MATHEMATICS OF COMPUTATION
Volume 75, Number 254, Pages 983–996
S 0025-5718(06)01826-6
Article electronically published on January 23, 2006

COMPLEXITY OF INVERTING THE EULER FUNCTION

SCOTT CONTINI, ERNIE CROOT, AND IGOR E. SHPARLINSKI

Abstract. Given an integer n, how hard is it to find the set of all integers m such that $\varphi(m) = n$, where φ is the Euler totient function? We present a certain basic algorithm which, given the prime number factorization of n, in polynomial time "on average" (that is, $(\log n)^{O(1)}$), finds the set of all such solutions m. In fact, in the worst case this set of solutions is exponential in $\log n$, and so cannot be constructed by a polynomial time algorithm. In the opposite direction, we show, under a widely accepted number theoretic conjecture, that the Partition Problem, an **NP**-complete problem, can be reduced in polynomial (in the input size) time to the problem of deciding whether $\varphi(m) = n$ has a solution, for polynomially (in the input size of the Partition Problem) many values of n (where the prime factorizations of these n are given). What this means is that the problem of deciding whether there even exists a solution m to $\varphi(m) = n$, let alone finding any or all such solutions, is very likely to be intractable. Finally, we establish close links between the problem of inverting the Euler function and the integer factorization problem.

1. Introduction

In this paper we study the complexity of inverting the *Euler function* $\varphi(m)$, which, as usual, for an integer $m \geq 1$, is defined by

$$\varphi(m) = \#(\mathbb{Z}/m\mathbb{Z})^{\times} = \prod_{p^{\alpha} \parallel m} p^{\alpha-1}(p-1).$$

Miller [19] showed that the problem of computing the Euler function $\varphi(n)$ is of equivalent difficulty to the problem of factoring n.[1] Here we concentrate on the dual question of inverting the Euler function, and we consider three types of problems of increasing difficulty:

Problem A. Given n, and given the prime factorization of n, determine whether there exists an integer m satisfying $\varphi(m) = n$.

Problem B. Given n, and given the prime factorization of n, determine the set $\Psi(n)$ of all solutions $m \geq 1$ to $\varphi(m) = n$.

Problem C. Given only n (and not its prime factorization), determine the set $\Psi(n)$ of all solutions $m \geq 1$ to $\varphi(m) = n$.

Received by the editor December 6, 2004 and, in revised form, April 26, 2005.
2000 *Mathematics Subject Classification.* Primary 11A51, 11Y16, 68Q17, 68Q25.
Key words and phrases. Euler function, integer factorisation, Partition problem, NP-completeness.

[1] The algorithm of [19] is not deterministic. Given the prime factorization of n, one can compute $\varphi(n)$ deterministically; however, Miller showed that given $\varphi(n)$, a certain randomized algorithm can factor n in t steps with probability $1 - \vartheta^t$, for a certain $\vartheta \in (0, 1)$, where each "step" requires only $(\log n)^{O(1)}$ bit operations.

We design an algorithm which solves PROBLEM B in exponential time in $\log n$ in the worst case, and in polynomial time for "almost all" n. An algorithm based on similar ideas is outlined in [8], [20], and [18], however no complexity analysis has been given. Because for infinitely many n the cardinality of $\Psi(n)$ is exponentially large, any algorithm for solving either PROBLEM B or PROBLEM C *must* run in time at least exponential in $\log n$ in the worst case. Indeed, P. Erdős [12] and C. Pomerance [22, Theorem 4.6] show that for infinitely many n,

$$\#\Psi(n) \geq n^{\gamma+o(1)},$$

where $\gamma > 0$ is any constant such that for any sufficiently large X there are at least $X^{1+o(1)}$ primes $p \leq X$ such that all prime divisors of $p-1$ are less than $X^{1-\gamma}$ (see also [21]). By Theorem 1 of [3] one can take $\gamma = 0.7039$.

A natural question is whether PROBLEM A is any easier; obviously, PROBLEM A is no harder than either PROBLEM B or PROBLEM C. We prove in Section 3 that, assuming a certain strong form of the famous *Hardy–Littlewood* prime k-tuplet conjecture (in the case $k = 2$), there is a one-to-many polynomial time reduction from the PARTITION PROBLEM, an **NP**-complete problem, to the question of whether $\varphi(m) = n$.[2] This shows that PROBLEM A likely cannot be solved in time polynomial in $\log n$ if one believes the famous conjecture that $\mathbf{P} \neq \mathbf{NP}$. Although at the present time the Hardy–Littlewood conjecture is out of reach, there are a number of results in this direction which leave little doubt that the conjecture is correct; for example, see [11] for the original formulation and [4] for more recent advances.

Furthermore, in Section 5 we show that if we could solve PROBLEM C in time polynomial in $\#\Psi(n) + \tau(n) + \log n$ (actually we show this under a slightly weaker condition), then we can factor almost all (relative density 1) integers n that are the product of two primes by a certain randomized algorithm whose expected running time is polynomial in $\log n$. This result seems weaker than that of Section 3 but is based on different assumptions and has an unconditional form. Assuming the Extended Riemann Hypothesis, however, we show that if we have such an algorithm for PROBLEM C, then we can factor *all* integers n that are the product of two primes in polynomial time.

Many aspects of the Euler function, such as growth, distribution in arithmetic progressions and in other special sets of elements, and many other similar questions, have extensively been studied in the literature; see [5, 6, 10, 13, 14, 15, 21, 22] and references therein. Nevertheless, the questions considered here seem to be new and have never been studied.

2. NOTATION

We use $\omega(m)$ and $\tau(m)$ to denote the total number of distinct prime and positive integer divisors of a positive integer m, respectively (we also define $\omega(1) = 0$, $\tau(1) = 1$).

