

# On Augmentation Algorithms for Linear and Integer-Linear Programming: From Edmonds-Karp to Bland and Beyond

Jesús A. De Loera

Department of Mathematics  
University of California  
One Shields Avenue  
Davis, CA 95616 USA

[deloera@math.ucdavis.edu](mailto:deloera@math.ucdavis.edu)

<http://www.math.ucdavis.edu/~deloera>

Raymond Hemmecke

Zentrum Mathematik  
Technische Universität München  
85747 Garching bei München, Germany  
[hemmecke@tum.de](mailto:hemmecke@tum.de)

<http://www-m9.ma.tum.de/Allgemeines/RaymondHemmecke>

Jon Lee

Department of Industrial and Operations Engineering  
University of Michigan  
1205 Beal Avenue  
Ann Arbor, MI 48109-2117 USA

[jonxlee@umich.edu](mailto:jonxlee@umich.edu)

<https://sites.google.com/site/jonleewebpage/>

Motivated by Bland’s linear-programming generalization of the renowned Edmonds-Karp efficient refinement of the Ford-Fulkerson maximum-flow algorithm, we discuss three closely-related natural augmentation rules for linear and integer-linear optimization. In several nice situations, we show that polynomially-many augmentation steps suffice to reach an optimum. In particular, when using “discrete steepest-descent augmentations” (i.e., directions with the best ratio of cost improvement per unit 1-norm length), we show that the number of augmentation steps is bounded by the number of elements in the Graver basis of the problem matrix, giving the first ever strongly polynomial-time algorithm for  $N$ -fold integer-linear optimization. Our results also improve on what is known for such algorithms in the context of linear optimization (e.g., generalizing the bounds of Kitahara and Mizuno for the number of steps in the simplex method) and are closely related to research on the diameters of polytopes and the search for a strongly polynomial-time simplex or augmentation algorithm.

*Key words:* augmentation, Graver basis, test set, circuit, elementary vector, linear program, integer program, Edmonds-Karp, steepest descent.

*MSC2000 subject classification:* Primary: 90C10; secondary: 65K05, 90C05, 52B55

---

**1. Introduction.** We consider a general framework for solving linear programs (LPs) and integer-linear programs (ILPs) of the form

$$\min \{ \mathbf{c}^\top \mathbf{x} : A\mathbf{x} = \mathbf{b}, \mathbf{0} \leq \mathbf{x} \leq \mathbf{u}, \mathbf{x} \in X \}, \quad (1)$$

where  $A \in \mathbb{Z}^{d \times n}$ ,  $\mathbf{b} \in \mathbb{Z}^d$ ,  $\mathbf{c} \in \mathbb{Z}^n$ , and where  $X = \mathbb{R}^n$  (LP) or  $X = \mathbb{Z}^n$  (ILP). We focus on solution algorithms that are based on an *augmentation procedure*. At each iteration, we have a current feasible point  $\mathbf{x}_k$ . An *augmentation direction*  $\mathbf{z}$  has  $\mathbf{x}_k + \alpha\mathbf{z}$  feasible for some  $\alpha > 0$  (which implies that  $\mathbf{z}$  is in the kernel of  $A$ ) and has  $\mathbf{c}^\top \mathbf{z} < 0$ . Together, the *augmentation* is  $\alpha\mathbf{z}$ , and we pass to the next feasible solution  $\mathbf{x}_{k+1} := \mathbf{x}_k + \alpha\mathbf{z}$ . We note the trivial fact that  $\mathbf{c}^\top \mathbf{x}_{k+1} < \mathbf{c}^\top \mathbf{x}_k$ . In this way, an augmentation procedure produces a sequence of feasible solutions  $\mathbf{x}_1, \mathbf{x}_2, \dots$  that are successive improvements on the value of the objective function, until an optimal solution is reached. Finally, an augmentation  $\alpha\mathbf{z}$  is *maximal* if, considering  $\mathbf{x}_k$  and  $\mathbf{z}$  to be fixed,  $\alpha$  is the largest value of  $t > 0$  for which  $\mathbf{x}_{k+1} := \mathbf{x}_k + t\mathbf{z}$  is feasible. If the augmentation is maximal,  $\mathbf{x}_{k+1}$  is the best point on the intersection of the feasible region with the half-line  $\{\mathbf{x}_k + t\mathbf{z} : t > 0\}$ . In what follows, we give upper bounds on the number of maximal augmentations required to reach an optimum, for both the LP and ILP cases.

Of course, augmentation algorithms are nothing new, and one of the best-known in the family is the classical simplex algorithm. For the case of LP, the simplex method is indeed an augmentation algorithm, where we start at a vertex of the polyhedron that is the feasible region, and the augmentation used at each iteration corresponds to an available edge direction at the current vertex. Of course, in the degenerate case, there can be considerable work to calculate an improving edge direction. Nonetheless, by limiting augmentation directions to available edge directions at the current vertex, and always choosing maximal augmentations, we insure that the next feasible solution is also a vertex, and so the simplex algorithm can continue. Similarly, the idea of augmentation has played a very important historical role in the algorithmic theory of network flows; in particular, there is the seminal and very well-known work of Edmonds and Karp (see [15]). They showed that for maximum-flow (essentially the same problem as maximizing the flow on a single arc of a flow-conservative network, subject to simple bounds on the other arcs), the number of augmentations (taken by the classic Ford-Fulkerson augmentation algorithm) is bounded by the number of arcs times the number of vertices, or slightly more crudely by the number of arcs squared, *when augmentations are always chosen to have the fewest number of arcs, and the augmentation is maximal*. In unpublished work, R. Bland ([6]) extended this, rather elegantly, to general LPs. Besides not being published by Bland in the 1970's, it is not well known even today. The result was implicitly alluded to in print in 1987 (see [4]), and mentioned more concretely in J. Lee's 1986 dissertation [25] (see Proposition 3.1 in the follow-on publication [26]). Bland himself made a concrete statement of it (still without proof) in 1992 (see [7]):

*"It was prompted by another result in the same Edmonds-Karp paper [15], that if one always augments on a shortest augmenting path, the number of augmentations in the Ford-Fulkerson maximum flow algorithm is less than the product of the numbers of nodes and edges. Fulkerson [17] had investigated the extent to which fundamental properties of networks generalize to broader classes of linear programming problems, where elementary vectors in the appropriate subspaces play the roles of circuits and cocircuits (minimal cutsets). Bland's dissertation [5] carried this further, and in later work he showed how the Edmonds-Karp result generalizes to arbitrary linear programming problems of the form*

$$\begin{aligned} & \text{maximize } x_0 \\ & \text{subject to } A\mathbf{x} = \mathbf{0}, \\ & \mathbf{l} \leq \mathbf{x} \leq \mathbf{u}. \end{aligned} \tag{2}$$

*Here the bound on the number of augmentations is the number of variables times the number of different lengths of normalized elementary vectors; in particular if  $A$  is totally unimodular the bound is the product of the dimensions of  $A$ , as in the max flow result of Edmonds and Karp, where  $A$  is the (totally unimodular) node-edge matrix of a directed graph. This seemed to be amusing, but without any obvious use, until late in 1978 when Paul Seymour [35]*

proved his remarkable decomposition theorem for unimodular matroids. Bland and Edmonds [8] used Seymour’s theorem to show how shortest augmentations could be computed in polynomial time when the constraint matrix  $A$  is totally unimodular. Cost scaling was used to extend the approach to a polynomial-time primal algorithm for totally unimodular linear programming problems

$$\begin{aligned} & \text{maximize } \mathbf{a}^\top \mathbf{x} \\ & \text{subject to } A\mathbf{x} = \mathbf{b}, \\ & \mathbf{l} \leq \mathbf{x} \leq \mathbf{u}. \end{aligned}$$

which are solved as a sequence of subproblems of the form (2). More recently Tardos [36] has given strongly polynomial algorithms for totally unimodular linear programming problems, and still more general classes of combinatorial optimization problems, using cost scaling in a far more clever way.”

Today, progress on the augmentation approach to LP continues: new families of augmentation algorithms for LP include those proposed in [12] (and rediscovered in [2]). Very recently, the work of Kern, Faigle, and Peis [23] rekindled our interest in Bland’s result as they, among other things, rediscovered it. Additionally, Kitahara and Mizuno [24] gave an upper bound for the number of different basic-feasible solutions generated by the simplex method. Their results are direct generalizations of Y. Ye’s work [37].

On the other hand, the study of augmentation algorithms for ILPs goes back at least to the 1970’s and was part of the study of “test sets” (see [18, 31, 32] and the references therein). More recently there have been important developments. In [33] the authors show that one can solve every integer-linear programming problem in polynomial time provided one can efficiently solve a special *directed-augmentation problem* (the directed-augmentation problem differs from the ordinary augmentation problem by splitting each direction into its positive and negative parts and considering linear objectives on each of these parts). These authors showed that if one can solve the directed-augmentation problem in polynomial time, then the original problem can be solved by a polynomial-time algorithm. Their main application is to specific combinatorial-optimization problems, such as the min-cost flow problem. Later [21] showed that if a “best-improving” augmentation  $\mathbf{z}$  in the “Graver basis” of the constraint matrix is chosen (the authors of [21] called this augmentation rule “greedy improvement”; here we will call this “deepest descent”), only polynomially-many augmentations (in the binary encoding length of the input data) must be performed in order to reach the optimal value. The paper [21] shows how to efficiently use an augmentation algorithm for separable convex objective functions and implicitly provided bounds like those presented here.

