A Note on the Approximability of Deepest-Descent Circuit Steps

Steffen Borgwardt[a], Cornelius Brand[b,*], Andreas Emil Feldmann[b], Martin Koutecký[b]

[a] *Department of Mathematical and Statistical Sciences, University of Colorado Denver P.O. Box 173364, CB 170 Denver, CO 80217-3364 USA*

[b] *Univerzita Karlova, Matematicko-fyzikální fakulta, Informatický ústav Univerzity Karlovy Malostranské nám. 25 118 00 Praha 1, Czech Republic*

Abstract

Linear programs (LPs) can be solved by polynomially many moves along the circuit direction improving the objective the most, so-called deepest-descent steps. Computing these steps is NP-hard (De Loera et al., arXiv, 2019), a consequence of the hardness of deciding the existence of an optimal circuit-neighbor (OCNP) on LPs with non-unique optima.

We prove OCNP is easy under the promise of unique optima, but already $O(n^{1-\varepsilon})$-approximating dd-steps remains hard even for totally unimodular n-dimensional 0/1-LPs with a unique optimum. We provide a matching n-approximation.

Keywords: circuits, linear programming, deepest-descent steps, complexity theory

1. Introduction

Linear programming is a fundamental tool in both the theory and applications of combinatorial optimization: We are given a system $A\mathbf{x} = \mathbf{b}$, $B\mathbf{x} \leq \mathbf{d}$ with $A \in \mathbb{R}^{m_A \times n}$, $B \in \mathbb{R}^{m_B \times n}$, $\mathbf{b} \in \mathbb{R}^{m_A}$ and $\mathbf{d} \in \mathbb{R}^{m_B}$ and a cost vector $\mathbf{c} \in \mathbb{R}^n$. We call an assignment $\mathbf{x} \in \mathbb{R}^n$ to the variables *feasible* if it satisfies the system of equalities and inequalities, and the set of these feasible assignments is a *polyhedron*, which will be denoted as P throughout. The goal is to find a feasible

*Corresponding author

Email addresses: `steffen.borgwardt@ucdenver.edu` (Steffen Borgwardt), `cbrand@iuuk.mff.cuni.cz` (Cornelius Brand), `feldmann.a.e@gmail.com` (Andreas Emil Feldmann), `koutecky@iuuk.mff.cuni.cz` (Martin Koutecký)

assignment $\mathbf{x} \in \mathbb{R}^n$ minimizing $\mathbf{c}^T\mathbf{x}$.

Linear programming has been known to be solvable in weakly-polynomial[1] time since the groundbreaking work of Khachiyan on the ellipsoid method [1] and Karmarkar on the interior point method [2]. The existence of a strongly polynomial algorithm for linear programming, that is, an algorithm which makes $\text{poly}(n, m_A, m_B)$ arithmetic operations and finds an optimal solution, is a major open problem. Exploring methods other than the ellipsoid and interior point methods is a possible pathway for a resolution of this important open problem.

One such family of methods are iterative augmentation methods [3] using the circuits of the matrix pair A, B, which are defined as follows:

Definition 1. Given matrices A, B, the set of *circuits* $\mathcal{C}(A, B)$ consists of all $\mathbf{g} \in \ker(A) \setminus \{\mathbf{0}\}$ normalized to coprime integer components for which $B\mathbf{g}$ is support-minimal over $\{B\mathbf{x} \mid \mathbf{x} \in \ker(A) \setminus \{\mathbf{0}\}\}$.

The set of circuits $\mathcal{C}(\mathcal{P})$ of the polyhedron $P = \{\mathbf{x} \in \mathbb{R}^n \mid A\mathbf{x} = \mathbf{b}, B\mathbf{x} \leq \mathbf{d}\}$ satisfies $\mathcal{C}(\mathcal{P}) = \mathcal{C}(A, B)$. A generic iterative augmentation method for linear programming over P starts from some initial feasible iteration with $i = 0, 1, \ldots,$ finds a circuit $\mathbf{g}^i \in \mathcal{C}(A, B)$ and a step-length $\lambda^i \in \mathbb{R}_+$ such that $\mathbf{x}^{i+1} = \mathbf{x}^i + \lambda^i\mathbf{g}^i$ is feasible and $\mathbf{c}^T\mathbf{g}^i < 0$.The specific choice of λ^i and $\mathbf{g}^i$ distinguishes the individual methods. For example, a *steepest-descent step* is one which minimizes $\mathbf{c}^T\mathbf{g}^i/\|\mathbf{g}^i\|_1$, where $\|\cdot\|_1$ is the 1-norm, and a *deepest-descent step* is one which minimizes $\lambda^i\mathbf{c}^T\mathbf{g}^i$. The set $\mathcal{C}(A, B)$ is, for instance, the set of all *potential* edge directions, arising from any polyhedron having A and B as their constraint matrices over the varying choices of the right-hand sides $\mathbf{b}$ and $\mathbf{d}$. This set contains the set of set of *actual* edge directions appearing on P with $\mathbf{b}$ and $\mathbf{d}$ fixed as a subset. To be precise, by an *edge direction*, we mean any (normalized) vector in a one-dimensional subspace spanned by the set of optimal points with respect to some cost vector. This means that considering

[1]From here on out, whenever we speak of a problem with instances containing numbers as inputs as being solvable in polynomial time, we intend this to mean weakly-polynomial time, unless explicitly stated otherwise.

all circuits in each iteration gives a potentially larger improvement than by only considering the edge directions, as is the case in the Simplex method. This gives rise to the circuit diameter conjecture [4], which states that for any d-dimensional polyhedron with f facets, the circuit diameter is bounded above by $f - d$; the circuit diameter is the smallest number of feasible circuit steps between two points of a polyhedron. The significance of studying the circuit-based iterative augmentation methods is also highlighted by recent success of Graver bases in the design of integer programming algorithms [5], since a Graver basis is essentially the integer programming analogue of the set of circuits.