We also use the Vinogradov symbols $\gg$, $\ll$ as well as the Landau symbols O and o with their regular meanings (we recall that $U \ll V$ and $U = O(V)$

[2]That is, for each instance of the PARTITION PROBLEM, we construct a series of numbers $n_1, ..., n_K$, such that the PARTITION PROBLEM has a solution if and only if one of these numbers is a totient. Furthermore, one cannot apply our algorithm which solves PROBLEM B to easily decide whether one of $n_1, ..., n_K$ is a totient (in the hopes of solving almost all instances of the PARTITION PROBLEM quickly), because each n_i has many divisors. Therefore it turns out that our algorithm for solving PROBLEM B takes a long time to find $\Psi(n_i)$, let alone to decide whether n_i is a totient.

are both equivalent to the inequality $|U| \le cV$ with some constant $c > 0$). The implied constants in the symbols O, $\gg$ and $\ll$ are always absolute unless indicated otherwise.

3. **NP**-hardness of Problem A

3.1. **Totient testing and the Partition Problem.** In this section we prove that the problem of deciding whether a given set of integers $\mathcal{S}$ contains a totient, where the prime factorizations of these numbers are also given, is **NP**-complete, if we assume a certain strong form of the *Hardy–Littlewood* prime k-tuplet conjecture. This result shows that if we had a polynomial time algorithm for solving Problem A, then we could solve **NP**-complete problems in polynomial time (assuming the Hardy–Littlewood conjecture).

We note that we need the Hardy–Littlewood conjecture in a slightly stronger uniform formulation than that which usually appears in the literature. Namely, we assume:

Conjecture 3.1. *There exists an integer $A > 0$ such that the following holds: Suppose that $(M_1x + a_1)(M_2x + a_2)$ has no fixed prime divisors as x runs through the integers, and that $M_1, M_2 > 0$, and $0 \le a_i < M_i$ for $i = 1$ and 2. Then, there exists an integer $x < \log^A(M_1M_2 + 1)$ such that both $M_1x + a_1$ and $M_2x + a_2$ are prime.*

We refer to [4] for several results in the direction of Conjecture 3.1 "on average". It is certainly an interesting open question whether such results can be used to obtain an unconditional proof of **NP**-completeness of Problem A.

We now focus our attention on the following NP-complete problem:

> Partition Problem: Given a sequence of $2k$ integers $x_1, \dots, x_{2k}$, where k is odd,[3] such that $S = x_1 + \cdots + x_{2k}$ is even, decide whether there exist k of them whose sum is $S/2$.

Assuming Conjecture 3.1, we give a polynomial time reduction of the Partition Problem to the problem of deciding whether there exists an integer m satisfying $\varphi(m) = n$, for a certain small set of values of n. More precisely, we prove the following theorem:

Theorem 3.2. *Suppose that B bits are required to describe the input $x_1, \dots, x_{2k}$ for the* Partition Problem. *Then, assuming Conjecture* 3.1*, we construct in polynomial time a set of $K = (Bk)^{O(1)}$ integers $n_1, \dots, n_K$ such that there exists a k-element subset of $\{x_1, \dots, x_{2k}\}$ summing to $S/2$ if and only if for some $i = 1, \dots, K$ we have that $\varphi(m) = n_i$ has a solution.*

In order to prove this theorem, we require the following proposition, proved in Section 3.2:

Proposition 3.3. *Given $x_1, \dots, x_{2k}$, we can construct in polynomial time a modulus M and a series of congruence classes $a_i \pmod M$ with $(a_i, M) = 1$, such that if $N_1, \dots, N_{2k}$ are any numbers satisfying*

$$N_i \equiv a_i \pmod M,$$

[3] We do not actually need the condition k odd, but it makes our proof simpler to include it. Also, we note that if we have a collection of integers for which this does not hold, then we can make it hold by adding two zeros to the set, thereby enlarging the set of $2k + 2$ integers.

and if $\{i_1, \dots, i_\ell\} \subset \{1, 2, \dots, 2k\}$*, with* $\ell \le k$*, then*

$$\gcd(2N_{i_1} \cdots N_{i_\ell} + 1, M) = 1 \quad \Longleftrightarrow \quad \ell = k, \text{ and } x_{i_1} + \cdots + x_{i_\ell} = S/2. \tag{1}$$

Moreover, we have that

$$N_i - 1 \nmid 4N_1 \cdots N_{2k}, \qquad i = 1, 2, \dots, 2k, \tag{2}$$

and

$$\gcd(4N_1 \cdots N_{2k} + 1, M) > 1 \quad \text{and} \quad \gcd(2N_1 \cdots N_{2k} + 1, M) > 1. \tag{3}$$

Proof of Theorem 3.2. Suppose that $x_1, \dots, x_{2k}$ are given, and suppose we have a set of primes $p_1, \dots, p_{2k}$ such that $p_i \equiv a_i \pmod M$. Then, if there is a solution to

$$\varphi(m) = 4p_1 \cdots p_{2k},$$

the integer m must be of the form

$$m = P_1 P_2 \quad \text{or} \quad m = 2P_1 P_2,$$

where P_1 and P_2 are primes of the special form

$$P_1 = 2q_1 \cdots q_k + 1 \quad \text{and} \quad P_2 = 2r_1 \cdots r_k + 1,$$

where $\{q_1, \dots, q_k\}$ and $\{r_1, \dots, r_k\}$ partition $\{p_1, \dots, p_{2k}\}$.

To see that this is the case, we first note that m cannot be a product of more than 2 distinct odd primes, since if it were, we would have that $8|\varphi(m)$. Also, m cannot be divisible by the square of an odd prime p, since it would imply that $p - 1|\varphi(m)$, which would violate (2).