The set of possible augmentation directions we use depends on the type of problem: for the LP case, we use as the set of possible augmentation directions the *circuits* of  $A$ .  $\mathcal{C}(A)$  consists of the normalized *elementary vectors* (or *circuits*) associated with  $\ker(A) \setminus \{\mathbf{0}\}$  (see [29]) — that is, the vectors having (set-wise) minimal support in  $\ker(A) \setminus \{\mathbf{0}\}$ . The set of elementary vectors of  $\ker(A) \setminus \{\mathbf{0}\}$  is a finite set of lines through the origin, with the origin excluded. Usually, it is convenient to normalize in an arbitrary manner, so that  $\mathcal{C}(A)$  comprises a single point and its negative from each such line. In our context, Bland’s normalization uses the objective function so that  $\mathbf{c}^\top \mathbf{z} = -1$  for every augmentation direction — in his terms, such a  $\mathbf{z}$  is *unit augmenting*. For the vectors in  $\mathcal{C}(A)$ , we choose on each line the (nonzero) integer point closest to the origin and its negative as the normalized representatives.

For the IP case, the set of possible augmentation directions is the *Graver basis* of  $A$ , denoted by  $\mathcal{G}(A)$ . We obtain Graver’s original finite set of  $\sqsubseteq$ -minimal elements in  $\ker(A) \cap \mathbb{Z}^n \setminus \{\mathbf{0}\}$ , where  $\mathbf{u} \sqsubseteq \mathbf{v}$  if and only if  $u^{(i)}v^{(i)} \geq 0$  and  $|u^{(i)}| \leq |v^{(i)}|$  (see [18]). In general,  $\mathcal{G}(A)$  has a nice sign-compatible representation property: every (integer vector)  $\mathbf{z} \in \ker(A)$  can be written as  $\mathbf{z} = \sum \alpha_i \mathbf{g}_i$ , with  $\mathbf{g}_i \in \mathcal{G}(A)$ ,  $\alpha_i > 0$ ,  $\alpha_i \in \mathbb{Z}$ , and  $\alpha_i \mathbf{g}_i \sqsubseteq \mathbf{z}$ , for all  $i$ , and the sum involves at most  $2n - 2$  terms.

In fact,  $\mathcal{G}(A)$  is an inclusion-minimal set with this property. It should be noted that due to the sign-compatible representation we have

$$\|\mathbf{z}\|_1 = \sum \alpha_i \|\mathbf{g}_i\|_1,$$

where  $\|\cdot\|_1$  is the usual 1-norm.

We remark that the all circuits are members of the Graver basis. But they also provide an elegant sign-compatible representation *over the reals* for all non-zero elements of  $\ker(A)$ . Specifically, every vector  $\mathbf{z} \in \ker(A)$  can be written as  $\mathbf{z} = \sum \alpha_i \mathbf{g}_i$ , with  $\mathbf{g}_i \in \mathcal{C}(A)$ ,  $\alpha_i \geq 0$ ,  $\alpha_i \in \mathbb{R}$ , and the sum involves at most  $n$  terms. Note that the (worst-case) number of summands is smaller than in the general Graver decomposition. It is worth remarking that Graver bases can also be defined for the general mixed-integer case (see [19, Section 3.3]) — but in the mixed-integer setting, the Graver basis is not generally finite, in sharp contrast to the LP and ILP cases; so we do not consider that situation here as, unlike the LP and ILP cases, it requires the existence of some kind of specialized oracle to generate the augmentations one by one. For an introduction to Graver bases and the latest on augmentation algorithms for integer programming see the books [13] and [27].

**1.1. Our Contributions** The main contribution of this paper is an extension of Bland's theorem beyond LP to ILP. Our main results and proofs extend what Bland did for circuits to Graver bases. We look at several augmentation rules. Although they are very similar, our results here are naturally divided into the LP and ILP cases, with small but important technical changes. Also, depending on whether variables are restricted to be integers or not, we rely on different sets as the set of allowable augmentation directions, which we denote by  $\mathcal{T}(A)$ . The set  $\mathcal{T}(A)$  will change depending on the case.

In what follows, we will employ  $\mathcal{T}(A)$  in augmentation algorithms that iteratively replace a feasible (either continuous or integer) solution  $\mathbf{x}_k$  to  $A\mathbf{x} = \mathbf{b}$ ,  $\mathbf{0} \leq \mathbf{x} \leq \mathbf{u}$ , by a better feasible solution  $\mathbf{x}_{k+1} := \mathbf{x}_k + \alpha\mathbf{z}$ , where  $\mathbf{c}^\top \mathbf{z} < 0$  and  $\alpha > 0$ . We consider three specific augmentation rules.

**DEFINITION 1 (DISCRETE DEEPEST DESCENT).** With respect to a feasible solution  $\mathbf{x}_k$ , we choose  $\mathbf{z}$  such that  $-\alpha\mathbf{c}^\top \mathbf{z}$  is maximized among all  $\mathbf{z} \in \mathcal{T}(A)$  and  $\alpha > 0$  such that  $\mathbf{x}_{k+1} := \mathbf{x}_k + \alpha\mathbf{z}$  is feasible (note that for ILP this means that  $\alpha \in \mathbb{Z}$ ).

**DEFINITION 2 (DISCRETE DANTZIG DESCENT).** With respect to a feasible solution  $\mathbf{x}_k$ , we choose  $\mathbf{z}$  such that  $-\mathbf{c}^\top \mathbf{z}$  is maximized among all  $\mathbf{z} \in \mathcal{T}(A)$  such that  $\mathbf{x}_k + \epsilon\mathbf{z}$  is feasible for some  $\epsilon > 0$  (note that for LP this means for all sufficiently small  $\epsilon > 0$ , and for ILP this means for  $\epsilon = 1$ ). Then we take a maximum augmentation in such a direction. That is, we let  $\mathbf{x}_{k+1} := \mathbf{x}_k + \alpha\mathbf{z}$ , where  $\alpha$  is the largest value for which  $\mathbf{x}_k + \alpha\mathbf{z}$  is feasible (note that for ILP this means that  $\alpha \in \mathbb{Z}$ ).

**DEFINITION 3 (DISCRETE STEEPEST DESCENT).** With respect to a feasible solution  $\mathbf{x}_k$ , we choose  $\mathbf{z}$  such that  $-\mathbf{c}^\top \mathbf{z} / \|\mathbf{z}\|_1$  is maximized among all  $\mathbf{z} \in \mathcal{T}(A)$  such that  $\mathbf{x}_k + \epsilon\mathbf{z}$  is feasible for some  $\epsilon > 0$  (note that for LP this means for all sufficiently small  $\epsilon > 0$ , and for ILP this means for  $\epsilon = 1$ ). Then we take a maximum augmentation in such a direction. That is, we let  $\mathbf{x}_{k+1} := \mathbf{x}_k + \alpha\mathbf{z}$ , where  $\alpha$  is the largest value for which  $\mathbf{x}_k + \alpha\mathbf{z}$  is feasible (note that for ILP this means that  $\alpha \in \mathbb{Z}$ ).

Note that all three augmentation rules produce maximal augmentations. Practically speaking, there are more situations in which discrete steepest descent and discrete Dantzig descent can be practically implemented, as compared to discrete deepest descent. Still, it is interesting to analyze and contrast these augmentation rules. We derive the following main results concerning these augmentation rules for the ILP and the LP cases. In some structured situations, these bounds will provide very good guarantees of performance. Our ILP theorem is new, while our LP results extend what Bland started early on, and greatly extends the applicability of the bounds of Kitahara and Mizuno.

Next we state the results. Note that in what follows the base of all logarithms is two.