Throughout this paper, we consider polyhedra in the general form $P = \{\mathbf{x} \in \mathbb{R}^n \mid A\mathbf{x} = \mathbf{b}, B\mathbf{x} \leq \mathbf{d}\}$, just as we already did up to this point. We assume that P is pointed, i.e., it has a vertex. This is required for some of our problem statements to be well-defined. A check whether P is pointed can be done efficiently through elementary linear algebra.

Let us formally define a deepest-descent step:

Definition 2. Let $P = \{\mathbf{x} \in \mathbb{R}^n \mid A\mathbf{x} = \mathbf{b}, B\mathbf{x} \leq \mathbf{d}\}$, let $\mathbf{c} \in \mathbb{R}^n$ and $\mathbf{x}^0 \in P$, and consider the LP $\min\{\mathbf{c}^T\mathbf{x} : \mathbf{x} \in P\}$. A $\mathbf{c}$*-deepest-descent step* $\mathbf{y}$ *from* $\mathbf{x}^0$ is a vector $\mathbf{y} = \alpha \cdot \mathbf{g}$ for some circuit $\mathbf{g} \in \mathcal{C}(A, B)$ that maximizes the objective function improvement $-\mathbf{c}^T(\alpha\mathbf{g})$ among all circuits $\mathbf{g} \in \mathcal{C}(A, B)$ and all $\alpha > 0$ with $\mathbf{x}^0 + \alpha\mathbf{g} \in P$.

When the context is clear, we simply refer to a deepest-descent step $\mathbf{y}$ (dd-step), dropping information about $\mathbf{c}$, P, or $\mathbf{x}^0$. We call the term $c_{\mathbf{y}} = -\mathbf{c}^T\mathbf{y}$ the *deepest-descent improvement* (dd-improvement). It is known that repeatedly taking deepest-descent steps converges to an optimal solution in $\mathcal{O}(n \log(\mathbf{b}, \mathbf{c}, \mathbf{d}))$ iterations [3]. A k*-approximate dd-step* $\mathbf{z}$ is a circuit step whose improvement is at least $1/k$ of the improvement of a dd-step, as measured by the objective value $c_{\mathbf{y}}$ of a dd-step versus $\mathbf{c}^T\mathbf{z}$. It is known [6] that iteratively augmenting k-approximate dd-steps takes at most k-times more iterations to converge to an optimum. Thus, we are interested in exact and approximate computations of a deepest-descent step. We formally denote this search as follows.

Deepest-Descent Step Problem (dd-SP)	
Input:	$\mathbf{c} \in \mathbb{R}^n$, polyhedron $P \subset \mathbb{R}^n$, $\mathbf{x}^0 \in P$
Find:	$\mathbf{c}$-deepest-descent circuit step $\mathbf{y}$ in P from $\mathbf{x}^0$.

The natural question leading to our results is then: How hard is it to compute a dd-step?

1.1. Our Contribution

Our first positive result with respect to this question pertains to the efficient approximability of dd-SP:

Theorem 1. dd-SP *can be approximated within a factor of n in polynomial time.*

This follows by an averaging argument on well-behaved decompositions of the difference of two solutions to an LP as a set of (scaled) circuits.

The obvious follow-up question is whether an n-approximation can be significantly improved. We answer this negatively, even for a fairly restricted family of LPs:

Theorem 2. *Even for LPs over $0/1$-polytopes defined by a totally unimodular matrix and with unique optima,* dd-SP *cannot be approximated within $O(n^{1-\epsilon})$ for any $\epsilon > 0$ in polynomial time, unless $\mathsf{P} = \mathsf{NP}$.*

In particular, this demonstrates that to obtain a better approximation ratio or even polynomial tractability, one would need to consider an even more restricted family of LPs.

Further, we turn to the complexity of computing dd-steps exactly. De Loera et al. [7] have recently shown that dd-SP is NP-hard. However, a closer look at their construction reveals that they in fact show hardness of detecting whether it is possible to get to some optimum in one circuit step from a given initial point $\mathbf{x}^0$. We call this problem OCNP:

Optimal Circuit-Neighbor Problem (OCNP)

Input: $\mathbf{c} \in \mathbb{R}^n$, polyhedron $P \subset \mathbb{R}^n$, $\mathbf{x}^0 \in P$

Decide: Is there an optimum $\mathbf{x}^*$ with respect to $\min\{\mathbf{c}^T\mathbf{x} : \mathbf{x} \in P\}$ such that $\mathbf{x}^* - \mathbf{x}^0$ is a circuit direction?

Somewhat surprisingly, we show:

Theorem 3. *OCNP is solvable in polynomial time for LPs with a unique optimum.*

The standard trick of slightly perturbing the objective $\mathbf{c}$ of an LP makes some optimum unique, and the set of objectives with non-unique optima has volume 0, so in a sense OCNP is easy "almost always." This is contrasted by De Loera et al. [7] showing the NP-hardness of general OCNP.

This raises the following question: What is the complexity of DD-SP for LPs with a unique optimum, given that OCNP is easy? Despite the encouraging polynomial-time solvability of OCNP for this special case, we obtain as a byproduct of Theorem 2 that, unlike OCNP, DD-SP remains hard, even for the same, restricted family of LPs:

Theorem 4. DD-SP *is* NP*-hard, even for LPs over* $0/1$*-polytopes defined by a totally unimodular matrix and with unique optima.*

1.2. Connections to Previous Work

There are two papers in the literature with an especially strong connection to ours. We detail this connection separately, and discuss other related work hereafter.

