DM898 Parameterized Algorithms Lars Rohwedder
Fall 2025 rohwedder@sdu.dk

Eisenbrand-Weismantel Algorithm

Lecture Notes

The results described here are based on Eisenbrand and Weismantel's algorithm from [1] that solves integer linear programs in FPT time in the number of constraints and the largest coefficient in a constraint. The emphasis of these notes is on simplicity and self-containedness, for which we accept slightly worse bounds and running times (although the difference is marginal). If you notice errors, please send them by email.

Overview

We consider integer linear programs in a form with equality constraints and encoded using matrices and vectors. This will be more convenient to work with and other types of constraints can be transformed into this by adding slack variables.

Given coefficients of the objective $c \in \mathbb{Z}^n$, a matrix $A \in \{-\Delta, \dots, \Delta\}^{m \times n}$, right-hand side $b \in \mathbb{Z}^m$ and lower and upper bounds $\ell_i \in \mathbb{Z} \cup \{-\infty\}, u_i \in \mathbb{Z} \cup \{\infty\}$, $i \in \{1, 2, \dots, n\}$, our task is to solve

$$\begin{aligned} &\min c^{\mathsf{T}} x \\ &Ax = b \\ &\ell_i \leq x_i \leq u_i, \quad x_i \in \mathbb{Z} \qquad\qquad \text{for all } i = 1, 2, \dots, n \end{aligned}$$

Each row of A encodes the coefficients of a constraint and the corresponding entry of b describes the constant term in the constraint. Each column of A corresponds to one variable.

Example. Consider a variant of the Knapsack problem with three items of profits $4, 5, 1$, weights $2, 3, 2$, and capacity 6, where in addition we are required to pick at most two items, then this can be modelled as the following integer linear program:

$$\begin{aligned} \min -4x_1 - 5x_2 - x_3& \\ 2x_1 + 3x_2 + 2x_3 + s_1 &= 6 \\ x_1 + x_2 + x_3 + s_2 &= 2 \\ x_1, x_2, x_3 &\in \{0, 1\} \\ s_1, s_2 &\in \mathbb{Z}_{\geq 0} \end{aligned}$$

Here the s_1 and s_2 are slack variables that transform the $\leq$ constraint into an equality constraint. In matrix-vector encoding we have the values:

$$A = \begin{pmatrix} 2 & 3 & 2 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 \end{pmatrix} \quad b = \begin{pmatrix} 6 \\ 2 \end{pmatrix} \quad c = \begin{pmatrix} 4 \\ 5 \\ 1 \\ 0 \\ 0 \end{pmatrix} \quad \ell = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad u = \begin{pmatrix} 1 \\ 1 \\ 1 \\ \infty \\ \infty \end{pmatrix}$$

LP relaxation. The *LP relaxation* of the ILP above is obtained by omitting the integrality constraints: The resulting linear program can be solved in polynomial time, but we are interested in an integer solution. As a first step, we can get a solution where not too many variables are non-integer.

Lemma 1. *Let $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$, and $\ell_i \in \mathbb{R} \cup \{-\infty\}, u_i \in \mathbb{R} \cup \{\infty\}$ for each $i \in \{1, 2, \dots, n\}$. If the linear program*

$$\begin{aligned} \min\ & c^\mathsf{T} x \\ Ax &= b \\ \ell_i \le x_i \le u_i, \quad & x_i \in \mathbb{R} \qquad\qquad \text{for all } i = 1, 2, \dots, n \end{aligned}$$

is feasible and bounded, then it has an optimal solution x^* with

$$|\{i \in \{1, \dots, n\} : \ell_i < x_i^* < u_i\}| \le m.$$

Such a solution can be found in polynomial time.

Note that with integer bounds, the other at least $n - m$ variables must be integer. Eisenbrand and Weismantel's algorithm crucially relies on the following theorem.

