Matrix Decomposition Problem is Complete for the Average Case [*]

Yuri Gurevich[†] University of Michigan.

Abstract

The first algebraic average-case complete problem is presented. It is arguably the simplest natural average-case complete problem to date.

1. Introduction

The theory of average-case completeness, pioneered by Leonid Levin [Le], aroused much enthusiasm, but only a few average-case complete problems have been found until now [Le, Gu1, VL, BCGL]. Will that theory remain a curiosity or will it be eventually used to prove average-case intractability of problems arising in applications? We believe in the latter alternative. It used to be very difficult to prove undecidability results, but by now the feel of undecidability is sharpen to such a degree and such a rich assortment of simple undecidable problems is accumulated that proving undecidability is often a routine exercise. You sense that the problem Π in question is undecidable, find an appropriate known undecidable problem Π_0 and reduce Π_0 to Π. A similar development took place in the theory of NP completeness. What we (this is a communal "we") need is to accumulate a wealth of diverse simple average-case complete problems and to develop a feel for average-case intractability.

In this paper, the first algebraic average-case complete problem is presented. In the center of our attention is the *modular group*, i.e., the multiplicative group $\mathrm{SL}_2(Z)$ of two-by-two integer matrices of determinant 1. By default, in this paper, matrices are elements of the modular group. Define the size $|A|$ of a matrix A in some natural way. It may be defined as the number of bits necessary to write the matrix down (using the binary notation for the entries). It may be defined as the log of the maximal absolute value of its entries. (Notice that $|A|$ is the size rather than the determinant of a matrix A.) What is the most natural (sort of *a priori*) probability distribution on matrices? One may require that matrices of the same size have the same probability and that the probability of a random matrix to be of size n equals the default probability of n (which is chosen, somewhat arbitrarily, to be proportional to $n^{-1} \cdot (\log n)^{-2}$ in this paper). This will be our default probability distribution on matrices. Fortunately, average-case completeness proofs are robust with respect to the exact definitions of the size functions and the probability distributions; see Section 2 in this connection.

A matrix pair (B, C) gives rise to an operator $T_{B,C}(X) = BXC$ over $\mathrm{SL}_2(Z)$ which is linear (even though $\mathrm{SL}_2(Z)$ is not closed under addition) in the following sense: If $X = \sum Y_i$ then $T_{B,C}(X) = \sum T_{B,C}(Y_i)$. Andreas Blass [Bl] showed that an operator T on $\mathrm{SL}_2(Z)$ is linear in that sense if and only if there are matrices B and C such that either $T(X) = BXC$ for all X or else $T(X)$ is the transpose of BXC for all X. Moreover, any linear operator T on $\mathrm{SL}_2(Z)$ uniquely extends to a linear operator on all two-by-two integer (or even complex) matrices; this gives rise to the standard representation of T by a four-by-four integer matrix. The two presentations are polynomial-time computable each from the other. Thus, an appropriate matrix pair (B, C) with one additional bit, indicating whether the transpose is applied, is a natural representation of the corre-

[*] 31st Annual Symposium on Foundations of Computer Science, IEEE Computer Society Press, Washington, 1990

[†] Partially supported by NSF grants DCR 85-03275 and CCR 89-04728. Address: Electrical Engineering and Computer Science Department, University of Michigan, Ann Arbor, MI 48109-2122.

sponding linear operator T on $\mathrm{SL}_2(Z)$. Respectively, define the size $|T|$ of T as $|B|+|C|$ and the default probability of T as $\mathbf{P}[B] \times \mathbf{P}[C]$.

Let σ be a positive integer. In the following definition, MD stands for Matrix Decomposition and 1 is the unit two-by-two matrix.

Definition. $\mathrm{MD}(\sigma)$ is the following randomized decision problem:

Instance: A matrix A, a sequence $S = (T_1, \ldots, T_\sigma$ of σ linear operators and a natural number n.

Size: $|(A,S,n)| = |A| + (\sum_{i=1}^{i=\sigma} |T_i|) + n$.

Question: Is there a product $P = T_1 \times \ldots \times T_m$ of $m \leq n$ linear operators $T_i \in S$ such that $A = P(1)$?

Probability: Choose A, the members of S and n independently with respect to the corresponding default probability distributions.

We prove that if σ is not too small then $\mathrm{MD}(\sigma)$ is average-case complete. It remains complete if the size of every matrix in S is bounded by $c \cdot \log |A|$ where c is a not too small positive number. Upper bounds on the minimal appropriate values of σ and c come from the proof of the average-case completeness of Randomized Post Correspondence Problem in [Gu1]. In particular, the bound on σ is something like 3 times the number of instructions of a universal Turing machine. The bounds were irrelevant in [Gu1], and one should be able to improve them if necessary.

One can get rid of n as a separate constituent and assume, for example, that $n = |A|$. And of course one can make σ variable. An appropriate probability distribution may be given by the following experiment: Choose A and σ (and n if it is a separate constituent) independently with respect to the default distributions and then choose σ linear operators independently with respect to the default distribution.

