

# Proximity results and faster algorithms for Integer Programming using the Steinitz Lemma

## Summary

Friedrich Eisenbrand, Robert Weismantel  
Samuel Bismuth

March 18, 2022

## 1 Abstract and Introduction

The paper shows two algorithms to solve integer programming (IP) problems.

( $\Delta$  is an upper bound of each absolute value of an entry in  $A$ .)

- (i) Consider the standard IP in the form,

$$\max\{c^T x : Ax = b, x \geq 0, x \in \mathbb{Z}^n\} \quad (1)$$

where  $A \in \mathbb{Z}^{m \times n}$ ,  $b \in \mathbb{Z}^m$  and  $c \in \mathbb{Z}^n$ .

**Paper algorithm running time:**  $(m \cdot \Delta)^{O(m)} \cdot \|b\|_\infty^2$ .

**Improvement:** The algorithm presented in this paper improves Papadimitriou bound on IP running time:  $(m \cdot \Delta)^{O(m^2)}$  which was the best bound since 1981. In addition, the absolute values of  $b$  doesn't need to be bounded by  $\Delta$  anymore, contrarily to Papadimitriou result.

- (ii) Consider the standard IP in the bounded form,

$$\max\{c^T x : Ax = b, 0 \leq x \leq u, x \in \mathbb{Z}^n\} \quad (2)$$

where  $A \in \mathbb{Z}^{m \times n}$ ,  $b \in \mathbb{Z}^m$ ,  $u \in \mathbb{N}^n$  and  $c \in \mathbb{Z}^n$  and  $|a_{ij}| \leq \Delta$  for each  $i, j$ .

**Paper algorithm running time:**  $n \cdot (\log \Delta)^2 \cdot O(m)^{m+2} \cdot O(\Delta)^{m(m+1)}$ .

**Improvement:** The paper shows that for the bounded and unbounded knapsack problems where all the items are of weight at most  $\Delta_a$ , the algorithm runs in time  $O(n \cdot \Delta_a^2)$  and  $O(n^2 \cdot \Delta_a^2)$  respectively. This improves Tamir bounds so far by a factor of  $n$ .

Both results relies on a lemma of Steinitz that we will define later.

The algorithm (ii) follows from two other results:

- (a) New bounds on the distance of an optimal vertex  $x^*$  of the LP-relaxation and an optimal solution of the integer program itself:

$$\|z^* - x^*\|_1 \leq m \cdot (2m\Delta + 1)^m$$

The new bound improve by a factor of  $n^2$  the classical bound of Cook for integer program in standard form and fixed  $m$ .

- (b) Generalization of Aliev bound on the absolute integrality gap:

$$\|C\|_\infty \cdot O(m)^{m+1} \cdot O(\Delta)^m$$

IP is really important and interesting since most combinatorial optimization and geometry of number can be formulated as IP. We recall that every IP in inequality form can be transformed on IP of the standard form by duplicating variables and introducing slack variables. I have personally met a lot of problems (in the field of fair-division and number partition) that can be formulated as IP. That is one of the reason that I chosed to work on this paper. One of the problem is explained in the following question.

## 2 Related work

IP is a really strong tool, and of course, there is already a lot of work on the field.

- Not Steinitz Lemma related.
  - It is proved by Lenstra that any IP in inequality form is solved in polynomial time if the number of variable is fixed. His algorithm shows a time bound of  $2^{O(n^2)}$ .
  - This has been improved by Kannan to  $2^{O(n \log(n))}$ , which is the best bound on the exponent 2 in 30 years.
  - Papadimitriou provide an algorithm for IP in the standard form that is complementary to Lenstra and Kannan result. His algorithm is pseudopolynomial if  $m$  is fixed. His algorithm is based on dynamic programming (DP). The DP is a maximum weight path problem on the graph. I decided to ignore the technical details here. We will come back on these details later. The upper bound of the running time is  $O(n^{2m+2} \cdot (m\Delta)^{(m+1)(2m+1)})$ .
- Steinitz Lemma related.
  - Dash (and others authors) have showns that IP can be solved in pseudopolynomial time if the number of rows is a function of  $m$ . The run time is less efficient but the interesting aspect of their algorithm is that it relies on linear programming techniques only.
  - Buchin (and others authors) have shown that  $m^{m/2-o(m)} \leq \text{number of rows} \leq m^{m-o(m)}$ , then Dash algorithm is pseudopolynomial in for fixed  $m$ , but doubly exponential in  $m$ .

### 3 Model / Preliminaries / Notation

- $\Delta$  is an upper bound on the entries of  $A$  only.

- $\|\cdot\|$  denotes an arbitrary norm of  $\mathbb{R}^m$ .

Norm definition:

We define by  $N$  a norm of a space vector of  $\mathbb{R}^m$  if for every  $x, y \in \mathbb{R}$ :

- $N(x) = 0$  iff  $x = 0$ .
- $N(x + y) \leq N(x) + N(y)$ .
- for all  $\lambda \in \mathbb{R}$ ,  $N(\lambda x) = |\lambda|N(x)$ .

- Absolute integrality gap: the maximum ratio between the solution quality of the integer program and of its relaxation.

- Steinitz theorem with my word (please see theorem 1.1 of the paper for a formal definition).

