Third IPCO Conference (1993) 11–21
G. Rinaldi and L. Wolsey, eds.

Optimal solution and value of parametric integer programs

Ravi Kannan

Bellcore, 445, South Street, Morristown, New Jersey 07962 and School of Computer Science, Carnegie-Mellon University, Pittsburgh, Pennsylvania 15213, USA
E-mail: ravi@bellcore.com

This paper considers parametric integer programming where the constraint matrix is fixed and we wish to solve the problem repeatedly for different right hand side vectors. This arises in many contexts, for example, in production planning, the constraint matrix may be determined by the technology etc. that generally remains fixed over a long period of time, whereas the right hand side vector may vary depending upon the demands or availability of resources both of which are more volatile. In these contexts, therefore, it makes sense to spend possibly a large amount of time preprocessing the matrix A so that then as each b comes in, we can rather quickly find the optimal or a feasible solution to the Integer Program. We show here the following result: Suppose we are given the $(m \times n)$ constraint matrix A, an objective function c and some affine set P in $\mathbb{R}^m$ over which the right hand side vector varies. Let j be the (affine) dimension of P. After "preprocessing" A and c for time bounded by (a polynomial in the length of the data)$^{n(n+j)}$, we are able to find the optimal solution to the Integer Program

$$\begin{array}{ll} \text{maximize} & c \cdot x \\ \text{subject to} & Ax \le b \end{array}$$

for each input right hand side vector b in P by a parallel algorithm that uses (a polynomial in the data)$^{n(n+j)}$ processors and takes time $O(n(\log(\text{length of data})))$. The parallel algorithm is derived from a structural result for Integer Programming which is in some sense analogous to the existence of a basic feasible (optimal) solution to a feasible (feasible and bounded) Linear Programming problem.

Key words: Parametric integer programming, parallel algorithm.

1. Introduction

For linear programming, we know that if the constraint matrix is fixed, the optimal solution varies in a piecewise linear fashion as the right hand side varies. This is so because, each basic solution can be expressed as a linear function of the right hand side vector and we know that if there is an optimal solution, there is a basic feasible one. Thus given a fixed constraint $m \times n$ matrix A, we can find matrices $T_1, T_2, \dots T_k$, such that for all b, the optimal solution to the linear program $\max c \cdot x : Ax \le b$ is the one of $T_1 b, T_2 b, \dots T_k b$, namely the one that is feasible and attains the best objective function value. We also know that k is at most $\binom{m}{n}$ and all the T_i can be found in time at most poly(m^n). (In fact, better upper bounds on k are available using the upper bound theorem.)

In this paper, we show an analogous result for Integer Programming which is used to derive a parallel algorithm. To state this result, we define a "(generalized) floor function" below. The definition is recursive.

Definition.

- Any affine (linear + constant) function $f : \mathbb{R}^m \to \mathbb{R}^n$ is a floor function.
- A constant times a floor function is a floor function. The sum of two floor functions is a floor function.
- The floor of a floor function is floor function.

Definition. The depth of a floor function is the depth of nesting of floors in the function. The size of the floor function is the length of its description, where all constants are written in binary. The depth can be better understood by thinking of the floor function as a circuit with gates of three types: sum gates (each with two inputs), multiplication by constant gates (each with one input) and floor gates (one input). The depth of the function is the minimum over all circuits that compute the function of the maximum number of floor gates on any path from an input to an output in the circuit.

In this paper, we show that given A, c which are fixed $m \times n$ and $1 \times n$ matrices of integers respectively, and a set P in $\mathbb{R}^m$ of affine dimension j over which b, the right hand side vector can vary, we can produce floor functions $f_1, f_2, \ldots f_k : \mathbb{R}^m \to \mathbb{Z}^n$ such that for all $b \in P$, the optimal solution to the Integer Program $\max c \cdot x : Ax \leq b$ is the one among $f_1(b), f_2(b), \ldots f_k(b)$ that is feasible and attains the best objective function value; each floor function is of depth at most $n+1$. Further the algorithm runs in time $(\text{poly}(\text{length of data}))^{n(n+j)}$. Of course this means that k is also bounded by this amount. Further, the size of each floor function is bounded above by $(\text{poly}(n))^n \times$ (length of input). Clearly, then the optimal solution value as a function of b can be expressed as $\max\{c \cdot f_i(b) : Af_i(b) \leq b\}$. Once we have these floor functions, it will be seen that using known techniques, we can get the parallel algorithms claimed.