Firstly, and most importantly, an inspiration for this note is the recent paper of De Loera et al. [7]. Our polynomial-time algorithm for OCNP (Theorem 3) stands in contrast to the results of De Loera et al. [7], where it is shown that finding optimal circuit-neighbors is NP-hard in general. Hence, the hardness of OCNP hinges on the existence of multiple optima. At this point, a flawed line of reasoning might become appealing:

The reduction in [7] comes from the directed Hamiltonian path problem. By introducing a negligible probability for one-sided error through the Isolation Lemma [8], we may assume that the reduction source instance $D = (N, F, s, t)$ on $n = |N|$ nodes, has a unique solution—that is, a unique Hamiltonian path from s to t. It is tempting to apply the reduction of [7], and use the above algorithm for OCNP to solve the produced instance. This is an optimization problem on the matching polytope $P_M(H)$ of some undirected bipartite graph H on $2n + 1$ vertices. We then have also solved the original instance of the (unique) Hamiltonian path problem in polynomial time. This fails, however, since the optima of the instance of OCNP are not in one-to-one correspondence with Hamiltonian paths in the input instance. Namely, the set of optima in the instance of OCNP is the set of matchings of size $n - 1$ in a graph H' obtained from H through the deletion of some edges. In particular, this set is not necessarily a singleton if the original graph D had a unique Hamiltonian path.

To save this approach, one might apply a perturbation to the cost vector of the produced LP on $P_M(H)$, to ensure uniqueness of solutions nonetheless (as remarked, uniqueness holds with probability 1). This perturbation, however, would have to retain precisely all optimal circuit neighbors, and not one of the other optima. Producing this perturbation would therefore require us to have at hand an optimal circuit neighbor in the first place.

To avoid confusion, we stress that "uniqueness" refers not to the solutions of OCNP itself, but to the LP that constitutes part of the input of OCNP (which implies uniqueness of the solution for OCNP). In other words, there might be a unique optimal circuit neighbor, while the LP has several optima.

Also note that while [7] discusses approximability, it does *not* concern DD-SP but a different problem: deciding what is the shortest path between two vertices of a polytope, either using the edges of the 1-skeleton, or using circuit steps. It is not clear to us whether any inapproximability of DD-SP follows from their construction.

Secondly, we make use of [9] for our positive results on OCNP and the n-

approximability of DD-SP. Most importantly, the set of circuit directions appear as a subset of the extreme rays of a polyhedral cone constructed from the original input [9, Theorem 3]. Recall that extreme rays are those not in the conic hull of any other rays in the object at hand.

Proposition 1. *Let $P = \{\mathbf{x} \in \mathbb{R}^n \mid A\mathbf{x} = \mathbf{b}, B\mathbf{x} \leq \mathbf{d}\}$ be a pointed polyhedron. The pointed cone*

$$C_{A,B} = \{(\mathbf{x}, \mathbf{y}^+, \mathbf{y}^-) \in \mathbb{R}^{n+2m_B} \mid A\mathbf{x} = \mathbf{0},\ B\mathbf{x} = \mathbf{y}^+ - \mathbf{y}^-,\ \mathbf{y}^+, \mathbf{y}^- \geq \mathbf{0}\}$$

is generated by the set of extreme rays $S \cup T'$, where:

1. *The set $S := \{(\mathbf{g}, \mathbf{y}^+, \mathbf{y}^-) \mid \mathbf{g} \in \mathcal{C}(A, B),\ y_i^+ = \max\{(B\mathbf{g})_i, 0\},\ y_i^- = \max\{-(B\mathbf{g})_i, 0\}\}$ gives the circuits of P.*
2. *The set $T' \subseteq T := \{(\mathbf{0}, \mathbf{y}^+, \mathbf{y}^-) \mid y_i^+ = y_i^- = 1$ for some $i \leq m_B,\ y_j^+ = y_j^- = 0$ for $j \neq i\}$ has size at most m_B.*

Informally, all circuits of P can be found as extreme rays of $C_{A,B}$: a projection of a vector in the set S onto its first n components gives the corresponding circuit $\mathbf{g}$. The 'non-circuits' in the set T are trivial to identify, and the corresponding projection just returns $\mathbf{0}$. Note further that the length of a bit encoding of $C_{A,B}$ is (in the order of) at most twice the bit encoding length of P. This implies that one can efficiently optimize linear objective functions over the set of (one-normed) circuits. Further, this allows the efficient computation of a conformal sum.

1.3. Related Work

Apart from the directly related papers mentioned in the previous subsection, there is vast literature revolving around pivoting rules for circuit augmentation algorithms, and circuits of linear programs in general. Without any pretense of being comprehensive, let us point to a couple of seminal works (below) and refer to [9] with respect to circuits, and to [7] for circuit augmentation and the references therein for a more extensive treatment.

The idea of performing augmenting steps in the direction of circuits instead of only edges during an execution of the simplex algorithm goes back at least to Bland's thesis [10] and is explored in detail in [3] and implemented, for example, in [11]. The notion of a circuit itself in turn was conceived only slightly before that by Rockafellar [12], and quite fruitfully [13, 14, 15] adapted to the integral case by Graver [16].

1.4. Outline

Our main contribution is a proof of the inapproximability of the computation of a dd-step within a factor of $O(n^{1-\epsilon})$, even when restricted to special classes of polyhedra. We begin by connecting to and generalizing previous results in the literature, in Sections 2 and 3. In Section 4, we prove our main result. In Section 5, we conclude with some open questions.

2. Efficiency of OCNP for LPs with unique optima

We begin by discussing the OCNP problem. De Loera et al. [7] showed that OCNP is NP-hard, and this implies that computing an optimal dd-step is NP-hard. (We call an optimization problem NP-hard if a corresponding decision version—is it possible to meet or exceed a given objective function value?—is NP-hard.) Recall the discussion in Section 1.2.

The proof in [7] is based on the underlying LP having multiple optima. While showing the claim under this assumption clearly is sufficient, note that for a given polyhedron P, the set $C_{\text{multi}} \subset \mathbb{R}^n$ of $\mathbf{c}$ for which there exist multiple optima has volume 0 in $\mathbb{R}^n$. Informally, it is enough to slightly perturb the objective function to create a unique optimum. We now show that this hardness does not hold if the underlying LP has a unique optimum.