If we have  $n$  vectors  $x_i$  in  $\mathbb{R}^m$  such that the sum of every  $x_i$  is 0 and for each  $i$ , the norm of  $x_i$  is smaller or equal to 1, then, there exists a permutation  $\pi$  such that all partial sum satisfy that the norm of the sum of every  $n$  vector  $x_\pi$  is smaller or equal to  $c(m)$ , we can say upper bounded by  $c(m)$ , where  $c(m)$  is a constant depending on  $m$  only. The paper then uses this bound to improve algorithm for IP. We ignore the proof of the theorem for now since it is not relevant for the algorithm.

### 4 Algorithm (i)

(We choose to write the algorithm without explanation first. Don't worry if you don't understand everything, we explain each step later.)

The algorithm works as follows:

- (1) check integer feasibility of (1).
- (2) run a single-source longest path algorithm from 0 to the other nodes  $D$ , in particular to  $b$ .
- (3) if the algorithm detects a cycle of positive weight, we assert that (1) is unbounded.
- (4) otherwise, the longest path from 0 to  $b$  corresponds to an optimal solution of (1).

### 5 Corectness of the algorithm (i)

In the following we explain the mathematics beyond the algorithm, steps by steps. This will prove the correctness of the algorithm and its running time.

## 5.1 (1)

(We let the mathematical explanation as they are in the paper since it is really well explained and there is nothing to summary since each single step is important to understand everything.)

Consider the feasibility problem:

We have to decide whether there exists a non-negative integer vector  $z^* \in \mathbb{Z}_{\geq 0}^n$  such that  $Az^* = b$  holds. The solution  $z^*$  gives rise to a sequence of vectors  $v_1, \dots, v_t$  such that each  $v_i$  is a column of  $A$  and

$$v_1 + \dots + v_t = b$$

The  $i$ -th column of  $A$  appears  $z_i^*$  times on the left of the equation and  $t = \|z^*\|_1$ . The equation can be rewritten as:

$$(v_1 - b/t) + \dots + (v_t - b/t) = 0$$

Observe that the infinity norm of each  $v_i - b/t$  is at most  $2\Delta$ . Steinitz lemma implies that there exists a permutation  $\pi$  of the number  $1, \dots, t$  such that all partial sum of the sequence

$$v_{\pi(1)} - b/t, \dots, v_{\pi(t)} - b/t$$

have infinity norm at most  $2m \cdot \Delta$ . In other words, for each  $j \in \{1, \dots, t\}$  one has

$$\|v_{\pi(1)} + \dots + v_{\pi(j)} - (j/t) \cdot b\|_{\infty} \leq 2m \cdot \Delta$$

This implies that each partial sum of the sequence

$$\|x - (j/t) \cdot b\|_{\infty} \leq 2m \cdot \Delta$$

is contained in the set  $S \subseteq \mathbb{Z}^m$  that consists of all points  $x \in \mathbb{Z}^m$  for which there exists a  $j \in \{1, \dots, t\}$  with

This set  $S$  contains at most  $(4m\Delta + 1)^m \cdot t$  elements. The partial sums of

$$v_{\pi(1)}, \dots, v_{\pi(t)}$$

corresponds to the nodes of a directed walk from 0 to  $b$  in the digraph  $D = (S, A)$  where one has a directed arc  $xy \in A$  from  $x \in S$  to  $y \in S$  if  $y - x$  is a column of  $A$ . If on the other hand there is a path from 0 to  $b$  in this digraph  $D$ , then the arcs of the path define a multiset of columns of  $A$  summing up to  $b$ .

Now, let us focus on the running time of the feasibility checking.

The number of vertices  $|S|$  of the digraph is equal to  $(4m \cdot \Delta + 1)^m \cdot \|b\|_1$ . The number of arcs  $|A|$  is bounded by  $|S|n$ . The integer feasibility problem is an unweighted single source shortest path problem. Bread First Search (BFS) solves it in linear time. Then the running time of (1) is

$$|S|n = O(m \cdot \Delta)^m \cdot \|b\|_1 \cdot n$$

## 5.2 (2)

Once the integer feasibility of the problem is checked, we describe in this section how to tackle the **optimization** problem of (1).

Introduce weights on the arcs if the digraph  $D = (S, A)$  as follows: the weight of the arc  $xy$  is  $c_i$  if  $y - x$  is the  $i$ -th column of  $A$ .

The longest path in the weighted digraph from 0 to  $b$  corresponds to an optimal solution of (1). Since  $|A| \leq |S|n$  the problem can be solved by Bellman-Ford in time  $O(|S| \cdot |A|) = O(n \cdot |S|^2)$ , if there are no positive cycles reachable from 0.

We now prove that there exists a positive cycle reachable from 0 if and only if the feasible integer program (1) is unbounded. It is known that (1) is unbounded iff there exists an integer solution of  $Ax = 0, x \geq 0, c^T x > 0$ . Let  $r^*$  be such a solution. Steinitz Lemma in the spirit of the rearrangement but with  $b = 0$ ,  $r^*$  corresponds to a cycle in  $D$  of positive length starting at 0.

## 5.3 (3) and (4)

Steps (3) and (4) are only a simple check of the output of the single-source longest path algorithm. That is, the running time is  $O(1)$ .

## 6 Algorithm (ii)

We chose to ignore this algorithm and only focus on the algorithm (i).

## 7 Future work

There any discussion of future work along the paper.

## 8 Link of the paper

<https://arxiv.org/abs/1707.00481>