So, m is of the form P, $2P$, $4P$ (P is a prime), P_1P_2, or $2P_1P_2$. Now, if $m = 2P$ or P, then P must be of the form

$$P = 4p_1 \cdots p_{2k} + 1.$$

However, from condition (3), we must have that $(P, M) > 1$, which would imply that P cannot be prime, and so m cannot be of the form P or $2P$. Finally, if $m = 4P$, then we must have that

$$P = 2p_1 \cdots p_{2k} + 1;$$

and again, from condition (3), we must have that $(P, M) > 1$, so m cannot be of the form $4P$. Thus, as claimed, m must be of the form $2P_1P_2$ or P_1P_2.

Now, these odd primes P_1 and P_2 must be of the form

$$P_1 = 2q_1 \cdots q_\ell + 1, \qquad P_2 = 2q_{\ell+1} \cdots q_{2k} + 1,$$

where $\{q_1, \dots, q_{2k}\} = \{p_1, \dots, p_{2k}\}$. If $\ell < k$, then by (1) we have that

$$\gcd(2q_1 \cdots q_\ell + 1, \ M) > 1,$$

and so P_1 could not be prime. Thus, $\ell = k$.

Now suppose that there is a subset $\{x_{i_1}, \dots, x_{i_k}\} \subseteq \{x_1, \dots, x_{2k}\}$ summing to $S/2$, and let $\{x_{j_1}, \dots, x_{j_k}\}$ be its complement. Let $\ell = 2, 3, \dots, k+2$ be arbitrary, and suppose we were lucky and have that $1 \in \{i_1, \dots, i_k\}$ and $\ell \in \{j_1, \dots, j_k\}$; certainly, for one of these values of ℓ we have that either this holds, or else that $1 \in \{j_1, \dots, j_k\}$ and $\ell \in \{i_1, \dots, i_k\}$. Let $\{t_1, \dots, t_{2k-2}\} = \{1, 2, \dots, 2k\} \setminus \{1, \ell\}$. Then, by the Hardy-Littlewood conjecture, we know that we can pick values for $t_1, \dots, t_{2k-2} < (Bk)^{O(1)}$ such that the numbers $a_i + Mt_i$ are all prime (the numbers a_i are coprime to M); moreover, we can pick such values in time $(Bk)^{O(1)}$, by picking t_1 first, then t_2, then t_3, and so on.

Now, we consider the polynomials

$$F(x) = 2(a_1 + Mx) \prod_{\substack{u \in \{i_1,\dots,i_k\} \\ u \neq 1}} (a_u + Mt_u) + 1$$

and

$$G(y) = 2(a_\ell + My) \prod_{\substack{u \in \{j_1,\dots,j_k\} \\ u \neq \ell}} (a_u + Mt_u) + 1.$$

By (1) these polynomials are coprime to M, and so have no fixed prime divisors; in fact, $(a_1+Mx)F(x)$ and $(a_\ell+My)G(y)$ have no fixed prime divisors. So, assuming the Hardy-Littlewood conjecture, if we run through the values of $x, y < (Bk)^{O(1)}$ that make $a_1 + Mx$ and $a_\ell + My$ both prime, then among these values of x and y, there must be a choice which makes all of

$$a_1 + Mx, \quad a_\ell + My, \quad F(x), \quad G(y)$$

prime. So, we have a set of primes $p_1, \dots, p_{2k}$ of the form

$$p_i = a_i + Mt_i, \qquad i = 2, \dots, \ell - 1, \ell + 1, \dots, 2k,$$

and

$$p_1 = a_1 + Mx, \quad p_\ell = a_\ell + My.$$

Note that these primes satisfy the congruence conditions $p_i \equiv a_i \pmod{M}$, and we have that

$$2p_{i_1} \cdots p_{i_k} + 1 = F(x) \qquad \text{and} \qquad 2p_{j_1} \cdots p_{j_k} + 1 = G(y)$$

are prime. So, if we let $n(x, y) = 4p_1 \cdots p_{2k}$, then we get a solution

$$\varphi(F(x)G(y)) = n(x, y).$$

So, by running through the choices for $x, y < (Bk)^{O(1)}$, we are guaranteed to hit upon a value of $n(x, y)$ having a solution m to $\varphi(m) = n(x, y)$, as long as there is a subset of $\{x_1, \dots, x_{2k}\}$ of cardinality k summing to $S/2$.

Conversely, if there is no subset of $\{x_1, \dots, x_{2k}\}$ of cardinality k summing to $S/2$, then either $F(x)$ or $G(y)$ is an odd composite, and so fails to satisfy

$$\varphi(F(x)G(y)) = n(x, y)$$

for all values of x and y.

It follows that we can reduce the Partition Problem to the problem of deciding whether there are solutions m to $\varphi(m) = n$ for a set of $K = (Bk)^{O(1)}$ values of n. □

3.2. **Proof of Proposition 3.3.**

3.2.1. *General outline.* First, suppose that $x_1, \dots, x_{2k}$ are given. We build up our congruence conditions $a_i \pmod{M}$ in two stages, which we refer to as "initial congruence restrictions" and "primary congruence restrictions". The number M we build up is the product of various moduli which define these congruence restrictions.

3.2.2. *Initial congruence restrictions.* First, we suppose that

$$a_i \equiv 1 \pmod 8, \qquad i = 1, 2, \dots, 2k.$$

This condition ensures that (2) holds, since if $N_i \equiv a_i \pmod M$, where $8|M$, then we have that $8|N_i - 1$, whereas $N_1 \cdots N_{2k}$ is odd.