Theorem 2 (Proximity theorem)**.** *Assume that an ILP of the form above is feasible and bounded. Let x^* be a optimal solution to the LP relaxation with at most m non-integral variables. Then there exists some optimal integer solution x with*

$$\|x - x^*\|_1 = \sum_{i=1}^{n} |x_i - x_i^*| \le (2m^2\Delta + 1)^m + m =: \text{prox} \ .$$

In particular,

$$\|x - \lfloor x^* \rfloor\|_1 = \sum_{i=1}^{n} |x_i - \lfloor x_i^* \rfloor| \le (2m^2\Delta + 1)^m + 2m =: \text{prox}'.$$

This bound might be surprising, because it is independent of n. In fact, if many of the variables were not already integers then there would be no hope to achieve something close to this. Consider the example

$$\begin{aligned} x_1 + \dots + x_{2n} &= n \\ x_1, \dots, x_{2n} &\in \{0, 1\} \ . \end{aligned}$$

One solution to the LP relaxation is $x^* = (1/2, \dots, 1/2)^\mathsf{T}$. However, any integer solution would need to have n many 1s and n many 0s, so it would be at distance n in ℓ_1-norm to x^*. This does not form a counter-example to the theorem above, because the theorem requires that only few variables are non-integer, which comes "for free" because of the previous lemma.

Before we prove the proximity theorem and the lemma, we will look at how we can exploit it in a dynamic program.

Dynamic Program

We proceed similar to the Knapsack dynamic program based on "dominance", see Figure 1. The dynamic program computes in each iteration i a set of tripels (C, B, k) with the meaning that C is the optimal value that can be achieved restricted to variables $1, \dots, i$ and a right-hand side B and a distance of k to $\lfloor x^* \rfloor$. More formally:

- compute optimum x^* to LP relaxation with $\leq m$ non-integral variables
- $\mathcal{T} \leftarrow \{(0,0,0)\}$ *// set of undominated (objective, right-hand side, distance-to-$\lfloor x^* \rfloor$) triples obtainable*
- for $i \in \{1, 2, \ldots, n\}$
  - $\mathcal{T}' \leftarrow \mathcal{T}$
  - $\mathcal{T} \leftarrow \emptyset$
  - for x_i in $\{\max\{\ell_i, \lfloor x_i^* \rfloor - \text{prox'}\}, \ldots, \min\{u_i, \lfloor x_i^* \rfloor + \text{prox'}\}\}$
    - $\mathcal{T} \leftarrow \mathcal{T} \cup \{(C + c_i x_i, B + A_i x_i, k + |\lfloor x_i^* \rfloor - x_i|) \mid (C, B, k) \in \mathcal{T}'\}$ *// A_i is the ith column of A*
  - for $(C, B, k), (C', B', k') \in \mathcal{T}$ with $C < C', B = B', k = k'$
    - $\mathcal{T} \leftarrow \mathcal{T} \setminus (C', B', k')$
  - for $(C, B, k) \in \mathcal{T}$ with $k >$ prox'
    - $\mathcal{T} \leftarrow \mathcal{T} \setminus (C, B, k)$
- return $\min\{C \mid (C, B, k) \in \mathcal{T}, B = b\}$

Figure 1: Algorithm ILP(n, A, b, c, ℓ, u)

Lemma 3. *After iteration i of the dynamic program, we have for each $B \in \mathbb{Z}^m$ and $k \in \mathbb{Z}_{\geq 0}$ that $\mathcal{T}$ contains (C, B, k) if and only if $k \leq \text{prox}'$ and*

$$\min \left\{ \sum_{j=1}^{i} c_j x_j \mid x_i \in \{\ell_i, \ldots, u_i\} \forall j \in \{1, 2 \ldots, i\}, \sum_{j=1}^{i} A_j x_j = B, \sum_{j=1}^{i} |x_j - \lfloor x_j^* \rfloor| = k \right\} \quad (1)$$

exists and is equal to C.

Proof. We prove this by induction over i. For $i = 1$ this is true because we enumerate all elements over which the minimum (1) is taken (all choices of of x_1) and then remove all suboptimal tripels (a removal only happens if $A_1 = 0$).

Assume now that $i > 1$. We first argue that we only have tripels (C, B, k) that correspond to solutions $x_1, \ldots, x_i$ with $\sum_{j=1}^{i} A_j x_j = B, \sum_{j=1}^{i} |x_j - \lfloor x_j^* \rfloor| = k$. This is because we derive each tripel from another tripel (C', B', k') with $C' = C - c_i x_i$, $B' = B - A_i x_i$, and $k' = k - |x_i - \lfloor x_i^* \rfloor|$. By induction, (C', B', k') corresponds to a solution $x'_1, \ldots, x'_{i-1}$ with $\sum_{j=1}^{i-1} c_j x'_j = C', \sum_{j=1}^{i-1} A_j x'_j = B'$, and $\sum_{j=1}^{i-1} |x'_j - \lfloor x_j^* \rfloor| = k'$. Thus, (C, B, k) corresponds to $x'_1, \ldots, x'_{i-1}, x_i$. Clearly, all tripels with $k > \text{prox}'$ are also removed.

