SIAM J. COMPUT.
Vol. 27, No. 4, pp. 960–971, August 1998
© 1998 Society for Industrial and Applied Mathematics
003

PLANAR INTEGER LINEAR PROGRAMMING IS NC EQUIVALENT TO EUCLIDEAN GCD*

D. F. SHALLCROSS†, V. Y. PAN‡, AND Y. LIN-KRIZ§

Abstract. It is not known if planar integer linear programming is P-complete or if it is in NC, and the same can be said about the computation of the remainder sequence of the Euclidean algorithm applied to two integers. However, both computations are NC equivalent. The latter computational problem was reduced in NC to the former one by Deng [*Mathematical Programming: Complexity and Application*, Ph.D. dissertation, Stanford University, Stanford, CA, 1989; *Proc. ACM Symp. on Parallel Algorithms and Architectures*, 1989, pp. 110–116]. We now prove the converse NC-reduction.

Key words. integer linear programming, Euclidean algorithm, greatest common divisor, parallel computational complexity

AMS subject classifications. 68Q15, 68Q22, 03D15

PII. S0097539794276841

1. Introduction.

Our main result is the proof of the NC-equivalence of the planar integer linear programming problem and the computation of the Euclidean remainder sequence, defined by the Euclidean algorithm for two integers, which terminates with their greatest common divisor (GCD). Hereafter, we will refer to these two major computational problems as 2-ILP and EUGCD, respectively. Both 2-ILP and EUGCD, as well as k-ILP, that is, the integer linear programming with k variables, for a fixed k, belong to the class P; specifically, sequential Boolean time $O(k^{9k}L\log L)$ suffices for solving k-ILP [7], [8], [12], and time $O(L^2)$ suffices for EUGCD [1], [11], where L denotes the input length (size).

On the other hand, very little is known about parallel complexity of both k-ILP and EUGCD. In particular, it is not known if either of these problems belongs to NC or if either of them is P-complete. (We recall that a computational problem is in NC if it can be solved by using $O((\log L)^c)$ time and $O(L^c)$ processors for a fixed c independent of the input length L; NC-reduction (or reduction in NC) of one problem to another is the reduction under the above bounds on time and on the processor number; NC-equivalence is the existence of NC-reductions in both directions; a P-complete problem is one whose solution in NC would imply the NC solution of all problems in P (see [4, Chapter 7], [5], [6], [10])).

There are relatively few computational problems for which, like for k-ILP and EUGCD, we neither know if they belong to NC nor if they are P-complete. Several computational problems of this kind are related to k-ILP and EUGCD via NC-reductions (see [2], [3], and Figure 1). These problems include GCD (computing the GCD of two integers), SGCD (solution of the set equation $m\mathbf{Z}+n\mathbf{Z}=d\mathbf{Z}$), expanding

*Received by the editors November 8, 1994; accepted for publication (in revised form) April 16, 1996; published electronically May 18, 1998. The results of this paper have been presented at the 3rd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '92) and at the 34th Annual IEEE Symposium on Foundations of Computer Science (FOCS '93).

http://www.siam.org/journals/sicomp/27-4/27684.html

†Bellcore, 445 South St., Morristown, NJ 07962 (davids@bellcore.com).

‡Department of Mathematics and Computer Science, Lehman College of the City University of New York, Bronx, NY 10468 (VPAN@lcvax.lehman.cuny.edu) and International Computer Science Institute, 1947 Center St., Berkeley, CA 94704. The work of this author was supported by NSF grant CCR-902690 and PSC CUNY awards 661340, 6622478, and 664334.

§1524 LSA Building, University of Michigan, Ann Arbor, MI 48109-1382 (ylinkriz@umich.edu).

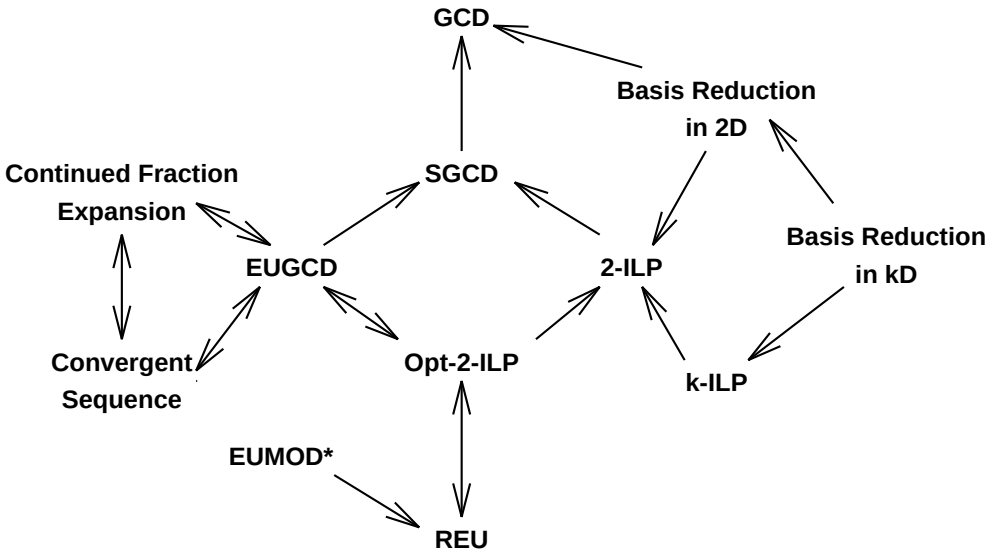

FIG. 1. *Arrowhead problems can be NC reduced to arrowtail problems.*

the continued fraction for the ratio of two integers, computing the sequence of their convergents, and reduction of a lattice of k dimensions for a fixed k.