Lemma 1. *Let* $\min\{\mathbf{c}^T\mathbf{x} : \mathbf{x} \in P\}$ *be an LP over a polyhedron* P *with a unique optimum* $\mathbf{x}^*$ *(that may not be known), and let* $\mathbf{x}^0 \in P$*. In polynomial time, it can be verified whether* $\mathbf{x}^* - \mathbf{x}^0$ *is a circuit direction.*

Proof. Generally, LPs are solvable in polynomial time. As the LP at hand has a unique optimum $\mathbf{x}^*$, this $\mathbf{x}^*$ can be found in polynomial time. Let $\mathbf{d} = \mathbf{x}^* - \mathbf{x}^0$. If $\mathbf{d} = \mathbf{0}$, there is nothing to prove: $\mathbf{x}^0$ itself already is optimal and we were able to verify so efficiently. Thus $\mathbf{d} \neq \mathbf{0}$ in the following.

Recall that the circuit directions of a polyhedron P appear as a subset S of the extreme rays of a polyhedral cone $C_{A,B}$, as in Proposition 1 [9]. We construct $\mathbf{d}_S := (\mathbf{d}, \mathbf{y}^+, \mathbf{y}^-)$, where $\mathbf{y}_i^{\pm} = \max\{\pm(B\mathbf{d})_i, 0\}$ as in the definition of S. The construction of $\mathbf{d}_S$ is efficient: $\mathbf{d}$ is copied over and $\mathbf{y}^{\pm}$ is derived from a matrix-vector product on the original input and component-wise comparisons.

Note that $\mathbf{d}_S \in C_{A,B}$, as $\mathbf{x}^*, \mathbf{x}^0 \in P$, and that $\mathbf{d}_S \notin T$ (as $\mathbf{d} \neq \mathbf{0}$). Thus, if $\mathbf{d}_S$ is an extreme ray of $C_{A,B}$, it can only be a member of S, which would imply that $\mathbf{d}$ is a circuit. A check whether a given $\mathbf{d}_S \in C_{A,B}$ is an extreme ray is possible in polynomial time: first, identify the set of active constraints of $\mathbf{d}_S$ with respect to $C_{A,B}$, i.e., check which constraints in the formulation of $C_{A,B}$ given in Proposition 1 are satisfied with equality, and construct the associated row submatrix of all active constraints; then perform a rank check for this submatrix – if its rank is precisely $(n + 2m_B) - 1$, then $\mathbf{d}_S$ lies in a one-dimensional face of $C_{A,B}$, i.e., in an extreme ray. These steps are possible in polynomial time because the bit encoding length of $C_{A,B}$ is at most twice the bit encoding length of P.

Summing up, $\mathbf{x}^*$ can be found efficiently, $\mathbf{d}_S$ can be constructed efficiently, and $\mathbf{d}_S$ is an extreme ray of $C_{A,B}$ if and only if $\mathbf{x}^* - \mathbf{x}^0$ is a circuit direction, and the required check is efficient, too. This proves the claim. □

As an immediate consequence, we obtain the following theorem.

Theorem 3. *OCNP is sovlable in polynomial time for LPs with a unique optimum.*

Because the set of objective functions for which there exist multiple optima for a given polyhedron has volume 0, Theorem 3 tells us that OCNP “almost always” can be decided efficiently.

3. n-Approximability of dd-SP

Next, we show that an efficient approximation of DD-SP with an error equivalent to the dimension of the underlying polyhedron is possible.

Lemma 2. *Let* $P = \{\mathbf{x} \in \mathbb{R}^n \mid A\mathbf{x} = \mathbf{b}, B\mathbf{x} \leq \mathbf{d}\}$*, let* $\mathbf{c} \in \mathbb{R}^n$ *and* $\mathbf{x}^0 \in P$*, and consider the LP* $\min\{\mathbf{c}^T\mathbf{x} : \mathbf{x} \in P\}$*. Then an* $(n-\operatorname{rank}(A))$*-approximation of a* $\mathbf{c}$*-deepest-descent circuit step in* P *from* $\mathbf{x}^0$ *can be computed in polynomial time.*

Proof. Let $\mathbf{y}$ be a $\mathbf{c}$-deepest-descent step $\mathbf{y}$ in P from $\mathbf{x}^0$ and let $\mathbf{x}^*$ be an optimum of $\min\{\mathbf{c}^T\mathbf{x} : \mathbf{x} \in P\}$. LPs generally are polynomial-time solvable, so an optimal $\mathbf{x}^*$ can be computed efficiently.

The vector $\mathbf{x}^* - \mathbf{x}^0$ can be written as a so-called *conformal sum* $\mathbf{x}^* - \mathbf{x}^0 = \sum_{i=1}^{n'} \alpha_i \mathbf{g}_i$, where $n' = n-\operatorname{rank}(A)$, $\alpha_i > 0$ and $\mathbf{g}_i$ is a circuit of P for all $i \leq n'$, and all the circuits $\mathbf{g}_i$ are sign-compatible with each other (and with $\mathbf{x}^* - \mathbf{x}^0$) [16, 17]. Such a conformal sum can be computed in polynomial time, see e.g. Algorithm 4 in [9].

Next, note that $\mathbf{c}^T\mathbf{y} \geq \mathbf{c}^T(\mathbf{x}^* - \mathbf{x}^0) = \sum_{i=1}^{n'} \mathbf{c}^T(\alpha_i \mathbf{g}_i)$. (Recall that $\mathbf{c}^T\mathbf{y}$ is negative, as LP is a minimization problem.) Thus, at least one of the $\alpha_i \mathbf{g}_i$ satisfies $\mathbf{c}^T(\alpha_i \mathbf{g}_i) \leq \frac{1}{n'}\mathbf{c}^T(\mathbf{x}^* - \mathbf{x}^0) \leq \frac{1}{n'}\mathbf{c}^T\mathbf{y}$. For a given conformal sum, it is efficient to find an $\alpha_i \mathbf{g}_i$ with smallest value $\mathbf{c}^T(\alpha_i \mathbf{g}_i)$.

