly $\sqrt{1 - \beta}\hat{s} \le s \le \sqrt{1 + \beta}\hat{s}$ using that $(1 - \beta)\mu\mathbf{1}_n \le xs \le (1 + \beta)\mu\mathbf{1}_n$ (Proposition 6.6). In particular, for all $z \in \mathbb{R}^n$

$$\sqrt{1 - \beta}\|x^{-1}z\|_2 \le \|\hat{x}^{-1}z\|_2 \le \sqrt{1 + \beta}\|x^{-1}z\|_2, \qquad \sqrt{1 - \beta}\|s^{-1}z\|_2 \le \|\hat{s}^{-1}z\|_2 \le \sqrt{1 + \beta}\|s^{-1}z\|_2. \tag{60}$$

The main advantage in the $\hat{x}, \hat{s}$ rescaling, is that that the subspaces $\hat{x}^{-1}W$ and $\hat{s}^{-1}W^\perp$ remain orthogonal since $\hat{x}\hat{s} = \mu\mathbf{1}_n$. This property often helps simplify the analyses. Even though $\hat{x}, \hat{s}$ contain square roots, we note that the computation of $(\Delta x, \Delta s)$ can be done in strongly polynomial time given appropriate representation of the subspaces U, V (see Proposition B.15). More precisely, $(\Delta x, \Delta s)$ will be functions of the diagonal inner products $\mathrm{diag}(\hat{x}^{-2}) = \mathrm{diag}(s/(x\mu))$ and $\mathrm{diag}(\hat{s}^{-2}) = \mathrm{diag}(x/(s\mu))$, which do not contain square roots. For the sake of notational simplicity, we assume below that the iterate (x, s) is on the central path, that is, $xs = \mu \mathbf{1}_n$. In this case, $\hat{x} = x$ and $\hat{s} = s$.

The Trust Region Step. We now explain the motivation for subspace LLS steps. The starting point for subspace LLS steps is the *trust region* step of Lan, Monteiro and Tsuchiya [LMT09], for which subspace LLS steps will yield suitably good and strongly polynomial computable approximation (the latter point is the main goal of this entire section). Given an iterate $(x, s) > 0$ and partition (B, N), $B, N \neq \emptyset$, the trust-region step computes:

$$\begin{aligned} \Delta x &:= \arg\min \left\{ \left\| x_N^{-1}(x_N + \Delta x_N) \right\|_2 : \left\| x_B^{-1} \Delta x_B \right\|_2 \leq \beta/100, \Delta x \in W \right\}, \\ \Delta s &:= \arg\min \left\{ \left\| s_B^{-1}(s_B + \Delta s_B) \right\|_2 : \left\| s_N^{-1}(\Delta s_N) \right\|_2 \leq \beta/100, \Delta s \in W^\perp \right\}. \end{aligned} \tag{61}$$

The above program models the desire for $(x_N + \alpha\Delta x_N, s_B + \alpha \Delta s_B)$ to "scale down" while enforcing that $(x_B + \alpha \Delta x_B, s_N + \alpha \Delta s_N)$ "barely move" (even for $\alpha = 1$!) as hard constraints. The constant $1/100$ above is arbitrary, indeed, any small enough constant would work (though β is meaningful as it encodes the width of the ℓ_2 neighborhood we wish to stay inside). In terms of the step length $\alpha \in [0, 1]$ one can achieve with these steps, if the maximum value of the primal and dual trust region programs is ε, where $0 \leq \varepsilon < \beta/200$, then one can set the step-length $\alpha = 1 - 100\varepsilon/\beta$ and satisfy $(x + \alpha \Delta x, s + \alpha \Delta s) \in \mathcal{N}^2(\beta)$ (see Proposition B.16). In particular, the primal-dual gap drops to $200\varepsilon/\beta\overline{\mu}(x, s)$ after this step, which is 0 if $\varepsilon = 0$ (indeed, this is generally how finite termination occurs).

The main idea of subspace LLS is to approximate trust region steps by restricting the directions $(\Delta x, \Delta s)$ to live in subspaces for which the norm constraint $\left\| x_B^{-1} \Delta x_B \right\| \leq \beta/100$, $\left\| s_N^{-1} \Delta s_N \right\| \leq \beta/100$ becomes redundant.

For this purpose, the definition of a *cheap lifting subspace* will be fundamental:

Definition B.4 (Cheap Lifting Subspace)**.** Let $W \subseteq \mathbb{R}^n$ be a subspace, $u \in \mathbb{R}^n_{++}$, and (B, N) be a non-trivial partition of $[n]$. Then, $V \subseteq W$ is a cheap lifting subspace for (W, u, B, N) with lifting cost $\tau \geq 0$ if

$$\|u_B x_B\|_2 \leq \tau \|u_N x_N\|_2, \forall x \in V.$$

Note that for the above inequality to hold, we must have that $\dim(\pi_N(V)) = \dim(V)$, since otherwise there exists a vector $x \in V$ with $x_B \neq \mathbf{0}_B$ and $x_N = \mathbf{0}_N$.

In the context of solving the primal trust-region program, if $V \subseteq W$ is a cheap lifting space with respect to (W, x^{-1}, B, N) with lifting cost $\tau = \frac{\beta}{100\sqrt{n}}$, the primal subspace LLS direction with respect to (W, V, N, B)

$$\Delta x^\ell := \arg\min_{\Delta x \in V} \|x_N^{-1}(x_N + \Delta x_N)\|_2$$

automatically satisfies the trust-region constraint $\|x_B^{-1} \Delta x_B^\ell\|_2 \leq \beta/100$. This is because $x_N^{-1} \Delta x_N^\ell$ is the orthogonal projection of $x_N^{-1} x_N = \mathbf{1}_N$ onto subspace $x_N^{-1} W$, and hence $\|x_N^{-1} \Delta x_N^\ell\|_2 \leq \|\mathbf{1}_N\|_2 \leq \sqrt{n}$. By the lifting cost condition on V, we then have $\|x_B^{-1} \Delta x_B^\ell\|_2 \leq \tau \|x_N^{-1} \Delta x_N^\ell\|_2 \leq \sqrt{n}\tau = \frac{\beta}{100}$, as needed. Analogously, any cheap lifting subspace $U \subseteq W$ for (W, s^{-1}, N, B) with lifting cost at most τ will also suffice for the dual subspace LLS direction Δs^ℓ as in Definition B.3 to be feasible.

While low lifting cost subspaces are sufficient to get feasible solutions to the trust-region program, they do not necessarily yield useful approximations. In particular, one can always choose the subspace $V = \{0\}$, which is trivially cheap. To make significant progress along a polarized segment, we will require that the use of cheap lifting subspaces of *maximum dimension* subject to a lifting cost threshold τ. The main idea will be to show that after every successful subspace LLS step, the dimension of cheap-lifting subspace will either increase by one in $\mathrm{poly}(n)$ additional IPM iterations or we will have past the end of the polarized segment. For more details on this argument see Appendix B.5.

We now explain how to find these cheap lifting subspaces and what their achievable dimensions are. This will be achieved by understanding the singular value decomposition of the lifting maps, as defined in Definition 6.2. We first recall the basic properties singular values and singular value decompositions.

Definition B.5 (Singular Value Decomposition)**.** A linear operator $T : U \to V$, where $U \subseteq \mathbb{R}^m$ and $V \subseteq \mathbb{R}^n$ are linear subspaces, admits a singular value decomposition (SVD) $T = \sum_{i=1}^{k} \sigma_i v_i u_i^\top$, $k = \mathrm{rank}(T)$, where $v_1, \dots, v_k \in V$ and $u_k, \dots, u_k \in U$ are orthonormal vectors in their respective subspaces and $\sigma_1 \geq \cdots \geq \sigma_k > 0$. We define $\sigma(T) := (\sigma_1, \dots, \sigma_k)$ to be the vector of singulars values of T listed in non-increasing order. By convention, we define $\sigma_i = 0$ for $i > \mathrm{rank}(T)$. For a matrix $M \in \mathbb{R}^{n\times m}$, we define its singular to be those of the induced linear operator from $\mathbb{R}^m$ to $\mathbb{R}^n$.

One of the most useful ways to characterize singular values is via the following variational characterization.

Proposition B.6 (Max-Min Principle for Singular Values)**.** *Let $T : U \to V$ be a linear operator as in Definition B.5. Then, for $1 \leq i \leq n := \dim(U)$, we have that*

$$\sigma_i(T) = \min_{\dim(S)=n-i+1} \max_{u\in S\setminus\{\mathbf{0}\}} \frac{\|Tx\|_2}{\|x\|_2} = \max_{\dim(S)=i} \min_{x\in S\setminus\{\mathbf{0}\}} \frac{\|Tx\|_2}{\|x\|_2}. \tag{62}$$

where $S \subseteq U$ ranges over the linear subspaces of the prescribed dimensions above.

With the above definitions, we link the achievable dimensions for cheap lifting subspaces as follows.

Lemma B.7. *Let $W \subseteq \mathbb{R}^n$ be a linear subspace, $u \in \mathbb{R}^n_{++}$, $B \cup N = [n]$ be a partition, $B, N \neq \emptyset$, $p = \dim(\pi_N(W))$. Let $L_N^{uW} : \pi_N(uW) \to W$ and $\ell_N^{uW} : \pi_N(W) \to \pi_B(W)$ be the lifting maps as Definition 6.2. Then, given a threshold $\tau \geq 0$, then the maximum dimension of a cheap lifting subspace $V \subseteq W$ for (W, u, B, N) with lifting cost τ is*

$$d = |\{i \in [p] : \sigma_i(\ell_N^{uW}) \leq \tau\}|.$$

Moreover, letting $\sum_{i=1}^{r} \sigma_i y_i w_i^T$, $r = \mathrm{rank}(\ell_N^{uW}) \leq p$ be the singular value decomposition of ℓ^{uW} and $\mathbf{C} = (w_1, \dots, w_p)$ be an extension of the right singular vectors to an orthonormal basis of $\pi_N(uW)$, then if $d > 0$, the subspace $V = u^{-1} L_N^{uW}(\mathrm{im}(\mathbf{C}_{\geq p-d+1}))$ is a cheap lifting subspace for (W, u, B, N) with lifting cost $\sigma_{p-d+1} \leq \tau$.