For the discrete deepest-descent rule, in the ILP case a polynomial time bound was first proved, but not explicitly stated as a proposition, in [21] and in [27] (see top of page 47 (end of proof of Lemma 3.10)). Here we make it very explicit:

**LEMMA 1 ((see also [21])).** *The number of discrete deepest-descent augmentations needed to reach an optimal solution from  $\mathbf{x}_0$  is bounded by  $(4n - 4) \log(\mathbf{c}^\top(\mathbf{x}_0 - \mathbf{x}_{\min}))$ .*

In addition, for the other two augmentation rules we prove:

**THEOREM 1 (ILP case).** *Let  $A \in \mathbb{Z}^{d \times n}$ ,  $\mathbf{b} \in \mathbb{Z}^d$  and  $\mathbf{c} \in \mathbb{Z}^n$  define the ILP*

$$\min \{ \mathbf{c}^\top \mathbf{x} : A\mathbf{x} = \mathbf{b}, \mathbf{0} \leq \mathbf{x} \leq \mathbf{u}, \mathbf{x} \in \mathbb{Z}^n \}.$$

*Let  $\mathbf{x}_0$  be an initial feasible solution, let  $\mathbf{x}_{\min}$  be an optimal solution, and let  $\gamma$  be the maximum non-zero entry (in absolute value) in any feasible solution. Then we have the following bounds on the number of augmentations to reach an optimal solution from  $\mathbf{x}_0$ .*

- (a) *The number of discrete Dantzig-descent augmentations needed to reach an optimal solution of the ILP is bounded by  $(4n - 4)\gamma \log(\mathbf{c}^\top(\mathbf{x}_0 - \mathbf{x}_{\min}))$ .*
- (b) *Any discrete steepest-descent direction (which by definition belongs to  $\mathcal{G}(A)$ ) is an overall steepest-descent direction (which could be any applicable direction from  $\mathbb{Z}^n$ ). Moreover, the number of discrete steepest-descent augmentations to reach an optimal solution of the given ILP is bounded by  $|\mathcal{G}(A)|$ .*

Next we present our results for augmentation algorithms in linear programming. The results for integer augmentations arguments in [21] and in [27] (see top of page 47 (end of proof of Lemma 3.10)) could be adapted to obtain an LP bound too, but the bound one obtains from [21] leads directly to an extra factor of  $n$  in the number of augmentations due to the strategy presented there to ensure termination. Our contribution is that we manage to get rid of this factor of  $n$ :

**LEMMA 2.** *The number of discrete deepest-descent augmentations needed to reach an optimal solution from  $\mathbf{x}_0$  is bounded by  $2n \log(\delta \mathbf{c}^\top(\mathbf{x}_0 - \mathbf{x}_{\min}))$ .*

Moreover we prove the following extensions to the other augmentation rules for LPs:

**THEOREM 2 (LP case).** *Let  $A \in \mathbb{Z}^{d \times n}$ ,  $\mathbf{b} \in \mathbb{Z}^d$  and  $\mathbf{c} \in \mathbb{Z}^n$  define the LP*

$$\min \{ \mathbf{c}^\top \mathbf{x} : A\mathbf{x} = \mathbf{b}, \mathbf{0} \leq \mathbf{x} \leq \mathbf{u}, \mathbf{x} \in \mathbb{R}^n \}.$$

*Let  $\mathbf{x}_0$  be an initial feasible solution, let  $\mathbf{x}_{\min}$  be an optimal solution, let  $\gamma$  be the maximum non-zero entry (in absolute value) in any feasible solution, and let  $\delta$  denote the least common multiple of all subdeterminants of  $A$ . Then we have the following bounds on the number of augmentations to reach an optimal solution from  $\mathbf{x}_0$ .*

- (a) *The number of discrete Dantzig-descent augmentations needed to reach an optimal solution from  $\mathbf{x}_0$  is no more than  $2n^2\delta\gamma \log(\delta \mathbf{c}^\top(\mathbf{x}_0 - \mathbf{x}_{\min}))$ .*
- (b) *Any discrete steepest-descent direction (which by definition belongs to  $\mathcal{C}(A)$ ) is an overall steepest-descent direction (which could be any applicable direction from  $\mathbb{R}^n$ ). Moreover, the number of discrete steepest-descent augmentations to reach an optimal solution of the given LP is bounded by  $|\mathcal{C}(A)|$ .*

While by nature LP augmentation algorithms cannot have cycling, as can happen for the simplex method, it should be noted that the LP case of the discrete deepest-descent and the discrete Dantzig-descent augmentation algorithms only guarantee that we reach a feasible solution  $\mathbf{x}_k$  with objective value “close enough” to the optimal value. Whether any given feasible solution is “close



enough” can be checked by generating a vertex of the feasible region with objective value that is at least as good as  $\mathbf{c}^\top \mathbf{x}_k$ . Finding such a vertex can be done by augmenting  $\mathbf{x}_k$  only along such (discrete deepest-/Dantzig-descent) circuit directions that lead to a better feasible solution with an additional component reaching its lower or upper bound. Geometrically, this corresponds to (iterative) augmentation within the smallest face of the polyhedron  $\{\mathbf{x} : A\mathbf{x} = \mathbf{b}, \mathbf{0} \leq \mathbf{x} \leq \mathbf{u}, \mathbf{x} \in \mathbb{R}^n\}$  that contains  $\mathbf{x}_k$ . (The circuits of the problem matrix  $A$  provide such a restricted optimality certificate). We recommend the example in [19]. Finally, again using the circuits of  $A$ , it can be checked whether the vertex found is optimal. The overall closeness-test requires at most  $n$  augmentations.

Finally we present the interesting consequences of the main theorems. Note that the bounds in part (c) of Theorems 1 and 2 only depend on  $A$ , but not on  $\mathbf{b}$ ,  $\mathbf{c}$ , and the particular initial feasible solution  $\mathbf{x}_0$  chosen. As a direct consequence of Part (b) of Theorem 1, we obtain the following corollaries.

**COROLLARY 1.** *For the pure 0/1 ILP  $\min\{\mathbf{c}^\top \mathbf{x} : A\mathbf{x} = \mathbf{b}, \mathbf{0} \leq \mathbf{x} \leq \mathbf{1}, \mathbf{x} \in \mathbb{Z}^n\}$  the number of discrete deepest-/Dantzig-/steepest-descent augmentations is bounded by  $O(n \log(\|\mathbf{c}\|_1))$ .*

We note that in the LP case, the proof of Theorem 2, part (c), immediately gives Bland’s fundamental result:

**COROLLARY 2 (Bland’s Theorem).** *The number of discrete steepest-descent augmentations needed to solve  $\min\{\mathbf{c}^\top \mathbf{x} : A\mathbf{x} = \mathbf{b}, \mathbf{0} \leq \mathbf{x} \leq \mathbf{u}, \mathbf{x} \in \mathbb{R}^n\}$  is bounded by the number of (different) positive values of  $-\mathbf{c}^\top \mathbf{z} / \|\mathbf{z}\|_1$  over all (elementary vectors)  $\mathbf{z} \in \mathcal{C}(A)$  times the number  $n$  of variables.*

As for totally-unimodular matrices  $A$ ,  $\mathcal{C}(A) = \mathcal{G}(A)$ , i.e., they coincide for the LP and ILP cases, the proof of Theorem 2, part (c), also implies:

**COROLLARY 3.** *For totally-unimodular matrix  $A$ , the number of discrete steepest-descent augmentations needed to solve  $\min\{\mathbf{c}^\top \mathbf{x} : A\mathbf{x} = \mathbf{b}, \mathbf{0} \leq \mathbf{x} \leq \mathbf{u}, \mathbf{x} \in \mathbb{Z}^n\}$  is bounded by the number of (different) positive values of  $-\mathbf{c}^\top \mathbf{z} / \|\mathbf{z}\|_1$  over all (elementary vectors)  $\mathbf{z} \in \mathcal{C}(A)$  times the number  $n$  of variables.*

For a totally-unimodular matrix  $A$ ,  $\mathcal{C}(A)$  consists only of vectors with (at most  $d + 1$ ) entries in  $\{-1, 0, 1\}$ . Thus, for  $\mathbf{z} \in \mathcal{C}(A)$  we have  $|\mathbf{c}^\top \mathbf{z}| \leq \|\mathbf{c}\|_1$  and  $\|\mathbf{z}\|_1$  takes on at most  $d + 1$  different values. Plugging this into Corollary 3 we get the following.

**COROLLARY 4.** *For totally-unimodular matrix  $A$ , the number of discrete steepest-descent augmentations needed to solve  $\min\{\mathbf{c}^\top \mathbf{x} : A\mathbf{x} = \mathbf{b}, \mathbf{0} \leq \mathbf{x} \leq \mathbf{u}, \mathbf{x} \in \mathbb{Z}^n\}$  is bounded by  $n(d + 1)\|\mathbf{c}\|_1$ .*

From this we immediately recover the complexity bound for the algorithm of Edmonds and Karp to find maximum flows in a network: let  $A$  be the node/arc-incidence matrix of a connected directed graph. Note that one row of  $A$  is linearly dependent on the other rows and thus can be removed from  $A$ . Then  $n = |E|$  and  $d = |V| - 1$ . As we maximize the flow on a specific (auxiliary) arc (from sink to source), we have  $\|\mathbf{c}\|_1 = 1$ . Thus,  $n(d + 1)\|\mathbf{c}\|_1 = |E| \cdot |V|$  bounds the number of discrete steepest-descent augmentations to solve the max-flow problem. This observation is not surprising, as the augmentation approach using Graver bases specializes to the algorithm by Edmonds and Karp in the setting of maximum flows.