Deng in [2], [3] proved NC-reduction from EUGCD to Opt-2-ILP, which is the (stronger) optimization version of 2-ILP (see Definition 1). (In Figure 1, we use 2-ILP for the feasibility integer linear programs.) The NC-reduction in the opposite direction, from Opt-2-ILP to EUGCD, turned out to be more elusive. Lin-Kriz and Pan in [13] established NC-equivalence between Opt-2-ILP and the problem they called REU, thus abbreviating relative EUGCD. REU amounts to solving both EUGCD and EUMOD*, the special case of MOD*, where MOD* denotes the problem of computing MOD*$(c, b_1, ..., b_n) = (...((c \bmod b_1) \bmod b_2)...) \bmod b_n$, for any set of natural numbers c, b_1, b_2, ..., b_n, whereas EUMOD* is the same problem where $b_1 > b_2$, $b_{i+2} = b_i \bmod b_{i+1}$, $i = 1, 2,...,n-2$; that is, in the input set for EUMOD*, the values b_3, ..., b_n have been generated as the successive remainders computed by the Euclidean algorithm for b_1 and b_2.

Although MOD* is a P-complete problem [9], we do not know if EUMOD* is in NC or if it is P-complete, so that the works [2], [3], and [13] still left establishing NC-equivalence between Opt-2-ILP and EUGCD as a research challenge. [13] proposed to try to reduce EUMOD* to EUGCD in NC, which would resolve the issue, but this has not worked so far.

The work on the version of [13] submitted for journal publication has brought in an additional coauthor (David Shallcross) and a stronger result, which ended up in the appearance of [14]. This paper combines the work of these two papers.

In this paper, we rely on applying a sequence of appropriate dissections and unimodular transformations of triangles, a method previously used in [16] to count the number of integer points in a polygon. We show that, used correctly, these techniques are powerful enough to arrive at the desired NC-reduction of Opt-2-ILP to EUGCD, which completes the long-awaited proof of NC-equivalence of these two problems.

2. Preliminaries. Hereafter, **Z**, **Q**, and **R**, as usual, denote the sets of integers, rationals, and reals, respectively. For (column) vectors a, b, c, ... from $\mathbf{Z}^2$ and $\mathbf{Q}^2$, we will let (a_1, a_2), (b_1, b_2), (c_1, c_2), ... denote the pairs of their coordinates. $\lfloor x \rfloor$ and $\lceil x \rceil$ denote the two integers closest to a rational x such that $\lfloor x \rfloor \le x \le \lceil x \rceil$.

EUGCD denotes the computational problem which requires, for an input pair a_0, a_1 of positive integers, to compute a sequence of integers a_2, a_3, ..., a_{n+2} such that

$$a_{i+2} = a_i - a_{i+1} \left\lfloor \frac{a_i}{a_{i+1}} \right\rfloor, \ i = 0, 1, ..., n, \qquad a_{n+1} > a_{n+2} = 0. \tag{1}$$

This is intimately related to continued fractions, as we can write

$$\frac{a_0}{a_1} = \left\lfloor \frac{a_0}{a_1} \right\rfloor + \frac{1}{\left\lfloor \frac{a_1}{a_2} \right\rfloor + \frac{1}{\lfloor \frac{a_2}{a_3} \rfloor + \cdots \frac{1}{a_n / a_{n+1}}}}. \tag{2}$$

Note that

$$n = O(\log a_0). \tag{3}$$

We will promiscuously mix representations of triangles by their vertices and by their facets or sides, because either representation can be transformed into the other in NC by performing some basic linear algebra operations.

Define a *unimodular* matrix to be an integer matrix with determinant either 1 or -1. Multiplying a unimodular matrix by an integer vector gives an integer vector, and since the inverse of a unimodular matrix is also unimodular, only an integer vector can multiply a unimodular matrix to give an integer vector.

DEFINITION 1. *An instance of Opt-2-ILP is the following: given $A \in \mathbf{Z}^{m \times 2}$, $v \in \mathbf{Z}^m$, $u = (u_1, u_2)^T \in \mathbf{Z}^2$, find $x = (x_1, x_2)^T \in \mathbf{Z}^2$ such that $Ax \le v$ that maximizes $u^T x = u_1 x_1 + u_2 x_2$, where v^T denotes the transpose of a vector v. That is, less formally, given a polygon P in $\mathbf{R}^2$ as an intersection of half-planes and a linear objective function, find the integer point that maximizes that function over all integer points in the polygon.*

We observe that a unimodular transformation of a triangle transforms a solution of Opt-2-ILP over this triangle to a solution of Opt-2-ILP over its image.

3. NC reduction of Opt-2-ILP to Opt-2-ILP over a triangle of a special form. The next two lemmas are from [13] (compare also [2], [3]); we clarify their proofs.