Proof. We first show that upper bound on d. Let $V \subseteq W$ be a cheap lifting subpace as above with $\dim(V) = d'$, and let $\widehat{V} := uV$ and $\widehat{W} := uW$. For any $y \in \widehat{V} \subseteq \widehat{W}$, by definition of the lifting map, we have that $(\ell_N^{\widehat{W}}(y_N), y_N) \in \widehat{W}$ and $\|y_B\| \leq \|\ell_N^{\widehat{W}}(y_N)\|$. Therefore, without loss of generality, we may assume that

$$\widehat{V} = \{(\ell_N^{\widehat{W}}(y_N), y_N) : y_N \in \pi_N(\widehat{V})\} = L_N^{\widehat{W}}(\pi_N(\widehat{V})).$$

since this can only decrease lifting cost while maintaining the dimension (recall that $\dim(\pi_N(V)) = \dim(V)$ for any cheap lifting subspace). Recall that $\ell^{\widehat{W}} : \pi_N(\widehat{W}) \to \pi_B(\widehat{W})$, $\dim(\pi_N(\widehat{W})) = p$ and that $\dim(\pi_N(\widehat{V})) = d'$ and satisfies $\pi_N(\widehat{V}) \subseteq \pi_N(\widehat{W})$. Therefore, the lifting cost of V satisfies

$$\tau \geq \max_{x\in V, x_N\neq \mathbf{0}_N} \frac{\|u_B x_B\|_2}{\|u_N x_N\|_2} = \max_{y\in \widehat{V}, y_N\neq 0} \frac{\|y_B\|_2}{\|y_N\|_2} = \max_{z\in\pi_N(\widehat{V})} \frac{\ell_N^{\widehat{W}}(z)}{\|z\|} \geq \sigma_{p-d'+1}(\ell_N^{\widehat{W}}),$$

where the last inequality follows from Proposition B.6. Since $\tau \leq \sigma_{p-d'+1}$, we have that $\{i \in [p] : \tau \geq \sigma_i(\ell_N^{\widehat{W}})\} \supseteq \{i \in [p] : i \geq p - d' + 1\}$ (recall that singular values are in non-increasing order). Since the first set has size d and the second has size d', we conclude that $d \geq d'$, as needed.

We now prove the part, which gives an explicit form for the lower bound. If $d > 0$, then clearly $\sigma_{n-d+1}(\ell_N^{\widehat{W}}) \leq \tau$. Then, using $\mathbf{C} = (w_1, \dots, w_p)$ as defined above, it is direct to verify, using the orthogonality properties of the SVD,

$$\max_{x\in \mathrm{im}(\mathbf{C}_{\geq n-d+1})\setminus\{\mathbf{0}_N\}} \frac{\ell_N^{\widehat{W}}(x)}{\|x\|_2} = \sigma_{n-d+1}(\ell_N^{\widehat{W}}).$$

By tracing the reduction from the first part backwards, we conclude that $u^{-1} L_N^{\widehat{W}}(\mathrm{im}(\mathbf{C}_{\geq n-d+1})) \subseteq W$ is a cheap lifting subspace for (W, u, N, B) with lifting cost at most $\sigma_{n-d+1}(\ell_n^{\widehat{W}}) \leq \tau$, as desired. □

We remark that so far we have been treating the tasks of finding primal and dual cheap lifting subspaces separately. Fortunately, it turns out the singular values of the corresponding primal and dual lifting maps are identical. Thus, the corresponding problems of finding cheap lifting subspace on both sides are intimately linked. The formal connection is given below.

Lemma B.8. *(Corollary of [ADL+23, Lemma 2.18]) Let $x, s \in \mathbb{R}^n_{++}$, $\mu = \langle x, s\rangle / n$, satisfy $xs = \mu \mathbf{1}_n$. Then, for $W \subseteq \mathbb{R}^n$ and partition $B \cup N = [n]$, $\sigma(\ell_N^{x^{-1}W}) = \sigma(\ell_B^{s^{-1}W^\perp})$. Furthermore, if $\sum_{i=1}^r \sigma_i u_i v_i^\top$ is the SVD of $\ell_N^{x^{-1}W}$, then $\sum_{i=1}^r \sigma_i v_i u_i^\top$ is the SVD of $\ell_B^{s^{-1}W^\perp}$.*

Given the above, computing cheap lifting subspaces for either the primal or dual, is closely tied to the problem of computing an *approximate* SVD of the lifting map. For the sake of strong polynomial algorithms, one must design such approximate SVDs to be numerically stable, e.g., to maintain bounded bit complexity, as well as to take a strongly polynomial number of arithmetic operations (i.e., depending only the dimension of the underlying matrix).

In Appendix B.7, we present a very simple approximtae SVD for this purpose based on a modification of the Gram-Schmidt orthogonalization procedure. We use this approximate SVD in Cheap-Lift-Subspace.

B.3 The Subspace LLS IPM

In this section, we describe the subspace LLS interior point method.

The associated partition Definition B.3 and Definition B.4 are applicable for any partition $B \cup N = [n]$, $B, N \neq \emptyset$ and $z \in \mathcal{N}^2(\beta)$. Our algorithm chooses a natural partition derived from the relative step lengths in the affine scaling step:

Definition B.9 (Associated partition)**.** For $z = (x, s) \in \mathcal{N}^2(\beta)$, let $(\Delta x^{\mathrm{a}}, \Delta s^{\mathrm{a}})$ be the affine scaling step as in (59) with $\nu = 0$. Let us define the associated partition $\widetilde{B}_z \cup \widetilde{N}_z = [n]$ as

$$\widetilde{B}_z := \left\{ i : \left|\frac{\Delta x_i^{\mathrm{a}}}{x_i}\right| < \left|\frac{\Delta s_i^{\mathrm{a}}}{s_i}\right| \right\} \qquad \widetilde{N}_z := [n] \setminus \widetilde{B}_z .$$

The Subspace LLS Algorithm. Let $\tau := \frac{1}{16\lceil\sqrt{n}\rceil}$ denote the lifting cost threshold. The algorithm is given as IPM with subspace LLS below.

B.4 Correctness

In this subsection, we prove that upon termination, the IPM with subspace LLS satisfies its output requirements, namely it outputs optimal primal and dual solutions that are close to the analytic centers of the corresponding optimal faces. This will mainly depend on the guarantees on the computed step lengths for subspace LLS and affine scaling, which are given in Proposition B.16 and Proposition B.14. We also show that Cheap-Lift-Subspace is correct, which relies on the guarantees of the approximate SVD in Appendix B.7 and the computation of lifting maps Proposition B.17.

Lemma B.10. *The output of IPM with subspace LLS is correct. Furthermore, each iteration of the algorithm runs in strongly polynomial time.*

Proof. We first show that the output $(x^\star, s^\star, v^\star, w^\star)$ satisfies the requirements listed in the output description.

To argue this, we first claim that during the last iteration of the while loop, either the affine scaling step length α^{a} or LLS step length α^ℓ is 1. To see this, note that by assumption $\overline{\mu}(x_0, s_0) > 0$, thus we at least enter the while loop. Secondly, for any iterate (x, s), by Proposition B.1(ii), the corrector step leaves $\overline{\mu}(x, s)$ unchanged, whereas the affine scaling step satisfies $\overline{\mu}(x + \alpha^{\mathrm{a}}\Delta x^{\mathrm{a}}, s + \alpha^{\mathrm{a}}\Delta x^{\mathrm{a}}) = (1 - \alpha^{\mathrm{a}})\overline{\mu}(x, s)$. Similarly, by Proposition B.16, $\overline{\mu}(x + \alpha^\ell \Delta x^\ell, x + \alpha^\ell) \geq 3/4(1 - \alpha^\ell)\overline{\mu}(x, s)$. Thus, the only to exit the loop, corresponding to the condition $\overline{\mu}(x, s) = 0$, is if $\max\{\alpha^{\mathrm{a}}, \alpha^\ell\} = 1$, as needed.

Let $(x, s) \in \overline{\mathcal{N}}^2(\beta)$ be the iterate just after we exit the while loop, and let $(\Delta x^{\mathrm{a}}, \Delta s^\ell)$, α^{a} and $(\Delta x^\ell, \Delta s^\ell)$, α^ℓ be the affine scaling and LLS step during the last iteration. Clearly, $(x^\star, s^\star) = (x, s)$. By line 20, if $\alpha^\ell = 1$, the previous iterate $(\bar{x}, \bar{s}) \in \mathcal{N}^2(\beta/2)$ satisfies $(\bar{x}, \bar{s}) := (x - \Delta x^\ell, s - \Delta s^\ell)$ and if $\alpha^\ell < \alpha^{\mathrm{a}} = 1$, $(\bar{x}, \bar{s}) := (x - \Delta x^{\mathrm{a}}, s - \Delta s^{\mathrm{a}})$.

Let $B := \{i \in [n] : x_i > 0\}$ and $N := \{i \in [n] : s_i > 0\}$ and let $\mu := \overline{\mu}(\bar{x}, \bar{s})$. Note that μ equals the value of the corresponding variable set on either line 25 or line 28.