It is worth recalling that although the complexity statements in Corollary 1 and Corollary 4 depend on the unary size of  $\mathbf{c}$ , these two results actually can be improved based on the results of Frank and Tardos [16]. Frank and Tardos used Diophantine approximation to replace  $\mathbf{c}$  with a new objective function  $\mathbf{c}'$  where the integer numbers occurring in the entries are small but define an equivalent problem with the same optima. Furthermore, the new weight function  $\mathbf{c}'$  can be found in strongly polynomial-time. In conclusion, if we are able to generate in polynomial time the corresponding augmentation elements of the Graver basis according to one of the three rules, we

obtain strongly polynomial-time algorithms. Of course, this is in general hard to do other than by computing the entire Graver basis, but for special matrices  $A$ , one can do much better.

Our results on augmentations for totally-unimodular matrices provide another interesting geometric result. For years, researchers have been looking at the diameter of the graph of polyhedra (i.e., the graph whose nodes are the vertices of the convex polyhedron  $P = \{\mathbf{x} : A\mathbf{x} = \mathbf{b}, \mathbf{0} \leq \mathbf{x} \leq \mathbf{u}\}$ ). It has been shown in [9] that the diameter of the graph of totally-unimodular  $d$ -dimensional polytopes (i.e., the length of the longest shortest path between a pair of nodes) is bounded above by  $d^{3.5} \log(d)$ . We wish to stress that in our circuit augmentations, we do not always follow edges of the polyhedron  $P$ . Rather, we may cut through the interior of  $P$  or the interior of some faces. This suggests the notion of *circuit diameter*. The circuit distance from  $v_1$  to  $v_2$  is the smallest number of circuit augmentations needed to go from  $v_1$  to  $v_2$ . We can then define the circuit diameter as the maximum number of steps along circuit basis directions that are needed to go from any vertex of the polyhedron to any other vertex of the polyhedron. This notion was first introduced and investigated in the article [10]. More recently other generalizations of the notion of diameter using circuits were introduced in [11]. One can show that the circuit diameter of a polyhedron is bounded from above by the usual combinatorial diameter of polyhedra (this is because arcs are themselves some of the possible augmentation directions, a subset of the circuits). In this context, our results show that one can bound the *circuit diameter* for totally-unimodular polytopes in standard form as follows

**COROLLARY 5.** *For a  $d \times n$  totally-unimodular matrix  $A$ , the circuit diameter of the polyhedron  $P := \{\mathbf{x} : A\mathbf{x} = \mathbf{b}, \mathbf{0} \leq \mathbf{x} \leq \mathbf{u}\}$  is bounded above by  $2(n(d+1)(n-d))$ .*

This corollary is a good general bound for totally-unimodular matrices, but we suspect it can be further improved, as we already know that for network-flow polytopes a better bound is possible. Indeed, Orlin [28] designed a polynomial-time primal network-simplex algorithm for the minimum-cost flow problem which gives a graph-diameter bound of  $O(|E||V|\log(|V|))$ , with  $E, V$  equal to the sets of arcs and nodes in the network.

*Proof:* To see this, take any vertex  $v$  of  $P$ . Choose a cost vector specific to  $v$ : namely, let  $c_i = 1$  if  $v_i = 0$  and  $c_i = -1$  if  $v_i = u_i$ . Otherwise, put  $c_i = 0$ . Thus, this new objective function  $\mathbf{c}$  has  $n - d = n - \text{rank}(A)$  many nonzero entries. Thus, by Corollary 4, with the steepest-descent rule, any other vertex is connected to  $v$  with no more than  $n(d+1)(n-d)$  many augmentations. Hence, the path between any pair of vertices of  $P$  is bounded by  $2n(d+1)(n-d)$ .  $\square$

Finally, let us consider optimization problems for which the constraint matrix is structured. Recall that an  $N$ -fold matrix is a matrix of the form

$$[A, B]^{(N)} := \begin{pmatrix} B & B & \cdots & B \\ A & O & & O \\ O & A & & O \\ & & \ddots & \\ O & O & & A \end{pmatrix}.$$

For fixed matrices  $A$  and  $B$ , the size of the Graver basis of  $[A, B]^{(N)}$  (and its binary encoding length) increases only polynomially with  $N$ . Combining this with Theorems 1 and 2, parts (c), we obtain the following result.

**THEOREM 3.** *For fixed matrices  $A$  and  $B$ , the associated families of  $N$ -fold LPs and ILPs can be solved in strongly polynomial time.*

This generalizes the results from [20, 21], which showed that for fixed matrices  $A$  and  $B$ , the corresponding  $N$ -fold ILPs could be solved in time polynomial in  $N$ , in fact in  $O(N^3)$  steps. Theorem 3 strengthens this to strong polynomiality. As a direct consequence we obtain the following results:

**COROLLARY 6.** *The following special cases of  $N$ -fold matrices can be solved in strongly polynomial time in the linear and integer case:*

- *All 2-way transportation problems with a fixed number of rows or columns.*
- *All 3-way transportation problems with two of the three dimensions fixed.*
- *All  $d$ -way transportation problems with  $d - 1$  dimensions of fixed constant value.*

**2. Proofs** We present the proofs to the above results in separate subsections. We will use the following lemma (a slight variation from Theorem 3.1 in [1]) which establishes the bounds we claim once we can guarantee sufficient improvement at each iteration:

**LEMMA 3.** *Let  $\epsilon > 0$  be given. Moreover, let  $H$  denote the difference between maximum and minimum objective-function values of the LP/ILP problem in  $n$  variables. Suppose that  $f^k = \mathbf{c}^\top \mathbf{x}_k$  is the objective-function value of the solution  $\mathbf{x}_k$  at the  $k$ -th iteration of an algorithm and that  $f^* = \mathbf{c}^\top \mathbf{x}_{\min}$  is the minimum objective-function value. Furthermore, suppose that the algorithm guarantees that for every augmentation  $k$ ,*

$$(f^k - f^{k+1}) \geq \beta(f^k - f^*)$$

*(i.e., the improvement at augmentation  $k + 1$  is at least  $\beta$  times the maximum possible improvement). Then the algorithm reaches a solution with  $f^k - f^* < \epsilon$  in no more than  $2 \log(H/\epsilon)/\beta$  augmentations.*

*Proof:* Without loss of generality, we assume that  $\mathbf{c}$  is an integer vector. (Thus for the ILP version the values  $f^k$  are decreasing successive integer values.) The biggest necessary change of value of the objective function occurs when we start the augmentation at an maximum point  $\mathbf{x}_{\max}$  and end with some minimum optimal point  $\mathbf{x}_{\min}$ . Thus, we have  $H = f^{\max} - f^{\min}$ .

If we had at every augmentation an improvement of at least  $\beta(f^{\max} - f^{\min})/2$ , then in no more than  $2/\beta$  augmentations we would have reached the optimum. But if this improvement is not achieved at each augmentation, say at the  $q$ -th augmentation we have  $f^q - f^{q+1} \leq \beta(f^{\max} - f^{\min})/2$  then, together with the hypothesis  $(f^q - f^{q+1}) \geq \beta(f^q - f^{\min})$ , we get that

$$f^q - f^{\min} \leq (f^{\max} - f^{\min})/2 = H/2.$$

In other words, the overall improvement reached so far is at least half of the maximum possible improvement  $H$ . In conclusion, after  $2/\beta$  augmentations, we have either reached the optimum or have at least divided the possible gap by 2. Therefore in no more than  $2/\beta \log_2(H/\epsilon)$  augmentations we reach a solution with  $f^k - f^* < \epsilon$ .  $\square$

It is important to note that in the ILP case  $f^k$  is integer, and we can apply Lemma 3 with  $\epsilon = 1$  and conclude that we can reach the optimum in  $O(\log(H)/\beta)$  augmentations. In the LP case, let  $\delta$  denote the least common multiple of all subdeterminants of  $A$ . Observe that once we find a feasible solution  $\mathbf{x}_k$  with objective value  $f^k$  satisfying  $f^k - f^* < \epsilon = 1/\delta$ , then any vertex with an objective value of at most  $f^k$  must be optimal. As explained above (right before the statement of Theorem 1), such a vertex can be found from  $\mathbf{x}_k$  in at most  $n$  additional augmentations. This leads to extra factors of  $n$  and of  $\log(\delta)$  in the bounds for the LP cases compared to the ILP cases.

**2.1. Proof of Lemma 1 and Theorem 1** Let us assume that  $\mathbf{x}_k$  is a non-optimal feasible solution, and let  $\mathbf{x}_{\min}$  be an optimal solution to the ILP. Then there exists a (sign-compatible) representation

$$\mathbf{x}_{\min} - \mathbf{x}_k = \sum \alpha_i \mathbf{g}_i,$$

with  $\alpha_i > 0$ ,  $\alpha_i \in \mathbb{Z}$  and with  $\alpha_i \mathbf{g}_i \subseteq \mathbf{x}_{\min} - \mathbf{x}_k$ . Moreover, due to Sebö's result [34], at most  $2n - 2$  summands are needed.