LEMMA 3.1. *Opt-2-ILP is NC-reducible to Opt-2-ILP over triangles.*

Proof. For an instance of Opt-2-ILP with feasible region P given in the usual manner as the intersection of half-planes, we may, in NC, compute the representation of P as a polygon given by vertices and edges. We will next show a relatively simple (although far from being the most efficient) method for doing this in NC. For every pair of half-planes compute the point where their boundary lines intersect (if they do intersect). For each such point, check whether it lies in each of the remaining half-planes and reject infeasible points. Eliminate duplicates. Finally, declare two points (now vertices) to be adjacent if they both lie on the boundary of the same half-plane. If unbounded P are allowed as input, we can add explicit bounds on the components of any finite optimal solution (see [15, Corollary 17.1c]).

We next triangulate this polygon in NC by taking an arbitrary vertex a, and, for each adjacent pair of vertices b, c, neither equal to a, producing the triangle abc. By

a note above, we can obtain in NC the equations of the sides of any triangle from its vertices. Now we solve the original optimization problem by, in parallel, solving the linear number of optimization problems over the triangles and taking the best of the optima. □

LEMMA 3.2. *Opt-2-ILP over triangles is NC-reducible to solving the following pair of computational problems*: *GCD and Opt-2-ILP over triangles of the form* $T =$*convex hull*(α, β, γ) *where* $\alpha = (\alpha_1, \alpha_2)^T$, $\beta = (\beta_1, \beta_2)^T$, $\gamma = (\gamma_1, \gamma_2)^T$ *are three points in* $\mathbf{Q}^2$ *such that* $\alpha_1 = \beta_1 < \gamma$, $\alpha_2 > \beta_2$, $\alpha_2 > \gamma$, *and* α *is the solution to the linear programming relaxation of the original Opt-2-ILP.*

Proof. We will rely on the proof of Lemma 5.2 of [13], which we will essentially reproduce with slightly changed notation for the triangles and their vertices and with some unnecessary claims removed.

We shall apply (unimodular) linear transformations

$$U \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} ax + by \\ cx + dy \end{pmatrix},$$

where $U = \left(\begin{smallmatrix} u_{11} & u_{12} \\ u_{21} & u_{22} \end{smallmatrix}\right)$ is a 2×2 matrix of the matrix class

$$GL_2(\mathbf{Z}) = \left\{ \begin{pmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{pmatrix} \ : \ g_{11}, g_{12}, g_{21}, g_{22} \in \mathbf{Z} \text{ and } g_{11}g_{22} - g_{21}g_{12} = \pm 1 \right\} .$$

Clearly, every such a transformation (with $U \in GL_2(\mathbf{Z})$) will map all the integral points into integral points; furthermore, if $U \in GL_2(\mathbf{Z})$, then $U^{-1} \in GL_2(\mathbf{Z})$, since

$$(\det\ U)U^{-1} = \begin{pmatrix} u_{22} & -u_{12} \\ -u_{21} & u_{11} \end{pmatrix}$$

and since

$$\det(U^{-1}) = \frac{u_{11}u_{22} - u_{12}u_{21}}{\det\ U} = 1 .$$

Now let A_0, B_0, C_0 denote the vertices of the triangle $T_0 = (A_0B_0C_0)$, which is the feasibility region of the input Opt-2-ILP. We now seek a linear transformation with the matrix $U \in GL_2(\mathbf{Z})$ that transforms the triangle $T_0 = (A_0B_0C_0)$ into a triangle $T = (ABC)$ satisfying the requirements of Lemma 3.2. Without loss of generality, we may assume that A_0 is an optimum for the LP relaxation of the input ILP and that all the angles of the triangle $T_0 = (A_0B_0C_0)$ are less than $\frac{\pi}{2}$ (Figure 2).

Next, we will transform the triangle $T_0 = (A_0B_0C_0)$ into a triangle $T = (ABC)$ so that $A - B$ will lie on the Y-axis and A (the image of A_0) will be the optimum point of correspondingly transformed linear program (see Figure 3). To formalize this stage, we let $A_0 - B_0 = \left(\begin{smallmatrix} z \\ t \end{smallmatrix}\right) = \vec{A_0B_0}$ denote the vector directed from B_0 to A_0, and we will seek a matrix

$$M = \begin{pmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{pmatrix} \in GL_2(\mathbf{Z})$$

such that

$$\begin{pmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{pmatrix} \begin{pmatrix} z \\ t \end{pmatrix} = \begin{pmatrix} 0 \\ v \end{pmatrix};$$

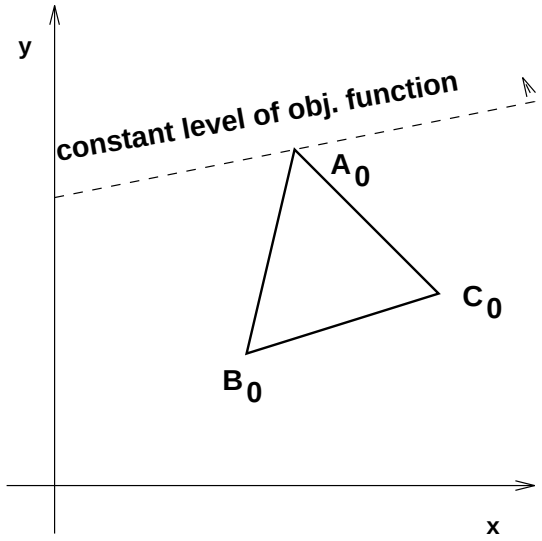

FIG. 2. *An arbitrary triangle* $A_0B_0C_0$.

