The NC Equivalence of Integer Linear Programming and Euclidean GCD

Victor Pan

TR-92-041

December 1992

The NC Equivalence of Integer Linear Programming and Euclidean GCD

David Shallcross * Victor Pan †

December 16, 1992

Summary

We show NC-reduction of integer linear programming with two variables to the evaluation of the remainder sequence arising in the application of the Euclidean algorithm to two positive integers. Due to the previous result of Deng, this implies NC-equivalence of both of these problems, whose membership in NC, as well as P-completeness, remain unresolved open problems.

1 Introduction

Consider k-ILP (the integer linear programming problem with k variables, for a fixed k) and EUGCD (the problem of computing the Euclidean remainder sequence, defined by the Euclidean algorithm for two integers and ending with their greatest common divisor, gcd). Both of these major computational problems belong to the class P, that is, can be solved in polynomial time, specifically, in sequential Boolean time $O(k^{9k} I \log I)$ for k-ILP ([6],[7],[11]) and $O(I^2)$ for EUGCD ([1], [10]), where I denotes the input size (length).

Both of these problems are quite special in the study of parallel computational complexity: k-ILP, for any fixed $k \geq 2$, and EUGCD are among the relatively few major computational problems in P that so far have successfully resisted numerous attempts to prove their P-completeness or their

*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.bitnet, and International Computer Science Institute, Suite 600, 1947 Center St., Berkeley, CA 94704.

membership in NC. (We recall, from [5] chapter 7, [9], and [4], that NC is a class of computational problems whose solution uses $O((\log I)^d)$ time and $O(I^d)$ processors, for any fixed d independent of the input size I. NC-reduction and NC-equivalence are defined as the reduction and the equivalence (defined as the reduction in both directions) of two computational problems to each other under the above restriction on the time and processor bounds. A P-complete problem is one whose solution can be used as an oracle in order to solve in NC any other problem in P.)

k-ILP and EUGCD turn out to be closely related (via NC-reductions and NC-equivalence) to several other major computational problems, such as GCD (computing the gcd of two integers), SGCD (solution of the set equation $m\mathbf{Z} + n\mathbf{Z} = d\mathbf{Z}$), expanding 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 (see [2],[3] and Figure 1). For all these problems, we do not know if any of them is P-complete or if any of them is in NC, and the current research is directed towards establishing NC-reductions among them.

Deng in [2],[3] showed several such reductions, most notably from EUGCD to Opt-2-ILP, the (stronger) optimization version of 2-ILP (see Definition 1). (In Figure 1, we use 2-ILP and k-ILP for the feasibility integer linear programs.) The NC-reduction into the opposite direction, from Opt-2-ILP to EUGCD, turned out to be more elusive. Yu Lin-Kriz and Victor 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 [8], 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.

In the present paper, we instead relied on applying a sequence of appropriate dissections and unimodular transformations of triangles, a method previously used by [14] to count the number of integer points in a polygon. We showed that, used correctly, these techniques are powerful enough to ar-

rive at the desired NC-reduction of Opt-2-ILP to EUGCD, which completed the long awaited proof of NC-equivalence of these two problems.

2 Preliminaries

Hereafter, $\mathbf{Z}$, $\mathbf{Q}$, and $\mathbf{R}$, 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 $\lfloor x \rfloor \leq x \leq \lceil x \rceil$.

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

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

Note that

$$n = O(\log a_0). \tag{2}$$

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: 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 $Ax \leq v$ 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 the proof of the first lemma.

Lemma 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 (though 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 optimum solution (see [12], Corollary 17.1b).

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 the equations of the sides of any triangle from its vertices in NC. 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 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 $\alpha = (\alpha_1, \alpha_2)$, $\beta = (\beta_1, \beta_2)$, $\gamma = (\gamma_1, \gamma_2)$ 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: See [13], Lemma 5.2. □

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 $\{(x_1, x_2) : ax_1 + bx_2 \leq c, x_1 \geq g, x_2 \geq h\}$, 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 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 3 *Let a, b be positive integers, and c, d, e, f, g be integers, such that*

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

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

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

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

Proof: If x^* defined above were 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 2.)