Note that sign-compatibility of the representation  $\mathbf{x}_{\min} - \mathbf{x}_k = \sum \alpha_i \mathbf{g}_i$  implies that for all  $i$  the vectors  $\mathbf{x}_k + \alpha_i \mathbf{g}_i$  and  $\mathbf{x}_{\min} - \alpha_i \mathbf{g}_i$  are all feasible solutions, since their components lie between the components of  $\mathbf{x}_k$  and of  $\mathbf{x}_{\min}$ . Moreover, we can observe that for all such sign-compatible representations  $\mathbf{x}_{\min} - \mathbf{x}_k = \sum \alpha_i \mathbf{g}_i$  we must have  $\mathbf{c}^\top \mathbf{g}_i \leq 0$  for all  $i$ , as otherwise  $\mathbf{x}_{\min} - \alpha_i \mathbf{g}_i$  would be a feasible solution with  $\mathbf{c}^\top(\mathbf{x}_{\min} - \alpha_i \mathbf{g}_i) = \mathbf{c}^\top \mathbf{x}_{\min} - \alpha_i \mathbf{c}^\top \mathbf{g}_i < \mathbf{c}^\top \mathbf{x}_{\min}$ , contradicting the minimality of  $\mathbf{x}_{\min}$ .

Next, we analyze what happens for each choice of augmentation rule:

**2.1.1. Proof of Lemma 1: Discrete deepest descent.** We observe that

$$0 > \mathbf{c}^\top(\mathbf{x}_{\min} - \mathbf{x}_k) = \mathbf{c}^\top \sum \alpha_i \mathbf{g}_i = \sum \alpha_i \mathbf{c}^\top \mathbf{g}_i \geq -(2n-2)\Delta$$

where  $\Delta > 0$  is the largest value of  $-\alpha \mathbf{c}^\top \mathbf{z}$  over all  $\mathbf{z} \in \mathcal{G}(A)$  and integer  $\alpha > 0$  for which  $\mathbf{x}_k + \alpha \mathbf{z}$  is feasible. Rewriting this, we get

$$\Delta \geq \frac{\mathbf{c}^\top(\mathbf{x}_k - \mathbf{x}_{\min})}{2n-2}.$$

Now let  $\alpha \mathbf{z}$  be the discrete deepest-descent augmentation applied to  $\mathbf{x}_k$ , leading to  $\mathbf{x}_{k+1} := \mathbf{x}_k + \alpha \mathbf{z}$ . Then we get  $\Delta = -\alpha \mathbf{c}^\top \mathbf{z}$  and

$$\mathbf{c}^\top(\mathbf{x}_k - \mathbf{x}_{k+1}) = -\alpha \mathbf{c}^\top \mathbf{z} = \Delta \geq \frac{\mathbf{c}^\top(\mathbf{x}_k - \mathbf{x}_{\min})}{2n-2}.$$

Thus, we have a factor of  $\beta = 1/(2n-2)$  of objective-function decrease at each augmentation, leading to the desired polynomial number of augmentations via Lemma 3 taking  $\epsilon = 1$ . In this case, we get the number of augmentations bounded by  $(4n-4) \log(\mathbf{c}^\top(\mathbf{x}_0 - \mathbf{x}_{\min}))$ .

**2.1.2. Proof of part (a) of Theorem 1: Discrete Dantzig descent.** We observe that

$$0 > \mathbf{c}^\top(\mathbf{x}_{\min} - \mathbf{x}_k) = \mathbf{c}^\top \sum \alpha_i \mathbf{g}_i = \sum \alpha_i \mathbf{c}^\top \mathbf{g}_i \geq -\Delta_0 \sum \alpha_i \geq -(2n-2)\Delta_0 \alpha_{\max},$$

where this time  $\Delta_0 > 0$  denotes the greatest value of  $-\mathbf{c}^\top \mathbf{z}$  over all  $\mathbf{z} \in \mathcal{G}(A)$  for which  $\mathbf{x}_k + \mathbf{z}$  is still feasible and where  $\alpha_{\max} = \max \{\alpha_i\}$ . Rewriting this, we get

$$\Delta_0 \geq \frac{\mathbf{c}^\top(\mathbf{x}_k - \mathbf{x}_{\min})}{(2n-2)\alpha_{\max}}.$$

Now let  $\alpha \mathbf{z}$  be the discrete Dantzig-descent augmentation applied to  $\mathbf{x}_k$ , leading to  $\mathbf{x}_{k+1} := \mathbf{x}_k + \alpha \mathbf{z}$ . Then we get

$$\mathbf{c}^\top(\mathbf{x}_k - \mathbf{x}_{k+1}) = -\alpha \mathbf{c}^\top \mathbf{z} = \alpha \Delta_0 \geq \Delta_0 \geq \frac{\mathbf{c}^\top(\mathbf{x}_k - \mathbf{x}_{\min})}{(2n-2)\alpha_{\max}} \geq \frac{\mathbf{c}^\top(\mathbf{x}_0 - \mathbf{x}_{\min})}{(2n-2)\gamma},$$

where  $\gamma$  is the maximum entry in any feasible integer solution (or, equivalently, in any vertex of  $P_I$ ). Thus, we have a factor of  $\beta = 1/((2n-2)\gamma)$  of objective-function decrease at each augmentation leading to the desired polynomial number of augmentations via Lemma 3 taking  $\epsilon = 1$ . In this case we get the number of augmentations bounded by  $(4n-4) \gamma \log(\mathbf{c}^\top(\mathbf{x}_0 - \mathbf{x}_{\min}))$ .

**2.1.3. Proof of part (b) of Theorem 1: Discrete steepest descent.** We begin with a series of lemmas. Note that the proof for the LP case follows exactly the same lines, since we do not use integrality of the components in our arguments.

**LEMMA 4.** *Let  $\mathbf{x}_k$  be a feasible solution, and let  $\mathbf{z}$  be an associated steepest descent direction. Then there is some augmentation direction  $\mathbf{g} \in \mathcal{G}(A)$  from  $\mathbf{x}_k$ , with  $-\mathbf{c}^\top \mathbf{g} / \|\mathbf{g}\|_1 \geq -\mathbf{c}^\top \mathbf{z} / \|\mathbf{z}\|_1$ .*

*Proof.* There is a sign-compatible representation  $\mathbf{z} = \sum \alpha_i \mathbf{g}_i$  via elements  $\mathbf{g}_i \in \mathcal{G}(A)$ . Observe that due to the sign-compatible representation,  $\mathbf{x}_k + \alpha_i \mathbf{g}_i$  is also feasible for all  $i$ . (The components of  $\mathbf{x}_k + \alpha_i \mathbf{g}_i$  lie between those of  $\mathbf{x}_k$  and  $\mathbf{x}_k + \mathbf{z}$ , implying that  $\mathbf{0} \leq \mathbf{x}_k + \alpha_i \mathbf{g}_i \leq \mathbf{u}$ .) In other words, all  $\alpha_i \mathbf{g}_i$  are applicable augmentations at  $\mathbf{x}_k$ .

It remains for us to show that there exists some index  $i$  with  $-\mathbf{c}^\top \mathbf{g}_i / \|\mathbf{g}_i\|_1 \geq -\mathbf{c}^\top \mathbf{z} / \|\mathbf{z}\|_1$ . Assume to the contrary that we have  $-\mathbf{c}^\top \mathbf{g}_i / \|\mathbf{g}_i\|_1 < -\mathbf{c}^\top \mathbf{z} / \|\mathbf{z}\|_1$  for all  $i$ . This yields

$$\begin{aligned} -\mathbf{c}^\top \mathbf{z} &= -\sum \alpha_i \mathbf{c}^\top \mathbf{g}_i \\ &= \sum \alpha_i \|\mathbf{g}_i\|_1 \frac{-\mathbf{c}^\top \mathbf{g}_i}{\|\mathbf{g}_i\|_1} \\ &< \sum \alpha_i \|\mathbf{g}_i\|_1 \frac{-\mathbf{c}^\top \mathbf{z}}{\|\mathbf{z}\|_1} \\ &= \frac{-\mathbf{c}^\top \mathbf{z}}{\|\mathbf{z}\|_1} \sum \alpha_i \|\mathbf{g}_i\|_1 \\ &= \frac{-\mathbf{c}^\top \mathbf{z}}{\|\mathbf{z}\|_1} \|\mathbf{z}\|_1 \\ &= -\mathbf{c}^\top \mathbf{z}, \end{aligned}$$

a contradiction.  $\square$

Lemma 4 states that among all steepest-descent directions applicable at a feasible solution  $\mathbf{x}_k$ , there is always one in  $\mathcal{G}(A)$ . Or in other words, the discrete steepest-descent rule is a steepest descent rule as claimed in Theorems 1 and 2, parts(c).