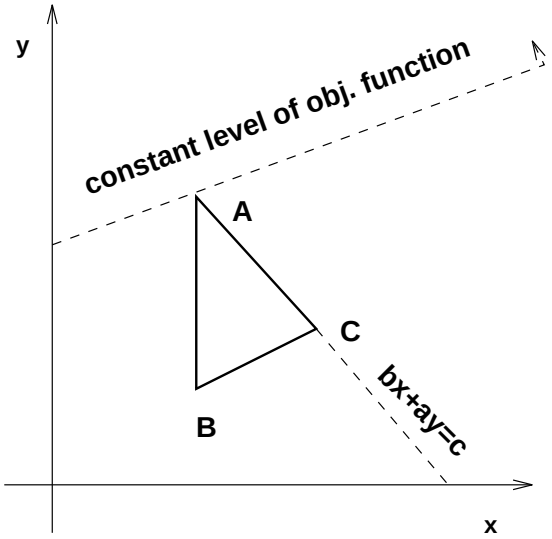

FIG. 3. *Triangle* $A_0B_0C_0$ *shifted to triangle* ABC.

that is,

$$m_{11}z + m_{12}t = 0 \ ,$$

$$m_{21}z + m_{22}t = v \ ,$$

$$m_{11}m_{22} - m_{12}m_{21} = \pm 1 \ .$$

Let $v = gcd(z,t)$. We can compute m_{21}, m_{22} by using the Euclidean algorithm or any algorithm that computes $v = gcd(z,t)$, since (as this can be immediately verified)

$$m_{11} = \frac{t}{v}\ , \quad m_{12} = -\ \frac{z}{v}\ ,$$

$$m_{11}m_{22} - m_{12}m_{21} = \frac{t}{v}\ m_{22} + \frac{z}{v}\ m_{21} = 1\ .$$

Let $T = (ABC)$ denote the resulting triangle.

During the entire transformation process, the image of A_0 (at the end represented by A'') remains the optimum of the correspondingly transformed linear program. Indeed, let $f(x, y) = ux + vy$ denote the objective function; then

$$ux + vy = \begin{pmatrix} u & v \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} u & v \end{pmatrix} M^{-1} M \begin{pmatrix} x \\ y \end{pmatrix},$$

and $M\binom{x}{y} = \binom{x'}{y'}$ is the optimum point in the new coordinate system, which is the image of the original optimum point. The vector $(u \quad v)M^{-1}$ is easy to calculate ($M^{-1} \in GL_2(\mathbf{Z})$ is the inverse of M), and it represents the direction of the constant level of the transformed objective function. Therefore, the optimum point and the optimum value are the transformation invariants.

Now we immediately observe that the requirements of Lemma 3.2 hold, except that we have yet to prove that $a > b$ in the equation for the edge $AC\ :\ bx + ay = c$ (Figure 3). Let k_{AC} denote the slope of AC; then all we need to prove is that $-\ \frac{\pi}{4} < k_{AC} \le 0$. We use the customary notation for the unimodular group of 2×2 matrices:

$$SL_2(\mathbf{Z}) = \left\{ \begin{pmatrix} s_{11} & s_{12} \\ s_{21} & s_{22} \end{pmatrix}\ :\ s_{11}s_{22} - s_{21}s_{12} = 1 \right\}\ .$$

Let $\alpha, \beta \in SL_2(\mathbf{Z})$ denote two such matrices:

$$\alpha = \begin{pmatrix} 1 & 0 \\ -1 & 1 \end{pmatrix}\ , \qquad \beta = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}\ ,$$

where $\alpha^{-1} = \beta$. Denote $\vec{CA} = C - A = \binom{g}{h}$; then

$$\text{(a)} \qquad \alpha \begin{pmatrix} g \\ h \end{pmatrix} = \begin{pmatrix} g \\ h - g \end{pmatrix}\ ,$$

$$\text{(b)} \qquad \beta \begin{pmatrix} g \\ h \end{pmatrix} = \begin{pmatrix} g \\ h + g \end{pmatrix}\ .$$

We can always assume that $g > 0$; otherwise we would have multiplied the resulting vector $\left[\begin{smallmatrix} g \\ h \end{smallmatrix}\right]$ by $\left(\begin{smallmatrix} -1 & 0 \\ 0 & 1 \end{smallmatrix}\right) \in GL_2(\mathbf{Z})$. If $h \ge g$, then we will multiply it by $\alpha^{\lfloor \frac{h}{g} \rfloor}$ and will arrive at the vector $\left[\begin{smallmatrix} g \\ h_1 \end{smallmatrix}\right]$, where $h_1 = h - g\lfloor \frac{h}{g} \rfloor < g$, as is required. Similarly, if $h < 0$, then we will multiply $\left[\begin{smallmatrix} g \\ h \end{smallmatrix}\right]$ by $\beta^{\lfloor \frac{-h}{g} \rfloor}$ and will arrive at $\left[\begin{smallmatrix} g \\ h_2 \end{smallmatrix}\right]$, where $h_2 = h + g\lfloor \frac{-h}{g} \rfloor > 0$. Thus, the requirements of Lemma 3.2 are satisfied for the triangle ABC. □

4. Restriction of the problem to the boundary of the convex hull of all the integer points of a superscribed right triangle. We will use the following definitions.