Next, we suppose that $3|M$ and that

$$a_i \equiv 2 \pmod 3.$$

This condition implies half of (3), since if $N_i \equiv a_i \pmod M$, we would have

$$2N_1 \cdots N_{2k} + 1 \equiv 2^{2k+1} + 1 \equiv 0 \pmod 3.$$

Then, we suppose that $5|M$ and that

$$a_i \equiv 4 \pmod 5.$$

This condition implies the other half of (3), since if $N_i \equiv a_i \pmod M$, we would have

$$4N_1 \cdots N_{2k} + 1 \equiv 4^{2k+1} + 1 \equiv 0 \pmod 5.$$

We note that the conditions we have so far also imply that for any $N_{i_1}, \dots, N_{i_k}$ satisfying $N_{i_j} \equiv a_{i_j} \pmod M$,

$$(2N_{i_1} \cdots N_{i_k} + 1,\ 8 \cdot 3 \cdot 5) = 1.$$

Next, we find integers $a_1, \dots, a_{2k}$ so that if $N_i \equiv a_i \pmod M$, and if

$$\{A_1, \dots, A_\ell\} \subset \{N_1, \dots, N_{2k}\},$$

where $\ell \le k$, then

$$(2A_1 \cdots A_\ell + 1,\ M) = 1$$

implies that $\ell = k$. This proves part of (1): Let $R_1, \dots, R_{k-1}$ be consecutive primes greater than k. Then, for $j = 1, 2, \dots, k-1$, we let g_j be any integer solution to

$$1 + jg_j \equiv R_j \pmod{2R_j}.$$

Clearly, since $j < k < R_j$, we have that if j is odd, then $g_j \equiv (R_j - 1)j^{-1} \pmod{2R_j}$ is a solution, and if j is even, then

$$g_j \equiv ((R_j - 1)/2)(j/2)^{-1} \pmod{R_j}$$

gives a solution. Next, we assume that

$$\prod_{j=1}^{k-1} \frac{2^{R_j} + 1}{3} \ \Bigg|\ M,$$

and our congruence conditions are

$$a_i \equiv 2^{g_j} \pmod{\frac{2^{R_j} + 1}{3}}, \qquad i = 1, 2, \dots, 2k,\ j = 1, 2, \dots, k-1.$$

We note that these moduli are all coprime, and are coprime to $8 \cdot 3 \cdot 5 = 120$, which is the modulus of the previous congruence conditions.

Now suppose that $N_i \equiv a_i \pmod M$, and suppose that

$$N = 2A_1 \cdots A_\ell + 1, \quad \text{for some } 1 \le \ell \le k-1,$$

where $\{A_1, \dots, A_\ell\} \subset \{N_1, \dots, N_{2k}\}$. Then, we have that

$$N = 2A_1 \cdots A_\ell + 1 \equiv 2^{1 + \ell g_\ell} + 1 \equiv 2^{R_\ell} + 1 \equiv 0 \pmod{\frac{2^{R_\ell} + 1}{3}}.$$

So,

$$\gcd(2A_1 \cdots A_\ell + 1,\ M) > 1,$$

as claimed.

We claim now that if

$$\{A_1, \ldots, A_k\} \subset \{N_1, \ldots, N_{2k}\},$$

then we have that $2A_1 \cdots A_k + 1$ is coprime to all the moduli we have so far. To show this, it suffices to prove that this number is coprime to $(2^{R_i} + 1)/3$ for $i = 1, 2, \ldots, k-1$: we have that

$$2A_1 \cdots A_k + 1 \equiv 2^{1+g_i k} + 1 \pmod{\frac{2^{R_i}+1}{3}}.$$

Now, the numbers $(2^{R_i}+1)/3$ and $2^{1+g_i k}+1$ are coprime unless $R_i | 1+g_i k$. But now, we know that $1 + g_i i \equiv 0 \pmod{R_i}$. So, the only way that $1 + g_i k \equiv 0 \pmod{R_i}$ could hold is if $R_i | (k-i)$, which can only hold if $i = k$. Thus, $2A_1 \cdots A_k + 1$ is coprime to the moudli we have so far.

3.2.3. *Primary congruence conditions.* Now we establish some congruence conditions which allow us to relate numbers a_i to the numbers x_i.

First, we require the following basic statement:

Lemma 3.4. *Suppose that $\{r_1, \ldots, r_n\}$ and $\{s_1, \ldots, s_n\}$ are two disjoint sets of odd prime numbers. Then, we have that the numbers*

$$Q_{i,j} = \frac{2^{r_i s_j} - 1}{(2^{r_i} - 1)(2^{s_j} - 1)}$$

are coprime.

Proof. It is well known that

$$\gcd(2^A - 1, 2^B - 1) = 2^{\gcd(A,B)} - 1.$$

Applying this to our problem, we find that

$$G = \gcd(2^{r_a s_c} - 1, 2^{r_b s_d} - 1) = 2^{\gcd(r_a s_c, r_b s_d)} - 1.$$

Now, if $a = b$, then we assume $c \neq d$, lest $r_a s_c = r_b s_d$; then we get that

$$G = 2^{r_a} - 1.$$

It is now obvious that $Q_{a,c}$ and $Q_{b,d}$ are coprime.

The same argument proves that $Q_{a,c}$ and $Q_{b,d}$ are coprime for $c = d$ (with $a \neq b$). □

Now let

$$A = \sum_{i=1}^{2k} |x_i|,$$

and let $U_1 < U_2 < \cdots < U_u$ be a set of primes of size $(k \log A)^{O(1)}$ which are coprime to the common modulus for the congruence conditions in Section 3.2.2; that is,

$$\gcd\left(8 \cdot 3 \cdot 5 \cdot \prod_{j=1}^{k-1} \frac{2^{R_j}+1}{3}, U_i\right) = 1, \qquad i = 1, \ldots, u.$$

Also, we suppose that

$$\prod_{i=1}^{u} U_i > 2A.$$

Then, let $v = U_u$, and let $V_1 < \cdots < V_v$ be primes greater than U_u and of size $(k \log A)^{O(1)}$.