There are earlier results of Blair and Jeroslow [2] (and many other authors - see for example [12]), that study the optimal solution value as a function b. [2] shows that the value function is the minimum of a finite number of floor functions each of finite depth and size; their proofs use cutting plane techniques and the floor functions that arise are called Gomory functions. Unfortunately, however, the actual bound on their number is exponential in the length of the data even for fixed n; one of the main contributions of this paper is to get bounds that are polynomial for fixed n. While our results have this superficial similarity with those of [2], the techniques are quite different and in a sense also the results: their results essentially get a hold of all the facets of the convex hull of the integral feasible points and then apply linear programming duality. (Hence we get the value function as a minimum.) Ours on the other hand, produce what we may call a "test set" for the integer program, i.e., a set of candidate solutions for the optimum, so that one of them (of course the best feasible one) is guaranteed to be an optimal solution. (Chvátal [3] defined the rank of a polytope to be (essentially) the minimum d such that all facets of the convex hull of the integer points in the polytope can be defined by floor functions of depth at most d. His main result in [3] was to show that all polytopes have finite rank. In a sense, here we are arguing that the rank is at most $n+1$; but where we are interested in a test set rather than the entire convex hull of the integer points.)

The arguments in this paper follow the lines of two papers–[6] and [7]. Many ideas are similar to these papers, but the emphasis there was a uniform description of the test set as b varied over P. To get this, those papers had to partition P into doubly exponential (in n, j) number of pieces and further the description of each piece was not amenable to be processed by a parallel algorithm. Here, the task is simpler - rather than a uniform description, we are looking for something which will work for each given right hand side in P. Thus we are able to get rid of the double exponential dependence; also we are able to get a closed form description with floor functions whereas there the descriptions were implicit with extra integer variables.

It would be interesting to use these results to do some sensitivity analysis - i.e.,figuring out what "local" changes to b leave the (feasible) optimal function f_i unchanged, so that a new optimal solution can be easily computed. This line of investigation will be pursued.

Tools from the geometry of numbers

Here, we introduce some tools needed.

Suppose K is a closed bounded convex set in $\mathbb{R}^n$ and v is an element of $\mathbb{R}^n$. The *width of K along v* is

$$\max\{v \cdot x : x \in K\} - \min\{v \cdot x : x \in K\}.$$

The *width of K* (with respect to the lattice $\mathbb{Z}^n$) is defined to be the minimum width of K along any nonzero integer vector. Note that this differs from the usual definition of the geometric width of K, where the minimum is over all vectors v of length 1, rather than all nonzero integer vectors. The width as defined here is greater than or equal to the geometric width since nonzero integer vectors have length at least one. The following theorem will be used.

Flatness Theorem [8]. *There is a universal constant c_o such that any closed bounded convex set K in $\mathbb{R}^n$ of width at least $c_o n^2$ contains a point of $\mathbb{Z}^n$.*

Remark. The constant c_o will be used throughout the paper. By looking at the case $n = 1$, we see that c_o must be at least 1, a fact that we will use.

In [9] it is shown that for any fixed $m \times n$ matrix A satisfying some nondegeneracy condition, there is a small finite set V of nonzero integer vectors such that for any "right hand side" b, there is some $v(b)$ belonging to V such that the polytope $K_b = \{x : Ax \leq b\}$ has approximately the smallest width along $v(b)$; more precisely, the width of K_b along $v(b)$ is at most twice the width of K_b along any nonzero integer vector. Section 2 of this paper proves from first principles a result in the same spirit. There are two differences - here, we do not assume any nondegeneracy condition. Secondly, in the result here, b is allowed to vary over some subset of $\mathbb{R}^m$ and the upper bound on the cardinality of V is in terms of the dimension of the affine hull of this subset. Letting the subset be the whole of $\mathbb{R}^m$, we can recover a result similar to [9].

Notation

$\mathbb{R}^n$ is Euclidean n space. The lattice of all integer vectors in $\mathbb{R}^n$ is denoted $\mathbb{Z}^n$. For any two sets $S, T \subseteq \mathbb{R}^n$, we denote by $S + T$ the set $\{s + t : s \in S; t \in T\}$. For any positive real, λ, we denote by λS, the set $\{\lambda s : s \in S\}$.

By a "rational polyhedron", we mean a polyhedron that can be described by a system of inequalities that have rational coefficients; the inequalities may have irrational right hand sides.

In much of the paper A will be a fixed $m \times n$ matrix. If the meaning of A is clear from the context, for any b in $\mathbb{R}^m$, the polyhedron $\{x \in \mathbb{R}^n : Ax \leq b\}$ will be denoted by K_b. In much of the paper, b will vary over some polyhedron in $\mathbb{R}^m$. Some bounds in the paper will be in terms of the affine dimension j_o of this polyhedron. The "size" of a rational matrix is the number of bits needed to express it. It is assumed that integers are written in binary notation, so it takes $O(\log M)$ length string to express an integer of magnitude M.