By sign-compatibility of the $\mathbf{g}_i$, for any index set $I \subset \{1, \ldots, n\}$, $\mathbf{x}^0 + \sum_{i \in I} \alpha_i \mathbf{g}_i$ lies in P. In particular, this holds for $|I| = 1$: each of the $\mathbf{g}_i$ allows for a (maximal-length) circuit step $\beta_i \mathbf{g}_i$ from $\mathbf{x}^0$ that stays in P, and where $\beta_i \geq \alpha_i$. Note $\mathbf{c}^T(\beta_i \mathbf{g}_i) \leq \mathbf{c}^T(\alpha_i \mathbf{g}_i)$

For a given $\mathbf{g}_i$, it is efficient to compute the maximal β_i such that $\mathbf{x}^0 + \beta_i \mathbf{g}_i \in P$: each facet of the polyhedron provides an upper bound on β_i and one picks the smallest from them. Thus a $\beta_i \mathbf{g}_i$ with $\mathbf{c}^T(\beta_i \mathbf{g}_i) \leq \frac{1}{n'}\mathbf{c}^T\mathbf{y}$ can be computed in polynomial time. This proves the claim. □

As an immediate consequence, we obtain the following corollary.

Theorem 1. DD-SP *can be efficiently approximated within a factor of* n.

4. $O(n^{1-\epsilon})$-Inapproximability of dd-SP

The efficiency of OCNP for LPs with a unique optimum (Section 2) is one of the reasons for our interest in a proof for the inapproximability (and implied NP-hardness) of DD-SP that does not rely on this restriction. In Section 3, we saw that there is an efficient n-approximation. In this section, we show that this is essentially the best one can expect.

We will provide a proof for the claimed inapproximability of DD-SP that holds even when restricted to special classes of polyhedra. We call a polyhedron $P = \{\mathbf{x} \in \mathbb{R}^n \mid A\mathbf{x} = \mathbf{b}, B\mathbf{x} \leq \mathbf{d}\}$ with totally unimodular constraint matrices A and B and integral right-hand sides a TU-polyhedron.

To this end, we will perform a reduction from the following problem.

Directed Weighted Longest Cycle Problem (DWLCP)

Input:	Directed graph $G = (V, E, c)$ with arc costs $\mathbf{c} \in \mathbb{Q}^{\|E\|}$
Find:	Directed cycle of maximal cost

DWLCP is a generalization of the Directed (Unweighted) Longest Cycle Problem (DLCP), where the number of arcs of a cycle is counted, i.e., $c_{ij} = 1$ for all $(i, j) \in E$. Note that $|V|$ is the largest possible cost of a simple cycle for any instance of DLCP. For a graph $G = (V, E, c)$, DLCP cannot be polynomial-time approximated within $|V|^{1-\epsilon}$ for any $\epsilon > 0$, unless $P = NP$ [18]. This hardness transfers immediately to DWLCP: the cost $c_{\mathbf{x}} = \mathbf{c}^T\mathbf{x}$ of a longest cycle $\mathbf{x}$ cannot be polynomial-time approximated within $|c_{\mathbf{x}}|^{1-\epsilon}$ for any $\epsilon > 0$.

Through a reduction from DWLCP, we will prove that a dd-step $\mathbf{y}$ also cannot be polynomial-time approximated within $|c_{\mathbf{y}}|^{1-\epsilon}$ for any $\epsilon > 0$. In our construction, we will guarantee that the underlying LP has a unique solution (and, even stronger, that this fact is known), which allows us to obtain inapproximability and hardness even for such LPs. To this end, we begin with the polynomial construction of an instance of DWLCP from DWLP where all cycles have different costs while retaining the original "hierarchy" of costs. We denote the length of a bit encoding of a weighted graph G as $\mathcal{I}_G$.

Lemma 3. *Let $G = (V, E)$ be a directed graph. It is possible to construct a set of arc costs $\mathbf{c} \in \mathbb{Q}^{|E|}$ in polynomial time such that all cycles in $G' = (V, E, \mathbf{c})$ have different cost, and the cost of a cycle exceeds the number of arcs by strictly less than one. Further, the bit encoding length of G' is polynomial in the bit encoding length of G.*

Proof. Let $G = (V, E)$ be an unweighted directed graph. Let $n = |V|$ and $m = |E|$. First, we complement G to a weighted graph $G' = (V, E, \mathbf{c}')$, where $c'_{ij} = 1$ for all $(i, j) \in E$. In this graph, the cost of a cycle is measured through the number of arcs. Cycles with the same number of arcs have the same cost. To simplify notation, we will refer to a cycle interchangeably either as a subset of E or as a 0/1-vector $\mathbf{x}$ with components 1 precisely for the arcs on the cycle (a unit flow along the cycle); e.g., for two cycles $C_1, C_2 \subseteq E$ represented by vectors $\mathbf{x}_1, \mathbf{x}_2$, by $\mathbf{x}_1 \setminus \mathbf{x}_2$ we mean the arc set $C_1 \setminus C_2$. Note that $\mathcal{I}_{G'}$ is polynomial in $\mathcal{I}_G$: for each arc, only a (constant-size/single-bit) encoding of the number 1 is needed.

We will prove the claim through a simple perturbation on $\mathbf{c}'$ to resolve any ties between cycles. The new, perturbed costs are called $\mathbf{c}$. We are going to show that the perturbation is efficient and changes $\mathcal{I}_{G'}$ only polynomially.