Assume first that $\alpha^\ell = 1$. Then by the guarantees of Proposition B.16 used on line 17, we must have that $\|\Delta x^\ell \Delta s^\ell\|_2 \leq \beta\mu/8$ and $\|(\bar{x} + \Delta x^\ell)(\bar{s} + \Delta s^\ell)\|_2 = 0$. For $i \in [n]$, using that $|\Delta x_i^\ell \Delta s_i^\ell| \leq \beta\mu/8 < (1 - \beta/2) \leq \bar{x}_i \bar{s}_i$

Algorithm 2: IPM WITH SUBSPACE LLS

Input : Instance of (LP) with primal constraint matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$, rank$(\mathbf{A}) = m$, and initial $(x^0, s^0) \in \mathcal{N}^2(\beta)$, $\beta \in (0, 1/6]$.

Output: $(x^\star, s^\star, v^\star, w^\star)$ satisfying:

1. $x^\star \in \mathcal{P}, s^\star \in \mathcal{D}, \langle x^\star, s^\star\rangle = 0$,
2. $B := \{i \in [n] : x^\star_i > 0\}, N := \{i \in [n] : s^\star_i > 0\}$ satisfy $B \cup N = [n]$,
3. $v^\star \in W^\perp, v^\star_B > 0, w^\star \in W, w^\star_N > 0$,
4. $\|(x^\star_B v^\star_B, s^\star_N w^\star_N) - \mathbf{1}_n\|_2 \le \beta$,

where $W := \ker(\mathbf{A})$, $W^\perp := \mathrm{im}(\mathbf{A}^\top)$.

1 Compute basis $\mathbf{P} \in \mathbb{R}^{n\times(n-m)}$ satisfying $\mathrm{im}(\mathbf{P}) = \ker(\mathbf{A})$; $\mathbf{D} \leftarrow \mathbf{A}^\top$;
2 $(x, s) \leftarrow (x^0, s^0)$;
3 **while** $\overline{\mu}(x, s) > 0$ **do**
4   Compute corrector direction $(\Delta x^c, \Delta s^c)$ at (x, s);
5   $x \leftarrow x + \Delta x^c, s \leftarrow s + \Delta s^c$;
6   Compute affine scaling direction $(\Delta x^{\mathrm{a}}, \Delta s^{\mathrm{a}})$ at (x, s);
7   Set α^{a} for $(\Delta x^{\mathrm{a}}, \Delta s^{\mathrm{a}})$ according to Proposition B.14 with parameter β;
8   $\widetilde{B} \leftarrow \left\{i : \left|\frac{\Delta x^{\mathrm{a}}_i}{x_i}\right| < \left|\frac{\Delta s^{\mathrm{a}}_i}{s_i}\right|\right\}, \widetilde{N} \leftarrow [n] \setminus \widetilde{B}$;
9   **if** $B, N \neq \emptyset$ **then**
10     $\mathbf{MP} \leftarrow$ Cheap-Lift-Subspace$(\mathbf{P}, \frac{1}{x}, \widetilde{B}, \widetilde{N}, \tau)$;
11     $\mathbf{MD} \leftarrow$ Cheap-Lift-Subspace$(\mathbf{D}, \frac{1}{s}, \widetilde{N}, \widetilde{B}, \tau)$;
12   **if MP** *or* **MD** *equals* $\mathbf{0}_n$ **then**
13       $\alpha^\ell \leftarrow 0$;
14   **else**
15       $(\widetilde{V}, \widetilde{U}) \leftarrow (\mathrm{im}(\mathbf{MP}), \mathrm{im}(\mathbf{MD}))$;
16   Compute the subspace LLS direction $(\Delta x^\ell, \Delta s^\ell)$ at (x, s) with respect to $(\widetilde{B}, \widetilde{N}, \widetilde{V}, \widetilde{U})$ using Proposition B.15 on input $(x, s, B, N, \mathbf{MP}, \mathbf{MD})$;
17   Set α^ℓ for $(\Delta x^\ell, \Delta s^\ell)$ according to Proposition B.16 with parameters $\mu = \overline{\mu}(x, s), \beta$;
18   **else**
19     $\alpha^\ell \leftarrow 0$;
20   **if** $\alpha^{\mathrm{a}} > \alpha^\ell$ **then**
21     $(x, s) \leftarrow (x + \alpha^{\mathrm{a}}\Delta x^{\mathrm{a}}, s + \alpha^{\mathrm{a}}\Delta s^{\mathrm{a}})$;
22   **else**
23     $(x, s) \leftarrow (x + \alpha^\ell\Delta x^\ell, s + \alpha^\ell\Delta s^\ell)$;
24 **if** $\alpha^\ell = 1$ **then**
25   $\bar{\mu} \leftarrow \langle x - \Delta x^\ell, s - \Delta s^\ell\rangle / n$;
26   $(v, w) \leftarrow (-\Delta s^\ell/\bar{\mu}, -\Delta x^\ell/\bar{\mu})$;
27 **else**
28   $\bar{\mu} \leftarrow \langle x - \Delta x^{\mathrm{a}}, s - \Delta s^{\mathrm{a}}\rangle / n$;
29   $(v, w) \leftarrow (-\Delta s^{\mathrm{a}}/\bar{\mu}, -\Delta x^{\mathrm{a}}/\bar{\mu})$;
30 **return** (x, s, v, w);

(Proposition 6.6) and $(\bar{x}_i + \Delta x^\ell_i)(\bar{s}_i + \Delta s^\ell_i) = 0$, we conclude that either $x_i = \bar{x}_i + \Delta x^\ell_i > 0$ and $s_i = \bar{s}_i + \Delta s^\ell_i = 0$, and thus $i \in B$, or that $x_i = \bar{x}_i + \Delta x^\ell_i = 0$ and $s_i = \bar{s}_i + \Delta s^\ell_i > 0$, and thus $i \in N$. In particular, we see that (B, N) partition $[n]$.

Given the above, for $(v, w) := (-\Delta s^\ell/\mu, -\Delta x^\ell/\mu)$ as set on 26, we have that $v_B = \bar{s}_B/\mu > 0$ and $u_N = \bar{x}_N/\mu > 0$. Clearly $(v^\star, u^\star) = (v, u)$. Furthermore,

$$
\begin{aligned}
\|(x_B v_B, s_N w_N) - \mathbf{1}_n\|_2 &= \left\|\frac{((\bar{x}_B + \Delta x^\ell_B)\bar{s}_B, (\bar{s}_N + \Delta s^\ell_N)\bar{x}_N)}{\mu} - \mathbf{1}_n\right\|_2 \\
&= \left\|\frac{\bar{x}\bar{s}}{\mu} - \mathbf{1}_n - \frac{\Delta x^\ell \Delta s^\ell}{\mu}\right\|_2 \le \left\|\frac{\bar{x}\bar{s}}{\mu} - 1_n\right\|_2 + \left\|\frac{\Delta x^\ell \Delta s^\ell}{\mu}\right\|_2 \\
&\le \beta/2 + \beta/8 \le \beta, \text{ as needed}. \qquad (63)
\end{aligned}
$$

If $\alpha^\ell < \alpha^{\mathrm{a}} = 1$, the analysis is essentially identical with the role $(\Delta x^\ell, \Delta s^\ell)$ and $(\Delta x^{\mathrm{a}}, \Delta s^{\mathrm{a}})$ switched. To

Algorithm 3: CHEAP-LIFT-SUBSPACE

Input : Matrix $\mathbf{M} \in \mathbb{R}^{n \times d}$, $\mathrm{rank}(\mathbf{M}) = d$, $u \in \mathbb{R}^n_{++}$, partition $B \cup N = [n]$, $B, N \neq \emptyset$, threshold $\tau > 0$.

Output: Matrix $\mathbf{C} \in \mathbb{R}^{n \times k}$, $k \geq 1$, such that for $W := \mathrm{im}(\mathbf{M})$, $V := \mathrm{im}(\mathbf{C})$ the following holds:

1. $V \subseteq W$.
2. $k = \mathrm{rank}(\mathbf{C})$ or $\mathbf{C} = (\mathbf{0}_n)$.
3. $\forall v \in V$, $\|u_B v_B\|_2 \leq (\tau/2)\|u_N v_N\|_2$.
4. $\dim(V) \geq |\{i \in [p] : \sigma_i(\ell_N^{uW}) \leq \frac{\tau}{2^{n+1}\sqrt{n}}\}|$, where ℓ_N^{uW} is in as in Definition 6.2 and $p = \dim(\pi_N(W))$.

// Compute matrix representation of lifting map L_N^{uW}

$\Pi^{uW} \leftarrow \mathrm{diag}(u)\mathbf{M}(\mathbf{M}^\top \mathrm{diag}(u)^2\mathbf{M})^{-1}\mathbf{M}^\top \mathrm{diag}(u)$, the orthogonal projection onto uW;

$\mathbf{L}^u \leftarrow (\Pi^{uW}_{[n],N})(\Pi^{uW}_{N,N})^+$, the lifting map L_N^{uW};

// Approximation of “Cheap Lift” subspace of ℓ_N^{uW}

$(\mathbf{P}, \mathbf{R}, \mathbf{Q}) \leftarrow$ WELL-CONDITIONED-GSO($\mathbf{L}^u_{B,N}$);

$l \leftarrow \min\{i \in [p] : \|\mathbf{Q}_i\|_2^2 \leq \frac{\tau^2}{4^{n+1}}\}$ or -1 if undefined;

if $l \neq -1$ **then**

  $\mathbf{C}^u \leftarrow (\mathbf{PR})_{\geq l}$;

  // Lift subspace into W and rescale

  $\mathbf{C} \leftarrow$ column basis of $\frac{1}{u}\mathbf{L}^u\mathbf{C}^u$;

  return $\mathbf{C}$;

else

  return $\mathbf{0}_n$;

see this, to ensure $\alpha^a = 1$ in Proposition B.14, we must have that $\|\Delta x^a \Delta s^a\|_2 = 0$. Since affine scaling step satisfying $\bar{s}\Delta x^a + \bar{x}\Delta s = -\bar{x}\bar{s}$, we have $(\bar{x} + \Delta x^a)(\bar{s} + \Delta s^a) = \Delta x^a \Delta s^a = 0$. The analysis now follows as above with $(v, w) := (-\Delta s^a/\mu, -\Delta x^a/\mu)$ as set on line (29). Note that by the stronger guarantee $\Delta x^a \Delta s^a = 0$, one can in fact reduce β in (63) to $\beta/2$.