DEFINITION 2. *Hereafter, a triangle T of the form $\{(x_1, x_2) : ax_1 + bx_2 \le c, x_1 \ge g, x_2 \ge h\}$, for five integers $a > 0$, $b > 0$, c, g, h, will be called a* right triangle.

DEFINITION 3. *For any set S, let δS denote the boundary of the convex hull of the integer points in S.*

To solve Opt-2-ILP over a triangle of Lemma 3.2, we will reduce (in NC) this problem to solving EUGCD and to solving Opt-2-ILP over a few right triangles with integer slopes (see the next sections). To solve the latter problem in NC (see section 6), we will need the following lemma.

LEMMA 4.1. *Let a, b be positive integers, and let c, d, e, f, g be integers such that*

$$T = \{(x_1, x_2) : ax_1 + bx_2 \le c,\ dx_1 + ex_2 \le f,\ x_1 \ge g\}$$

is a nonempty bounded triangle. Let $u \in \mathbf{Z}^2$ be such that maximum over T of $u^T x$ occurs at the vertex $x^* = (g, (c - ag)/b)$. (This is the upper-left-hand corner of T.) Then the maximum over $T \cap \mathbf{Z}^2$ of $u^T x$ occurs among the integer points of $\delta T'$, the boundary of the convex hull H of integer points in the right triangle T', where

$$T' = \{(x_1, x_2) : ax_1 + bx_2 \le c,\ x_1 \ge g,\ x_2 \ge h\},$$

and h is an integer chosen so that $T \subset T'$.

Proof. If x^* defined above was an integer, then $x = x^*$ would optimize $u^T x$ over $T \cap \mathbf{Z}^2$ and would be a vertex on $\delta T'$. Otherwise let $x = z^*$ maximize $u^T x$ over $T \cap \mathbf{Z}^2$. Now suppose that $z^* \notin \delta T'$, but rather $z^* = (z_1^*, z_2^*) \in$ interior H, and we shall obtain a contradiction. (See Figure 4.)

Let K be the open wedge $\{(x_1, x_2) : x_1 < z_1^*,\ az_1^* + bz_2^* < ax_1 + bx_2\}$, and let P be the parallelogram $K \cap T'$. For all $x \in P$, $x_2 > z_2^*$, so P is in fact a subset of T. All points of K (and hence all of P) have a better objective value than z^*. In particular, by choice of z^*, P contains no integer points.

Since x^* is a vertex of T' but not an integer point, $x^* \notin H$. Thus since $z^* \in$ interior H, the diagonal of P from z^* to x^* intersects $\delta T'$ at some point t on an edge between two integer points w and y. Because they are integer points, neither w nor y can lie in P. They are in T', so they cannot lie anywhere else in K, but because the edge between them contains a point of P, one of these, say y, satisfies $y_1 < z_1^*$, $ay_1 + by_2 < az_1^* + bz_2^*$, and the other, w, satisfies $w_1 > z_1^*$, $aw_1 + bw_2 > az_1^* + bz_2^*$. Furthermore, either $w_2 > z_2^*$ or else $y_2 > z_2^*$.

Let $\bar{x} = w + y - z^*$. By the above, and since w and y are in T', we can see that $\bar{x}_1 > g$, and $a\bar{x}_1 + b\bar{x}_2 < c$. We can also see that $\bar{x}_2 > h$, so that $\bar{x} \in T'$. The points w and y are opposite vertices of a parallelogram with integer vertices y, $\bar{x}$, w, and z^* in T'. Since the point t between w and y lies on the boundary of the convex hull of integer points of T', we have the desired contradiction. Thus x^* lies on the boundary of H. ∎

5. Recursive partition of a triangle into unimodular images of right triangles.

LEMMA 5.1. *For integers k, l, $p > 0$, $q > 0$, r, the right triangle*

$$S = \left\{(x_1, x_2) : x_2 \le \frac{r}{q} - \frac{p}{q}x_1,\ x_1 \ge l,\ x_2 \ge k\right\}$$

(if nonempty) is the union (with disjoint interiors) of the right triangle

$$R = \left\{(x_1, x_2) : x_2 \le l' - \left\lfloor \frac{p}{q} \right\rfloor x_1,\ x_1 \ge l,\ x_2 \ge k\right\},$$