Let $\mathbf{c} = \mathbf{c}' + \boldsymbol{\delta}$, where $\boldsymbol{\delta} = (\delta_1, \dots, \delta_m)^T$ and $\delta_i = 2^{-i}$. Informally $\delta_1 = \frac{1}{2}$, $\delta_2 = \frac{1}{4}$, $\delta_3 = \frac{1}{8}$, and so on. Each δ_i can be encoded in at most $m + 1$ bits, due to being the inverses of powers of 2. Thus, each $c_i = c'_i + \delta_i$ can be encoded in at most $m + 2$ digits and $\mathcal{I}_{G'} \leq (m + 2)\mathcal{I}_G$. As $\mathcal{I}_G \geq m$, the change in encoding length is polynomial. Further, $\mathbf{c}$ can be constructed in polynomial time.

It remains to prove that all cycles in G' are of different cost with respect to $\mathbf{c}$ and that the cost of cycles has increased by less than one. The latter is immediately clear from $\sum_{i=1}^{m} \delta_i < 1$. Note that the number of arcs of a cycle is $\mathbf{c}'^T\mathbf{x}$ and the cost with respect to $\mathbf{c}$ is $\mathbf{c}^T\mathbf{x}$. Let $\mathbf{x}_1$, $\mathbf{x}_2$ be two cycles and assume $\mathbf{c}'^T\mathbf{x}_1 > \mathbf{c}'^T\mathbf{x}_2$, which in particular implies $\mathbf{c}'^T\mathbf{x}_1 \geq \mathbf{c}'^T\mathbf{x}_2 + 1$. As $\mathbf{c}^T\mathbf{x}_2 < \mathbf{c}'^T\mathbf{x}_2 + \sum_{i=1}^{m} \delta_i$ and $\sum_{i=1}^{m} \delta_i < 1$, we have $\mathbf{c}^T\mathbf{x}_1 \geq \mathbf{c}'^T\mathbf{x}_1 > \mathbf{c}^T\mathbf{x}_2$.

Finally, consider two cycles $\mathbf{x}_1 \neq \mathbf{x}_2$ with $\mathbf{c}'^T\mathbf{x}_1 = \mathbf{c}'^T\mathbf{x}_2$. The cycles have the same number of arcs, so $\mathbf{x}_1 \backslash \mathbf{x}_2 \neq \emptyset$ and $\mathbf{x}_2 \backslash \mathbf{x}_1 \neq \emptyset$. Let index k be smallest among all arcs in $\mathbf{x}_1 \backslash \mathbf{x}_2$, and let l be smallest among all arcs used in $\mathbf{x}_2 \backslash \mathbf{x}_1$. Without loss of generality, assume $k < l$. Note $\delta_k > \sum_{i=k+1}^{m} \delta_i$. Thus $\mathbf{c}^T \mathbf{x}_1 - \mathbf{c}^T \mathbf{x}_2 \geq \delta_k - (\sum_{i=k+1}^{m} \delta_i) > 0$, i.e., $\mathbf{c}^T \mathbf{x}_1 > \mathbf{c}^T \mathbf{x}_2$. This proves the claim. □

Remark 1. It is natural to ask whether it is necessary to introduce numbers of exponential size into $\mathbf{c}$ in the Lemma above. In other words, does every integer vector $\mathbf{c}$ which preserves exactly one optimum of $\mathbf{c}'$ and does not introduce any new optima have some entry of order 2^n? This is open, but observe that if we require something stronger, the answer is "yes."

We show that every integer $\mathbf{c}$ which breaks all ties between cycles of the same length must have exponential entries. Clearly the number of cycles of length n can be $\Omega(2^n)$. Denote $c_{\max} = \|\mathbf{c}\|_\infty$. In order to get a distinct value $\mathbf{c}^T \mathbf{x}$ for every cycle $\mathbf{x}$ of length n, $c_{\max} \in \Omega(2^n)$, as otherwise there are not enough distinct values $\mathbf{c}^T \mathbf{x}$ since clearly $0 \leq \mathbf{c}^T \mathbf{x} \leq n \cdot c_{\max}$.

We are now ready to prove our main claim.

Theorem 2. *Let $P = \{\mathbf{x} \in \mathbb{R}^n \mid A\mathbf{x} = \mathbf{b}, B\mathbf{x} \leq \mathbf{d}\}$ with $A, B \in \mathbb{R}^{m \times n}$, let $\mathbf{c} \in \mathbb{R}^n$ and $\mathbf{x}^0 \in P$, and consider the LP $\min\{\mathbf{c}^T \mathbf{x} : \mathbf{x} \in P\}$. A deepest-descent circuit step $\mathbf{y}$ in P from $\mathbf{x}^0$ cannot be approximated within $O(n^{1-\epsilon})$ for any $\epsilon > 0$ in polynomial time, unless* P = NP. *The hardness holds for LPs with unique optima, over 0/1-polytopes, TU-polyhedra, or any combination thereof.*

Proof. We will prove the claim through a reduction from the Directed Longest Cycle Problem (DLCP), for which it was shown in [18] that no $|V|^{1-\epsilon}$-approximation can be computed for any $\epsilon > 0$ in polynomial time, unless P = NP, even in graphs of constant maximum out-degree Δ^+. By Lemma 3, for a given graph $G = (V, E)$ it is possible to efficiently construct a weighted graph $G' = (V, E, \mathbf{c})$ in which all cycles have a different cost and their cost lies strictly between the number of arcs of the cycle and that number plus one. The graph G' can be used as input for a Directed Weighted Longest Cycle Problem (DWLCP) and also has constant maximum out-degree Δ^+. If there was an efficient $|V|^{1-\epsilon}$-

approximation for DWLCP on G', then there would be an efficient $|V|^{1-\epsilon}$-approximation for DLCP on G. We will show that if there exists an algorithm to efficiently $O(n^{1-\epsilon})$-approximate DD-SP, then there exists an efficient $|V|^{1-\epsilon}$-approximation for DWLCP, and in turn DLCP – a contradiction unless $P = NP$. Further, the move from G to G' will allow us to show that we retain this hardness even for LPs with unique optima.