A simpler version of $\mathrm{MD}(\sigma)$ is obtained by making S a sequence of matrices rather than linear operators. The question becomes whether A can be represented as a product of at most n S-matrices. We doubt that this bounded word problem for $\mathrm{SL}_2(Z)$ with the natural probability distribution (so that the constituents of a random instance are chosen independently with respect to the corresponding default probability distributions) is complete for the average case.

2. Basics of the average-case completeness theory

In the theory of NP completeness, it is common to encode problem instances by strings [GJ]. The length of the encoding string becomes the size of the encoded instance. We find it more convenient to define the size of a problem instance directly. In this connection, we introduce size functions.

Definition. A function F from a set U to natural numbers is a *size function* on U if each subset $\{x : F(x) \leq n\}$ of U is finite.

Speaking about a set with a size function, we will always presuppose a fixed representation format for the elements which is appropriate for reading by algorithms. The intention is that the size function reflects roughly the number of bits in a standard representation. The notation $|x|$ will be used to denote the size of an element x.

Remark. It may be convenient to allow noninteger sizes, but the generalization is unnecessary in this paper. Notice also that if the size is exactly the number of binary bits in the fixed representation format then the number of elements of size n is bounded by 2^n.

A function T from a set with a size function to nonnegative reals is *polynomially bounded* if $T(x)$ is bounded by a polynomial of $|x|$.

Definition. A *domain* is a set (the *universe* of the domain) with a size function and a probability distribution. A function T from a domain to nonnegative reals is *polynomial on average* (or *polynomially bounded on average*) if, for some $\varepsilon > 0$, $\sum (Tx)^\varepsilon |x|^{-1} \mathbf{P}[x] < \infty$.

Here the sum is over elements x of positive size and $\mathbf{P}[x]$ abbreviates $\mathbf{P}[\{x\}]$. (Recall that a probability distribution is defined on subsets.) The notion of polynomiality on average is due to Levin [Le]; it is motivated and discussed in [Jo, Gu1, BCGL, Gu2].

Definition. A domain D is *regular* if there exists a polynomial $p(n)$ such that, for every n, the probability of the event $\{x : |x| = n\}$ is either zero or bounded by $1/p(n)$ from below.

All specific domains in this paper will be regular. The following lemma allows the convenience of dealing with instances of one size.

Lemma 2.1 [Gu1]. Let D be a regular domain and $\mathbf{P}_n[x]$ be the conditional probability $\mathbf{P}[x \mid |x| = n]$; if $\mathbf{P}\{x : |x| = n\} = 0$, let $\mathbf{P}_n[x] = 0$. A function T from D to nonnegative reals is polynomial on average if and only if there exists $\varepsilon > 0$ such that $\sum_{|x|=n}(Tx)^{\varepsilon}\mathbf{P}_n[x]$ is bounded by a polynomial of n.

The following lemma justifies the use of more convenient size functions:

Lemma 2.2. Suppose that D_1, D_2 are two domains with the same universe U and the same probability distribution. Let S_i be the size function of D_i and T be a function from U to nonnegative reals.

1. Suppose that S_1 is bounded by a polynomial of S_2 and T is polynomial on average with respect to D_1. Then T is polynomial on average with respect to D_2.
2. If S_1, S_2 are bounded each by a polynomial of the other then T is polynomial on average with respect to D_1 if and only if it is polynomial on average with respect to D_2.

Definition. Let μ and ν be two probability distributions on the same set with a size function. Then ν *dominates* μ if there exists a function g, polynomial on μ-average, such that $\mu(x) \leq g(x) \cdot \nu(x)$.

Lemma 2.3. If ν dominates μ and T is polynomial on ν-average then it is polynomial on μ-average.

A proof of Lemma 2.3 may be found in [Gu1, Section 1]. The term used there is "weak domination"; the term "domination" is restricted in [Le, Gu1] to the case when g is polynomially bounded.

Definition. Let f be a function from a domain A to a domain B such that $p = \mathbf{P}_B(\text{range}(f)) > 0$. We define the notion of domination with respect to f. First suppose that f is injective, i.e., one-to-one. Then $\mathbf{P}_B$ *dominates* $\mathbf{P}_A$ with respect to f if the probability function $\nu(x) = \mathbf{P}_B(fx)/p$ on A dominates $\mathbf{P}_A$. In the general case, $\mathbf{P}_B$ *dominates* $\mathbf{P}_A$ with respect to f if there exists a probability function ν on A such that $\mathbf{P}_B(y)/p = \sum_{fx=y}\nu(x)$ for all $y \in \text{range}(f)$ and ν dominates $\mathbf{P}_A$.

Definition. A function defined on a domain D is AP-*time* or AP-*time computable* (AP stands for "Average-case Polynomial") if it is computable in time polynomial on average with respect to D.

Definition. A function f from a domain A to a domain B is a *reduction* of A to B if it has the following two properties:

Efficiency: f is AP-time.

Domination: $\mathbf{P}_B$ dominates $\mathbf{P}_A$ with respect to f.

A decision problem can be seen as a set with a size function and a distinguished subset (of positive instances). A *randomized decision problem* (shortly r.d.p.) is a domain with a distinguished subset. An r.d.p. is AP if it is decidable in time polynomial on average.