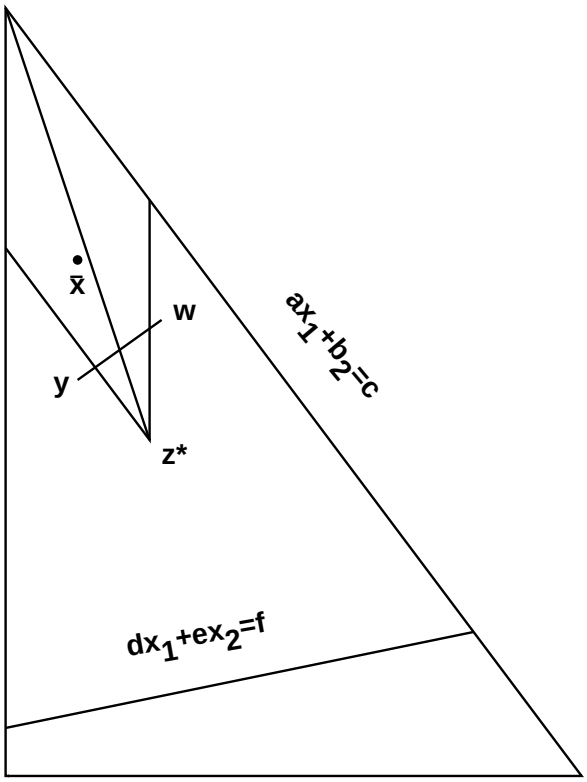

FIG. 4.

sharing the horizontal edge with the triangle S, and the unimodular image $U = MW$ of the triangle

$$W = \left\{ (x_1, x_2) : x_2 \le \frac{r}{q'} - \frac{p'}{q'} x_1,\ x_1 \ge l',\ x_2 \ge k' \right\},$$

where $p' = q$, $q' = p - q\lfloor \frac{p}{q} \rfloor$, $l' = k(\frac{q'}{p}) + \lfloor \frac{p}{q} \rfloor r/p$, $k' = l$, and

$$M = \begin{pmatrix} 0 & 1 \\ 1 & -\left\lfloor \frac{p}{q} \right\rfloor \end{pmatrix}.$$

Proof. We just divide S along the line of slope $-\lfloor \frac{p}{q} \rfloor$ that goes through the point $(\frac{r}{p} - (\frac{q}{p})k, k)$, the lower-right-hand corner of S. (See Figure 5.) This gives $S = R \cup U$ where R is as above, and

$$U = \left\{ (x_1, x_2) : l' - \left\lfloor \frac{p}{q} \right\rfloor x_1 \le x_2 \le \frac{r}{q} - \frac{p}{q} x_1,\ x_1 \ge l \right\}.$$

Now perform an elementary unimodular transformation $(x_1, x_2) \to (x_1, \lfloor \frac{p}{q} \rfloor x_1 + x_2)$ to map U to the triangle

$$V = \left\{ (x_1, x_2) : l' \le x_2 \le \frac{r}{q} - \left(\frac{p}{q} - \left\lfloor \frac{p}{q} \right\rfloor \right) x_1,\ x_1 \ge l \right\}.$$

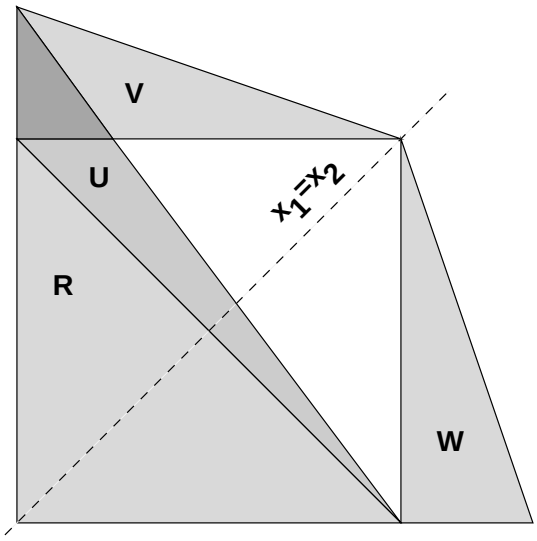

FIG. 5. *Lemma* 5.1 *for* $k = l = 0$.

Notice that we have $(\frac{p}{q}) - \lfloor \frac{p}{q} \rfloor = \frac{q'}{p'}$. Reflecting V over the line $x_1 = x_2$ gives us W above. □

LEMMA 5.2. *Given a sequence* $a_0, a_1, \ldots, a_{n+1}$ *of integers satisfying* (1) *and three rationals* h, k_0, *and* l_0 *such that* $a_0 l_0 + a_1 k_0 \leq h$, *we can compute in NC three sequences of rationals* k_i, l_i, *and unimodular matrices* M_i *such that if*

$$T^* = \{(x_1, x_2) : a_0 x_1 + a_1 x_2 \leq h,\ x_1 \geq l_0,\ x_2 \geq k_0\}$$

and

$$V_i = \left\{(x_1, x_2) : x_2 \leq l_{i+1} - \left\lfloor \frac{a_i}{a_{i+1}} \right\rfloor x_1,\ x_1 \geq l_i,\ x_2 \geq k_i \right\},$$

then

$$T^* = \bigcup_{i=0}^{n} M_i V_i.$$

Proof. First we give recursions for the sequences k_i, l_i, and M_i. Then we will show that these sequences actually meet the requirements of the lemma.