For each $i = 1, 2, \ldots, u$, let

$$\{\vartheta(i,1), \ldots, \vartheta(i, U_i - 1)\} = \{0, \ldots, U_i - 1\} \setminus \{S/2 \pmod{U_i}\};$$

that is, the values $\vartheta(i,j)$ run through all the residue classes modulo U_i, except the class $S/2 \pmod{U_i}$. Also, let

$$\delta_{i,j} \equiv k^{-1} \pmod{U_i V_j}, \qquad 0 \le \delta_{i,j} \le U_i V_j - 1.$$

Our final set of congruence conditions on $a_1, \ldots, a_{2k}$ is as follows: For $i = 1, \ldots, u$, $j = 1, \ldots, U_i - 1$, and $\ell = 1, \ldots, 2k$,

$$a_\ell \equiv -2^{V_j x_\ell + \delta_{i,j}(V_j \vartheta(i,j) - 1)} \pmod{\frac{2^{U_i V_j} - 1}{(2^{U_i} - 1)(2^{V_j} - 1)}}.$$

From Lemma 3.4, we have that all these moduli are coprime, and so the Chinese Remainder Theorem tells us that such a_ℓ exist (and can be easily computed).

Let M be the product of all the moduli in our congruence conditions, that is,

$$M = 8 \cdot 3 \cdot 5 \cdot \prod_{\nu=1}^{k-1} \frac{2^{R_\nu} + 1}{3} \cdot \prod_{i=1}^{u} \prod_{j=1}^{U_i - 1} \frac{2^{U_i V_j} - 1}{(2^{U_i} - 1)(2^{V_j} - 1)},$$

and suppose that $N_i \equiv a_i \pmod{M}$, $i = 1, \ldots, 2k$. We claim that

$$x_{n_1} + \cdots + x_{n_k} = S/2$$

if and only if $2N_{n_1} \cdots N_{n_k} + 1$ is coprime to M.

First, let us suppose that $x_{n_1} + \cdots + x_{n_k} \ne S/2$. Then, for one of our primes $U_1, \ldots, U_u$, we have that

$$x_{n_1} + \cdots + x_{n_k} \equiv \vartheta(i,j) \pmod{U_i},$$

for some $j = 1, 2, \ldots, U_i - 1$. Letting

$$T = \frac{2^{U_i V_j} - 1}{(2^{U_i} - 1)(2^{V_j} - 1)},$$

we see that

$$\begin{aligned} 2N_{n_1} \cdots N_{n_k} + 1 &\equiv (-1)^k 2^{1 + V_j(x_{n_1} + \cdots + x_{n_k} - k\delta(i,j)\vartheta(i,j)) - k\delta(i,j)} + 1 \\ &\equiv -2^{V_j U_i I} + 1 \equiv 0 \pmod{T}, \end{aligned}$$

where I is some integer.

On the other hand, if

$$x_{n_1} + \cdots + x_{n_k} = S/2,$$

then for any $i = 1, 2, \ldots, u$, $j = 1, 2, \ldots, U_i - 1$,

$$\begin{aligned} 2N_{n_1} \cdots N_{n_k} + 1 &\equiv (-1)^k 2^{V_j(x_{n_1} + \cdots + x_{n_k} - \vartheta(i,j))} + 1 \\ &\equiv -2^{V_j I} + 1 \pmod{T}, \end{aligned}$$

where $\gcd(I, U_i) = 1$. To show that this last quantity is coprime to T, we observe that

$$\gcd(2^{V_j I} - 1,\ 2^{U_i V_j} - 1) = 2^{V_j} - 1,$$

and so,

$$\gcd(2^{V_j I} - 1,\ T) = \gcd(2^{V_j} - 1, T) = 1.$$

To see this last equality, we observe that

$$\frac{2^{U_i V_j} - 1}{2^{V_j} - 1} = \sum_{j=0}^{U_i - 1} 2^{jV_j} \equiv \sum_{j=0}^{U_i - 1} 2^j = 2^{U_i} - 1 \pmod{2^{V_j} - 1}.$$

Therefore, since $\gcd(2^{U_i} - 1, 2^{V_j} - 1) = 1$, it follows that $\gcd(2^{V_j} - 1, T) = 1$, as claimed. □

4. Algorithm for Problem B

As we have seen in Section 3, Problem A is likely to be **NP**-complete, and certainly Problem B is no easier. Nevertheless, here we show in some sense "on average" Problem B can solved in polynomial time.

We also remark that one cannot get an efficient "on average" algorithm for the Partition Problem from a combination of the reduction of Section 3 and the algorithm of this section. This is because the reduction does not produce uniformly distributed instances of Problem B.

Given an algorithm for solving Problem B, we then have an algorithm for solving Problem C, whose first step is just to factor n by using a probabilistic factoring algorithm (see [9]); however, for most integers n, the factoring step dominates the overall complexity of the algorithm. In the worst case, where $\Psi(n)$ is "large", the running time of the rest of the algorithm would dominate this factoring step.

In this section we prove the following result which is based on some kind of "intelligent exhaustive search".