Definition. A randomized decision problem Π_1 *reduces* to a randomized decision problem Π_2 if there exists a reduction f from the domain of Π_1 to the domain of Π_2 with the following additional property:

Correctness: For every instance x of Π_1 of positive probability, x is a positive instance of Π_1 if and only if $f(x)$ is a positive instance of Π_2.

The reducibility relation is transitive, and an r.d.p. Π_1 is AP if it reduces to an r.d.p. Π_2 which is AP [Gu1]. It may be desirable to liberalize the notion of reduction by allowing the reducing machine to flip coins [VL, Gu1, BCGL]. We defer the use of such *randomizing* reductions to the last section.

Definition. RNP (for "Randomized NP") is the class of randomized NP problems Π such that the probability distribution of Π is dominated by a polynomial time computable distribution [Le,Gu1]. An RNP problem Π is RNP *complete* if every RNP problem reduces to Π.

Remark. As a rule, the restriction on probability distributions is satisfied in natural cases, but it may conceivably be violated if instances of Π are generated by an adversary. Ben-David, Chor, Goldreich and Luby studied more general *samplable* distributions [BCGL]. Impagliazzo and Levin showed that, for appropriately generalized reductions, problems complete for RNP remain complete for the more general class [IL]. For simplicity, we stick here to the original restriction.

On a set of nonzero finite cardinality m, the uniform probability distribution assigns the probability $1/m$ to each element. We define a default probability distribution on natural numbers and generalize the notion of uniform distribution to an infinite set with a size function.

Definition. ω is a domain of natural numbers such that $|n| = n$ and the probability of a positive n is proportional to $n^{-1} \cdot (\log n)^{-2}$. A domain D (or, rather, its probability distribution) is *uniform* if elements of the same size have the same probabilities and $\mathbf{P}_D[\{x : \ |x| = n\}] = \mathbf{P}_\omega[n]$.

Definition. BS (or BS1) is the uniform domain of binary strings where the size of a string is its length.

BS will be our main domain of binary strings. We will view BS not only as a domain but also as a monoid (i.e. a semigroup with a unit) of binary strings. For technical reasons, we introduce an alternative domain of binary strings.

Definition. Let R be an order on a set U which gives rise to an order isomorphism ι from (U, R) to the naturally ordered set of natural numbers. The probability distribution $\mathbf{P}_\omega[\iota(x)]$ on U will be said to be *imposed* by R.

Definition. Order binary strings lexicographically (more exactly, first by length and then lexicographically). BS2 is the domain of binary strings where the size of a string is its length and the probability distribution is imposed by the lexicographical order.

Lemma 2.4. The probability distributions of BS1 and BS2 dominate each other.

Definition. The direct product $A \times B$ of domains A and B is the domain of pairs (a, b), where $a \in A$ and $b \in B$, such that $|(a, b)| = |a| + |b|$ and $\mathbf{P}(a, b) = \mathbf{P}_A(a) \times \mathbf{P}_B(b)$.

Definition. Let C be a set of instances of an r.d.p. Π such that $\mathbf{P}[C] > 0$. The *restriction* $\Pi|C$ of Π to C is obtained from Π by the following modification of the probability distribution: $\mathbf{P}_{\Pi|C}[x]$ is the conditional probability $\mathbf{P}_\Pi[x \mid x \in C]$.

3. Positive matrices

Call an arbitrary matrix (i.e. an element of $\mathrm{SL}_2(Z)$) *positive* if it has no negative entries. Positive matrices form a monoid $PM = \mathrm{SL}_2(N)$. In this section, a column is a column of two relatively prime nonnegative integers; for notational simplicity, we view a positive matrix as the pair of its columns. If u is a column, let u_1 be the upper and u_2 the lower components of u. Partially order columns componentwise: $u \leq v$ if $u_1 \leq v_1$ and $u_2 \leq v_2$, and $u < v$ if $u \leq v$ and either $u_1 < v_1$ or $u_2 < v_2$. Define $\max(X)$ to be the maximal entry of a positive matrix X.

Lemma 3.1. Let A and B be the matrices $\begin{matrix} 1 & 0 \\ 1 & 1 \end{matrix}$ and $\begin{matrix} 1 & 1 \\ 0 & 1 \end{matrix}$.

1. $(u, v) \times A = (u + v, v)$, and $(u, v) \times B = (u, u + v)$.

2. If A is a right divisor of a positive matrix (u, v) in PM then $u > v$, and if B is a right divisor of (u, v) in PM then $u < v$.

3. If $m = \max(u, v)$ appears in two or more places of a positive matrix (u, v) then $m = 1$.

Proof. (1) is obvious, and (2) follows from (1).

(3) If m occurs twice in the same row or the same column, then it divides the determinant 1 and therefore $m = 1$. If $v_1 = u_2 = m$ then the determinant cannot be positive. If $u_1 = v_2 = m$ then $1 = u_1v_2 - v_1u_2 \geq m^2 - (m-1)^2 = 2m - 1$ and therefore $m = 1$. QED

The second statement of Lemma 3.1 implies that the monoid generated by the matrices A and B is free. This fact is noticed in [Ei, Chapter VI, Section 12]. The following theorem should be known too, but we don't have an appropriate reference.

Theorem 3.1. The monoid PM is isomorphic to the monoid BC of binary strings. The two indecomposable nonunit elements are the matrices A and B of Lemma 3.1.