LEMMA 5. *Let  $\mathbf{x}_k$  be a feasible solution, let  $\alpha \mathbf{s}$  be a steepest-descent augmentation relative to  $\mathbf{x}_k$ , and let  $\mathbf{x}_{k+1} := \mathbf{x}_k + \alpha \mathbf{s}$ . Let  $\beta \mathbf{t}$  be a steepest-descent augmentation relative to  $\mathbf{x}_{k+1}$ . Then we have  $-\mathbf{c}^\top \mathbf{s} / \|\mathbf{s}\|_1 \geq -\mathbf{c}^\top \mathbf{t} / \|\mathbf{t}\|_1$ .*

*Proof.* Suppose on the contrary that  $-\mathbf{c}^\top \mathbf{s} / \|\mathbf{s}\|_1 < -\mathbf{c}^\top \mathbf{t} / \|\mathbf{t}\|_1$ . First observe that  $\alpha \mathbf{s} + \beta \mathbf{t}$  is an applicable augmentation at  $\mathbf{x}_k$ . Moreover, we have

$$\begin{aligned} -\mathbf{c}^\top (\alpha \mathbf{s} + \beta \mathbf{t}) &= \alpha \|\mathbf{s}\|_1 \frac{-\mathbf{c}^\top \mathbf{s}}{\|\mathbf{s}\|_1} + \beta \|\mathbf{t}\|_1 \frac{-\mathbf{c}^\top \mathbf{t}}{\|\mathbf{t}\|_1} \\ &> \alpha \|\mathbf{s}\|_1 \frac{-\mathbf{c}^\top \mathbf{s}}{\|\mathbf{s}\|_1} + \beta \|\mathbf{t}\|_1 \frac{-\mathbf{c}^\top \mathbf{s}}{\|\mathbf{s}\|_1} \\ &= (\alpha \|\mathbf{s}\|_1 + \beta \|\mathbf{t}\|_1) \frac{-\mathbf{c}^\top \mathbf{s}}{\|\mathbf{s}\|_1}, \\ &\geq (\|\alpha \mathbf{s} + \beta \mathbf{t}\|_1) \frac{-\mathbf{c}^\top \mathbf{s}}{\|\mathbf{s}\|_1}, \end{aligned}$$

and therefore

$$\frac{-\mathbf{c}^\top (\alpha \mathbf{s} + \beta \mathbf{t})}{\|\alpha \mathbf{s} + \beta \mathbf{t}\|_1} > \frac{-\mathbf{c}^\top \mathbf{s}}{\|\mathbf{s}\|_1}.$$

This contradicts the fact that  $\mathbf{s}$  was a steepest descent direction for  $\mathbf{x}_k$ .  $\square$

Lemma 5 states that the steepness of steepest-descent augmentations never increases.

LEMMA 6. *Let  $\mathbf{x}_k$  be a feasible solution and let  $\alpha_1 \mathbf{z}_1, \dots, \alpha_j \mathbf{z}_j$  be the following steepest-descent augmentations applied to  $\mathbf{x}_k$ . If  $\mathbf{z}_1$  and  $\mathbf{z}_j$  do not have the same sign-pattern from  $\{\leq 0, \geq 0\}^n$ , then we have  $-\mathbf{c}^\top \mathbf{z}_1 / \|\mathbf{z}_1\|_1 > -\mathbf{c}^\top \mathbf{z}_j / \|\mathbf{z}_j\|_1$ .*

*Proof.* Assume on the contrary that  $-\mathbf{c}^\top \mathbf{z}_1 / \|\mathbf{z}_1\|_1 \leq -\mathbf{c}^\top \mathbf{z}_j / \|\mathbf{z}_j\|_1$ . By monotonicity, Lemma 5, we must have  $\frac{-\mathbf{c}^\top \mathbf{z}_1}{\|\mathbf{z}_1\|_1} = \frac{-\mathbf{c}^\top \mathbf{z}_2}{\|\mathbf{z}_2\|_1} = \dots = \frac{-\mathbf{c}^\top \mathbf{z}_j}{\|\mathbf{z}_j\|_1}$ . We conclude that

$$\begin{aligned} -\mathbf{c}^\top \left( \sum_{i=1}^j \alpha_i \mathbf{z}_i \right) &= \sum_{i=1}^j \alpha_i \|\mathbf{z}_i\|_1 \frac{-\mathbf{c}^\top \mathbf{z}_i}{\|\mathbf{z}_i\|_1} \\ &= \sum_{i=1}^j \alpha_i \|\mathbf{z}_i\|_1 \frac{-\mathbf{c}^\top \mathbf{z}_1}{\|\mathbf{z}_1\|_1} \\ &= \left( \sum_{i=1}^j \alpha_i \|\mathbf{z}_i\|_1 \right) \frac{-\mathbf{c}^\top \mathbf{z}_1}{\|\mathbf{z}_1\|_1} \\ &> \left\| \sum_{i=1}^j \alpha_i \mathbf{z}_i \right\|_1 \frac{-\mathbf{c}^\top \mathbf{z}_1}{\|\mathbf{z}_1\|_1}, \end{aligned}$$

since  $\mathbf{z}_1$  and  $\mathbf{z}_j$  do not have the same sign-pattern. This implies that

$$\frac{-\mathbf{c}^\top \left( \sum_{i=1}^j \alpha_i \mathbf{z}_i \right)}{\left\| \sum_{i=1}^j \alpha_i \mathbf{z}_i \right\|_1} > \frac{-\mathbf{c}^\top \mathbf{z}_1}{\|\mathbf{z}_1\|_1}.$$

As by assumption  $\sum_{i=1}^j \alpha_i \mathbf{z}_i$  is an applicable augmentation at  $\mathbf{x}_k$ . This contradicts the fact that  $\mathbf{z}_1$  was a steepest-descent augmentation for  $\mathbf{x}_k$ .  $\square$

Lemma 6 states that once the sign pattern of a steepest-descent augmentation strictly changes, the steepness must decrease. This lemma has a surprising consequence if we only apply steepest-descent directions from  $\mathcal{G}(A)$ :

**COROLLARY 7.** *For discrete steepest descent, no direction from  $\mathcal{G}(A)$  is chosen twice as an augmenting direction. Therefore, the number of steepest-descent augmentations needed to reach an optimal solution is bounded by  $|\mathcal{G}(A)|$  and thus is independent of  $\mathbf{b}$ ,  $\mathbf{c}$  and the initial solution  $\mathbf{z}_0$ .*

*Proof.* Let  $\mathbf{x}_k$  be a feasible solution and let  $\alpha_1 \mathbf{z}_1, \dots, \alpha_j \mathbf{z}_j$ , with  $\mathbf{z}_1, \dots, \mathbf{z}_j \in \mathcal{G}(A)$ , be the following steepest-descent augmentations applied to  $\mathbf{x}_k$  with  $\alpha_i$  chosen maximally in each augmentation. Moreover assume that  $\mathbf{z}_j = \mathbf{z}_1$ . Then, by Lemma 6, all intermediate augmentations must have the same sign-pattern as  $\mathbf{z}_1$ , as otherwise the steepness of the augmentations would have dropped. As all vectors  $\alpha_1 \mathbf{z}_1, \dots, \alpha_j \mathbf{z}_j$  have the same sign-pattern, the components of  $\mathbf{x}_k + \alpha_1 \mathbf{z}_1 + \alpha_j \mathbf{z}_j$  lie between the components of  $\mathbf{x}_k$  and  $\mathbf{x}_k + \sum_{i=1}^j \alpha_i \mathbf{z}_i$  and therefore also between  $\mathbf{0}$  and  $\mathbf{u}$ . Thus,  $\mathbf{x}_k + \alpha_1 \mathbf{z}_1 + \alpha_j \mathbf{z}_j = \mathbf{x}_k + (\alpha_1 + \alpha_j) \mathbf{z}_1$  is a feasible solution. This contradicts the fact that  $\alpha_1$  was chosen maximally.  $\square$

**2.2. Proof of Lemma 2 and Theorem 2** Let us assume that  $\mathbf{x}_k$  is a non-optimal feasible solution, and let  $\mathbf{x}_{\min}$  be an optimal solution to the LP. Then there exists a (sign-compatible) representation

$$\mathbf{x}_{\min} - \mathbf{x}_k = \sum \alpha_i \mathbf{g}_i,$$

with  $\alpha_i > 0$  and with  $\alpha_i \mathbf{g}_i \sqsubseteq \mathbf{x}_{\min} - \mathbf{x}_k$ . Moreover, due to Carathéodory theorem, at most  $n$  summands are needed in such a representation. Again, sign-compatibility of the representation  $\mathbf{x}_{\min} - \mathbf{x}_k = \sum \alpha_i \mathbf{g}_i$  implies that for all  $i$  the vectors  $\mathbf{x}_k + \alpha_i \mathbf{g}_i$  and  $\mathbf{x}_{\min} - \alpha_i \mathbf{g}_i$  are all feasible solutions.