Theorem 4.1. Problem B *can be solved in time*

$$T(n) \le \left(\#\Psi^*(n) + \tau(n) + \log n\right)^{O(1)},$$

where

$$\Psi^*(n) = \bigcup_{d \mid n} \Psi(d).$$

Proof. Basically, we form a sequence of sets $\mathcal{E}_1, \mathcal{E}_2, \ldots$, where $\mathcal{E}_i$ is the set of all $2i$-tuples of the form $(\ell_1, \ldots, \ell_i, \gamma_1, \ldots, \gamma_i)$, where the $\ell_1 < \cdots < \ell_i$ are all prime, and where $\gamma_1, \ldots, \gamma_i$ are positive integers satisfying

$$\prod_{j=1}^{i} \ell_j^{\gamma_j - 1}(\ell_j - 1) \mid n.$$

It is clear that we cannot construct this sequence $\mathcal{E}_i$ forever, because $\#\Psi(n)$ is finite; moreover, we finish with $\mathcal{E}_k$, where

$$k = O(\log n).$$

Now, every integer m satisfying $\varphi(m) = n$ corresponds to some vector in one of these sets $\mathcal{E}_i$, namely the vector $(\ell_1, \ldots, \ell_i, \gamma_1, \ldots, \gamma_i)$ satisfying

$$m = \prod_{j=1}^{i} \ell_j^{\gamma_j}.$$

Also, it is easy to see that $\mathcal{E}_1, \ldots, \mathcal{E}_k$ contain

$$\#\Psi^*(n) = \sum_{d|n} \#\Psi(d)$$

vectors among them. Thus, given $\mathcal{E}_1, \ldots, \mathcal{E}_k$, at most

$$(\#\Psi^*(n) + \log n)^{O(1)}$$

bit operations are required to scan through the sets $\mathcal{E}_i$ to locate vectors corresponding to solutions $\varphi(m) = n$.

Finally, we describe how to build the list $\mathcal{E}_i$, given that we already have the list $\mathcal{E}_{i-1}$: basically, we run through $\mathcal{E}_{i-1}$ of vectors $(\ell_1, \ldots, \ell_{i-1}, \gamma_1, \ldots, \gamma_{i-1})$, and we locate all primes $\ell_i > \ell_{i-1}$, and all integers $\gamma_i \geq 1$ such that

$$\ell_i^{\gamma_i - 1}(\ell_i - 1) \mid n_i,$$

where

$$n_i = \frac{n}{\prod_{j=1}^{i-1} \ell_j^{\gamma_j - 1}(\ell_j - 1)}.$$

To do this, we search through the divisors of n_i to see which are of the form $\ell_i^{\gamma_i - 1}(\ell_i - 1)$. The number of bit operations needed to find all such divisors is

$$(\tau(n_i) + \log n)^{O(1)} = (\tau(n) + \log n)^{O(1)}.$$

Now, for each such vector $(\ell_1, \ldots, \ell_{i-1}, \gamma_1, \ldots, \gamma_{i-1})$ and for each such pair (ℓ_i, γ_i), $\ell_i > \ell_{i-1}$, we add the vector $(\ell_1, \ldots, \ell_i, \gamma_1, \ldots, \gamma_i)$ to the new list $\mathcal{E}_i$.

It is easy to see that the number of bit operations required to build the list $\mathcal{E}_1, \ldots, \mathcal{E}_k$ is then

$$(\#\Psi^*(n) + \tau(n) + \log n)^{O(1)},$$

and so our theorem is proved. □

To address the average performance of the algorithm, we require the following bound:

Theorem 4.2.

$$\sum_{n \leq x} \#\Psi^*(n) \ll x \log x.$$

Proof. We have that

$$\sum_{n \leq x} \#\Psi^*(n) = \sum_{n \leq x} \sum_{d|n} \#\Psi(d) \leq x \sum_{d \leq x} \frac{\#\Psi(d)}{d}. \tag{4}$$

Now,

$$\sum_{d \leq x} \#\Psi(d) = \#\{n \geq 1 \ : \ \varphi(n) \leq x\} = \left(\frac{\zeta(2)\zeta(3)}{\zeta(6)} + o(1)\right) x;$$

see [7]. So, by partial summation, we conclude that

$$\sum_{d \leq x} \frac{\#\Psi(d)}{d} = O(\log x),$$

which, together with (4), finishes the proof. □

An almost immediate corollary of Theorem 4.2, together with the well-known bound

$$\sum_{n\le x}\tau(n)=O(x\log x) \tag{5}$$

(see Theorem 2 in Section I.3.2 of [23], and Theorem 4.1) is the following.

Corollary 4.3. *For every $A>0$, there exists $B>0$, so that for all but at most $O(x/\log^A x)$ integers $n\le x$ we have that the algorithm in Theorem 4.1 finds $\Psi(n)$ in time $\log^B n$.*

5. Problem C is as hard as factoring

Here we show that if we had an "efficient" algorithm for solving Problem C, then we can factor density 1 of the integers in $\mathcal{P}_2$, which is the set of integers that are the product of exactly two prime numbers.

This result seems weaker than the result in Theorem 3.2; however, Theorem 3.2 assumes a strong version of the Hardy-Littlewood conjecture, whereas here we make no such assumptions.

Let $\mathcal{P}_2(x)$ denote the set of $n\in\mathcal{P}_2$ with $n\le x$. It is well known that

$$\#\mathcal{P}_2(x)=(1+o(1))\frac{x\log\log x}{2\log x}.$$

As usual, we say that a randomized algorithm factors n in polynomial time, if the algorithm has access to a random number generator, and factors n in time $Ck\log^A n$ (for some constants $A,C>0$) steps with probability at least $1-2^{-k}$.

Our next result, connecting solvability of Problem C with integer factorization, is the following:

Theorem 5.1. *Given an algorithm that, for each integer N, finds the set $\Psi(N)$ in $(\#\Psi^*(N)+\tau(N)+\log N)^{O(1)}$ steps, without being given the prime factorization of N, one can factor using a randomized algorithm in polynomial time every $n\in\mathcal{P}_2(x)\backslash\mathcal{E}(x)$, for some set $\mathcal{E}(x)$ such that*

- *$\#\mathcal{E}(x)=O(x/\log^2 x)$, unconditionally,*
- *$\mathcal{E}(x)=\varnothing$, under the Extended Riemann Hypothesis.*

Proof. Let $\pi(X,r,a)$ denote the number of primes $\ell\le X$ with $\ell\equiv a \pmod r$. We need the following result which is a greatly relaxed version of Theorem 2.1 of [2]. Namely, for every $X>X_0$, we have that for all but at most $O(1)$ prime numbers r with $X^{1/3}\ge r\ge\log X$,

$$\pi(X,4r,a)\ge\frac{X}{2\varphi(4r)\log X}=\frac{X}{4r\log X} \tag{6}$$

holds for every integer a with $\gcd(a,4r)=1$. It is also well known that (6) holds under the Extended Riemann Hypothesis for all r satisfying $X^{1/3}\ge r\ge 2$.