Proof. Define $\mathit{weight}(u) = u_1 + u_2$ and $\mathrm{weight}(u, v) = \mathrm{weight}(u) + \mathrm{weight}(v)$. It suffices to prove that every nonunit positive matrix (u, v) is a product of matrices A and B. The proof is an induction on $s = \mathrm{weight}(u, v)$. Since the entries of the main diagonal are not zero, $s \geq 2$.

The case $s \leq 3$ is easy: A and B are the only nonunit matrices of weight ≤ 3. Suppose that $s > 3$. Then $m = \max(u, v) > 1$. Exploiting the symmetry, we may suppose that m appears in u. If $u_1 = m$ then $1/m = (u_1v_2 - v_1u_2)/m > v_2 - u_2$ and therefore $u_2 \geq v_2$. Similarly, if $u_2 = m$ then $u_1 \geq v_1$. Thus, the column $u - v$ has nonnegative entries. The determinant of $(u-v, v)$ equals 1 and therefore $(u - v, v)$ is an element of $\mathrm{SL}_2(N)$. By the induction hypothesis, $(u - v, v)$ is a product of matrices A and B. By Lemma 3.1(1), $(u, v) = (u - v, v) \times A$. QED

Corollary. If a positive matrix (u, v) is not the unit matrix then one of the two columns is greater than the other.

Proof. The fact has been established in the proof of Theorem 3.1. QED

Call the greater column of a nonunit positive matrix *major*; in the case of the unit matrix, call either column *major*. The other column of the matrix will be called *minor*.

Lemma 3.2. The major column and one bit indicating whether it is the first or the second column uniquely define the minor column.

Proof. Without loss of generality, the given matrix (u, v) is not the unit matrix. It follows that both components of the major column are positive. By virtue of symmetry, suppose that u is the major column. We show that the minor column v is the least column such that $v < u$ and $u_1v_2 - u_2v_1 = 1$. Let w be any column such that $w < u$ and $u_1w_2 - u_2w_1 = 1$. Then $u_1(w_2 - v_2) = u_2(w_1 - v_1) = u_1u_2k$ for some k; hence $w_1 = v_1 + ku_1$ and $w_2 = v_2 + ku_2$. If $k < 0$ then either w_1 or w_2 is negative. Hence $k \geq 0$ and therefore $w_1 \geq v_1$, $w_2 \geq v_2$. QED

Let $\mathrm{lh}(n)$ be the length of the binary notation for n.

Definition. We define a domain structure on the monoid PM. It is the uniform domain with the size function $|X| = \mathrm{lh}(\max(X))$. Thus, PM (and also PM1) is the monoid and domain of positive matrices.

Lemma 3.3. The relative probability $\mathbf{P}_{\mathrm{PM}}[X \mid |X| = l] = \Theta(2^{-2l})$.

Proof. Let $g(l) \approx f(l)$ mean that $g(l) = \Theta(f(l))$, i.e., that there exist positive constants c, c' and l_0 such that $cf(l) \leq g(l) \leq c'f(l)$ for all $l \geq l_0$ [Kn2]. It suffices to prove that the number $N(l)$ of positive matrices of size l is $\Theta(2^{2l})$. Recall that $\phi(m)$ is the number of positive integers $n \leq m$ that are prime to m, and that $\Phi(m) = \phi(1) + \ldots + \phi(m) = 3m^2/\pi^2 + O(m \cdot \log m)$ [HW, Theorem 330]. Thus, $N(l) = \sum_{\mathrm{lh}(m)=l} \phi(m) \approx \Phi(2^l - 1) - \Phi(2^{l-1}) \approx \Theta(2^l)$. QED

For technical reasons, we introduce an alternative domain of positive matrices. Fix some linear ordering of the four positions of a two-by-two matrix.

Definition. Order positive matrices first by the maximal entry, then by the (highest) position of the maximal entry and then by the other entry of the major column. PM2 is the domain of positive matrices with the size function $|X| = \mathrm{lh}(\max(X))$ and the probability distribution imposed by the described lexicographical order.

Lemma 3.5. The probability distributions of PM1 and PM2 dominate each other.

By Theorem 3.1, PM is isomorphic to BC. There are exactly two isomorphisms of PM onto BC. One of them takes A to 0 and B to 1 while the other one takes A to 1 and B to 0. Let I be the isomorphism that takes A to 0, and let J be the corresponding isomorphism I^{-1} from BC to PM. Notice that the size of a matrix X may be quite different from the length of the corresponding string $I(X)$. It is easy to see that the isomorphism I is not computable in polynomial time: A matrix $A^n = \begin{matrix} 1 & 0 \\ n & 1 \end{matrix}$ is of size $\mathrm{lh}(n)$ whereas the string $0^n = I(A^n)$ is of length n. We will see in the next section that I is AP. The isomorphism J is P-time computable but $\mathbf{P}_{\mathrm{PM}}$ does not dominate $\mathbf{P}_{\mathrm{BS}}$ with respect to J and thus J fails to reduce BS to PM.

Theorem 3.2 [Bl]. $\mathbf{P}_{\mathrm{PM}}$ does not dominate $\mathbf{P}_{\mathrm{BS}}$ with respect to J.