It remains to argue that for each $B \in \mathbb{Z}^m$ and $k \leq \text{prox}'$ we add the optimal tripel, assuming (1) exists, which then also implies that all non-optimal tripels do not remain in the tripel set. Let x be an optimal solution to (1) and let $C = \sum_{j=1}^{i} c_j x_j$. Define $k' = k - |x_i - \lfloor x_i^* \rfloor|$, $B' = B - A_i x_i$, and $C' = C - c_i x_i$. By induction, after iteration $i - 1$ the tripel set must have contained (C'', B', k') for the optimal C'' corresponding to a solution $x'_1, \ldots, x'_{i-1}$. This is because the minimum exists with $x_1, \ldots, x_{i-1}$ being one of the choice. In particular, $C' \geq C''$. Thus, in iteration i we add $(C'' + c_i x_i, B' + A_i x_i, k' + |x_i - \lfloor x_i^* \rfloor|)$. We have that $C'' + c_i x_i \leq C' + c_i x_i = C$ by the previous inequalities. Furthermore, we have $C'' + c_i x_i \geq C$ because otherwise $y_1, \ldots, y_{i-1}, x_i$ would have a smaller objective than x, contradicting the optimality of x. Thus, (C, B, k) is indeed contained in $\mathcal{T}$ after iteration i. ☐

It follows that the algorithm is correct. We now bound its running time.

Lemma 4. *The dynamic program runs in time $n \cdot (m\Delta)^{O(m^2)}$.*

Proof. It suffices to bound the size of $\mathcal{T}$ by $(m\Delta)^{O(m^2)}$ at the beginning of each iteration. Notice that by Lemma 3 there is at most one tripel (C, B, k) for each B, k. However, many B, k have no tripel. If there is a tripel (C, B, k), then $k \leq \text{prox}'$ and there is a solution $x_1, \dots, x_i$ with $A_1x_1 + \cdots A_ix_i = B$ and $\sum_{j=1}^{i} |x_j - \lfloor x_j^* \rfloor| = k$. Thus,

$$\|B - (\underbrace{A_1\lfloor x_1^* \rfloor + \cdots + A_i\lfloor x_i^* \rfloor}_{:=B^*})\|_\infty = \|A_1x_1 + \cdots + A_ix_i - (A_1\lfloor x_1^* \rfloor + \cdots + A_i\lfloor x_i^* \rfloor)\|_\infty$$

$$\leq \sum_{i=1}^{n} \|A_i\|_\infty (x_i - \lfloor x_i \rfloor) \leq \Delta \sum_{i=1}^{n} (x_i - \lfloor x_i \rfloor) \leq k\Delta \ .$$

There only exist $(2k\Delta + 1)^m \leq (2\text{prox'}\Delta + 1)^m$ integer vectors B with $\|B - B^*\|_\infty \leq k\Delta$. Thus, at the beginning of an iteration $|\mathcal{T}| \leq (2\text{prox'}\Delta + 1)^m \cdot \text{prox'} \leq (m\Delta)^{O(m^2)}$ $\square$

Few non-tight variables

In this section we prove Lemma 1. Let x be any optimal solution to the linear program, which can be found in polynomial time using known algorithms. Denote by $i_1 < \cdots < i_h$ the indices of variables with $\ell_{i_j} < x_{i_j} < u_{i_j}$. If $h \leq m$ then we simply return x. Otherwise, denote by $A_i \in \mathbb{R}^m$ the ith column vector of A. Since $A_{i_1}, \dots, A_{i_h}$ are more than m vectors in dimension m, they must be linearly dependent. This means there are $\lambda_1, \dots, \lambda_h \in \mathbb{R}$ (that can be computed in polynomial time using standard linear algebra), not all zero such that

$$\lambda_1 A_{i_1} + \cdots + \lambda_h A_{i_h} = 0 \ .$$

Assume without loss of generality that $\lambda_1 c_{i_1} + \cdots + \lambda_h c_{i_h} \leq 0$. Otherwise, negate all λ_j.