If A, b, c are $m \times n, m \times 1, 1 \times n$ matrices respectively, then we denote by $LP(A, b, c)$ the linear program

$$\max c \cdot x \text{ subject to } Ax \leq b.$$

We denote by $IP(A, b, c)$ the integer program obtained by adding the restriction that all x_i be integers to the above linear program.

In talking about certain bounds (on running times or sizes), we will say the bound is poly(s, t, u) etc. This is abbreviation for saying that there exists a polynomial $p(\cdot, \cdot, \cdot)$ such that the bound is at most $p(s, t, u)$ where s, t, u are certain parameters associated with the problem.

2. Vectors along which K_b have small width

For each fixed b, there is a nonzero integer direction that achieves the minimum width of K_b. The main result of this section is Lemma 2.1 which says that we can compute, given A, a small number of nonzero integer directions such that as b varies over a bounded set, for each K_b, one of our directions achieves close to minimum width.

Lemma 2.1. *Suppose A is an $m \times n$ matrix of integers of size ϕ. For each $b \in \mathbb{R}^m$, we denote by K_b the polyhedron $\{x : Ax \leq b\}$. Let P be a polytope in $\mathbb{R}^m$ of affine dimension j_o such that for all $b \in P$, K_b is nonempty and bounded. Let M be $\max\{|b| : b \in P\}$. There is an algorithm that finds nonzero integer vectors $v_1, v_2, \ldots, v_r$, each of size at most poly$(n)(\log M + \phi)$ where r is at most poly$(m, n, \phi, \log M)^{n+j_o}$ such that for all $b \in P$, we have $\exists i, 1 \leq i \leq r$ such that one of the following is true:*

$$Width_{v_i}(K_b) \leq 1$$

or

$$\forall u \neq 0, u \in \mathbb{Z}^n,\ Width_{v_i}(K_b) \leq 2\, Width_u(K_b).$$

Further, the algorithm works in time bounded by

$$poly(m, n, \phi, \log M)^{(n+j_o)n}.$$

Proof. The first m of the v_i's will be the rows of A. We note that every K_b of zero volume has width 0 along one of these m vectors. Also, if a K_b has width at most 1 along one of these m directions, it is "taken care of" by that direction. So we only need the rest of the vectors to take care of full-dimensional K_b with width at least 1 along each of the m facet directions.

Since K_b is bounded, we have that K_b is contained in a ball of radius $M2^{4n^2\phi}$ ([13], Theorem 10.2) around the origin. Also, K_b has a centroid—say—x_o. (The centroid x_o is the unique point such that $\int_{K_b}(x - x_o)dx = 0$.) Consider $K_b - x_o$. Let this be $\{x : Ax \leq b'\}$. Note that b' belongs to $P' = P+($ column space of A $)$ which is a set of affine dimension at most $n + j_o$. By the above, $0 < b'_i \leq M2^{5n^2\phi}\forall i$. By a property of the centroid (namely, if y_o is the centroid of a bounded convex set K in $\mathbb{R}^n$, then for any $z \in K$, we have $(1 + \frac{1}{n})y_o - \frac{1}{n}z \in K$.) , and the lower bound of 1 on the width of K_b in any of the facet directions, we have that

$$\frac{1}{(n+1)} \leq b'_i \leq M2^{5n^2\phi}\forall i.$$

Let $R \subseteq \mathbb{R}^m$ be the rectangular solid $\{y : \frac{1}{(n+1)} \leq y_i \leq M2^{5n^2\phi} \forall i\}$. Applying Lemma (2.2) with Q = the affine hull of P', we get a finite set V' in $\mathbb{R}^m$ such that for each $y \in R \cap P'$, there is a $y' \in V'$ with $y' \leq y \leq 2y'$. Also, by that lemma, the size of each $y' \in V'$ is at most $n(\log M + 5n^2\phi + \log(n+1)) \leq \text{poly}(n)(\log M + \phi)$. (Note that by that lemma, the set V' can be found in time $\text{poly}(m, n, \phi, \log M)^{n+j_o}$.) For each y' in V' such that $K_{y'}$ is full dimensional, we find the nonzero integer vector that attains the width of $K_{y'}$. This set of nonzero integer vectors suffices as our set of v_i's. This is so because $y' \leq y \leq 2y'$ implies that $K_{y'} \subseteq K_y \subseteq K_{2y'}$ which implies that for any nonzero vector v, $\text{Width}_v(K_{y'}) \leq \text{Width}_v(K_y) \leq \text{Width}_v(K_{2y'})$. (The nonzero integer vector along which the width of a polyhedron is minimised, can be found in time bounded by $\text{poly}^{O(n)}$—see [8] (1986 version).) We note that the bound on r follows easily from Lemma 2.2.