4. Matrix Correspondence Problem

The direct product PM × PM is a domain and monoid of positive matrix pairs. If S is a set or sequence of positive matrix pairs, let S^n comprise products $P_1 \times \ldots \times P_m$ where $m \leq n$ and each P_i is a member of S. Let σ be a positive integer. In the following definition, MC stands for Matrix Correspondence and 1 is the unit matrix.

Definition. $\mathrm{MC}(\sigma)$ is the r.d.p. with domain $\mathrm{PM} \times [\mathrm{PM} \times \mathrm{PM}]^{\sigma} \times \omega$ where an instance (A, S, n) is positive if and only if there exists a pair (X, Y) in S^n such that $AX = Y$. An instance (A, S, n) of $MC(\sigma)$ is *robust* if either $AX = Y$ for some pair (X, Y) in S^n or else the whole submonoid of $\mathrm{PM} \times \mathrm{PM}$ generated by S has no pair (X, Y) with $AX = Y$. $\mathrm{RMC}(\sigma, c)$ is the restriction of $\mathrm{MC}(\sigma)$ to robust instances (A, S, n) such that the size of every pair in S is bounded by $c \cdot \log |A|$.

Theorem 4.1. Some $\mathrm{RMC}(\sigma, c)$ is RNP-complete.

Remark. Replacing PM with BS in the definition of $\mathrm{RMC}(\sigma, c)$ gives a variant $\mathrm{RPCP}(\sigma, c)$ of the Post Correspondence Problem which is RNP-complete for not too small σ and c [Gu1]. If we ignore probabilities and deal with decision problems only then the isomorphism J of Section 3 gives rise to a polynomial time reduction of $\mathrm{RPCP}(\sigma, c)$ to $\mathrm{MC}(\sigma, c)$. Unfortunately, this reduction fails to have the domination property (see Theorem 3.2) and it is difficult to be altered in any way: The correctness property of the reduction is too closely related to fact that J is an isomorphism. Theorem 4.1 is not proved by a reduction from $\mathrm{RPCP}(\sigma, c)$, but the proof of completeness of $\mathrm{RPCP}(\sigma, c)$ is used in an essential way.

Proof. The rest of this section is devoted to proving Theorem 4.1. We start with recalling the notion of a (simple) continued fraction [HW]. Every rational number r can be uniquely represented by a continued fraction $[a_l, \ldots, a_0]$; the numbers a_i are called partial quotients. If r is an integer then $l = 0$ and $a_0 = r$; otherwise $l > 0$, $a_l = \lfloor r \rfloor$ and $[a_{l-1}, \ldots, a_0]$ is the continued fraction for the rational number s such that $r = a_l + 1/s$. Notice that the denominator in the irreducible fraction for s is smaller than the denominator in the irreducible fraction for r, so the process terminates.

Lemma 4.1. Suppose that x is a nonempty binary string and let $m \leq n$ be the two entries of the major row of $J(x)$. Then $|x|$ equals the sum $s(n, m)$ of the partial quotients in the continued fraction for n/m.

Proof. If $|x| = 1$ then $m = n = 1$ and $s(n, m) = 1 = |x|$. Suppose that $|x| > 1$. By virtue of symmetry, we may suppose that $x = y0$; the other case is similar. Let (i, j) be the major row of $J(y)$. By Lemma 3.1(1), the major row (n, m) of $J(x)$ is $(i + j, j)$. It suffices to prove that if $i \leq j$ then $s(n, m) = s(j, i) + 1$, and if $i \geq j$ then $s(n, m) = s(i, j) + 1$. Since $J(y)$ is not the unit matrix, neither i nor j is zero.

Case $i = j$. $s(i,j) = i/j = 1$ and $s(n,m) = n/m = 2$.

Case $i < j$. $n/m = (i+j)/i = 1 + 1/(j/i)$, hence $s(n,m) = s(i,j) + 1$.

Case $i > j$. Let $[a_l, \ldots, a_0]$ be the continued fraction for i/j. Then $n/m = (i+j)/j = [a_l + 1, \ldots, a_0]$, so that $s(n,m) = s(j,i) + 1$. QED

Lemma 4.2. $|I(X)|$ is polynomial on average with respect to PM.

Proof. Let $s(n,m)$ be as in Lemma 4.1. We use the following strong result of Yao and Knuth [YK]: $\sum_{m=1}^{m=n} s(n,m) = (6n/\pi^2)(\ln n)^2 + O(n(\log n)(\log\log n)^2) = \Theta(n(\log n)^2)$. By Lemma 2.1, we may restrict attention to matrices of a given size $l > 0$. Let $a(X) < b(X)$ form the major row of a matrix X. Then $\sum_{b(X)=n} s(b(X), a(X)) = \Theta(n(\log n)^2)$. By Lemma 4.1, $\sum_{b(X)=n} |I(X)| = \Theta(n(\log n)^2)$ and therefore $\sum_{|X|=l} |I(X)|\Omega(2^l \cdot l^2 2^l)$. Now use Lemma 3.3 to check that the expectation of $|I(X)|$ with respect to the conditional probability $\mathbf{P}_{\mathrm{PM}}[X \mid |X| = l]$ is bounded by a polynomial of l. QED

Definition. Let T be a nondeterministic Turing machine with binary input alphabet. The *bounded halting problem* BH(T) is the randomized decision problem with domain BS$\times\omega$ such that an instance (x,n) is positive if and only if T has a halting computation of length $\le n$ on x. Call an instance (x,n) of BH(T) *robust* if either T has a halting computation of length $\le n$ on x or else T has no halting computation on x at all. RBH(T) is the restriction of BH(T) to robust instances.