We define $\mathcal{E}(x)$ as the set of $n=pq\in\mathcal{P}_2(x)$ such that p or q (or both) is a prime for which (6) fails. Clearly $\mathcal{E}(x)$ satisfies the required properties.

Now, assume we are given sufficiently large odd $n=pq\in\mathcal{P}_2(x)\backslash\mathcal{E}(x)$.

Clearly, we can also assume (for both unconditional and conditional results) that $n\ge x^{1/2}$.

We choose two positive integers $k_1,k_2\le x^3$ and we also consider the product $4(2k_1+1)(2k_2+1)n$.

Let D be a divisor of $4(2k_1+1)(2k_2+1)n$. We remark that if $D=\varphi(m)$ for some m, then m has at most two odd prime divisors, and for every prime divisor $\ell|m$ we have $\ell-1|D$. Clearly, the odd part of m must be one of the forms

$$(2q_1\cdots q_s+1)^a(2r_1\cdots r_t+1)^b, \qquad (2q_1\cdots q_s+1)^a, \qquad (4q_1\cdots q_s+1)^b,$$

where $q_1\cdots q_sr_1\cdots r_t|D$ and $q_1,\ldots,q_s,r_1,\ldots,r_t$ are odd primes (which are possibly not distinct). Given D, $q_1,\ldots,q_s$ and $r_1,\ldots,r_t$, there is at most one choice for $a,b\ge 1$ such that $\varphi(m)=D$.

Since there are at most $\tau(D)$ possibilities for each of the products $q_1\cdots q_s$ and $r_1\cdots r_t$, we obtain

$$\#\Psi(D)=O(\tau(D)^2)=O\left(\tau(4(2k_1+1)(2k_2+1)n)^2\right)$$

(this crude estimate can easily be improved). This implies that

$$\begin{aligned}\#\Psi^*(4(2k_1+1)(2k_2+1)n) &= O\left(\tau(4(2k_1+1)(2k_2+1)n)^3\right)\\ &= O\left(\tau(2k_1+1)^3\tau(2k_2+1)^3\right).\end{aligned}$$

We see from (5) that the total number of positive integers $k\le Y$ with $\tau(k)\ge \log^3 Y$ is $O(Y\log^{-2}Y)$. Thus from (6), applied with

$$X=4x^3p+2p+1, \qquad r=p, \qquad a=2p+1,$$

we derive that, for a sufficiently large x, there are at least

$$\frac{4x^3p}{4p\log(4x^3p+2p+1)}+O(x^3\log^{-2}x)\ge\frac{x^3}{4\log x}+O(x^3\log^{-2}x)\ge\frac{x^3}{5\log x}$$

positive integers $k_1\le x^3$ for which simultaneously $2(2k_1+1)p+1$ is prime and $\tau(2k_1+1)<8\log^3 n$ (recall that $x^{1/2}\le n\le x$ and $k_1\le x^3$). Similarly, we have at least the same number of positive integers $k_2\le x^3$ for which simultaneously $2(k_2+1)q+1$ is prime and $\tau(2k_2+1)\le 8\log^3 n$.

For each such pair of integers k_1,k_2 we see that the cardinality of the set $\Psi^*(4(2k_1+1)(2k_2+1)n)$ is polynomially bounded, namely,

$$\#\Psi^*(4(2k_1+1)(2k_2+1)n)=O(\log^{18}n),$$

and that $\Psi(4(2k_1+1)(2k_2+1)n)$ contains a solution of the form

$$m=(2(2k_1+1)p+1)(2(2k_2+1)q+1) \tag{7}$$

from which, together with the equation $n=pq$, the primes p and q can be trivially found. Now we simply try all values of $m\in\Psi(4(2k_1+1)(2k_2+1)n)$ in order to find the one of the form (7).

These considerations naturally lead to the following probabilistic algorithm which finds the above pair of k_1,k_2 and thus the primes p and q.

Suppose that the inverting algorithm outputs the set $\Psi(N)$ in time which is bounded by $(\#\Psi^*(N)+\tau(N)+\log N)^A$ for some constant $A>0$. We choose integers k_1,k_2 uniformly at random in the interval $[1,x^3]$ and use the algorithm to compute $\Psi(4(2k_1+1)(2k_2+1)n)$. If the time it takes exceeds $\log^{20A}N$, this means that $\#\Psi^*(4(2k_1+1)(2k_2+1)n)\ge\log^{19}N$, and we simply terminate the algorithm and choose another pair k_1,k_2. It is clear that after the expected number of $O(\log^6 x)=O(\log^6 n)$ (since $n\ge x^{1/2}$) trials (that is, random choices of k_1,k_2), we find the desired pair of k_1,k_2. $\square$

Recalling Theorem 4.2 we see that the algorithm requested in Theorem 5.1 to find $\Psi(N)$ is supposed to run in polynomial time for almost all integers N. Furthermore, Theorem 5.1 admits a version with an algorithm which, in polynomial time, finds $\Psi(N)$ provided $\Psi(N)+\tau(N) = (\log N)^{O(1)}$ (and is allowed to give a wrong answer or no answer for the remaining integers N).