We conclude that the algorithm's output is correct in all cases.

For the strongly polynomial bound on each iteration, we must simply show that all the computations performed within the while loop are strongly polynomial. In particular, we must check the corrector, affine scaling and subspace LLS steps can all be computed in strongly polynomial, and that calls to CHEAP-LIFT-SUBSPACE run in strongly polynomial time. These claims are verified in Proposition B.14 (corrector and affine scaling), Proposition B.15 (subspace LLS direction), Proposition B.16 (subspace LLS step length) and Lemma B.11 (CHEAP-LIFT-SUBSPACE). Thus, each iteration runs in strongly polynomial time as needed. □

Lemma B.11. *CHEAP-LIFT-SUBSPACE is correct and runs in strongly polynomial time.*

Proof. The strongly polynomial running time is direct from the strong polynomiality of Algorithm WELL-CONDITIONED-GSO, and that elementary matrix operations are strongly polynomial. We thus focus on correctness. The algorithm is essentially an algorithmic implementation of the second part of Lemma B.7. The main difference is that one has access only to approximate SVD instead of the exact SVD, and that we have access to matrix representations of the lifting maps (which affects the behavior of approximate SVDs). We give the analysis below.

We run through the steps of the algorithm. We first compute Π^{uW} the orthogonal projection on uW. We the compute $\mathbf{L}^u = \Pi^{uW}(\Pi^{uW}_{NN})^+$. By Proposition B.17, $\mathbf{L}^u \in \mathbb{R}^{n \times N}$ is the matrix that implements the lifting map L_N^{uW} and $\mathbf{L}^u_{B,N}$ implements ℓ_N^{uW}.

Let $p = \dim(\pi_N(uW))$, $(\mathbf{P}, \mathbf{Q}, \mathbf{R})$ be the output of WELL-CONDITIONED-GSO on $\mathbf{L}^u_{B,N}$. Let $r := |\{i \in [p] : \sigma_i(\ell_N^{uW}) \leq \frac{\tau}{2^{n+1}\sqrt{n}}\}|$.

We claim that if $r \geq 1$, then $l := \min\{j \in [p] : \|\mathbf{Q}_{\bullet,j}\|_2^2 \leq \frac{\tau^2}{4^{n+1}}\}$ is well-defined and that $p - l + 1 \geq r$. If $r \geq 1$, then $h := p - r + 1 \in [p]$ satisfies $h = \min\{i \in [p] : \sigma_i(\ell_N^{uW}) \leq \frac{\tau}{2^{n+1}\sqrt{n}}\}$. From here, by Lemma B.19, for $h \leq i \leq p$, we have that

$$\|\mathbf{Q}_i\|_2 \leq \sqrt{n}\sigma_i(\mathbf{L}^u) = \sqrt{n}\sigma_i(\ell_N^{uW}) \leq \sqrt{n}\sigma_h(\ell_N^{uW}) \leq \frac{\tau}{2^{n+1}}.$$

In particular, $l \leq h$, and thus $p - l + 1 \geq r$ as needed. If $l = -1$, then by the above, we must have $r = 0$, and thus the algorithm correctly returns $\mathbf{0}_n$. Assume now $l \in [p]$. We must now show that algorithm returns a cheap lifting subspace for (W, u, B, N) of lifting cost at most $\tau/2$ and dimension at least r.

By Lemma B.19, the matrix $\mathbf{C}^u_l := (\mathbf{PR})_{\bullet,\geq i}$ where satisfies that

$$\max_{x \in \mathrm{im}(\mathbf{C}^u_{\geq l}) \setminus \{0\}} \frac{\|\mathbf{L}^u(x)\|_2}{\|x\|_2} \leq 2^n \|\mathbf{Q}_i\|_2 \leq \tau/2. \tag{64}$$

We will need the following claim, whose proof we leave as an exercise.

Claim B.12. *Let $V \subseteq \mathbb{R}^N$ be any subspace containing $\ker(\mathbf{L}^u_{B,N})$. Then $\mathbf{L}^u(V) = \mathbf{L}^u(V \cap \pi_N(uW))$ and $\dim(V \cap \pi_N(uW)) = \dim(V) + \dim(\pi_N(uW)) - |N|$.*

Assuming the claim, let $V = \mathrm{im}(\mathbf{C}_{\geq l})$ as used by the algorithm. First, we note that V contains the kernel of the lifting matrix $\mathbf{L}^u_{B,N}$. If the kernel is trivial, there is nothing to prove. Otherwise, note that the kernel equals $\mathrm{im}(\mathbf{C}_{\geq k})$ where $k = \min\{i \in [|N|] : \mathbf{Q}_{\bullet,i} = \mathbf{0}_B\}$, where clearly $l \leq k$.

Therefore, letting $V' = V \cap \pi_N(uW)$, we use the claim to deduce that

$$\dim(V') = \dim(V) + \dim(\pi_N(W)) - |N| = (|N| - l + 1) + p - |N| = p - l + 1,$$

and that $\mathbf{L}^u(V') = \mathbf{L}^u(V)$. In particular, the lifting cost of V' with respect (W^u, B, N) is at most $\tau/2$ by (64), and the dimension of V' is $p - l + 1 \geq r$ by the first part. Therefore, the subspace $u^{-1}\mathbf{L}^u(V)$ is a cheap lifting subspace for (W, u, B, N) with lifting cost $\tau/2$ and dimension at least r as needed. Lastly, we make sure to compute a basis $\mathbf{C}$ of $u^{-1}\mathbf{L}^u(V) = u^{-1}\mathbf{L}^u(\mathbf{C}_{\geq l})$, to ensure the outputted matrix has full column rank.

□

B.5 Iteration Complexity

We now quickly sketch why the modified subspace LLS algorithms run an $O(n)$ factor slower in terms of iterations than [ADL+23]. As noted previously, this is immaterial to our main results, and in fact, any $\mathrm{poly}(n)$ slowdown of our IPM, for the sake of achieving strongly polynomial iterations would be sufficient. We restrict the discussion below to this goal.

The main insight structural insight in [ADL+23] is that the central path can be decomposed in at most

$$O\left(\sum_{i=1}^{n} \mathrm{SLC}_\eta(x^{\mathrm{m}}_i)\right)$$

$\gamma := \eta/\mathrm{poly}(n)$-polarized segments, where polarization is as in Definition B.2. Thus, the main challenge is to show that in at most $\mathrm{poly}(n \log(1/\eta))$ iterations, the subspace LLS IPM can traverse any γ-polarized segment.

Let $\mathrm{CP}[\mu_0, \mu_1]$, $0 \leq \mu_0 \leq \mu_1$ be an γ-polarized segment with polarization partition $B \cup N$, $B, N \neq \emptyset$. Note that $\log(\mu_0/\mu_1)$ can be arbitrarily large on such a segment (in fact, infinite as $\mu_1 = 0$ is possible), thus standard path following does not suffice to traverse these segments $O(\mathrm{poly}(n)\log(1/\gamma))$ iterations. To be able to argue rapid progress on such segments, one uses a different, in fact combinatorial quantity, to serve as a potential. Specifically, one uses the number of singular values of the lifting map above a threshold as our potential.

For an iterate $z = (x, s) \in \mathcal{N}^2(\beta)$, define the lifting maps $\ell_z = \ell_N^{\hat{x}^{-1}W}$ and $\ell_z^\perp = \ell_B^{\hat{s}^{-1}W^\perp}$, where $\hat{x}, \hat{s}$ are the normalized iterates as in Definition B.3. As the potential, one examines

$$\zeta(z) :== \left|\left\{i : \sigma_i(\ell_z) > \frac{\tau}{n^{1.5}}\right\}\right|,$$

where $\tau := \frac{1}{16\lceil\sqrt{n}\rceil}$, as above. Recall that the singular values of ℓ_z and $\ell_z^\perp$ are identical, and hence the above serves as a global potential, measuring progress along a segment.

A main property that drives the analysis is that the singular values evolve predictably over the course of a polarized segment. This is given the lemma below.

Lemma B.13 (Stability of singular values on polarized segments)**.** *Let* CP$[\mu_1, \mu_0]$ *be a* γ*-polarized segment of the central path with partition* $B \cup N = [n]$*. Let* $z, \widetilde{z} \in \mathcal{N}^2(\beta)$ *for* $\beta \in (0, 1/6]$*, such that* $\mu := \overline{\mu}(z)$ *and* $\widetilde{\mu} := \overline{\mu}(\widetilde{z})$ *satisfy* $\mu_0 \ge \mu \ge \widetilde{\mu} \ge \mu_1$*. Then we have:*

$$\frac{\gamma^2}{4n^2} \cdot \frac{\widetilde{\mu}}{\mu} \sigma(\ell_z) \le \sigma(\ell_{\widetilde{z}}) \le \frac{4n^2}{\gamma^2} \cdot \frac{\widetilde{\mu}}{\mu} \sigma(\ell_z). \tag{65}$$

In particular, the singular values scale down proportional with the parameter decrease. From here, the crucial property of our IPM is as follows. After $O(\mathrm{poly}(n) \log(1/\gamma))$ iterations, either we have crossed the end of the polarized segment, or the number of large singulars values has dropped by at least one. Clearly, the latter case can occur at most n times per polarized segment, since the lifting map has rank at most n.