Definition. WBS is the domain of binary strings where the size of a binary string x is its length and the probability of x equals $\mathbf{P}_{PM}(Jx)$. Let T be a nondeterministic Turing machine with binary input alphabet. The *weird halting problem* WH(T) and its robust version RWH(T) are similar to BH(T) and RBH(T) except the domain is WBS rather than BS.

Lemma 4.3. For a certain U, RWH(U) is RNP-complete.

Proof. Some RBH(T) is RNP-complete, by Corollary 1 of Theorem 4.1 in [Gu1]. (Actually, a slightly different version of bounded halting problems was considered in [Gu1]. It was supposed there that $n > |x|$ and $\mathbf{P}[(x,n)] \propto n^{-3}2^{|x|}$. However the same proof works. Also, the identity function reduces that older version of every RBH(T) to the new one.) Thus, it suffices to reduce a given RBH(T) to an appropriate RWH(U) .

One may be tempted to take $U = T$ and to use the identity mapping as a reduction. By Theorem 3.2, the identity function fails to do the job. For each binary string x, let $f(x)$ be the the positive matrix X whose lexicographic number equals the lexicographical number of string x. Given a binary string y, the desired U computes $x = f^{-1}(Jy)$, turns itself into T and then runs on input x. Let $g(x)$ be the time that U needs to recover x from $y = I(fx)$ and to turn itself into T. The desired reduction is

$$h(x,n) = (I(f(x)), g(x) + n).$$

Clearly, h has the correctness property. Both functions f and g are P-time computable. Section 3 provides the following recursive algorithm for computing $I(X)$. Suppose that X differs from the unit matrix and $X = (u,v)$. If u is the major column, $w = u - v$ and $Y = (w,v)$ then $I(X) = I(Y)0$, and if v is the major column, $w = v - u$ and $Y = (u,v)$ then $I(X) = I(Y)1$. The computation time of that algorithm is proportional to $|I(X)|$. By Lemma 4.2, $I(X)$ is AP-time. The composition of a P-time function $X = f(x)$ and an AP-time function $I(X)$ is an AP-time function $I(fx)$ [Gu1, Lemma 1.2]. Thus, h is AP-time.

It remains to check that h has the domination property, namely, that $\mathbf{P}_{\mathrm{BS}}(x) \times \mathbf{P}_\omega(n)$ is dominated by $\mathbf{P}_{\mathrm{PM}}(f(x)) \times \mathbf{P}_\omega(gx+n)$. Since $gx+n$ is bounded by a polynomial of $|x|+n$, there exists a polynomial p_1 such that $p_1(|x|+n) \cdot \mathbf{P}_\omega(gx+n) \ge 1$. Hence it suffices to prove that $\mathbf{P}_{\mathrm{BS}}$ is dominated by $\mathbf{P}_{\mathrm{PM}}$ with respect to f. By Lemma 2.4, $\mathbf{P}_{\mathrm{BS}}$ is dominated by $\mathbf{P}_{\mathrm{BS2}}$. In the obvious way, the latter is dominated by $\mathbf{P}_{\mathrm{PM2}}$ with respect to f. By Lemma 3.4, $\mathbf{P}_{\mathrm{PM2}}$ is dominated by $\mathbf{P}_{\mathrm{PM}}$. QED

Fix a Turing machine U witnessing Lemma 4.3. We will reduce RWH(U) to RMC(σ, c) for appropriate σ and c. The variant RPCP(σ, c) of the Post Correspondence Problem was defined in a remark above. According [Gu1, Section 5], there exists a P-time reduction

$$F(x, n) = (xx', K(x), p(n))$$

of RBH(U) to some RPCP(σ, c) where $|x'| \leq c \cdot \log|A|$. Extend the isomorphism J to sequences of pairs of binary strings. The function

$$G(x, n) = (J(xx'), J(K(x)), p(n))$$

is the desired reduction of RWH(U) to RMC(σ, c). Clearly, G has the correctness and the efficiency properties. Ignoring factors bounded by a polynomial of $|x| + n$ from above and by an inverse polynomial of $|x| + n$ from below, we have:

$$\mathbf{P}_{\mathrm{RMC}(\sigma,c)}[G(x, n)] = \mathbf{P}_{\mathrm{PM}}[J(x)] = \mathbf{P}_{\mathrm{RWH}}(x, n).$$

Theorem 4.1 is proved.