One can easily improve the bound of Theorem 5.1 on $\#\mathcal{E}(x)$ as

$$\#\mathcal{E}(x) \le x \exp\left(-A\frac{(\log\log x)^{3/2}}{(\log\log\log x)^{1/2}}\right)$$

for any constant $A > 0$, if before applying our reduction one tries to find a small factor of $n \in \mathcal{P}_2(x)$ by using the algorithm of [17]; see also [16]. Moreover, for the cryptographically most interesting class $\widetilde{\mathcal{P}_2}(x)$ of $n = pq \in \mathcal{P}_2(x)$ with $p < q < 2p$, the exceptional set $\widetilde{\mathcal{E}}(x)$ is of the size $\#\widetilde{\mathcal{E}}(x) = O(x^{1/2})$.

We also remark that if the algorithm of Theorem 5.1 ever fails for some $n_* \in \mathcal{P}_2$, then after this for all other $n \in \mathcal{P}_2$ one can first compute $\gcd(n, n_*) = 1$ and either factor n immediately or guarantee that the algorithm of Theorem 5.1 succeeds.

Acknowledgments

We would like to thank Kevin Ford for mentioning to us several references which had been left out of an earlier version of the paper. We also thank Carl Pomerance for several fruitful discussions.

References

[1] M. Agrawal, N. Kayal and N. Saxena, 'PRIMES is in **P**', Ann. of Math. (2) **160** (2004), 781–793. MR2123939

[2] W. R. Alford, A. Granville and C. Pomerance, 'There are infinitely many Carmichael numbers', *Annals of Math.*, **140** (1994), 703–722. MR1283874 (95k:11114)

[3] R. C. Baker and G. Harman, 'Shifted primes without large prime factors', *Acta Arith.*, **83** (1998), 331–361. MR1610553 (99b:11104)

[4] A. Balog, 'The prime k-tuplets conjecture on average', *Analytic Number Theory*, Progress in Mathematics **85**, Birkhäuser, Boston, 1990, 47–75. MR1084173 (92e:11105)

[5] W. Banks, J. B. Friedlander, C. Pomerance and I. E. Shparlinski, 'Multiplicative structure of values of the Euler function', *High Primes and Misdemeanours: Lectures in Honour of the 60th Birthday of Hugh Cowie Williams*, Fields Institute Communications, vol. 41, Amer. Math. Soc., 2004, 29–48. MR2075645 (2005f:11217)

[6] W. Banks, K. Ford, F. Luca, F. Pappalardi and I. E. Shparlinski, 'Values of the Euler function in various sequences', *Monatsh. Math.*, **146** (2005), 1–19.

[7] P. T. Bateman, 'On the distribution of values of the Euler function', *Acta Arith.*, **21** (1972), 329–345. MR0302586 (46:1730)

[8] W. Bosma, 'Some computational experiments in number theory', *Preprint*, 2004.

[9] R. Crandall and C. Pomerance, *Prime numbers: A Computational perspective*, Springer-Verlag, Berlin, 2001. MR1821158 (2002a:11007)

[10] T. Dence and C. Pomerance, 'Euler's function in residue classes', *The Ramanujan J.*, **2** (1998), 7–20. MR1642868 (99k:11148)

[11] L. E. Dickson, 'A new extension of Dirichlet's theorem on prime numbers', *Messenger of Mathematics*, **33** (1904), 155-161.

[12] P. Erdős, 'On the normal number of prime factors of $p-1$ and some related problems concerning Euler's ϕ-function', *Quart. J. Math.*, **6** (1935), 205–213.

[13] P. Erdős and C. Pomerance, 'On the normal number of prime factors of $\varphi(n)$', *Rocky Mountain J. Math.*, **15** (1985), 343–352. MR0823246 (87e:11112)

[14] K. Ford, 'The number of solutions of $\varphi(x) = m$', *Annals of Math.*, **150** (1999), 283–311. MR1715326 (2001e:11099)

[15] K. Ford, S. Konyagin and C. Pomerance, 'Residue classes free of values of Euler's function', *Proc. Number Theory in Progress*, Walter de Gruyter, Berlin, 1999, 805–812. MR1689545 (2000f:11120)
[16] H. W. Lenstra, Jr., 'Factoring integers with elliptic curves' *Annals of Math.*, **126** (1987), 649–673. MR0916721 (89g:11125)
[17] H. W. Lenstra, Jr., J. Pila and C. Pomerance, 'A hyperelliptic smoothness test, I', *Phil. Trans. of the Royal Society of London, Ser. A.*, **345** (1993), 397–408. MR1253501 (94m:11107)
[18] N. S. Mendelsohn, 'The equation $\varphi(x) = k$', *Math. Magazine*, **49** (1976), 37-39. MR0396385 (53:252)
[19] G. L. Miller, 'Riemann's hypothesis and tests for primality', *J. Comput. System Sci.*, **13** (1976), 300-317. MR0480295 (58:470a)
[20] L. L. Pennesi, 'A method for solving $\varphi(x) = n$', *Amer. Math. Monthly*, **74** (1957), 497-499.
[21] C. Pomerance, 'Popular values of Euler's function', *Mathematika*, **27** (1980), 84–89. MR0581999 (81k:10076)
[22] C. Pomerance, 'Two methods in elementary analytic number theory', *Number theory and application*, R. A. Mollin, ed., Kluwer Acad. Publ., Dordrecht, 1989, 135–161. MR1123073 (92j:11107)
[23] G. Tenenbaum, *Introduction to analytic and probabilistic number theory*, Cambridge University Press, UK, 1995. MR1342300 (97e:11005b)

Department of Computing, Macquarie University, Sydney, New South Wales 2109, Australia
E-mail address: scontini@ics.mq.edu.au

School of Mathematics, Georgia Institute of Technology, Atlanta, Georgia 30332
E-mail address: ecroot@math.gatech.edu

Department of Computing, Macquarie University, Sydney, New South Wales 2109, Australia
E-mail address: igor@ics.mq.edu.au