Let $G = (V, E)$ be a directed graph underlying an instance of DLCP and $G' = (V, E, \mathbf{c})$ the corresponding weighted directed graph with perturbed costs constructed as in Lemma 3, in turn an instance of DWLCP. Next, specify capacities $u_{ij} = 1$ for each $(i, j) \in E$ to obtain a network $G'' = (V, E, \mathbf{c}, \mathbf{u})$. The costs c_{ij} remain unchanged for all $(i, j) \in E$, i.e., they are the same as in G'.

This input can be used to specify a circulation problem. Recall that a circulation problem is a special case of a minimum-cost-flow problem and has a natural representation as an LP. Using the negative costs $-c_{ij}$ (recall the c_{ij} are positive), we obtain

$$\begin{aligned} \min\ & -\mathbf{c}^T\mathbf{x} \\ \text{s.t. } & A\mathbf{x} = \mathbf{0} \\ & \mathbf{0} \leq \mathbf{x} \leq \mathbf{1}, \end{aligned} \tag{LP}$$

where A is the node-arc incidence matrix of G', and $\mathbf{0}$ and $\mathbf{1}$ are vectors of all-zeros and all-ones of appropriate dimensions, respectively. The all-ones vector gives the capacity constraints. Let P refer to the polyhedron forming the feasible region of (LP). As node-arc incidence matrices are totally unimodular, and as the right-hand side vectors are the integral $\mathbf{0}$ and $\mathbf{1}$, P is a 0/1-polytope in $\mathbb{R}^n$, with $n = |E|$. There always exists an optimal vertex to an LP on a bounded polytope, so an optimal objective function value for (LP) is defined through a selection of arcs forming a circulation in G'. By the same argument as in Lemma 3, any subset of arcs sums up to a different total cost. Thus (LP) has a unique optimal solution.

Next, consider a trivial feasible flow $\mathbf{x}^0$ defined by $x^0_{ij} = 0$ for each $(i, j) \in E$. We are going to show that an efficient approximation of the dd-step in P from

$\mathbf{x}^0$ would imply an efficient approximation of DLCP.

Recall that the set of circuits of a node-arc incidence matrix A corresponds precisely to the simple *undirected* cycles underlying the network; a corresponding vector $\mathbf{g} \in \{-1, 0, 1\}$ would have entry 1 for each directed arc used in the 'correct' direction and -1 for each directed arc used in the 'wrong', opposite direction. The same holds for the circuits of $P = \{x \in \mathbb{R}^n \mid A\mathbf{x} = \mathbf{0}, \mathbf{0} \le \mathbf{x} \le \mathbf{1}\}$, as the inequality constraints $\mathbf{0} \le \mathbf{x} \le \mathbf{1}$ are represented through a constraint matrix $B = \binom{I}{-I}$, where I is the identity matrix; recall Definition 1. Since we have $x^0_{ij} = 0$ and $u_{ij} = 1$ for each $(i, j) \in E$, the step length α can always be set to 1 for any valid circuit, i.e., if there exists $\alpha > 0$ with $\mathbf{x}^0 + \alpha\mathbf{g} \in P$ for a circuit $\mathbf{g}$, then $\mathbf{x}^0 + \mathbf{g} \in P$ and $\mathbf{x}^0 + \beta\mathbf{g} \notin P$ for any $\beta > 1$. Further, any circuit $\mathbf{g}$ with $\mathbf{x}^0 + \mathbf{g} \in P$ can only have $0, 1$ entries, as $x^0_{ij} = 0$ for each $(i, j) \in E$. This means that edges can only be used in the correct direction. Therefore, an optimal dd-step $\mathbf{y}$ for (LP) from $\mathbf{x}^0$ is in one-to-one correspondence to a simple *directed* cycle of maximum length (as (LP) minimizes over negative arc costs).

Recall that by the hardness result in [18], we may assume that the maximum out-degree of G' is some fixed constant Δ^+, so in particular $|E| \le \Delta^+|V|$. Assume we had an algorithm that for a given $\epsilon > 0$ finds an $(n/\Delta^+)^{1-\epsilon}$-approximate dd-step $\mathbf{y}_\epsilon$ of the best dd-step $\mathbf{y}$, with dd-improvement $c_{\mathbf{y}_\epsilon}$ and $c_{\mathbf{y}}$, respectively. Then we have that $\frac{c_{\mathbf{y}}}{c_{\mathbf{y}_\epsilon}} \le (n/\Delta^+)^{1-\epsilon} = (|E|/\Delta^+)^{1-\epsilon} \le |V|^{1-\epsilon}$. We know that $c_{\mathbf{y}} = -\mathbf{c}^T\mathbf{y} = -\mathbf{c}^T\mathbf{g}$ and $c_{\mathbf{y}_\epsilon} = -\mathbf{c}^T\mathbf{y}_\epsilon = -\mathbf{c}^T(\alpha\mathbf{g}_\epsilon)$ for some $\alpha \in (0, 1]$ and some circuits $\mathbf{g}$ and $\mathbf{g}_\epsilon$. By the above, we may assume that $\alpha = 1$, so $c_{\mathbf{y}_\epsilon} = -\mathbf{c}^T(\mathbf{g}_\epsilon)$. Since $\frac{-\mathbf{c}^T\mathbf{g}}{-\mathbf{c}^T\mathbf{g}_\epsilon} = \frac{c_{\mathbf{y}}}{c_{\mathbf{y}_\epsilon}} \le |V|^{1-\epsilon}$, $\mathbf{g}_\epsilon$ corresponds to a cycle in G'' that approximates the longest cycle within a factor of $|V|^{1-\epsilon}$ (since by construction of the cost vector $\mathbf{c}$, a cycle has maximum cost if and only if it has maximum length). This would imply a polynomial-time $|V|^{1-\epsilon}$-approximation algorithm for general DLCP.