5. The modular group

In this section, a column is a column of two relatively prime (not necessarily positive) integers, and a matrix (i.e. an element of $\mathrm{SL}_2(Z)$) is seen as the pair of its columns. Call a matrix or a column positive (resp. negative) if all its entries are non-negative (resp. non-positive). If u is a column then u_1, u_2 are the upper and the lower entries of u, and $|u|$ is the positive column v such that $v_i = |u_i|$. Positive columns are ordered componentwise, like in Section 3. If u is a column then $\max(u) = \max(|u_1|, |u_2|)$. Any component of a column u with the absolute value $\max(u)$ is *major*, and the other component is *minor*. If X is a matrix (u, v) then $\max(X) = \max(\max(u), \max(v))$. Any entry of a matrix X with the absolute value $\max(X)$ is the *major* entry. If u, v are the two columns of a matrix X and $|u| > |v|$ then u is the *major* column and v is the *minor*; in the case of the unit matrix, both columns are *major* and both are *minor*.

Lemma 5.1. For every matrix $X = (u, v)$,

1. It is impossible that one of the numbers $u_1 v_2$, $u_2 v_1$ is positive and the other is negative. If they are both positive then $|u_1 v_2| - |u_2 v_1| = 1$, and if they are both negative then $|u_2 v_1| - |u_1 v_2| = 1$.
2. If X is not the unit matrix then either $(|u|) > (|v|)$ or $(|u|) < (|v|)$.

Lemma 5.2. If $\max(u, v) > 1$ then (u, v) has only one major entry.

Lemma 5.3. For every two matrices (u, v) and (u, v'), there exists an integer k, such that $v' = v + ku$.

Lemma 5.4. Let $X = (u, v)$ be any matrix with $\max(X) > 1$. If u (resp. v) is the major column of X then there exists exactly one additional matrix of the form (u, v') (resp. (u', v)) where the column v' (resp. u') is minor. Moreover, $v' = v \pm u$ (resp. $u' = u \pm v$). If the major column is positive or negative then one of the two possible minor columns is positive and the other one is negative.

Proof. It suffices to consider the case when u is the major column because if (u, v) is a counterexample with a major column on the right then $(-v, u)$ is a counterexample with the major column on the left. Further, it suffices to consider the case when the major entry is positive because if (u, v) is a counterexample with a negative major entry then $(-u, -v)$ is a counterexample with a positive major entry. Let u_i be the major entry of u and (u, v') be another matrix with major column u. By Lemma 5.3, $v' = v + ku$ for some k. Since $u_i > 1$, $v_i \neq 0$. If $v_i > 0$ then $k = -1$, and if $v_i < 0$ then $k = 1$. Notice also that if $v_i 0 >$ resp. $v_i < 0$) then indeed u is the major column of the matrix $(u, v - u)$ (resp. $(u, v + u)$). Now suppose that u is positive. Obviously, $u_1 > 0$ and $u_2 > 0$. By Lemma 5.1(1), v is either negative or positive. If v is positive (resp. negative) then v' is negative (resp. positive). QED

Definition. MG is the modular group and the uniform domain with the size of X equal to $\mathrm{lh}(\max(X))$.

Lemma 5.5. Let X be a random matrix in MG with $\max(X) > 1$. The probability that X is positive is 1/8, and the probability that X is the inverse of a positive matrix is 1/8 as well.

Proof. It suffices to prove the lemma for a fixed value of $\max(X)$. Fix $m > 1$ and let S_0 be the collection of matrices X with $\max(X) = m$. The inverse of a matrix $\begin{matrix} a & c \\ b & d \end{matrix}$ is the matrix $\begin{matrix} d & -c \\ -b & a \end{matrix}$; thus $\max(X^{-1}) = \max(X)$ and S_0 is closed under inversion. It follows, that the number of positive matrices in S_0 equals the number of the inverses of positive matrices. Hence it suffices to prove only the first statement of the lemma.

Let S_1 be the collection of S_0 matrices X such that the major entry of X is positive. For every (u, v) in S_0, exactly one of the two matrices (u, v), $(-u, -v)$ belongs to S_1. It remains to prove that the probability of a random S_1 matrix to be positive is 1/4.

Since the major entry of an S_1 matrix exceeds 1, the minor component of the major column is not zero. Let S_2 be the collection of S_1 matrices such that the minor component of the major column is positive. For every S_1 matrix X, let X' is the result of multiplying the diagonal of X, which contains the minor component of the major row, by -1. Exactly one of the two matrices X, X' belongs to S_2. It follows that S_2 contains exactly one half of the elements of S_1. It remains to prove that the probability of a random S_2 matrix to be positive is 1/2. Now use the previous lemma. QED

Let σ and c witness Theorem 4.1 and let LO be the domain MG×MG×Bool where Bool is domain with universe $\{0, 1\}$. We view elements $(B, C, 0)$ and $(B, C, 1)$ of LO as linear operators $X \rightsquigarrow BXC$ and $X \rightsquigarrow (BXC)^t$ respectively; $(BXC)^t$ is the transpose of BXC. Thus LO is not only a domain but also a monoid. Recall that, as it was explained in the Introduction, every linear operator over the modular group belongs to LO. If S is a sequence of linear operators, let S^n be the set of products $T_m \dots T_1$ where $m \le n$ and each $T_i \in S$. We restate the definition of Matrix Decomposition Problem. Let σ be a positive integer.

Definition. MD(σ) is the r.d.p. with domain MG × (LO)^s × ω such that an instance (A, S, n) is positive if and only if there exists $P \in S$ with $A = P(1)$. Here 1 is the unit matrix. Let MD(σ, c) be the restriction of MD(σ) to instances (A, S, n) where the size of every matrix in S is bounded by $c \cdot \log |A|$.