Note that the above improvement is easy to achieve if the first singular value above the threshold has size at most $\mathrm{poly}(n)$. In this case, it suffices to use regular affine scaling steps to bring the singular value below the threshold quickly. Thus, the main difficulty is when the singular values themselves are quite polarized, namely either all at most $1/\mathrm{poly}(n)$ or at least $\mathrm{poly}(n)$.

In this setting, it is shown that for the current iterate $z = (x, s)$ on the polarized segment, that the associated partition $(\widetilde{B}, \widetilde{N})$ will match the partition of the current segment (B, N). Secondly, if the LLS subspaces $\widetilde{V}, \widetilde{U}$ have *maximum dimension*, then in the next iteration, the first singular value above the threshold will drop to $\mathrm{poly}(n)$. The only point where our iteration complexity is affected by the weaker SVD is the bound on how small the singular values below the threshold need to be before the SVD picks them up. In particular, if one have a Γ-approximate SVD at hand, as long as all the singular values below the threshold have size at most $\beta/(\Gamma\,\mathrm{poly}(n))$, the SVD will be guaranteed to pick them up. For this gap to appear, we must potentially wait for an addition $\mathrm{poly}(n) \log(\Gamma)$ iterations, which is the cause of the slowdown. Thus, if $\Gamma = 2^n$, the slowdown is propertional to $\log(\Gamma) = n$. This completes the proof sketch.

B.6 Computing Steps in Strongly Polynomial Time

In this section, we show that all the steps used in the algorithm can be computed in strongly polynomial time. In this section, we show that this is the case of corrector, affine scaling, and subspace LLS steps. We also explain how to compute lifting maps, as these are needed for LLS steps.

Proposition B.14 (Computing Corrector and Affine Scaling Steps)**.** *Let* $\mathbf{A} \in \mathbb{R}^{m \times n}$, $\mathrm{rank}(\mathbf{A}) = m$, $W = \ker(\mathbf{A})$ *and* $W^\perp = \mathrm{im}(\mathbf{A}^\top)$*. Let* $z := (x, s)$*, where* $x \in W, x > 0$*,* $s \in W^\perp, s > 0$ *and* $t \in \mathbb{R}^n$*. Then, the solution to* $s\Delta x + x\Delta s = t$*,* $\Delta x \in W$*,* $\Delta s \in W^\perp$ *can be expressed as*

$$\begin{aligned} \Delta s &= \mathbf{A}^\top (\mathbf{A}\,\mathrm{diag}(x/s)\mathbf{A}^\top)^{-1} \mathbf{A} \frac{t}{s} \\ \Delta x &= \frac{t}{s} - \frac{x}{s} \mathbf{A}^\top (\mathbf{A}\,\mathrm{diag}(x/s)\mathbf{A}^\top)^{-1} \mathbf{A} \frac{t}{s}. \end{aligned}$$

Moreover, both the corrector step $(\Delta x^{\mathrm{c}}, \Delta s^{\mathrm{c}})$ *and affine scaling step* $(\Delta x^{\mathrm{a}}, \Delta s^{\mathrm{a}})$*, corresponding to* $t = \nu\overline{\mu}(z)\mathbf{1}_n - xs$ *for* $\nu \in \{0, 1\}$ *respectively, can be computed in strongly polynomial time. Furthermore, given* $\Delta x, \Delta s \in \mathbb{R}^n$ *and* $\beta \in (0, 1/6]$*, one can in strongly polynomial time compute an affine scaling step-length* $\alpha^{\mathrm{a}} \in (0, 1]$ *satisfying*

$$\max\left\{\frac{\beta}{3\sqrt{n}}, 1 - \frac{3\|\Delta x^{\mathrm{a}} \Delta s^{\mathrm{a}}\|_2}{\beta\overline{\mu}(z)}\right\} \le \alpha^{\mathrm{a}} \le \max\left\{\frac{\beta}{2\sqrt{n}}, 1 - \frac{2\|\Delta x^{\mathrm{a}} \Delta s^{\mathrm{a}}\|_2}{\beta\overline{\mu}(z)}\right\}.$$

Proof. It is directly verified by inspection that $s\Delta x + x\Delta s = t$ and that $\mathbf{A}\Delta x = \mathbf{0}$ and $\Delta s \in \mathrm{im}(\mathbf{A}^\top)$.

To compute the corrector and affine scaling directions, we proceed as follows. First, we solve the linear system $(\mathbf{A}\,\mathrm{diag}(x/s)\mathbf{A}^\top)y = \mathbf{A}t/s$ for $t = \nu\mu(z)\mathbf{1}_n - xs$, and then we let $\Delta s = \mathbf{A}^\top y$ and $\Delta x = \frac{t}{s} - \frac{x}{s}\Delta s$. Since computing matrix vector products as well as solving linear systems can be done in strongly polynomial time, both steps are strongly polynomially computable.

For the affine scaling step-length, we first compute $a = \frac{\beta}{\lceil 2\sqrt{n} \rceil}$, where $\lceil 2\sqrt{n} \rceil$ is computed by binary search in $O(\log n)$ time. Next, we compute $r = \|\Delta x_i^{\mathrm{a}} \Delta s_i^{\mathrm{a}}\|_\infty$ and $b = 1 - \lceil 2\|\Delta x^{\mathrm{a}} \Delta s^{\mathrm{a}}\|_2/r \rceil r/(\beta\overline{\mu}(z))$, where $\lceil 2\|\Delta x^{\mathrm{a}} \Delta s^{\mathrm{a}}\|_2/r \rceil \in [2, \lceil 2\sqrt{n} \rceil]$ can also be computed in $O(\log n)$ time via binary search. We then return $\alpha^{\mathrm{a}} := \max\{a, b\}$. Since $\lceil 2\sqrt{n} \rceil \in [2\sqrt{n}, 3\sqrt{n}]$ and $\lceil 2\|\Delta x^{\mathrm{a}} \Delta s^{\mathrm{a}}\|_2/r \rceil r \in [2\|\Delta x^{\mathrm{a}} \Delta s^{\mathrm{a}}\|_2, 3\|\Delta x^{\mathrm{a}} \Delta s^{\mathrm{a}}\|_2]$, we have that $\frac{\beta}{3\sqrt{n}} \le a \le \frac{\beta}{2\sqrt{n}}$ and $1 - \frac{3\|\Delta x^{\mathrm{a}} \Delta s^{\mathrm{a}}\|_2}{\beta\overline{\mu}(z)} \le b \le 1 - \frac{2\|\Delta x^{\mathrm{a}} \Delta s^{\mathrm{a}}\|_2}{\beta\overline{\mu}(z)}$. Thus, α^{a} satisfies the requirement. □

Similarly to the affine scaling and corrector steps, subspace LLS steps can be computed in strongly polynomial time given an appropriate representation of the subspaces. The formulas for the LLS step directions are given in the next proposition. These are computed by solving the linear system which sets the gradient of the quadratic optimization problems to zero.

Proposition B.15 (Computing the LLS Step Direction)**.** *Given an iterate $z = (x, s)$, partition $B \cup N = [n]$, $B, N \neq \emptyset$, and matrices $\mathbf{MP} \in \mathbb{R}^{n \times k_p}, \mathbf{MD} \in \mathbb{R}^{n \times k_d}$ satisfy $V := \mathrm{im}(\mathbf{MP}) \subseteq W$, $U := \mathrm{im}(\mathbf{MD}) \subseteq W^\perp$, and* $\dim(\pi_N(U)) = \mathrm{rank}(\mathbf{MP}_N) = k_p$, $\dim(\pi_B(U)) = \mathrm{rank}(\mathbf{MD}_B) = k_d$*. Then subspace LLS direction $(\Delta x^\ell, \Delta s^\ell)$ at z with respect to (B, N, U, V) can be computed as follows:*

$$\Delta x^\ell = \mathbf{MP}\left(\mathbf{MP}_N^\top \,\mathrm{diag}(s_N/x_N)\mathbf{MP}_N\right)^{-1}\mathbf{MP}_N s_N,$$
$$\Delta s^\ell = \mathbf{MP}\left(\mathbf{MP}_B^\top \,\mathrm{diag}(x_B/s_B)\mathbf{MP}_B\right)^{-1}\mathbf{MP}_B x_B.$$

The next proposition explains how to compute the step length associated with directions which go very far down the path. We will apply this to compute the subspace LLS step lengths.

Proposition B.16 (Step-length estimate for general directions)**.** *Let $z = (x, s) \in \mathcal{N}^2(\beta/2)$, $\mu := \overline{\mu}(z)$, $\beta \in (0, 1/6]$. Consider directions $\Delta x \in W$, $\Delta s \in W^\perp$ that satisfy $\|\Delta x \Delta s\|_2 \leq \beta\mu/8$. Let*

$$\gamma := \frac{\|(x + \Delta x)(s + \Delta s)\|_2}{\mu} \leq \frac{\beta}{9}.$$

Then $(x + \alpha\Delta x, s + \alpha\Delta s) \in \overline{\mathcal{N}}^2(\beta)$ and $\overline{\mu}(x + \alpha\Delta x, s + \alpha\Delta s) \in [1 \pm 1/8](1 - \alpha)\mu$, for all $0 \leq \alpha \leq 1 - \frac{8\gamma}{\beta}$. Furthermore, given $\Delta x \in \mathbb{R}^n, \Delta s \in \mathbb{R}^n, \mu > 0, \beta > 0$, one can in strongly polynomial time output a step-length $\alpha^\ell \in [0, 1]$ satisfying $1 - \frac{9\gamma}{\beta} \leq \alpha^\ell \leq 1 - \frac{8\gamma}{\beta}$ if $\|\Delta x \Delta s\|_2 \leq \beta\mu/8$ and $\gamma := \|(x + \Delta x)(s + \Delta s)\|_2/\mu \leq \frac{\beta}{9}$, and output $\alpha^\ell = 0$ otherwise.