For each $j \in \{1, \dots, h\}$ define

$$\delta_j = \begin{cases} \frac{u_{i_j} - x_{i_j}}{\lambda_j} & \text{if } \lambda_j > 0 \\ \frac{\ell_{i_j} - x_{i_j}}{\lambda_j} & \text{if } \lambda_j < 0 \\ \infty & \text{if } \lambda_j = 0 \end{cases}$$

Intuitively, δ_j describes the largest value such that $x_{i_j} + \delta_j\lambda_j$ still satisfies the variable bounds of the i_jth variable. Note that $\delta_j > 0$ for all $j \in \{1, \dots, h\}$ and $\delta \neq \infty$ for at least one $j \in \{1, \dots, h\}$. Thus,

$$\delta := \min\{\delta_j : j \in \{1, \dots, h\}\}$$

is finite and non-negative. Consider the solution x' with

$$x'_i = \begin{cases} x_i + \delta\lambda_j & \text{if } i = i_j \\ x_i & \text{if } i \notin \{i_1, \dots, i_h\} \end{cases}$$

We will argue that x' is an optimal solution to the linear program with at least one more variable at one of its bounds. Regarding the optimality we have

$$\sum_{i=1}^{n} c_i x'_i = \sum_{i=1}^{n} c_i x_i + \sum_{j=1}^{h} c_{i_j} \delta\lambda_j \leq \sum_{i=1}^{n} c_i x_i \ .$$

Regarding feasibility, because of the choice of λ_j, we have

$$Ax' = Ax + \sum_{j=1}^{h} A_{i_j} \delta\lambda_j = Ax + \delta \sum_{j=1}^{h} \lambda_j A_{i_j} = b \ .$$

Finally, consider the variable bounds. Let $j \in \{1, \dots, h\}$. If $\lambda_j > 0$ then

$$\begin{aligned} x'_{i_j} &= x_{i_j} + \delta\lambda_j \geq x_{i_j} \geq \ell_{i_j} \text{ and} \\ x'_{i_j} &= x_{i_j} + \delta\lambda_j \leq x_{i_j} + \delta_j\lambda_j = x_{i_j} + u_{i_j} - x_{i_j} = u_{i_j} \ . \end{aligned}$$

If $\lambda_j < 0$ then

$$\begin{aligned} x'_{i_j} &= x_{i_j} + \delta\lambda_j \geq x_{i_j} + \delta_j\lambda_j = x_{i_j} + \ell_{i_j} - x_{i_j} = \ell_{i_j} \text{ and} \\ x'_{i_j} &= x_{i_j} + \delta\lambda_j \leq x_{i_j} \leq u_{i_j} \ . \end{aligned}$$

Notice that for at least one j we have that $\delta = \delta_j$. In this case, the second inequality in the first case or the first inequality in the second case would become an equality and therefore the variable x'_{i_j} is now at one of its bounds. We repeat the procedure above until $h \leq m$.

Proof of proximity theorem

We will now prove Theorem 2. Throughout this section denote by x^* an optimal solution to the LP relaxation with at most m non-integral variables. Let x be an optimal integer solution with $\|x - x^*\|_1$ minimal.

We think about the vector $x - x^*$ as the total change to move from the fractional to the integer solution. To better analyze it, let us break $x - x^*$ into smaller pieces. The first piece is the vector $r \in \mathbb{Z}^n$, which simply rounds x^* towards x in every non-integer component. Formally, define

$$r_i = \begin{cases} \lceil x_i^* \rceil - x_i^* & \text{if } x_i^* \leq x_i \\ \lfloor x_i^* \rfloor - x_i^* & \text{if } x_i^* > x_i \end{cases}$$

Thus, $x^* + r$ is an integer vector, but it may not be feasible, that is, $A(x^* + r) \neq b$ is very likely and even if it was feasible, it may not be optimal.