Let us now analyze what happens for each choice of augmentation rule:

**2.2.1. Proof of Lemma 2 for Discrete deepest descent.** We observe that

$$0 > \mathbf{c}^\top (\mathbf{x}_{\min} - \mathbf{x}_k) = \mathbf{c}^\top \sum \alpha_i \mathbf{g}_i = \sum \alpha_i \mathbf{c}^\top \mathbf{g}_i \geq -n\Delta$$

where  $\Delta > 0$  is the largest value of  $-\mathbf{c}^\top \mathbf{z}$  over all  $\mathbf{z} \in \mathcal{C}(A)$  and  $\alpha > 0$  for which  $\mathbf{x}_k + \alpha \mathbf{z}$  is feasible. Rewriting this, we get

$$\Delta \geq \frac{\mathbf{c}^\top (\mathbf{x}_k - \mathbf{x}_{\min})}{n}.$$

Now let  $\alpha \mathbf{z}$  be the discrete deepest-descent augmentation applied to  $\mathbf{x}_k$ , leading to  $\mathbf{x}_{k+1} := \mathbf{x}_k + \alpha \mathbf{z}$ . Then we get  $\Delta = -\alpha \mathbf{c}^\top \mathbf{z}$  and

$$\mathbf{c}^\top(\mathbf{x}_k - \mathbf{x}_{k+1}) = -\alpha \mathbf{c}^\top \mathbf{z} = \Delta \geq \frac{\mathbf{c}^\top(\mathbf{x}_k - \mathbf{x}_{\min})}{n}.$$

Thus, we have a factor of  $\beta = 1/n$  of objective-function value decrease at each augmentation. Applying Lemma 3 with  $\epsilon = 1/\delta$  then yields a solution  $\bar{\mathbf{x}}$  with  $|\mathbf{c}^\top(\bar{\mathbf{x}} - \mathbf{x}_{\min})| < 1/\delta$  within  $2n \log(\delta \mathbf{c}^\top(\mathbf{x}_0 - \mathbf{x}_{\min}))$  many augmentations. Due to the definition of  $\delta$  as the least common multiple of all subdeterminants of  $A$ , any vertex with an objective value of at most  $\mathbf{c}^\top \bar{\mathbf{x}}$  must be optimal. As explained right before the statement of Theorem 1, such a vertex can be found from  $\bar{\mathbf{x}}$  in at most  $n$  additional augmentations. Finally, note that once the discrete deepest-descent augmentation makes progress in objective value less than  $\epsilon/(2n-2)$ , we have

$$|\mathbf{c}^\top(\mathbf{x}_k - \mathbf{x}_{\min})| = \sum |\alpha_i \mathbf{c}^\top \mathbf{g}_i| < (2n-2) \cdot \epsilon/(2n-2) = \epsilon.$$

Hence, we can decide effectively when we should stop making discrete deepest-descent augmentations and should rather find a nearby vertex.

**2.2.2. Proof of part (a) of Theorem 2: Discrete Dantzig descent.** In order to avoid zig-zagging (see example in Section 4 of [19] and the details of how to avoid this), we must augment to a vertex with lower objective-function value after each discrete Dantzig-descent augmentation. For this, we need at most  $n$  discrete Dantzig-descent augmentations within smaller and smaller faces of  $P$ . Let  $\mathbf{x}_k$  be a vertex of the given polyhedron. Again we observe that

$$0 > \mathbf{c}^\top(\mathbf{x}_{\min} - \mathbf{x}_k) = \mathbf{c}^\top \sum \alpha_i \mathbf{g}_i = \sum \alpha_i \mathbf{c}^\top \mathbf{g}_i \geq -\Delta_0 \sum \alpha_i \geq -n\Delta_0 \alpha_{\max},$$

where  $\Delta_0 > 0$  is the greatest value of  $-\mathbf{c}^\top \mathbf{z}$  over all  $\mathbf{z} \in \mathcal{C}(A)$  for which  $\mathbf{x}_k + \lambda \mathbf{z}$  is feasible for sufficiently small  $\lambda > 0$  and where  $\alpha_{\max} = \max \{\alpha_i\}$ . Rewriting this, we get

$$\Delta_0 \geq \frac{\mathbf{c}^\top(\mathbf{x}_k - \mathbf{x}_{\min})}{n\alpha_{\max}}.$$

Now, let  $\alpha \mathbf{z}$  be the discrete Dantzig-descent augmentation applied to  $\mathbf{x}_k$ , leading to  $\mathbf{x}_{k+1} := \mathbf{x}_k + \alpha \mathbf{z}$ . As  $\mathbf{x}_k$  is a vertex and  $\mathbf{z}$  an edge direction, we have  $\mathbf{x}_k, \mathbf{z} \in \frac{1}{\delta} \mathbb{Z}^n$ . As  $\mathbf{x}_{k+1}$  lies on the intersection of the half-line  $\{\mathbf{x} = \mathbf{x}_k + \lambda \mathbf{z} : \lambda \geq 0\}$  with some facet of the polyhedron, we get that  $\mathbf{x}_{k+1} \in \frac{1}{\delta^2} \mathbb{Z}^n$ . Consequently,  $\alpha \geq 1/\delta$ .

Thus, we get

$$\mathbf{c}^\top(\mathbf{x}_k - \mathbf{x}_{k+1}) = -\alpha \mathbf{c}^\top \mathbf{z} = \alpha \Delta_0 \geq \frac{1}{\delta} \Delta_0 \geq \frac{\mathbf{c}^\top(\mathbf{x}_k - \mathbf{x}_{\min})}{n\delta\alpha_{\max}} \geq \frac{\mathbf{c}^\top(\mathbf{x}_0 - \mathbf{x}_{\min})}{n\delta\gamma},$$

where  $\gamma$  is the maximum entry in any feasible solution (or, equivalently, in any vertex). Thus, we have a factor of  $\beta = 1/(n\delta\gamma)$  of objective-function decrease at each augmentation. Applying Lemma 3 with  $\epsilon = 1/\delta$  then yields the desired bound on the number of augmentations  $2n^2\delta\gamma \log(\delta \mathbf{c}^\top(\mathbf{x}_0 - \mathbf{x}_{\min}))$  to reach a vertex  $\bar{\mathbf{x}}$  with  $|\mathbf{c}^\top(\bar{\mathbf{x}} - \mathbf{x}_{\min})| < 1/\delta$ . This vertex must be optimal.

**2.2.3. Proof of part (b) of Theorem 2: Discrete steepest descent.** The proof here follows exactly the same lines of the proof to Theorem 1, part (c).

**2.3. Proof of Theorem 3.** If we keep  $A$  and  $B$  fixed and let  $N$  vary, the binary encoding lengths of the Graver bases of  $[A, B]^{(N)}$  are bounded by a polynomial in  $N$ , see [14]. More precisely, there is a constant  $g(A, B)$ , the so-called *Graver complexity* of  $A$  and  $B$  (see [22, 30, 3]), given by

$$\max \{ \|\mathbf{g}\|_1 : \mathbf{g} \in \mathcal{G}(BG(A)) \},$$

such that

$$|\mathcal{G}([A, B]^{(N)})| \leq \binom{N}{g(A, B)} |\mathcal{G}([A, B]^{(g(A, B))})| \in O(N^{g(A, B)}).$$

This means that we can find a steepest-descent direction in  $\mathcal{G}([A, B]^{(N)})$  in time polynomial in  $N$  and by Theorems 1 and 2, parts (c), the number of steepest-descent augmentations to pass from any feasible solution  $\mathbf{x}_0$  to an optimal solution is bounded polynomially in  $N$ . As the input of an  $N$ -fold LP/ILP contains  $\Theta(N)$  integer numbers (to encode  $\mathbf{b}$  and  $\mathbf{c}$ ), we can augment  $\mathbf{x}_0$  to optimality in strongly polynomial-time. It remains for us to show how we can find such a feasible solution  $\mathbf{x}_0$  in strongly polynomial-time. Note that by a shift of coordinates, we may assume without loss of generality that  $\mathbf{l} = \mathbf{0}$ .

To find such a feasible solution  $\mathbf{x}_0$ , consider the extended  $N$ -fold LP/ILP with problem matrix

$$\begin{pmatrix} B & O & O & I_{d_B} & -I_{d_B} & B & O & O & I_{d_B} & -I_{d_B} & \cdots & B & O & O & I_{d_B} & -I_{d_B} \\ A & I_{d_A} & -I_{d_A} & O & O & & & & & & & & & & & \\ & & & & & A & I_{d_A} & -I_{d_A} & O & O & & & & & & \\ & & & & & & & & & & \ddots & & & & & \\ & & & & & & & & & & & A & I_{d_A} & -I_{d_A} & O & O \end{pmatrix}.$$

This is an  $N$ -fold matrix composed using the matrices  $\bar{A} = (A \ I_{d_A} \ -I_{d_A} \ O \ O)$  and  $\bar{B} = (B \ O \ O \ I_{d_B} \ -I_{d_B})$ . As the right-hand side of our LP/ILP, we choose the same right-hand side vector  $\mathbf{b}$ , and as objective vector we use a 0/1-vector with zeros in the original components and with ones in the auxiliary components. All variables get lower bounds of 0, and the original variables get upper bounds specified by  $\mathbf{u}$ . Due to the special form of the matrix, we can immediately write down a feasible solution to this problem. (Simply assign zeros to the original variables and set the auxiliary slack variables according to the positive and negative parts of the right-hand side values.)

Optimizing this special linear objective function can now be done in strongly polynomial-time, since  $\bar{A}$  and  $\bar{B}$  are constant the running time of this auxiliary  $N$ -fold LP/ILP is bounded polynomially in  $N$ , but does not depend on the right-hand side or the objective vector. If the optimal value of this auxiliary LP/ILP is 0, a feasible solution to our original problem has been found (simply drop the auxiliary components). If the optimal value is positive, our original problem is infeasible.