We also need to argue a bound on the sizes of the v_i. To this end, suppose y is some element of V' and w is the integer vector that achieves the width of K_y. Suppose $w \cdot x$ is maximized at a vertex p of K_y. Then, by linear programming duality theory, there is a nonsingular $n \times n$ submatrix A_1 of A (whose rows form tight constraints at p) such that wA_1^{-1} is a nonnegative vector and $wA_1^{-1}y$ is the maximum value of $w \cdot x$ over K_y. Similarly, there is a nonsingular $n \times n$ submatrix A_2 of A such that $-wA_2^{-1}$ is a nonnegative vector and $wA_2^{-1}y$ is the minimum value of $w \cdot x$ over K_y. So w is the (an) optimal solution to the Integer Program

$$\min w \cdot (A_1^{-1} - A_2^{-1})y : wA_1^{-1} \geq 0 \ ; \ wA_2^{-1} \leq 0 \ ; \ w \text{ integers} .$$

A_1^{-1}, A_2^{-1} have size at most $\text{poly}(n)\phi$ and so by standard results, w has size at most $\text{poly}(n)(\log M + \phi)$. (The results only tell us that there exists a w with these size bounds; we take such a w as our v_i.) □

Lemma 2.2. *Let $R \subseteq \mathbb{R}^m$ be the rectangle $\{y : \alpha \leq y_i \leq \beta \forall i\}$ where $0 < \alpha \leq \beta$ are arbitrary rationals. Let Q be any affine subspace of $\mathbb{R}^m$ with dimension say t. Then there exist a finite set V' in $\mathbb{R}^m$ with $|V'| \leq \left(2m(\log_2 \frac{\beta}{\alpha} + 1)\right)^t$ such that for each $y \in R \cap Q$, there is a $y' \in V'$ with $y' \leq y \leq 2y'$. Further, for each y' in V', each coordinate of y' is α times an integral power of 2.*

Further, given R, Q, the set V' can be found in time $\text{poly}(m, \log \frac{\beta}{\alpha})^t$.

Proof. Divide R into sub-rectangles each of the form

$$\{z : \alpha 2^{p_i} \leq z_i \leq \alpha 2^{p_i+1} \text{ for } i = 1, 2 \ldots, m\}$$

where $p_1, p_2 \ldots, p_m$ are natural numbers between 0 and $l = \log_2(\beta/\alpha)$. I will show by induction on the pair t, m that $Q \cap R$ is contained in the union of some

$$2^t m^t (l+1)^t$$

subrectangles of R which clearly proves the lemma.

The case $t = 0$ is clear for all m. The case $m = 0$ is trivial. For higher t, note that if Q intersects a subrectangle, it intersects the boundary of the subrectangle. For each $i, 1 \leq i \leq m$ and each $p_i, 0 \leq p_i \leq l$, consider the $(m-1)$-dimensional rectangle $R' = R \cap \{z : z_i = 2^{p_i}\alpha\}$ and the division of it into subrectangles "induced" by the division of R. Also, let $Q \cap \{z : z_i = 2^{p_i}\alpha\}$ be Q'. If for any i and any p_i, such a Q' equals Q, we have the lemma by induction on m. Assume this is not the case. Then, Q' is a $(t-1)$-dimensional affine space. Applying the ctive assumption, we know that there are $(2(m-1)(l+1))^{t-1}$ subrectangles whose union ains $Q' \cap R'$. Each such subrectangle is a facet of 2 subrectangles of R. Thus there are

2.$(2(m-1)(l+1))^{t-1}\, m\,(l+1)$ subrectangles of R whose union contains $Q \cap R$.

To get the required algorithm, note that in the case where some Q' equals Q, we get one "problem of size " $t, m-1$ and in the other case, we get $m(l+1)$ problems each of "size" $t-1, m-1$. The time bound follows by routine analysis of the recursive algorithm. □

Lemma 2.3. *Suppose K is a rational polyhedron in $\mathbb{R}^n$ and $v \in \mathbb{Z}^n \setminus \{0\}$ satisfies*

$$\textit{either } Width_v(K) \le 1$$

or

$$\forall u \neq 0, u \in \mathbb{Z}^n,\ Width_v(K) \le 2\, Width_u(K).$$

Suppose also that y is in K. Let $s = 2c_o n^2 + 1$ (where c_o is the constant from the Flatness Theorem of Section 1). Then, $K \cap \mathbb{Z}^n$ is nonempty iff there exists $x \in K \cap \mathbb{Z}^n$ with $|v\cdot(x-y)| \le s$.