We split the remaining change, that is, $x - (x^* + r)$, into a sequence of changes $d^{(1)}, \dots, d^{(h)} \in \{-1, 0, 1\}^n$ that increase or decrease single variables by one: For every variable index $i \in \{1, \dots, n\}$, if $x_i^* + r_i \leq x_i$ then we add $x_i - (x_i^* + r_i)$ many vectors $d^{(j)} = (0, 0, \dots, 0, 1, 0, \dots, 0)$, which have a 1 at index i and zeroes everywhere else; if $x_i^* + r_i > x_i$ then we add $x_i^* + r_i - x_i$ many vectors $d^{(j)} = (0, 0, \dots, 0, -1, 0, \dots, 0)$, which have a -1 at index i and zeroes everywhere else.

By construction we have that

$$x = x^* + r + \sum_{j=1}^{h} d^{(j)} \ .$$

If we start with x^* and we now only add a subset of the changes $\{r, d^{(1)}, \dots, d^{(\ell)}\}$, then we will obtain a variable assignment that at least respects the variable bounds. Other than that it may still be infeasible, suboptimal or even fractional (if it does not contain r).

Lemma 5. *For every $P \subseteq \{r, d^{(1)}, d^{(2)}, \dots, d^{(\ell)}\}$ and $z = x^* + \sum_{p \in P} p$ it holds that*

$$\ell_i \leq z_i \leq u_i \quad \text{for all } i \in \{1, 2, \dots, n\} \ .$$

Proof. Let $i \in \{1, 2, \dots, n\}$. If $x_i^* \leq x_i$ then all vectors are non-negative in component i, that is, $r_i \geq 0$ and $d_i^{(j)} \geq 0$ for all $j \in \{1, 2, \dots, h\}$. Thus,

$$\ell_i \leq x_i^* \leq \underbrace{x_i^* + \sum_{p \in P} p_i}_{= z_i} \leq x_i^* + r_i + \sum_{j=1}^{h} d_i^{(j)} = x_i \leq u_i \ .$$

Conversely, if $x_i^* > x_i$ then all vectors are non-positive in component i and therefore

$$\ell_i \leq x_i = x_i^* + r_i + \sum_{j=1}^{h} d_i^{(j)} \leq \underbrace{x_i^* + \sum_{p \in P} p_i}_{=z_i} \leq x_i^* \leq u_i \ . \qquad \square$$

Later, we will show that if h is large enough, there must be a non-trivial subset of changes that maintains feasibility of all constraints. As we see in the next lemma, this forms then a contradiction to either the optimality of x^* or the fact that x is the closest integer optimal solution to x^*. This will then give a bound on h and indirectly on the proximity.

Lemma 6. *Let $\emptyset \neq P \subsetneq \{r, d^{(1)}, \dots, d^{(\ell)}\}$. Then*

$$\sum_{p \in P} Ap \neq 0 \ .$$

Proof. Assume towards contradiction that $\sum_{p \in P} Ap = 0$. Without loss of generality, assume that $r \notin P$. Otherwise, replace P by $P' = \{r, d^{(1)}, \dots, d^{(\ell)}\} \setminus P$, which also satisfies

$$\sum_{p \in P'} Ap = A(x - x^*) - \sum_{p \in P} Ap = b - b - 0 = 0 \ .$$

If the change in objective induced by P is negative, that is, $\sum_{p \in P} c^\mathsf{T} p < 0$, then $z = x^* + \sum_{p \in P} p$ would be a feasible fractional solution with

$$c^\mathsf{T} z = c^\mathsf{T}(x^* + \sum_{p \in P} p) < c^\mathsf{T} x^* \ .$$

Hence, x^* would not be optimal, a contradiction. If on the other hand $\sum_{p \in P} c^\mathsf{T} p \geq 0$, then $x' = x - \sum_{p \in P} p$ would be an integer solution and

$$c^\mathsf{T} x' = c^\mathsf{T}(x - \sum_{p \in P} p) \leq c^\mathsf{T} x \ .$$

Therefore, x' would also be an integer optimal solution and it would be closer to x^* than x because $P \neq \emptyset$ and all negated changes applied to x reduce the distance to x^*. This is a contradiction to the choice of x. $\square$

Notice that in total the changes keep the constraints valid, that is, $Ar + \sum_{j=1}^{h} Ad^{(j)} = 0$. Each of the vectors Ar, $Ad^{(1)}, \dots, Ad^{(h)}$ is integer and their components are bounded by $m\Delta$ in absolute value. We will explain this in detail later. The proximity theorem is now reduced to the following question:

> How many integer vectors with bounded entries that sum to zero can there be without any non-trivial subset already summing to zero?