Let K be the open wedge $\{(x_1, x_2) : x_1 < z_1^*,\ az_1^* + bz_2^* < ax_1 + bx_2\}$, and 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 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 4 *For integers k, l, $p > 0$, $q > 0$, r, the right triangle*

$$S = \{(x_1, x_2) : x_2 \leq \frac{r}{q} - \frac{p}{q}x_1,\ x_1 \geq l,\ x_2 \geq k\}$$

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

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

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

$$W = \{(x_1, x_2) : x_2 \leq \frac{r}{q'} - \frac{p'}{q'}x_1,\ x_1 \geq l',\ x_2 \geq k'\}$$

where $p' = q$, $q' = p - q\lfloor p/q \rfloor$, $l' = k(q'/p) + \lfloor p/q \rfloor r/p$, $k' = l$, 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 $-\lfloor p/q \rfloor$ that goes through the point $(r/p - (q/p)k, k)$, the lower right-hand corner of S. (See figure 3.) This gives $S = R \cup U$ where R is as above, and

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

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

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

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

Lemma 5 *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

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

then

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

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 $0 \leq i < n$,

$$k_{i+1} = l_i, \tag{3}$$

$$l_{i+1} = k_i \frac{a_{i+2}}{a_i} + \left\lfloor \frac{a_i}{a_{i+1}} \right\rfloor \frac{h}{a_i}, \tag{4}$$

$$\tilde{M}_i = \begin{pmatrix} 0 & 1 \\ 1 & -\left\lfloor \frac{a_i}{a_{i+1}} \right\rfloor \end{pmatrix}, \tag{5}$$

$$M_{i+1} = M_i \tilde{M}_i. \tag{6}$$

Due to (2)-(6), the maximum magnitude μ of l_i and of the entries of M_i (over all i) satisfies the relation

$$\log \mu = (\log(a_0 + a_1))^{O(1)}. \tag{7}$$

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

$$S_i = \{(x_1, x_2) : x_2 \leq \frac{h}{a_{i+1}} - \frac{a_i}{a_{i+1}} x_1,\ x_1 \geq l_i, x_2 \geq k_i\},$$

so $S_0 = T^*$. Applying Lemma 4, 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 (7) and since the values k_i and the 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 $i \geq 2$, 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_0a_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 *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 $\alpha_2 \geq \gamma_2$, 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 3 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 partial fraction representation (1) of the slope a_0/a_1 of T'. We now apply Lemma 5 to T', computing in NC the unimodular matrices M_i and right triangles T_i with integer slopes, such that $T' = \cup_i M_i T_i$. According to Lemma 3, 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 $\cup_i \delta(M_i T_i)$, so $T \cap \delta T'$ is contained in $T \cap \cup_i \delta(M_i T_i)$, which equals $\cup_i (T \cap M_i \delta T_i)$.

Each δT_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:

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

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

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

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

(See Figure 4.) Naturally, $M_i \delta T_i$ 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 1 *Opt-2-ILP can be NC-reduced to EUGCD.*

Proof: By Lemma 1, Opt-2-ILP can be NC-reduced to Opt-2-ILP over triangles. By Lemma 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 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, J.D. Ullman, *The Design and Analysis of Computer Algorithms*, Addison-Wesley, Reading, 1974.

[2] X. Deng, "Mathematical Programming: Complexity and Application", PhD Dissertation, Stanford University, 1989.

[3] X. Deng, "On parallel complexity of integer linear programming", *Proc. ACM Symp. on Parallel Algorithms and Architectures*, pp 110-116, 1989.

[4] R. Greenlaw, H.J. Hoover, W.L. Ruzzo, "A compendium of problems complete for P" Tech Report TR 91-11, CSD, University of Alberta, Edmondton and TR 99-05-01, CS and Eng. Dept, University of Washington, Seattle, 1991.

[5] A. Gibbons, W. Rytter, *Efficient Parallel Algorithms*, Cambridge University Press, 1988.

[6] R. Kannan, "A polynomial algorithm for the two-variable integer linear programming problem", *J. of ACM*, vol 27, pp. 118-122, 1980.

[7] R. Kannan, "Improved algorithms for integer linear programming and related lattice problems", *Proc. 15th Annual ACM Symp. on Theory of Computing*, pp. 193-206, 1983.

[8] H.H. Karloff, W.L. Ruzzo, "The complexity of iterated MOD function", *Information and Computation*, vol 80, pp. 193-204, 1989.

[9] R.M. Karp, V. Ramachandran, "A survey of parallel algorithms for shared-memory machines", *Handbook of Theoretical Computer Science*, North-Holland, Amsterdam, pp. 869-941, 1990.

[10] D.E. Knuth, *The Art of Computer Programming Vol. 2: Seminumerical Algorithms*, Addison-Wesley, Reading, 1981.

[11] H.W. Lenstra, Jr., "Integer linear programming with a fixed number of variables", *Mathematics of Operations Research*, vol 8, pp 538-548, 1983.

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

[13] Yu Lin-Kriz, V. Pan, "On parallel complexity of integer linear programming, gcd, and the iterated mod function", *Proc. 3rd Ann. ACM-SIAM Symp. on Discrete Algorithms*, 124-137, 1992.

[14] L. Ya. Zamanskii, V. L. Cherkasskii, "A formula for finding the number of integer points under a straight line and its aplication" (in Russian), *Èkonom. i Mat. Metody*, vol 20, 1132-1138 (1984).