Proof. Suppose z belongs to $K \cap \mathbb{Z}^n$ and $|v \cdot (z-y)| > s$. Then of course the width of K along v is greater than s. Let K' be obtained by shrinking K about y by a factor of $\text{Width}_v(K)/s$, i.e., ...

$$K' = \frac{s}{\text{Width}_v(K)}(K-y)+y.$$

Then, $\text{Width}_v(K') = s$ and also, $\text{Width}_v(K') \le 2\text{Width}_u(K')$ for all nonzero integer u. This implies that $K' \cap \mathbb{Z}^n$ is nonempty by the Flatness Theorem. □

3. The main result for bounded right hand sides

Theorem 3.1. *Let A be an $m \times n$ matrix and c an $1 \times n$ matrix of integers, with size of $\binom{A}{c}$ equal to ϕ. Let Q be a set in $\mathbb{R}^m$ of affine dimension j_o such that for all $b \in Q$, the set $K_b = \{x : Ax \le b\}$ is nonempty and bounded.* [1] *Let $M = (\max_{b \in Q}(|b|+1))$. There is an algorithm that yields floor functions $f_i : \mathbb{R}^m \to \mathbb{Z}^n, 1 \le i \le r$ each of depth n and size $(poly(n))^n(\phi + \log M)$ with $r \le poly(m, n, \phi, \log M)^{n(n+j_o)}$ such that for all $b \in Q$, the best feasible solution to $IP(A, b, c)$ among $f_1(b), f_2(b), \ldots f_r(b)$ is the optimal solution to $IP(A, b, c)$. Further, the size of each floor function is at most $poly(m, n, \phi, \log M)$. The running time of the algorithm is bounded by $poly(m, n, \phi, \log M)^{n(n+j_o)}$. [In particular, if none of the $f_i(b)$ is feasible to $IP(A, b, c)$, then the integer program is infeasible.]*

Proof. We use induction on n. For $n = 1$, it is easy to see that we can assume without loss of generality that all A_i are nonzero. We can take $r = 2m$ and define for $1 \le i \le m$, functions $f_{2i} = \lfloor (b_i/A_i) \rfloor$ and $f_{2i-1} = \lfloor (b_i/A_i) \rfloor + 1$. It is easy to see that the theorem is true with these functions.

Now we go to general n. We apply Lemma 2.1 with the matrix $\binom{A}{-c}$ and P in $\mathbb{R}^{m+1}$ defined by $P = \{(b, c_o) : b \in Q, c_o \in \mathbb{R}\}$. This gives us nonzero integer vectors $v_1, v_2, \ldots v_p$ with $p \le \text{poly}(m, n, \phi, \log M)^{n+j_o}$ such that for all (b, c_o) in P, the polytope $K_{(b,c_o)} = \{x : Ax \le b\ ;\ c \cdot x \ge -c_o\}$ has small width (i.e., width at most 1 or at most twice the minimum width) in

[1] From linear programming theory, we know of course that the property that K_b is bounded is independent of b.

one of the directions $v_1, v_2, \ldots v_p$. Now suppose for some $b \in Q$, the integer program $IP(A,b,c)$ is feasible and so it has an optimal solution whose value is say, $-c_o$. Then the polytope $K_{(b,c_o)}$ is free of interior lattice points, and has a lattice point on its boundary, namely the optimal solution to $IP(A,b,c)$. The width of this polytope is at most $s/2$ by the Flatness Theorem (where $s = 2c_o n^2 + 1$) and so its width along one of the v_i is at most s. This shows that for any optimal solution y to the linear program $LP(A,b,c)$, there is some optimal solution z to $IP(A,b,c)$ that we have $|v_i \cdot (z-y)| \leq s$ for some i (since, of course y belongs to $K_{(b,c_o)}$). There are at most m^n matrices $T_k, k = 1,2,\ldots$ such that for each b, an optimal solution to $LP(A,b,c)$ is $T_k b$ for some k. (In linear programming terminology, these are the basis inverse matrices suitably augmented by 0 columns.)

Now consider the $p.(2s+1).m^n$ choices of i,j,k such that $1 \leq i \leq p$, $-s \leq j \leq s$ and $1 \leq k \leq m^n$. For each choice consider the integer program:

$$\max c \cdot x : Ax \leq b \; ; \;\; v_i \cdot x = \lfloor v_i \cdot T_k b \rfloor + j \;\; x \text{ integers.}$$