The polytope we used in this construction is a 0/1-polytope with a TU-matrix, and the LP at hand has a unique optimum and this fact is known apriori; see above. This shows that the hardness of approximation holds even for LPs adhering to all these restrictions. This proves the claim. □

As an immediate consequence, we obtain NP-hardness of DD-SP from this inapproximability result.

Theorem 4. DD-SP *is* NP*-hard, even for LPs over* $0/1$*-polytopes defined by a TU matrix and with a unique optimum.*

A direct proof of the NP-hardness of DD-SP would be possible through a reduction from Hamiltonian cycle instead of DWLCP, following a similar line of arguments as in the proof of Theorem 2. A perturbation of the arc costs would not be necessary, and neither would be the careful connection of $|V|$ and $|E|$ through the inapproximability of DWLCP for graphs with fixed maximum out-degree. However, to obtain the final part of Theorem 4 – that hardness persists even for LPs with unique optima – one would have to reduce from a variant of Hamiltonian cycle where the underlying graph has a *unique* circulation with a maximal number of arcs and one has the apriori information that there exists such a circulation. (This property is what would guarantee the existence of a unique optimum in (LP), and apriori knowledge thereof.) To the best of the authors' knowledge, hardness of this variant has not been studied yet in the literature.

5. Open Problems

We conclude with two open problems related to our results. First, Theorem 1 shows how to n-approximate DD-SP. However, for the purposes of solving an LP using dd-steps, this is irrelevant, as the first step of the algorithm is to completely solve the LP itself. Is there a combinatorial n-approximation of DD-SP, i.e., an algorithm, which does not use the polynomial solvability of an LP as a black-box? Actually, this would yield a new algorithm for linear programming, so to make the question well-posed, we ask whether there is a combinatorial n-approximation of DD-SP for some non-trivial class of constraint matrices? Secondly, we have shown strong inapproximability of DD-SP. What are (natural) classes of LP instances for which DD-SP admits, e.g., $\log(n)$- or even c-approximation, for some constant $c \in \mathbb{R}_+$? Potential candidate classes include uni– or bimodular LPs, and more generally, LPs with minors of bounded absolute value. Also, structurally restricted classes of LPs might be of interest. In particular, for n-fold LPs, which have a special block-structure, an approximation ratio for DD-SP polynomially depending only on the parameters (that is, block size) would be desirable, and would break below the barrier proved in this paper.

Acknowledgments

S. Borgwardt gratefully acknowledges support of this work through NSF award 2006183 *Circuit Walks in Optimization*, Algorithmic Foundations, CCF, Division of Computing and Communication Foundations, and through Simons Collaboration Grant 524210 *Polyhedral Theory in Data Analytics* before. C. Brand was supported by project 0016976 of OP VVV call 02_18_053. M. Koutecký and A. Feldmann were partially supported by Charles University project UNCE/SCI/004 and by the project 19-27871X of GA ČR.

References

[1] L. G. Khachiyan, A polynomial algorithm in linear programming, in: Doklady Akademii Nauk, Vol. 244, Russian Academy of Sciences, 1979, pp. 1093–1096.

[2] N. Karmarkar, A new polynomial-time algorithm for linear programming, in: Proceedings of the sixteenth annual ACM symposium on Theory of computing, 1984, pp. 302–311.

[3] J. A. De Loera, R. Hemmecke, J. Lee, On augmentation algorithms for linear and integer-linear programming: from Edmonds-Karp to Bland and beyond, SIAM Journal on Optimization 25 (4) (2015) 2494–2511.

[4] S. Borgwardt, E. Finhold, R. Hemmecke, On the circuit diameter of dual transportation polyhedra, SIAM Journal on Discrete Mathematics 29 (1) (2016) 113–121.

[5] F. Eisenbrand, C. Hunkenschröder, K. Klein, M. Koutecký, A. Levin, S. Onn, An Algorithmic Theory of Integer Programming, eprint arXiv:1904.01361 (2019). arXiv:1904.01361.

[6] K. Altmanová, D. Knop, M. Kouteckỳ, Evaluating and tuning n-fold integer programming, Journal of Experimental Algorithmics (JEA) 24 (1) (2019) 1–22.

[7] J. A. De Loera, S. Kafer, L. Sanità, Pivot Rules for Circuit-Augmentation Algorithms in Linear Optimization, eprint arXiv:1909.12863 (2019).

[8] L. G. Valiant, V. V. Vazirani, NP is as easy as detecting unique solutions, Theor. Comput. Sci. 47 (3) (1986) 85–93.

[9] S. Borgwardt, C. Viss, A polyhedral model for enumeration and optimization over the set of circuits, Discrete Applied Mathematics, in print (2019).

[10] R. G. Bland, Complementary orthogonal subspaces of n-dimensional euclidean space and orientability of matroids., Ph.D. thesis (1974).

[11] S. Borgwardt, C. Viss, An implementation of steepest-descent augmentation for linear programs, Operations Research Letters 48 (3) (2020) 323–328.

[12] R. T. Rockafellar, The elementary vectors of a subspace of $\mathbb{R}^N$, in: Combinatorial Mathematics and its Applications, University of North Carolina Press, 1969, pp. 104–127.

[13] R. Hemmecke, On the computation of Graver test sets, Tech. Report (1999).

[14] R. Hemmecke, On the computation of Hilbert bases of cones, in: Mathematical Software, ICMS 2002, World Scientific, 2002, pp. 307–317.

[15] R. Hemmecke, On the positive sum property and the computation of Graver test sets, Mathematical Programming, Ser. B 96 (2003) 247–269.

[16] J. E. Graver, On the foundation of linear and integer programming I, Mathematical Programming 9 (1975) 207–226.

[17] S. Onn, Nonlinear Discrete Optimization, Zurich Lectures in Advanced Mathematics, European Mathematical Society, 2010.

[18] A. Björklund, T. Husfeldt, S. Khanna, Approximating longest directed paths and cycles, in: Lecture Notes in Computer Science, Vol. 3142, 2004, pp. 222–233.