Theorem 5.1. Some MD(σ, c) is RNP complete.

Proof. We reduce MC(σ, c) to MD(σ, c). If S is a sequence of positive matrix pairs, let S' be the result of replacing each pair (B, C) in S with the triple $(C, B^{-1}, 0)$. The desired reduction is $f(A, S, n) = (A, S', n)$. To check the correctness property, note that $A \cdot B_1 \cdot \ldots \cdot B_m = C_1 \cdot \ldots \cdot C_m$ if and only if $A = C_1 \cdot \ldots \cdot C_m B_m^{-1} \cdot \ldots \cdot B_1^{-1}$.

Obviously, f is P-time computable. By Lemma 5.5, f has the domination property. QED

6. Randomizing reductions

Let σ and c witness Theorem 5.1 and let MD$'(\sigma, c)$ be the alteration of MD(σ, c) with the domain MG × (LO)$^\sigma$ such that an instance (A, S) of MD$'(\sigma, c)$ is positive if and only if $(A, S, |A|)$ is a positive instance of MD(σ, c). MD$'(\sigma, c)$ is flat in the sense of [Gu1] where it is proved that no flat r.d.p. can be RNP-complete with respect to deterministic reductions of Section 2 unless deterministic exponential space equals nondeterministic exponential space. To prove the completeness of MD$'(\sigma, c)$, we use randomizing reductions [VL]. A simple way to introduce them was indicated in [Gu1].

Definition. Let Π be a randomized decision problem with a domain D, and let p be a P-time computable function from D to natural numbers. The p-dilation Π_p of Π is the following randomized decision problem E:

Instance: A pair (x, y) where $x \in D$ and y is a binary string of length $p(|x|)$.

Size: $|(x, y)| = |x| + |y|$.

Question: Is x a positive instance of Π ?

Probability: $\mathbf{P}_E(x, y) = \mathbf{P}_D(x) \cdot 2^{-|y|}$.

Dilation allows to generalize deterministic reductions of Section 2.

Definition. A *randomizing reduction* of r.d.p. Π_1 to an r.d.p. Π_2 is a deterministic reduction of some dilation of Π_1 to Π_2.

Theorem 6.1. $\mathrm{MD}'(\sigma)$ and $\mathrm{MD}'(\sigma, c)$ are RNP-complete with respect to randomizing reductions.

Remark. Instead of setting $n = |A|$, one can set $n = \pi(|A|)$ where π is any P-time computable nondecreasing function such that the inverse function $\pi^{-1}(j) = \min_i[\pi(i) \geq j]$ is polynomially bounded; see [Gu1, Section 9] in this connection.

Acknowledgement. We are grateful to George Bergman for his comments on Theorem 3.1 and to Kevin Compton for number-theoretic references. We are especially grateful to Andreas Blass for being so generous with his time and to Leonid Levin for urging us to find an algebraic problem complete for the average case.

References

[BCGL] Shai Ben-David, Benny Chor, Oded Goldreich and Michael Luby, "On the Theory of Average Case Complexity", 21st Annual ACM Symposium on Theory of Computing, ACM, 1989, 204–216.

[Bl] Andreas Blass, Private communication.

[Ei] Samuel Eilenberg, "Automata, Languages, and Machines", Vols. A and B, Academic Press, NY & London, 1974 and 1976, xvi+451pp. and xiii+387 pp.

[Gu1] Yuri Gurevich, "Average Case Complexity", J. Computer and System Sciences (a special issue on FOCS'87,) to appear.

[Gu2] Yuri Gurevich, "The Challenger-Solver Game", Bulletin of Europ. Assoc. for Theor. Comp. Sci, Oct. 1989.

[GJ] Michael R. Garey and David S. Johnson, "Computers and Intractability: A Guide to the Theory of NP-Completeness", Freeman, New York, 1979.

[HW] G. H. Hardy and E. M. Wright, "An introduction to the theory of numbers", Oxford University Press, 5th edition, 1988 printing.

[IL] Russel Impagliazzo and Leonid A. Levin, "No Better Ways to Generate Hard NP Instances than Picking Uniformly at Random", 31st Annual Symposium on Foundations of Computer Science, IEEE Computer Society Press, 1990.

[Jo] David S. Johnson, "The NP-Completeness Column", Journal of Algorithms 5 (1984), 284-299.

[Kn] Donald E. Knuth, "The Art of Computer Programming", Vol. 2, 2nd edition, Addison-Wesley, Reading, Massachusetts, 1973.

[Kn2] Donald E. Knuth, "Big Omicron and Big Omega and Big Theta", SIGACT News, Apr.–June, 1976, 18–24.

[Le] Leonid A. Levin, "Average Case Complete Problems", SIAM Journal of Computing, 1986.

[VL] Ramarathnam Venkatesan and Leonid Levin, " Random Instances of a Graph Coloring Problem are Hard", 20th Symp. on Theory of Computing, ACM, 1988.

[YK] Andrew C. Yao and Donald E. Knuth, "Analysis of the subtractive algorithm for greatest common divisors", Proc. Nat. Acad. Sci USA 72:12 (1975), 4720–4722.