Also, we have that the size of v_i is at most poly$(n)(\log M + \phi)$. Each of these is an $n-1$ dimensional problem. First, we will rewrite them with $n-1$ variables. To this end, first we remark that the algorithm of [8] gives us actually a basis containing the vector v_i as the first vector. Writing the basis vectors as the rows of a unimodular matrix U (whose size is bounded by poly$(n)(\log M + \phi)$) and making the substitution $y = Ux$ into the Integer Program, we get the following IP:

$$\max cU^{-1} \cdot y : AU^{-1} y \leq b \; ; \; y_1 = b_{m+1} \;\; y \text{integers}$$

where we have let $b_{m+1} = \lfloor v_i \cdot T_k b \rfloor + j$. Making the substitution $y_1 = b_{m+1}$, we get an Integer program in $n-1$ variables $y_2, y_3, \ldots y_n$ where it is easy to see that the sizes of all coefficients (including the right hand sides) is at most poly$(n)(\log M + \phi)$. Further, the right hand side varies over an affine space of dimension at most $j_o + 1$. (The extra 1 is because of the parameter b_{m+1} which is not expressible as an affine function of $b_1, b_2, \ldots b_m$.) Now we apply the inductive assumption on the $n-1$ dimensional problem to get the floor functions each of depth $n-1$ in the parameters $b_1, b_2, \ldots b_{m+1}$; thus they are of depth at most n in the parameters $b_1, b_2, \ldots b_m$. Also, the size of the floor functions is at most (poly$(n-1))^{n-1}$poly$(n)(\log M + \phi)$ which is at most (poly$(n))^n(\log M + \phi)$ as claimed. The bound on the number of floor functions follows easily. □

4. The case of unbounded right hand sides

In this section, we will prove the theorem of the last section as b varies over an unbounded set. To do so, we will show that for any $b \in P$, the optimal solution to IP(A,b,c) can be easily obtained from the optimal solution to IP(A,d,c) where d has all its components of size bounded by poly(size of A). Further, we will show that d is a "piecewise affine" function of b ; i.e., that b can be partitioned into polynomially many pieces such that for each piece in the partition, there is an affine function that maps b to d. This proof will use Lemma 4.2. First, we state the theorem.

Theorem 4.1. *Let A be an $m \times n$ matrix and c an $1 \times n$ matrix of integers, with size of $\binom{A}{c}$ at most ϕ. Let Q be a set in $\mathbb{R}^m$ of affine dimension j_o, such that for all $b \in Q$, the set $K_b = \{x : Ax \leq b\}$ is nonempty and bounded. There is an algorithm that yields floor functions $f_i : \mathbb{R}^m \to$*

$\mathbb{Z}^n, 1 \leq i \leq r$ each of depth $n+1$ with $r \leq (poly(m,n,\phi))^{n(n+j_o)}$ and size $poly(m,n,\phi)$ such that for all $b \in Q$, the best feasible solution to $IP(A,b,c)$ among $f_1(b), f_2(b), \ldots f_r(b)$ is the optimal solution to $IP(A,b,c)$. The running time of the algorithm is bounded by $poly(m,n,\phi)^{n(n+j_o)}$. [In particular, if none of the $f_i(b)$ is feasible to $IP(A,b,c)$, then the integer program is infeasible.]

First, we need some general results. Suppose $v \cdot x = v_o$ is a hyperplane in Euclidean space. It partitions space into two "regions" - $\{x : v \cdot x \leq v_o\}$ and $\{x : v \cdot x > v_o\}$. Similarly, a set of l hyperplanes in $\mathbb{R}^m$ partition $\mathbb{R}^m$ into (at most) 2^l "regions"-each region being determined by which side of each hyperplane it is on. There is another well-known upper bound on the number of regions - it is

$$\sum_{k=0}^{m} \binom{l}{k}.$$

For $l \leq m$, the sum is 2^l and the result is obvious. For $l > m$, we proceed by induction. The number of regions formed by the first $l-1$ of the hyperplanes is at most $\sum_{k=0}^{m}\binom{l-1}{k}$ by induction. Now imagine adding the l th hyperplane. we claim that the number of existing regions that the l th hyperplane intersects is at most $\sum_{k=0}^{m-1}\binom{l-1}{k}$ - to see this, note that the intersections of the existing regions with the l th hyperplane form a partition of the l th hyperplane (an $m-1$ dimensional affine space). Each region intersected by the l th hyperplane is divided into two by it. So we get the total number of regions formed by all the l hyperplanes is at most

$$\sum_{k=0}^{m}\binom{l-1}{k} + \sum_{k=0}^{m-1}\binom{l-1}{k} = \sum_{k=1}^{m}\left(\binom{l-1}{k-1} + \binom{l-1}{k}\right) + 1$$

which proves the claim. The lemma below follows immediately.

Lemma 4.2. *Suppose V is a j dimensional affine subspace of $\mathbb{R}^m$. For any set of l hyperplanes in $\mathbb{R}^m$, the number of regions in the partition of $\mathbb{R}^m$ by the l hyperplanes that V intersects is at most*

$$\sum_{k=0}^{j}\binom{l}{k} \leq l^j.$$

Further, we can find the regions intersected by V in time bounded by $O(\text{size of data}^j)$.