Proof. Let $z_\alpha := (x + \alpha\Delta x, s + \alpha\Delta s)$ for $0 \leq \alpha \leq 1 - \frac{8\gamma}{\beta}$, $\alpha < 1$. We first bound the centrality error using the estimate $(1 - \alpha)\mu$ for $\mu(z_\alpha)$ as follows:

$$\begin{aligned}
&\left\|\frac{(x + \alpha\Delta x)(s + \alpha\Delta s)}{(1 - \alpha)\mu} - \mathbf{1}\right\|_2 \\
&= \left\|\frac{(1 - \alpha)xs + \alpha(x + \Delta x)(s + \Delta s) - \alpha(1 - \alpha)\Delta x\Delta s}{(1 - \alpha)\mu} - \mathbf{1}\right\|_2 \\
&\leq \left\|\frac{xs}{\mu} - \mathbf{1}\right\|_2 + \frac{\alpha}{1 - \alpha}\left\|\frac{(x + \Delta x)(s + \Delta s)}{\mu}\right\|_2 + \alpha\left\|\frac{\Delta x\Delta s}{\mu}\right\|_2 \\
&\leq \beta/2 + \frac{\alpha}{1 - \alpha}\gamma + \alpha\frac{\beta}{8} \leq \frac{3}{4}\beta,
\end{aligned}$$

where the last inequality follows since $\frac{\alpha}{1-\alpha}\gamma \leq \beta/8$ for $0 \leq \alpha \leq 1 - 8\gamma/\beta$ and $\alpha < 1$ (needed to ensure the denominator $(1 - \alpha)$ is positive).

By Lemma 6.7 and the above bound, we get that $\overline{\mu}(z_\alpha) \in [1 \pm \frac{3}{4}\beta/\sqrt{n}](1 - \alpha)\mu \subseteq [1 \pm \frac{1}{8}](1 - \alpha)\mu$, and $z_\alpha \in \mathcal{N}\left(\frac{\frac{3}{4}\beta}{1 - \frac{3}{4}\beta}\right) \subseteq \mathcal{N}^2(\beta)$, for $\beta \in (0, 1/6]$. If $\gamma = 0$, letting $\alpha \to 1^-$, we conclude by continuity that $z_1 := (x + \Delta x, s + \Delta s) \in \overline{\mathcal{N}}^2(\beta)$ and $\overline{\mu}(z_1) = 0$, as needed.

For the last part, note that the conditions $\|\Delta x\Delta s\|_2 \leq \beta\mu/8$ and $\|(x + \Delta x)(s + \Delta s)\|_2/\mu \leq \beta/9$ can be checked in strongly polynomial time by squaring both sides. If this check fails, output $\alpha^\ell = 0$. Otherwise, compute $r = \|(x + \Delta x)(s + \Delta s)\|_\infty$ (i.e., the largest entry in absolute value) and compute $\nu = \lceil 8\|(x + \Delta x)(s + \Delta s)\|_2/r\rceil \in [8, \lceil 8\sqrt{n}\rceil]$ via binary search in $O(\log n)$ time, and return $\alpha^\ell = 1 - \frac{8\nu r}{\mu\beta}$. For correctness, note that $\|(x + \Delta x)(s + \Delta s)\|_2 \leq \nu(r/8) \leq \frac{9}{8}\|(x + \Delta x)(s + \Delta s)\|_2$, and thus the desired inequalities follow recalling that $\gamma := \|(x + \Delta x)(s + \Delta s)\|_2/\mu$. □

We now show how to compute lifting maps.

Proposition B.17 (Computing Lifting Maps)**.** *Let $\mathbf{U} \in \mathbb{R}^{n \times d}$, $\mathrm{rank}(U) = d$, and $B \cup N = [n]$ be a partition with $B, N \neq \emptyset$. Then, for $W = \mathrm{im}(U)$, $x \in \mathbb{R}^N$, we have that*

$$L_N^W(x) = \Pi_{[n],N}\Pi_{N,N}^+ x, \qquad \ell_N^W(x) = \Pi_{B,N}\Pi_{N,N}^+ x,$$

where $\Pi = \mathbf{U}(\mathbf{U}^\top\mathbf{U})^{-1}\mathbf{U}^\top$ is the orthogonal projection onto W. Furthermore, the matrix $\Pi_{[n],N}\Pi_{N,N}^+$ can be computed in strongly polynomial time.

Proof. Let $y \in \Pi_N(W) \subseteq \mathbb{R}^N$. Since $\mathrm{im}(\Pi) = W$, we clearly have that $y = \mathrm{im}(\Pi_{N,\bullet})$. Furthermore, since Π is positive semidefinite $\mathrm{im}(\Pi_{N,\bullet}) = \mathrm{im}(\Pi_{N,N})$. Thus, there exists $x \in \mathbb{R}^N$ such that $y = \Pi_{N,N}x$. We now claim that $L_N^W(y) = \Pi_{\bullet,N}x$. Let $\hat{y} = \Pi_{\bullet,N}x$ and take any $\bar{y} \in W$ with $\bar{y}_N = y$. We wish to show that $\|\bar{y}\|_2 \leq \|\hat{y}\|_2 \Rightarrow y' = y$.

To see this, notice that

$$\|\bar{y}\|_2^2 = \|(\bar{y} - \hat{y}) + \hat{y}\|_2^2 = \|(\bar{y} - \hat{y})\|_2^2 + 2\langle \bar{y} - \hat{y}, \hat{y}\rangle + \|\hat{y}\|_2^2.$$

To prove the claim, it suffices to show that $\langle \bar{y} - \hat{y}, \hat{y}\rangle = 0$. Let $\bar{x} \in \mathbb{R}^n$ denote the vector with $\bar{x}_N = x$ and $\bar{x}_B = \mathbf{0}_B$. Then, noting that $\hat{y} = \Pi\bar{x}$, we have that

$$\langle \bar{y} - \hat{y}, \hat{y}\rangle = \langle \bar{y} - \hat{y}, \Pi\bar{x}\rangle = \langle \Pi(\bar{y} - \hat{y}), \bar{x}\rangle = \langle \bar{y} - \hat{y}, \bar{x}\rangle = 0,$$

where the last equality follows since $\mathrm{supp}(\bar{y} - \hat{y}) \subseteq B$ and $\mathrm{supp}(\bar{x}) \subseteq N$. This proves the claim.

From the representation above, we note that $y \in \mathfrak{I}(\Pi_{N,N})$, then by the properties of pseudoinverses, we have that $y = \Pi_{N,N}\Pi_{N,N}^{+}y$. In particular, $L_N^W(y) = \Pi_{\bullet,N}\Pi_{N,N}^{+}y$, as needed.

We note that since matrix products and pseudoinverses can be computed in strongly polynomial time, the lifting map can be computed in strongly polynomial time. □

B.7 A strongly polynomial singular value approximation

In this section we propose a simple strongly polynomial algorithm to approximate the singular values of a matrix within a factor of $\sqrt{n}2^n$. As mentioned previously, while more sophisticated strongly polynomial SVDs indeed exists [DTK23, DR24], they are both somewhat complicated (and [DTK23] is randomized). The simple deterministic alternative we provide below is sufficient for our purposes, and allows for a more self-contained analysis.

Recall that the vector $\sigma(\mathbf{M})$ is the vector of singular values of $\mathbf{M}$, where the vector is in non-increasing order.

For a matrix $\mathbf{M} \in \mathbb{R}^{m\times n}$, assuming $\mathbf{M}$ has full column rank for simplicity, consider the following procedure (Algorithm 4). For all columns j of $\mathbf{M}$, consider the projection of the column $\mathbf{M}_{\bullet,j}$ onto the column-space of all the other columns $\mathbf{M}_{\bullet,[n]\setminus\{j\}}$. Then, remove the column $j_{\min}$ from the matrix for which the norm of this projection is the smallest and recurse on the remaining matrix $\mathbf{M}_{\bullet,[n]\setminus\{j_{\min}\}}$. When this process finishes, we obtain a permutation $\mathbf{MP}$ of the columns, given by the order in which they were removed from the matrix. It turns out, that the norms of the columns of the orthogonal matrix $\mathbf{Q}$ obtained from a Gram-Schmidt process on the permuted matrix $\mathbf{MP}$ (the first column removed from $\mathbf{M}$ is the last column in the reordering $\bar{\mathbf{M}}$) provide an exponential approximation of the singular values of the original matrix $\mathbf{M}$. The main observation for the proof is that the matrix $\mathbf{R}$ in the Gram-Schmidt process, uniquely defined by $\mathbf{MPR} = \mathbf{Q}$, has only exponential condition number.

It is not hard to see that the output of Well-Conditioned-GSO satisfies that $\mathbf{Q}$ is precisely the result of Gram Schmidt orthogonalization (GSO) on the matrix $\mathbf{MP}$, i.e., the matrix $\mathbf{M}$ after its columns have been permuted according to $\mathbf{P}$.

The following proposition explains why the GSO computed by Well-Conditioned-GSO is "well-conditioned".