Let

$$M_0 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix},$$

and, for $0 \leq i < n$,

$$k_{i+1} = l_i, \tag{4}$$

$$l_{i+1} = k_i \frac{a_{i+2}}{a_i} + \left\lfloor \frac{a_i}{a_{i+1}} \right\rfloor \frac{h}{a_i}, \tag{5}$$

$$\tilde{M}_i = \begin{pmatrix} 0 & 1 \\ 1 & -\left\lfloor \frac{a_i}{a_{i+1}} \right\rfloor \end{pmatrix}, \tag{6}$$

$$M_{i+1} = M_i \tilde{M}_i. \tag{7}$$

Due to (3)–(7), the maximum magnitude μ of l_i and of the entries of M_i (over all i) satisfies the relation

$$\log \mu = (\log(a_0 + a_1))^{O(1)}. \tag{8}$$

For $0 \le i \le n$, define the triangle

$$S_i = \left\{ (x_1, x_2) : x_2 \le \frac{h}{a_{i+1}} - \frac{a_i}{a_{i+1}} x_1,\ x_1 \ge l_i, x_2 \ge k_i \right\},$$

so $S_0 = T^*$. Applying Lemma 5.1, for $S = S_i$, $p = a_i$, $q = a_{i+1}$, $r = h$, $V = V_i$, $l = l_i$, $k = k_i$, $l' = l_{i+1}$, $W = S_{i+1}$, $p' = a_{i+1}$, $q' = a_{i+2}$, $M = \tilde{M}_i$, we can see that S_i is the union of the two triangles (with disjoint interiors) V_i and $\tilde{M}_i S_{i+1}$, where $S_{n+1} = \emptyset$. By induction, we easily deduce that $T = \bigcup_{i=0}^{n-1} M_i V_i$.

Now, due to (8), and since the values k_i and l_i are defined by a linear recurrence with coefficients depending only on h and the a_i, we may compute these values k_i and l_i in NC using the prefix algorithm on their transformation matrix. Similarly, the prefix algorithm allows one to compute in NC the M_i as the products of the $\tilde{M}_i$. □

Remark. If we let $u_i = (h - a_{i+1}k_i)/a_i$, then it can be verified, for $i \ge 2$, that

$$l_i = \left\lfloor \frac{a_{i-1}}{a_i} \right\rfloor u_{i-1} + l_{i-2},$$

$$u_i = \left\lfloor \frac{a_{i-1}}{a_i} \right\rfloor l_{i-1} + u_{i-2}.$$

Thus, if h, k_0, and l_0 are integers, u_0 and l_1 can be expressed as fractions with denominator a_0, and all later terms can be expressed as fractions with denominator $a_0 a_1$. Likewise, if h, k_0, and l_0 are fractions with denominators d_h, d_k, and d_l, all terms can be expressed as fractions with denominator $d_h d_k d_l a_0 a_1$.

6. Final reduction of Opt-2-ILP to EUGCD.

LEMMA 6.1. *Opt-2-ILP over a triangle of the form T = convex hull(α, β, γ), where α, β, and γ are points in $\mathbf{Q}^2$ such that $\alpha_1 = \beta_1 < \gamma_1$, $\alpha_2 > \beta_2$, and $\alpha_2 \ge \gamma_2$, and with objective vector $u \in \mathbf{Z}^2$ such that $u^T\alpha > u^T\beta$ and $u^T\alpha > u^T\gamma$, can be NC-reduced to EUGCD.*

Proof. Given such a triangle by its vertices, we easily convert it in NC to the representation used in the statement of Lemma 4.1 for T. We then trivially produce the parameters for the triangle T' in the statement of that lemma. Using one call to EUGCD, we produce the sequence $a_0, a_1, \ldots, a_n, a_{n+1}$ according to the relation (1), loosely speaking, the continued fraction representation of the slope a_0/a_1 of T'. We now apply Lemma 5.2 to T', computing in NC the unimodular matrices M_i and right triangles V_i with integer slopes such that $T' = \cup_i M_i V_i$. According to Lemma 4.1, the maximum of $u^T x$ over $x \in T$ occurs in $\delta T'$ and so in $T \cap \delta T'$. The set $\delta T'$ is contained in $\cup_i \delta(M_i V_i)$, so $T \cap \delta T'$ is contained in $T \cap \cup_i \delta(M_i V_i)$, which equals $\cup_i (T \cap M_i \delta V_i)$.

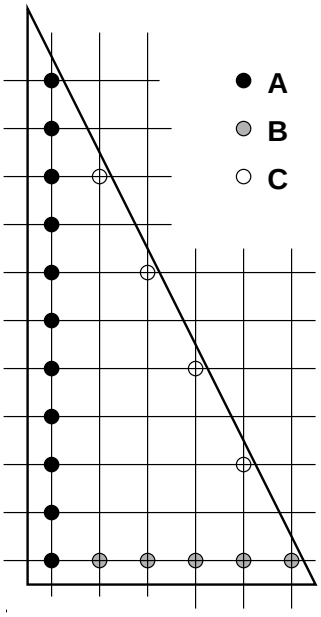

FIG. 6.