In order to bound it, we will use the following result.

Theorem 7 (Steinitz Lemma)**.** *Let $v_1, \dots, v_n \in \mathbb{R}^m$ with $v_1 + \dots + v_n = 0$. Then there exists a permutation $\sigma \in \mathcal{S}_n$ such that for every $i \in \{1, \dots, n\}$ it holds that*

$$\|v_{\sigma(1)} + \dots + v_{\sigma(i)}\| \leq m \cdot \max_{i=1}^{n} \|v_i\| \ .$$

Here $\|\cdot\|$ is an arbitrary norm.

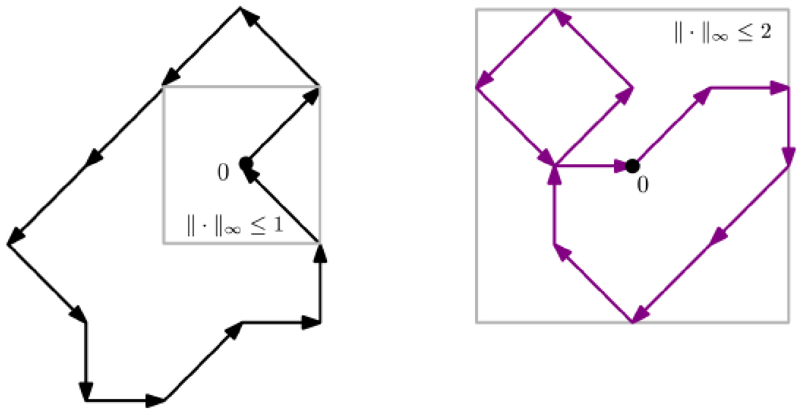

Figure 2: Steinitz Lemma. Source: [1]

It is quite easy to see that the Steinitz Lemma holds in one dimension, where the vectors are simply single reals of bounded absolute value: One can construct the permutation greedily. Choose the first real arbitrarily. Then, if the current sum is lower than zero, choose next a positive real, otherwise a negative one. Before proving it for arbitrary dimension, we will show how using it we can now conclude the proof of the proximity theorem.

Proof of proximity theorem. Define $v_1 = Ad^{(1)}, \dots, v_h = Ad^{(h)}$, and $v_{h+1} = Ar$. Then

$$v_1 + \cdots + v_{h+1} = A(x - x^*) = 0 \ .$$

Furthermore, $Ay^{(j)}$ is either a column of A or a negated column of A for each $j \in \{1, 2, \dots, h\}$. Thus, $\|v_j\|_\infty = \|Ad^{(j)}\|_\infty \leq \Delta$. Vector Ar is the sum of at most m columns of A (corresponding to the fractional variables of x^*), multiplied by a scalar between -1 and 1. Thus, $\|v_{h+1}\|_\infty = \|Ar\|_\infty \leq m\Delta$. Finally, $v_1, \dots, v_{h+1}$ are all integer vectors. For $v_1, \dots, v_h$ this is clear and for v_h it must also hold because the sum of all is zero.

We apply the Steinitz Lemma to $v_1, \dots, v_{\ell+1}$ and obtain a permutation $\sigma \in \mathcal{S}_{\ell+1}$ such that for all $i \in \{1, 2, \dots, h+1\}$ it holds that

$$\|v_{\sigma(1)} + \cdots + v_{\sigma(i)}\|_\infty \leq m^2\Delta \ .$$

Notice that the number of integer vectors with entries bounded in absolute value by $m^2\Delta$ is $(2m^2\Delta + 1)^m$. Therefore, if $h + 1 > (2m^2\Delta + 1)^m$, then there would be two distinct $i < i'$ with

$$v_{\sigma(1)} + \cdots + v_{\sigma(i)} = v_{\sigma(1)} + \cdots + v_{\sigma(i')}$$

Therefore $v_{\sigma(i+1)} + \cdots + v_{\sigma(i')} = 0$. By Lemma 6 this cannot be and hence $h + 1 \leq (2m^2\Delta + 1)^m$. We conclude

$$\|x - x^*\|_1 \leq \|r\|_1 + \sum_{i=1}^{h} \|p^{(i)}\|_1 \leq m + h \leq m + (2m^2\Delta + 1)^m \ . \qquad \square$$