□

**3. Concluding remarks** Through the notion of Graver basis, we have obtained extensions of classical results of Bland, Edmonds, and Karp. Our new version applies now to the case of integer-linear programs. As a consequence we have also derived the first-ever strongly polynomial-time algorithm for  $N$ -fold integer-linear optimization. Our new results also show that Kitahara-Mizuno-style bounds [24] hold in larger generality to include augmentations that go through the interior of the polytope and are not restricted to edges. Theorem 1 is in fact an ILP extension of those bounds too.

There are at least three interesting directions for improvement and further research. First, we remark the numbers  $n$  and  $2n - 2$  from Carathéodory's and Sebö's theorems we used in many of the arguments can be further improved to be  $n - \text{rank}(A)$  and  $2(n - \text{rank}(A)) - 2$  respectively because the arguments can be modified to use the dimension of the kernel of  $A$ . Such small improvement slightly strengthens several of the results of the paper, but we leave the details to the reader.



Second, as was demonstrated in [10, 11] the estimation of number of augmentations contribute to the estimation of the diameters of polytopes. Third, it would be interesting to improve our strongly polynomial algorithm for  $N$ -fold matrices. Currently, the number of augmentations does not depend on  $b$  and  $u$ , but is a polynomial of degree  $O(g(A,B))$  with  $g(A,B)$  the Graver complexity (see [22, 30, 3]) Thus the degree depends on the fixed matrices  $A$  and  $B$ . It would be desirable to arrive to a lower exponent algorithm, like the one of [20] which is not strongly polynomial yet, the number of augmentations is linear in the binary encoding of  $b$  and  $u$ , and it is only cubic in  $N$ ,  $O(N^3)$ .

**Acknowledgments** We are grateful for the comments from Shmuel Onn. The first and the third author are grateful for the hospitality received by the Isaac Newton Institute for Mathematical Sciences at the University of Cambridge, UK during the program “Polynomial Optimization”. The first author is thankful to the Technical University of Munich for the support and hospitality received during his visit in Spring 2013 as a John von Neumann professor and to the NSF for the support received through grant DMS-0914107. J. Lee was partially supported by NSF grant CMMI-1160915 and ONR grant N00014-14-1-0315.

## References

- [1] Ahuja, R.K., T.L. Magnanti, J.B. Orlin. 1993. *Network flows*. Prentice Hall Inc., Englewood Cliffs, NJ. Theory, algorithms, and applications.
- [2] Bárász, M., S. Vempala. 2010. A new approach to strongly polynomial linear programming. <http://www.cc.gatech.edu/~vempala/papers/affine.pdf>.
- [3] Bernstein, Y., S. Onn. 2009. The Graver complexity of integer programming. *Annals of Combinatorics* **13** 289–296.
- [4] Bixby, R. E., O. M.-C. Marcotte, L. E. Trotter, Jr. 1987. Packing and covering with integral feasible flows in integral supply-demand networks. *Math. Programming* **39**(3) 231–239.
- [5] Bland, R. G. 1974. *Complementary orthogonal subspaces of  $n$ -dimensional Euclidean space and orientability of matroids*. Thesis (Ph.D.)–Cornell University.
- [6] Bland, R. G. 1976. On the generality of network flow theory. Presented at the ORSA/TIMS Joint National Meeting, Miami, Florida, Fall 1976.
- [7] Bland, R. G., D. L. Jensen. 1992. On the computational behavior of a polynomial-time network flow algorithm. *Math. Programming* **54**(1, Ser. A) 1–39.
- [8] Bland, R.G., J. Edmonds. 1982. Fast primal algorithms for totally unimodular linear programming. Presented at XIth International Symposium on Mathematical Programming, Bonn, West Germany (August 1982).
- [9] Bonifas, N., M. Di Summa, F. Eisenbrand, N. Hähnle, M. Niemeier. 2014. On sub-determinants and the diameter of polyhedra. *Discrete and Computational Geometry* **52**(1) 102–115.
- [10] Borgwardt, S., E. Finhold, R. Hemmecke. 2014. On the circuit diameter of dual transportation polyhedra. To appear in *SIAM Discrete Math*. Available as Arxiv:1405.3184.
- [11] Borgwardt, S., De Loera J.A., E. Finhold. 2014. Edges vs circuits: a hierarchy of diameters in polyhedra. Available as ArXiv:1409.7638.
- [12] Cardoso, D., J. Clímaco. 1992. The generalized simplex method. *Operations Research Letters*. **12** 337–348.
- [13] De Loera, J. A., R. Hemmecke, M. Köppe. 2013. *Algebraic and geometric ideas in the theory of discrete optimization, MOS-SIAM Series on Optimization*, vol. 14. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA.
- [14] De Loera, J. A., R. Hemmecke, S. Onn, R. Weismantel. 2008.  $N$ -fold integer programming. *Discrete Optimization* **5**(2) 231–241. In Memory of George B. Dantzig.

- [15] Edmonds, J., R. M. Karp. 1972. Theoretical improvements in algorithmic efficiency for network flow problems. *J. ACM* **19**(2) 248–264.
- [16] Frank, A., E Tardos. 1987. An application of simultaneous diophantine approximation in combinatorial optimization. *Combinatorica* **7**(1) 49–65.
- [17] Fulkerson, D. R. 1968. Networks, frames, blocking systems. *Mathematics of the Decision Sciences, Part 1 (Seminar, Stanford, Calif., 1967)*. Amer. Math. Soc., Providence, R.I., 303–334.
- [18] Graver, J. E. 1975. On the foundation of linear and integer programming I. *Mathematical Programming* **9** 207–226.
- [19] Hemmecke, R. 2003. On the positive sum property and the computation of Graver test sets. *Math. Programming, Series B* **96** 247–269.
- [20] Hemmecke, R., S. Onn, L. Romanchuk. 2013.  $N$ -fold integer programming in cubic time. *Mathematical Programming* **137** 325–341.
- [21] Hemmecke, R., S. Onn, R. Weismantel. 2011. A polynomial oracle-time algorithm for convex integer minimization. *Mathematical Programming* **126** 97–117.
- [22] Hoşten, S., S. Sullivant. 2007. A finiteness theorem for Markov bases of hierarchical models. *J. Combin. Theory Ser. A* **114**(2) 311–321.
- [23] Kern, W., U. Faigle, B. Peis. 2012. Unpublished manuscript.
- [24] Kitahara, T., S. Mizuno. 2013. A bound for the number of different basic solutions generated by the simplex method. *Math. Program.* **137**(1-2, Ser. A) 579–586.
- [25] Lee, J. 1986. Subspaces with well-scaled frames. Ph.D. thesis, Cornell University.
- [26] Lee, J. 1989. Subspaces with well-scaled frames. *Linear Algebra and its Applications* **114-115** 21–56. Special Issue Dedicated to Alan J. Hoffman.
- [27] Onn, S. 2010. *Non-Linear Discrete Optimization*. Zurich Lectures in Advanced Mathematics, European Mathematical Society.
- [28] Orlin, J.B. 1997. A polynomial time primal network simplex algorithm for minimum cost flows. *Math. Programming* **78**(2, Ser. B) 109–129. Network optimization: algorithms and applications (San Miniato, 1993).
- [29] Rockafellar, R. T. 1969. The elementary vectors of a subspace of  $R^N$ . *Combinatorial Mathematics and its Applications (Proc. Conf., Univ. North Carolina, Chapel Hill, N.C., 1967)*. Univ. North Carolina Press, Chapel Hill, N.C., 104–127.
- [30] Santos, F., B. Sturmfels. 2003. Higher Lawrence configurations. *J. Comb. Theory Ser. A* **10** 151–164.
- [31] Scarf, H. E. 1997. Test sets for integer programs. *Mathematical Programming, Series A* **79** 355–368.
- [32] Schulz, A.S, R. Weismantel. 1999. An oracle-polynomial time augmentation algorithm for integer programming. *Proceedings of the 10th Annual ACM-SIAM Symposium on Discrete Algorithms*. 967–968.
- [33] Schulz, A.S, R. Weismantel. 2002. The complexity of generic primal algorithms for solving general integer program. *Mathematics of Operations Research* **27** 681 – 692.
- [34] Sebö, A. 1990. Hilbert bases, Caratheodory’s theorem and combinatorial optimization. Ravi Kannan, William R. Pulleyblank, eds., *IPCO*. University of Waterloo Press, 431–455.
- [35] Seymour, P. D. 1980. Decomposition of regular matroids. *J. Combin. Theory Ser. B* **28**(3) 305–359.
- [36] Tardos, É. 1986. A strongly polynomial algorithm to solve combinatorial linear programs. *Operations Research* **34**(2) pp. 250–256.
- [37] Ye, Y. 2011. The simplex and policy-iteration methods are strongly polynomial for the markov decision problem with a fixed discount rate. *Math. Oper. Res.* **36**(4) 593–603.