Each δV_i can be expressed simply as the union of three pieces A_i, B_i, C_i, each consisting of an arithmetic sequence of integer points, as follows:

$$\delta V_i = A_i \cup B_i \cup C_i,$$

$$A_i = \left\{ (\lceil l_i \rceil, \lceil k_i \rceil + j), \right.$$
$$\left. j = 0, \ldots, \lfloor l_{i+1} \rfloor - \left\lfloor \frac{a_{i+1}}{a_i} \right\rfloor \lceil k_i \rceil - \lceil k_i \rceil \right\},$$

$$B_i = \left\{ (\lceil l_i \rceil + j, \lceil k_i \rceil), \right.$$
$$\left. j = 0, \ldots, \frac{\lfloor l_{i+1} \rfloor - \lceil k_i \rceil}{\left\lfloor \frac{a_{i+1}}{a_i} \right\rfloor} - \lceil l_i \rceil \right\},$$

$$C_i = \left\{ \left(\lceil l_i \rceil + j, \lfloor l_{i+1} \rfloor - \left\lfloor \frac{a_{i+1}}{a_i} \right\rfloor \lceil l_i \rceil - \left\lfloor \frac{a_{i+1}}{a_i} \right\rfloor j \right), \right.$$
$$\left. j = 0, \ldots, \frac{\lfloor l_{i+1} \rfloor - \lceil k_i \rceil}{\left\lfloor \frac{a_{i+1}}{a_i} \right\rfloor} - \lceil l_i \rceil \right\}.$$

(See Figure 6.) Naturally, $M_i \delta V_i$ is the union of $M_i A_i$, $M_i B_i$, and $M_i C_i$. Thus we can maximize $u^T x$ over $x \in T$ by taking the maximum (for $i = 0, 1, ..., n$) of the maxima of $u^T x$ over x in $T \cap M_i A_i$, in $T \cap M_i B_i$, and in $T \cap M_i C_i$. □

THEOREM 6.2. *Opt-2-ILP can be NC-reduced to EUGCD.*

Proof. By Lemma 3.1, Opt-2-ILP can be NC-reduced to Opt-2-ILP over triangles. By Lemma 3.2, Opt-2-ILP over triangles can be NC-reduced to EUGCD plus Opt-2-ILP over triangles of a particular form. Finally, by Lemma 6.1 Opt-2-ILP can be NC-reduced to EUGCD. □

Since [2], [3] give a reduction of EUGCD to Opt-2-ILP, we obtain that Opt-2-ILP and EUGCD are NC-equivalent.

REFERENCES

[1] A. V. AHO, J. E. HOPCROFT, AND J. D. ULLMAN, *The Design and Analysis of Computer Algorithms*, Addison–Wesley, Reading, MA, 1974.

[2] X. DENG, *Mathematical Programming: Complexity and Application*, Ph.D. dissertation, Stanford University, Stanford, CA, 1989.

[3] X. DENG, *On parallel complexity of integer linear programming*, in Proc. ACM Symp. on Parallel Algorithms and Architectures, Association for Computing Machinery, New York, 1989, pp. 110–116.

[4] A. GIBBONS, W. RYTTER, *Efficient Parallel Algorithms*, Cambridge University Press, London, 1988.

[5] R. GREENLAW, H. J. HOOVER, AND W. L. RUZZO, *A compendium of problems complete for P*, Tech. report TR 91-11, Dept. of Computer Science, University of Alberta, Edmonton; Tech. report TR 99-05-01, CS and Engineering Dept., University of Washington, Seattle, 1991.

[6] R. GREENLAW, *Polynomial completeness and parallel computation*, in Synthesis of Parallel Algorithms J. H. Reif, Ed., Morgan-Kaufman Publishers, San Mateo, CA, 1993, pp. 901–954.

[7] R. KANNAN, *A polynomial algorithm for the two-variable integer linear programming problem*, J. Assoc. Comput. Mach., 27 (1980), pp. 118–122.

[8] R. KANNAN, *Improved algorithms for integer linear programming and related lattice problems*, in Proc. 15th Annual ACM Symp. on Theory of Computing, Association for Computing Machinery, New York, 1983, pp. 193–206.

[9] H. H. KARLOFF AND W. L. RUZZO, *The complexity of iterated MOD function*, Inform. Comput., 80 (1989), pp. 193–204.

[10] R. M. KARP AND V. RAMACHANDRAN, *A survey of parallel algorithms for shared-memory machines*, in Handbook of Theoretical Computer Science, North–Holland, Amsterdam, 1990, pp. 869–941.

[11] D. E. KNUTH, *The Art of Computer Programming Vol.* 2: *Seminumerical Algorithms*, Addison–Wesley, Reading, MA, 1981.

[12] H. W. LENSTRA, JR., *Integer linear programming with a fixed number of variables*, Math. Oper. Res., 8 (1983), pp. 538–548.

[13] YU LIN-KRIZ AND V. Y. PAN, *On parallel complexity of integer linear programming, gcd, and the iterated mod function*, in Proc. 3rd Ann. ACM-SIAM Symp. on Discrete Algorithms, 1992, pp. 124–137.

[14] D. F. SHALLCROSS, V. Y. PAN, AND YU LIN-KRIZ, *The NC equivalence of planar integer linear programming and Euclidean GCD*, in Proc. 34th Ann. Symp. on Foundations of Computer Science, IEEE Computer Soc. Press, Los Alamitos, CA, 1993, pp. 557–564.

[15] A. SCHRIJVER, *Theory of Linear and Integer Programming*, John Wiley and Sons, New York, 1986.

[16] L. YA. ZAMANSKII AND V. L. CHERKASSKII, *A formula for finding the number of integer points under a straight line and its application*, Èkonom. i Mat. Metody, 20 (1984), pp. 1132–1138 (in Russian).