Proof of Steinitz Lemma

We give an algorithmic proof by determining $\sigma(n), \sigma(n-1), \cdots, \sigma(1)$ iteratively in this order. The proof uses linear programming arguments and, in fact, heavily relies on Lemma 1. Let $U_k = \{1, 2, \dots, n\} \setminus \{\sigma(n), \sigma(n-1), \dots, \sigma(k+1)\}$ be the indices of unassigned vectors before we determine $\sigma(k)$. The initial set $U_n = \{1, 2, \dots, n\}$ contains all vector indices. Consider the linear program

$$
\begin{aligned}
\sum_{i \in U_k} (v_i)_j x_i = 0 & \qquad \text{for all } j \in \{1, \dots, m\} \\
\sum_{i \in U_k} x_k = |U_k| - m & \\
x_i \in [0, 1] & \qquad \text{for all } i \in U_k
\end{aligned}
\tag{LP$_i$}
$$

Note that LP_n, that is, the linear program for the first iteration, is feasible, because $x_1 = x_2 = \cdots = x_n = (n-m)/n$ is a solution. We will maintain throughout the algorithm the invariant that before determining $\sigma(k)$, the linear program LP_k is feasible.

Suppose that $k > m$ and we have already determined $\sigma(n), \dots, \sigma(k+1)$. Let U_k be the unassigned vector indices. Let x be a solution to LP_k, which we assume by the invariant exists. Define $x' = x \cdot (|U_k| - m - 1)/(|U_k| - m)$ to be a rescaled version of x. Then x' is a solution to

$$
\begin{aligned}
\sum_{i \in U_k} (v_i)_j x_i = 0 & \qquad \text{for all } j \in \{1, \dots, m\} \\
\sum_{i \in U_k} x_k = |U_i| - m - 1 & \\
x_i \in [0, 1] & \qquad \text{for all } i \in U_k
\end{aligned}
\tag{LP$'_k$}
$$

Because this LP is feasible and has $m+1$ constraints, by Lemma 1 it also has a solution x^* with at most $m+1$ fractional variables. Therefore there are at least $|U_k| - m - 1$ variables with value zero or one. Not all of them can be one, since otherwise $\sum_{i \in U_k} x_i > |U_k| - m - 1$. Let $i \in U_k$ with $x_i^* = 0$. We set $\sigma(k) = i$ and therefore $U_{k-1} = U_k \setminus \{i\}$. By dropping the variable x_i from LP'_k, which is safe because it is zero, the linear program becomes LP_{k-1} and $(x^*_{i'})_{i' \in U_{k-1}}$ attests its feasibility.

When $k = m$ then this construction no longer works, since $|U_k| - m - 1$ would become negative. Then only $|U_m| = m$ vectors are left and we assign them arbitrarily to $\sigma(m), \dots, \sigma(1)$.

It remains to analyze the size of each partial sum with the permutation as we constructed it. If $k \le m$ then

$$
\|v_{\sigma(1)} + \cdots + v_{\sigma(k)}\| \le \|v_{\sigma(1)}\| + \cdots + \|v_{\sigma(k)}\| \le k \cdot \max_{i=1}^{n} \|v_i\| \le m \cdot \max_{i=1}^{n} \|v_i\| \ .
$$

Now assume that $k > m$. Then $\{\sigma(1), \dots, \sigma(k)\} = U_k$. Let x be the solution to LP_k. It follows that

$$
\begin{aligned}
\|v_{\sigma(1)} + \cdots + v_{\sigma(k)}\| &= \|\sum_{i \in U_k} v_i\| \\
&= \|\sum_{i \in U_k} x_i v_i + \sum_{i \in U_k} (1 - x_i) v_i\| \\
&\le \|\sum_{i \in U_k} x_i v_i\| + \sum_{i \in U_k} (1 - x_i) \|v_i\| \\
&\le 0 + \sum_{i \in U_k} (1 - x_i) \max_{j=1}^{n} \|v_j\| \\
&= (|U_k| - \sum_{i \in U_k} x_i) \max_{j=1}^{n} \|v_j\| \\
&= m \cdot \max_{j=1}^{n} \|v_j\|.
\end{aligned}
$$

References

[1] Friedrich Eisenbrand and Robert Weismantel. Proximity results and faster algorithms for integer programming using the steinitz lemma. *ACM Trans. Algorithms*, 16(1), November 2019.