Proposition B.18. *Let $\mathbf{R} \in \mathbb{R}^{n\times n}$ be an upper triangular matrix with diagonal $\mathbf{1}_n$ with entries of absolute value at most 1. Then $\mathbf{R}^{-1}$ is upper triangular with diagonal $\mathbf{1}_n$ and $|\mathbf{R}^{-1}_{ij}| \leq \max\{1, 2^{j-i-1}\}$, $i,j \in [n]$, $i \geq j$. In particular, $\|\mathbf{R}^{-1}\|_2 \leq 2^n$.*

Proof. The proof goes proceeds via induction on n. The base case $n = 1$ is trivial, so assume $n > 1$. Then, it is direct to verify that

$$\mathbf{R}^{-1} = \begin{pmatrix} \mathbf{R}^{-1}_{[n-1],[n-1]} & -\mathbf{R}^{-1}_{[n-1],[n-1]}\mathbf{R}_{n,[n-1]} \\ \mathbf{0}_{n-1}^{\top} & 1 \end{pmatrix}.$$

By the induction hypothesis, the statement holds for $\mathbf{R}^{-1}_{i,j}$, $i,j \in [n-1]$, $i \geq j$. Note that $\mathbf{R}^{-1}$ is upper triangular as claimed. We now prove the coefficient bound for $\mathbf{R}^{-1}_{i,n}$, for $i \geq n$. If $i = n$, $\mathbf{R}^{-1}_{i,i} = 1$, as

Algorithm 4: WELL-CONDITIONED-GSO

Input : Matrix $\mathbf{M} \in \mathbb{R}^{m\times n}$ with full column rank.

Output: Matrix $\mathbf{Q} \in \mathbb{R}^{n\times n}$ with orthogonal columns, upper triangular matrix $\mathbf{R} \in \mathbb{R}^{n\times n}$ with diagonal $\mathbf{1}_n$, permutation matrix $\mathbf{P} \in \mathbb{R}^{n\times n}$ satisfying:

1. $\mathbf{MPR} = \mathbf{Q}$.
2. $\|\mathbf{Q}_{\bullet,t}\| = \min_{j\in[t]} \min_{x\in R^t, x_j=1} \|\mathbf{MP}_{\bullet,[t]}x\|$.
3. The entries of $\mathbf{R}$ have absolute value at most 1.

Compute $B_n \subseteq [n]$, $\mathrm{rank}(\mathbf{M}_{\bullet,B_n}) = \mathrm{rank}(\mathbf{M})$;
$J_n \leftarrow [n]$; $k \leftarrow n - |B_n|$;
for $t = n$ *down to* $n - k + 1$ **do**
  Pick $i \in J_t \setminus B_t$; $u \leftarrow \mathbf{0}_n$;
  $(u_i, u_{B_t}) \leftarrow (1, -\mathbf{M}^{-1}_{\bullet,B_t}\mathbf{M}_{\bullet,i})$;
  $\pi(t) \leftarrow \arg\max_{j\in\mathrm{supp}(v)} |u_j|$;
  $q_t \leftarrow \mathbf{0}_m$; $v_t := v_{t\pi(t)} \leftarrow u/u_{\pi(t)}$;
  $(J_{t-1}, B_{t-1}) \leftarrow (J_t \setminus \{\pi(t)\}, (B_t \cup \{i\}) \setminus \{\pi(t)\})$;
for $t = n - k$ *down to* 1 **do**
  for $i \in J_t$ **do**
    $v_{ti} \leftarrow \arg\min_{v\in\mathbb{R}^n, v_i=1, \mathrm{supp}(v)\subseteq J_t} \|\mathbf{M}v\|_2$;
    $p_{ti} \leftarrow \mathbf{M}v_{ti}$;
  $\pi(t) \leftarrow \arg\min_{j\in[J_t]} \|p_{tj}\|_2$;
  $q_t \leftarrow p_{t\pi(t)}$, $v_t \leftarrow v_{t\pi(t)}$;
  $J_{t-1} \leftarrow J_t \setminus \{\pi(t)\}$;
$\mathbf{P} \leftarrow (1_{i=\pi(j)})_{i,j\in[n]}$;
$\mathbf{R} \leftarrow \mathbf{P}^\top \begin{bmatrix} v_1 & \dots & v_n \end{bmatrix}$;
$\mathbf{Q} \leftarrow \begin{bmatrix} q_1 & \dots & q_n \end{bmatrix}$;
return $(\mathbf{P}, \mathbf{R}, \mathbf{Q})$

needed. For $i > n$, using that $\mathbf{R}$ has entries at most 1 we get that

$$|\mathbf{R}^{-1}_{in}| = |(\mathbf{R}^{-1}_{[n-1],[n-1]}\mathbf{R}_{n,[n-1]})_i| = |\sum_{j=i}^{n-1} \mathbf{R}^{-1}_{i,j}\mathbf{R}_{n,j}| \le \sum_{j=i}^{n-1} |\mathbf{R}^{-1}_{i,j}|$$
$$\le \sum_{j=i}^{n-1} \max\{1, 2^{j-i-1}\} = 1 + \sum_{l=0}^{n-i-2} 2^l = \max\{1, 2^{n-i-1}\},$$

as needed. For the last statement, by a direct calculation

$$\|\mathbf{R}^{-1}\|^2 \le \|\mathbf{R}^{-1}\|_F^2 \le \sum_{i=1}^{n}\sum_{j=i}^{n} \max\{1, 4^{j-i-1}\} = \frac{2}{3}n - 1/9 + 4^n/9 \le 4^n,$$

as needed. $\square$

We now show how the output of WELL-CONDITIONED-GSO can be used to compute $\sqrt{n}2^n$-approximation of the singular values and subspaces of $\mathbf{M}$.

Lemma B.19. *Let $\mathbf{M} \in \mathbb{R}^{m\times n}$ be a matrix. Let $(\mathbf{P}, \mathbf{R}, \mathbf{Q})$ be the output of* WELL-CONDITIONED-GSO *on $\mathbf{M}$. Let $\mathbf{C}_{\ge i} := (\mathbf{PR})_{\bullet,\ge i}$, $V_{\ge i} := \mathrm{im}(\mathbf{C}_i)$ and similarly $\mathbf{C}_{\le i} := (\mathbf{PR})_{\bullet,\le i}$, $V_{\le i} := \mathrm{im}(\mathbf{C}_{\le i})$, $i \in [n]$. Then,*

$$\sigma_i(\mathbf{M}) \le \max_{v\in V_{\ge i}\setminus\{\mathbf{0}_n\}} \frac{\|\mathbf{M}v\|_2}{\|v\|_2} \le 2^n \|\mathbf{Q}_{\bullet,i}\|_2 \le \sqrt{n}2^n \sigma_i(\mathbf{M}) \tag{66}$$

$$\sigma_i(\mathbf{M}) \ge \min_{v\in V_{\le i}\setminus\{\mathbf{0}_n\}} \frac{\|\mathbf{M}v\|_2}{\|v\|_2} \ge \frac{\|\mathbf{Q}_{\bullet,i}\|_2}{n} \ge \frac{\sigma_i(\mathbf{M})}{n2^n}. \tag{67}$$

Proof. Without loss of generality, we may assume that $\mathbf{P} = \mathbf{I}_n$. The inequalities $\sigma_i(\mathbf{M}) \le \max_{v\in V_{\ge i}\setminus\{\mathbf{0}_n\}} \frac{\|\mathbf{M}v\|_2}{\|v\|_2}$ and $\sigma_i(\mathbf{M}) \ge \min_{v\in V_{\le i}\setminus\{\mathbf{0}_n\}} \frac{\|\mathbf{M}v\|_2}{\|v\|_2}$ are direct consequence of the variational characterization of singular values (62) using that $\dim(V_{\ge i}) = \mathrm{rank}(\mathbf{R}_{\bullet,\ge i}) = n - i + 1$ and $\dim(V_{\le i}) = i$.

For the first upper bound, we have that

$$\max_{v\in V_{\geq i}} \frac{\|\mathbf{M}v\|_2}{\|v\|_2} = \max_{x\in\mathbb{R}^{n-i+1}\setminus\{0\}} \frac{\|\mathbf{M}\mathbf{R}_{\bullet,\geq i}x\|_2}{\|\mathbf{R}_{\bullet,\geq i}x\|_2} = \max_{x\in\mathbb{R}^{n-i+1}\setminus\{0\}} \frac{\|\mathbf{Q}_{\bullet,\geq i}x\|_2}{\|\mathbf{R}_{\bullet,\geq i}x\|_2}$$
$$\leq \max_{x\in\mathbb{R}^{n-i+1}\setminus\{0\}} \|\mathbf{Q}_{\bullet,i}\|_2 \frac{\|x\|_2}{\|\mathbf{R}_{\bullet,\geq i}x\|_2} \quad \left(\mathbf{Q} \text{ orthogonal}, \max_{j\geq i}\|\mathbf{Q}_{\bullet,j}\| = \|\mathbf{Q}_{\bullet,i}\|\right)$$
$$\leq \|\mathbf{Q}_{\bullet,i}\|_2\|\mathbf{R}^{-1}\|_2 \leq 2^n\|\mathbf{Q}_{\bullet,i}\|_2,$$

where the last inequality follows by Proposition B.18. It remains to prove that $\|\mathbf{Q}_{\bullet,i}\|_2 \leq \sqrt{n}\sigma_i(\mathbf{M})$, $\forall i \in [n]$. By (62), there exists a subspace $U_i \subseteq \mathbb{R}^n$ such that $\max_{x\in U_i\setminus\{0\}} \frac{\|Mx\|_2}{\|x\|_2} = \sigma_i(\mathbf{M})$ and $\dim(U_i) = n-i+1$. By dimension counting, $\exists\bar{x} \in U_i \setminus \{0\}$ such that $\mathrm{supp}(\bar{x}) \subseteq [i]$. Therefore,

$$\sigma_i(\mathbf{M}) \geq \frac{\|\mathbf{M}\bar{x}\|_2}{\|\bar{x}\|_2} \geq \frac{1}{\sqrt{n}}\frac{\|\mathbf{M}\bar{x}\|_2}{\|\bar{x}\|_\infty} \geq \frac{\|\mathbf{Q}_{\bullet,i}\|_2}{\sqrt{n}},$$

since $\pm\bar{x}/\|\bar{x}\|_\infty$ is a candidate solution for one of the least squares programs defining $\|\mathbf{Q}_{\bullet,i}\|$ (property 2 of the output description).