Proof. The first part is already proved. For the algorithm, we go again to the first part of the proof and see that a problem with parameters l, j is reduced to two problems one with parameters $l-1, j$ and the other $l-1, j-1$. If the running time of the algorithm is $T(l,j)$, we get the recurrence $T(l,j) \leq T(l-1,j) + T(l-1,j-1) + O(1)$ which solves to $T(l,j)$ is in $O(l^j)$. □

Suppose P is a set of affine dimension j_o in $\mathbb{R}^m$. Consider each of the (at most m^n) nonsingular $n \times n$ submatrices B of A. For each of these we can define an $n \times m$ matrix T by augmenting B^{-1} with 0 columns so that the possible corners of any K_b are of the form Tb for such T. For each such T, and each i, $1 \leq i \leq m$, consider the hyperplane $\{b : a^{(i)}Tb = b_i\}$ in $\mathbb{R}^m$. (Reminder: $a^{(i)}$ is the i th row of A.) There are at most m^{n+1} such hyperplanes and so by Lemma 4.2, we have that P intersects at most $m^{(n+1)j_o}$ of the regions that these hyperplanes partition $\mathbb{R}^m$ into. It is not difficult to see that we can find these regions in time bounded by poly(^{n}). For each such region U, there is a T_U such that for all $b \in U$, $T_U b$ belongs to K_b and maximizes $c \cdot x$ over all x in K_b. Let

$$b' = b - A\lfloor T_U b \rfloor.$$

Then K'_b is a translate of K_b by $-\lfloor T_U b \rfloor$ and so the optimal solution to $IP(A, b, c)$ is given by $\lfloor T_U b \rfloor$ plus the optimal solution to $IP(A, b', c)$. We will argue below that any component of b' exceeding a certain bound can be reduced without changing the optimal solution value to $IP(A, b', c)$, then we will have bounded b and we can appeal to the previous section. We use the following theorem ([4], quoted in Theorem 17.2 of [13]) to argue that the b'_i exceeding a certain value can be reduced.

Theorem 4.3 ([4]). *Let A be an integral matrix such that each subdeterminant of A is at most Δ in absolute value and suppose both $LP(A, b, c)$ and $IP(A, b, c)$ have finite optimal solutions. Then for any optimal solution y to the linear program, there is an optimal solution z to the integer program such that $|z - y|_\infty \leq n\Delta$. ($n$ is the number of columns of A.)*

Consider the m hyperplanes $(b - AT_U b)_i = n^2 2^{3\phi} + 2n2^\phi$ for $i = 1, 2, \ldots m$. By applying Lemma 4.2, we see that U intersects at most m^{j_o} of the regions that these m hyperplanes partition space into. We partition U into these m^{j_o} or less parts. Thus we have found a partition of P into polyhedra $P_1, P_2, \ldots P_t$ (each P_i is one of the parts of some U) with $t \leq m^{(n+2)j_o}$ and associated with polyhedron P_k in the above partition, we have an affine transformation $T(P_k)$ and an $I(P_k) \subseteq \{1, 2, \ldots m\}$ such that for all $b \in P_k$,

$$0 \leq (b - AT(P_k)b)_i \leq n^2 2^{3\phi} + 2n2^\phi \forall i \in I(P_k)$$

and

$$(b - AT(P_k)b)_i > n^2 2^{3\phi} + 2n2^\phi \forall i \notin I(P_k).$$

For each $b \in P_k$, let b' be $b - A\lfloor T(P_k)b \rfloor$, let b'' be defined by $b''_i = b'_i$ for $i \in I(P_k)$ and $b''_i = n^2 2^{3\phi} + n2^\phi$ for other i.

Note that for $i \notin I(P_k)$, since $(b - AT(P_k))_i > n^2 2^{3\phi} + 2n2^\phi$, we have that $(b - A\lfloor T(P_k)b \rfloor)_i > n^2 2^{3\phi} + n2^\phi$ and so $b'' \leq b'$ which implies that $K_{b''} \subseteq K_{b'}$. Now, $T(P_k)b - \lfloor T(P_k)b \rfloor = y$ (say) is the optimal solution to $LP(A, b', c)$. By Theorem 4.3, we know that $IP(A, b', c)$ has an optimal solution z with $|z - y|_\infty \leq n\Delta$. Since $|y|_\infty \leq 1$, we have then $|z|_\infty \leq (n\Delta + 1)$. Thus, we have that $|Az|_\infty \leq n2^\phi(n\Delta + 1) \leq n^2 2^{3\phi} + n2^\phi$ (since we have $\Delta \leq 2^{2\phi}$ for example from ([13], Theorem 3.2)). This implies that z belongs to $K_{b''}$ and so we have that any optimal solution to $IP(A, b'', c)$ is an optimal solution to $IP(A, b', c)$.

To apply the theorem of the last section, observe that all b'' we get in the above process satisfy the following three conditions:

- $|b''_i| \leq n^2 2^{3\phi} + 3n2^\phi \forall i \in I(P_k)$
- $b''_i = n^2 2^{3\phi} + n2^\phi \forall i \notin I(P_k)$
- $\exists b \in P_k, y \in \mathbb{R}^n$ such that $u_i = (b + Ay)_i \forall i \in I(P_k)\}$

Let P'_k be the polytope consisting of all vectors b'' in $\mathbb{R}^m$ satisfying the above three conditions. the first two conditions imply that P'_k is contained in a sphere of radius $n^2 2^{3\phi} + 3n2^\phi$ and two imply that the affine dimension of P'_k is at most $j_o + n$. We will apply the theorem (of section) with this P'_k, with $M = n^2 2^{3\phi} + 3n2^\phi$ and with the given A and c. That theorem the floor functions that it promises. Note that the arguments to the floor functions are actually $b - A\lfloor T(P_k)b \rfloor$, but clearly, these can be looked as floor functions with only b as the argument at the cost of increasing the depth by 1. Finally, we do this process for each P_k, $k = 1, 2, \ldots t$ and take all the floor function so obtained.

5. The parallel algorithm

To get the parallel algorithm, we recall that arithmetic operations (additions, subtractions, multiplications and divisions) can all be done with poly(operand size) processors and O(log(operand size)) time by standard algorithms [1]. The floor functions are all evaluated in parallel. Each floor function is evaluated sequentially except for the arithmetic operations in it which are each done in parallel. Since we have a bound of $(\text{poly}(n))^n(\log M + \phi)$ on the size of the floor function, we can evaluate it in parallel time n(log(length of data)) as claimed.

One of the more interesting open problems in parallel complexity seems to be the question of whether the greatest common divisor of two integers can be found by a parallel algorithm that runs in polylogarithmic time with a polynomial number of processors. An even simpler question is the following: given an integer a, after preprocessing it for time bounded by a polynomial in the number of bits of a, can one then find the greatest common divisor of a and any given b in polylogarithmic time with a polynomial number of processors? [Note if we are allowed to factor a into its prime factors, the problem would be trivial.] It is possible that we can write a set of linear constraints involving a polynomial number of variables where the coefficients depend in some (possibly nonlinear) way on a and the right hand side vector depends linearly on b such that the solution to the IP gives us the greatest common divisor of a and b. If this is so, then the results of this paper obviously solve the second problem.

References

[1] P. Beame, S.A. Cook and H.J. Hoover, "Log depth circuits for division and related problems," *SIAM Journal on Computing* 15 (1986) 994–1003.

[2] C.E. Blair and R.G.Jeroslow, "The value function of an integer program," *Mathematical Programming* 23 (1982) 237–273.

[3] V. Chavátal, "Edmonds polytopes and a hierarchy of combinatorial problems," *Mathematical Programming* 5 (1973) 305–337.

[4] W. Cook, A.M.H. Gerards, A. Schrijver and E. Tardos, "Sensitivity theorems in integer linear programming," *Mathematical Programming* 34 (1986) 251–264.

[5] M. Grötschel, L. Lovász and A. Schrijver, *Geometric algorithms and combinatorial optimization* (Springer-Verlag, 1988).

[6] R. Kannan, "Lattice translates of a polytope and the Frobenius problem," *Combinatorica* 12 (1992) 161–177.

[7] R. Kannan, "Test sets for Integer programs, ∀∃ sentences," in *Polyhedral Combinatorics*, W. Cook and P.D. Seymour, eds., DIMACS series in Discrete Mathematics and Theoretical Computer Science, Volume 1, 1990, American Mathematical Society, pp. 39–47.

[8] R. Kannan and L. Lovász, "Covering minima and lattice point free convex bodies," in *Lecture Notes in Computer Science* 241, K.V. Nori, ed., (Springer-Verlag, 1986) pp. 193–213. Final version in *Annals of Mathematics* 128 (1988) 577–602. (see Math. Rev. 89b:11055, 89i:52020)

[9] R. Kannan, L. Lovász and H.E. Scarf, "The shapes of polyhedra," Cowles Foundation Discussion paper No. 883, September (1988). To appear in *Mathematics of Operations Research.*

[10] H.W. Lenstra, "Integer programming with a fixed number of variables," *Mathematics of Operations Research* 8 (1983) 538–548.

[11] L. Lovász, "Geometry of Numbers and Integer Programming," Proceedings of the 13th International Symposium on Mathematical Programming, M. Iri and K. Tanabe, eds., (Mathematical Programming, 1989) pp. 177–201.

[12] G.L. Nemhauser and L.A. Wolsey, *Integer and Combinatorial Optimization* (Wiley-Interscience, 1988).
[13] A. Schrijver, *Theory of Linear and Integer Programming* (Wiley, 1986).