For the second lower bound,

$$\min_{v\in V_{\leq i}} \frac{\|\mathbf{M}v\|_2}{\|v\|_2} = \min_{x\in\mathbb{R}^{i}\setminus\{0\}} \frac{\|\mathbf{M}\mathbf{R}_{\bullet,\leq i}x\|_2}{\|\mathbf{R}_{\bullet,\leq i}x\|_2} = \min_{x\in\mathbb{R}^{i}\setminus\{0\}} \frac{\|\mathbf{Q}_{\bullet,\leq i}x\|_2}{\|\mathbf{R}_{\bullet,\leq i}x\|_2}$$
$$\geq \min_{x\in\mathbb{R}^{i}\setminus\{0\}} \|\mathbf{Q}_{\bullet,i}\|_2 \frac{\|x\|_2}{\|\mathbf{R}_{\bullet,\leq i}x\|_2} \quad \left(\mathbf{Q} \text{ orthogonal}, \min_{j\leq i}\|\mathbf{Q}_{\bullet,j}\| = \|\mathbf{Q}_{\bullet,i}\|\right)$$
$$\geq \|\mathbf{Q}_{\bullet,i}\|_2\|\mathbf{R}\|_2^{-1} \geq \frac{\|\mathbf{Q}_{\bullet,i}\|_2}{n} \geq \frac{\sigma_i(\mathbf{M})}{n2^n},$$

where $\|\mathbf{R}\|_2 \leq n$ follows from $\mathbf{R}$ having entries of absolute value at most 1 and $\sigma_i(\mathbf{M}) \leq 2^n\|\mathbf{Q}_{\bullet,i}\|_2$ comes from the first part. This concludes the proof. □

Lemma B.20. *WELL-CONDITIONED-GSO is correct and runs in strongly polynomial time.*

Proof. We start with correctness. We by show by induction for $n-k \leq t \leq n$ that $B_t \subseteq J_t$ forms a basis the column span of $\mathbf{M}$. Firstly, B_t does not change size during the course of the for loop, so it suffices to show that B_t stays independent. By definition, since B_t is a basis, for any $i \in J_t \setminus B_t$, $\mathbf{M}_{\bullet,i}$ can be uniquely expressed as a linear combinations of the columns of $\mathbf{M}_{\bullet,B_t}$. Thus, u is well-defined and $\mathbf{M}u = \mathbf{0}_n$. Since $\mathrm{supp}(u) = B_t \cup \{i\}$, by the basis exchange property $B_t \leftarrow (B_t \cup \{i\}) \setminus \{j\}$ remains a basis for any $j \in \mathrm{supp}(u)$. Note that $B_t \subseteq J_t$ since the element i we add to B_t is already contained in J_t. Lastly, since $|B_t| = |B_n| = n-k$ and $|J_t| = n-t$, we see that $J_{n-k} = B_{n-k}$. In particular, J_{n-k} is a basis of the column span of $\mathbf{M}_{\bullet,J_{n-k}}$.

Using the linear independence of the columns of $\mathbf{M}_{J_t}$, $1 \leq t \leq n-k$, we explain how to interpret and solve the least squares problem

$$v_{ti} := \operatorname*{arg\,min}_{v\in\mathbb{R}^n, v_i=1, \mathrm{supp}(v)\subseteq J_t} \|\mathbf{M}v\|_2 \tag{68}$$

for $i \in J_t$, $t \in [n]$. We first note that v_{it} is indeed uniquely defined by linear independence of $\mathbf{M}_{\bullet,J_t}$. Secondly, letting $J_t^i := J_t \setminus \{i\}$, it is direct to verify that $\mathbf{M}_{\bullet,J_t^i}(v_{ti})_{J_t^i} = -\Pi_W(\mathbf{M}_{\bullet,i})$, where Π_W is the orthogonal projection onto $W = \mathrm{im}(\mathbf{M}_{\bullet,J_t^i})$, and hence $\mathbf{M}v_{ti} = \mathbf{M}_{\bullet,i} - \Pi_W(\mathbf{M}_{\bullet,i}) = \Pi_{W^\perp}(\mathbf{M}_{\bullet,i})$. Using that

$$\Pi_W = \mathbf{M}_{\bullet,J_t^i}(\mathbf{M}_{\bullet,J_t^i}^\top\mathbf{M}_{\bullet,J_t^i})^{-1}\mathbf{M}_{\bullet,J_t^i}^\top,$$

where linear independence yields the invertibility of $(\mathbf{M}_{\bullet,J_t^i}^\top\mathbf{M}_{\bullet,J_t^i})^{-1}$, we see that

$$(v_{ti})_{J_t^i} = -(\mathbf{M}_{\bullet,J_t^i}^\top\mathbf{M}_{\bullet,J_t^i})^{-1}\mathbf{M}_{\bullet,J_t^i}^\top\mathbf{M}_{\bullet,i}.$$

Recalling that $(v_{ti})_i = 1$, the only other non-zero entry, this gives a strongly polynomial way to solve the least squares problems.

We now verify the properties required of the output. Firstly, note that by construction $Mv_i = q_i$, $i \in [n]$. Therefore $\mathbf{MPR} = \mathbf{MPP}^\top[v_1, \ldots, v_n] = \mathbf{M}[v_1, \ldots, v_n] = [q_1, \ldots, q_n] = \mathbf{Q}$ as needed. To show that $\mathbf{R} = \mathbf{P}^\top[v_1, \ldots, v_n]$ is upper triangular diagonal $\mathbf{1}_n$, note that $\mathrm{supp}(v_t) \subseteq J_t$, $(v_t)_{\pi(t)} = 1$ and that $J_t = \{\pi(i) : i \in [t]\}$. Therefore, letting π^{-1} denote the inverse permutation, we get $\mathrm{supp}(\mathbf{P}^\top v_t) \subseteq \pi^{-1}(J_t) = \{1, \ldots, t\}$ and $(\mathbf{P}^\top v_t)_t = (v_t)_{\pi(t)} = 1$, for $t \in [n]$, as needed. To see that $\mathbf{Q}$ has orthogonal columns, it is sufficient to show that q_t is orthogonal to $q_1, \ldots, q_{t-1}$, for $2 \leq t \leq n-k$, noting that $q_t = \mathbf{0}_m$ for $t > n-k$. By construction, $q_1, \ldots, q_{t-1}$ are in $\mathrm{im}(\mathbf{M}_{\bullet, J_{t-1}})$, while by the previous by the paragraph, $q_t = q_{t\pi(t)}$ is orthogonal to this subspace, noting that $J_{t-1} = J_t \setminus \{\pi(t)\} = J_t^{\pi(t)}$. We now show the second property of output that characterizes $\|\mathbf{Q}_{\bullet,t}\|_2$. By construction, recall that $J_t = \{\pi(1), \ldots, \pi(t)\}$, $t \in [n]$. For $1 \leq t \leq n-k$, by definition of q_t, we have

$$\|\mathbf{Q}_{\bullet,t}\|_2 = \|q_t\|_2 = \min_{j \in J_t} \min_{x \in \mathbb{R}^n : \mathrm{supp}(x) \subseteq J_t, x_j = 1} \|\mathbf{M}x\|_2 = \min_{j \in [t]} \min_{x \in \mathbb{R}^t : x_j = 1} \|\mathbf{M}\mathbf{P}_{\bullet,[t]} x\|_2.$$

Similarly, for $n-k+1 \leq t \leq n$, since by construction $(v_t)_{\pi(t)} = 1$, we have that

$$\|\mathbf{Q}_{\bullet,t}\|_2 = \|q_t\|_2 = 0 = \|\mathbf{M}v_t\|_2 = \min_{j \in [t]} \min_{x \in \mathbb{R}^t : x_j = 1} \|\mathbf{M}\mathbf{P}_{\bullet,[t]} x\|_2.$$

We now show that the $|\mathbf{R}_{jt}| \leq 1$, $\forall j, t \in [n]$. Since by construction $\mathbf{R}_{jt} = (v_t)_{\pi(j)}$, it suffices to show that $\|v_t\|_\infty \leq 1$, for all $t \in [n]$. For $t \in [n]$, examine $i \in \mathrm{supp}(v_t) \subseteq J_t$. Then

$$|(v_t)_i| = \frac{\|q_t\|_2}{\left\|\frac{q_t}{|(v_t)_i|}\right\|_2} \leq \frac{\|q_t\|_2}{\|q_{ti}\|_2} \leq 1,$$

where in the first inequality we used that $v_t/(v_t)_i$ is a feasible solution for the least squares program defining q_{ti} (since $i \in J_t$) having value $\|q_t/|(v_t)_i|\|_2 \geq \|q_{ti}\|_2$, and for the second inequality we used that $\|q_t\|_2 := \arg\min_{j \in J_t} \|q_{tj}\|_2$. We note that in this last proof it is crucial that $q_{ti} \neq \mathbf{0}_m$, $i \in J_t$, which follows by linear independence of $\mathbf{M}_{\bullet,J_t}$.

For the strongly polynomial guarantees, it remains to check that all the intermediate iterates have polynomial bit complexity. This follows directly by noting that each of the computed quantities vectors corresponds to a well-described linear system in the original matrix $\mathbf{M}$. Alternatively, one can use that the algorithm simply performs a "reverse order" GSO on the matrix $\mathbf{MP}$, and it is well known that that the intermediate iterates of GSO have polynomial bit complexity (see [GLS88]). □