A $p + 1$ Method of Factoring

By H. C. Williams

Abstract. Let $N$ have a prime divisor $p$ such that $p + 1$ has only small prime divisors. A method is described which will allow for the determination of $p$, given $N$. This method is analogous to the $p - 1$ method of factoring which was described in 1974 by Pollard. The results of testing this method on a large number of composite numbers are also presented.

1. Introduction. In 1974 Pollard [8] introduced a method of factorization which has since been called the $p - 1$ factorization technique. Actually, the test was known to D. N. and D. H. Lehmer many years before this but it was never published because, without a fast computer, it was not possible to determine how effective it would be in practice. For the convenience of the reader we give a brief description of this test.

Suppose $A7$ is a number to be factored and that $N$ has a prime factor $p$ such that

\begin{equation}
\begin{aligned}
p = \left( \prod_{i=1}^{k} q_i^{a_i} \right) + 1,
\end{aligned}
\end{equation}

where $q_i$ is the $i$th prime and $q_i^{a_i} \leq B_1$. Let $q_i^{k_i}$ be that power of $q_i$ such that $q_i^{k_i} \leq B_1$ and $q_i^{k_i+1} > B_1$ and put

\begin{equation}
\begin{aligned}
R = \prod_{i=1}^{k} q_i^{k_i}.
\end{aligned}
\end{equation}

Clearly, $p - 1 \mid R$ and since $a^{p-1} \equiv 1 \pmod{p}$ when $(N, a) = 1$, we have $a^R \equiv 1 \pmod{p}$. Thus, $p \mid (N, a^R - 1)$.

The algorithm now proceeds as follows. For a given $B_1$ put

\begin{equation}
\begin{aligned}
R = r_1 r_2 r_3 \cdots r_m,
\end{aligned}
\end{equation}

(for example, $m = k, r_i = q_i^{a_i}$), $a_0 = a$, where $(a, N) = 1$ and define

\begin{equation}
\begin{aligned}
a_i &\equiv a_{i-1}^{r_i} \pmod{N} \quad (i = 1, 2, 3, \ldots, m).
\end{aligned}
\end{equation}

The values of $a_i$ can be easily calculated by a power algorithm such as those mentioned in Knuth [4, p. 441ff.]. We now evaluate $a_m \equiv a^R \pmod{N}$ and $(a_m - 1, N)$. Even for fairly small values of $B_1$ it frequently occurs that $(a_m - 1, N)$ yields a nontrivial factor of $N$.

Pollard also gives in [8] two versions of a second step which can be appended to the above algorithm. We give one of these here.

Suppose instead of (1.1) we have

\begin{equation}
\begin{aligned}
p = s \left( \prod_{i=1}^{k} q_i^{a_i} \right) + 1,
\end{aligned}
\end{equation}

Received March 30, 1981; revised September 28, 1981.

1980 Mathematics Subject Classification. Primary 10A25.

©1982 American Mathematical Society

0025-5718/82/0000-0063/$03.25
where \( s \) is a prime and \( B_1 < s \leq B_2 \). In this case we have \( p \mid (a_m^s - 1, N) \). Let \( \{s_j: j = 1, 2, \ldots, k \} \) be the ordered set of all primes such that \( B_1 < s_j \leq B_2 \), and put \( 2d_j = s_{j+1} - s_j \). Since the differences between successive primes increase very slowly, we see that there will not be very many distinct values for the \( d_j \)'s. In fact, if we let \( d(x) \) be the largest value of \( d_j \) for all primes between \( 1 \) and \( x \), we have \( d(200000) = 43 \), \( d(10^6) = 57 \), and \( d(4.444 \times 10^{12}) = 326 \); see Brent [1]. Thus, it is not too difficult to tabulate \( a_m^{2d_j} \) for all the distinct \( d_j \). In fact, it is not really necessary to tabulate these values for all \( d_j < d(x) \). The larger \( d_j \) occur very seldom and the method is almost as fast if the table extends to only \( K \log x \) for some moderate value of \( K \) instead of \( d(x) \) which seems to be \( O((\log x)^2) \). This remark applies also to the second step of the \( p + 1 \) method.

We calculate \( b_1 \equiv a_m^{s_1} \pmod{N} \) and define

\[
b_{j+1} \equiv a_m^{2d_j} b_j \pmod{N}.
\]

We now compute

\[
G_t = \left( \prod_{i=0}^{c}(b_{t+i} - 1), N \right)
\]

for \( t = 1, c + 1, 2c + 1, \ldots, \lfloor B_2/c \rfloor c + 1 \).

Since \( b_j \equiv a_m^{s_j} \pmod{N} \), we see that \( p \) must divide some \( G_j \). Because greatest common divisors are more expensive to evaluate than products, we usually have \( c > 1 \).

In [3] Guy and Conway suggest that, by using Lucas functions, the first step of the \( p - 1 \) method can be converted into a factorization algorithm for finding a prime divisor \( p \) of \( N \) when \( p + 1 \) has only small prime factors. In this paper we give a description of how this can be done. We also present a number of new factorizations which have been obtained by using either the \( p - 1 \) or \( p + 1 \) method. It should be mentioned here that John Brillhart and Earl Ecklund have also implemented a version of the first step of the \( p + 1 \) method. However, in their few computer runs they were only able to find factors that had been previously discovered by the \( p - 1 \) method. We also point out that a version of the method using the finite field of \( p^2 \) elements is also possible, if the reader wishes to avoid Lucas functions. Indeed, the author has been informed that R. P. Brent has an implementation of the \( p + 1 \) method based on this interpretation.

2. The Lucas Functions. In order to develop the method and formulas required in the next section, we give here a description of some of the basic properties of the Lucas functions.

Let \( P, Q \) be integers, and let \( \alpha, \beta \) be the zeros of \( x^2 - Px + Q \). We define the Lucas functions by

\[
U_n(P, Q) = \frac{(\alpha^n - \beta^n)}{(\alpha - \beta)}, \quad V_n(P, Q) = \alpha^n + \beta^n.
\]

We also put \( \Delta = (\alpha - \beta)^2 = P^2 - 4Q \). When there is no doubt as to the values of the arguments \( P \) and \( Q \), we often omit them. These functions satisfy a large number of identities. We will require those given below

\[
\begin{align*}
U_{n+1} & = PU_n - QU_{n-1}, \\
V_{n+1} & = PV_n - QV_{n-1}.
\end{align*}
\]
A \( p + 1 \) METHOD OF FACTORING

\[ U_{2n} = V_n U_n, \]
\[ V_{2n} = V_n^2 - 2Q^n, \]

(2.3)

\[ U_{2n-1} = U_n^2 - QU_{n-1}^2, \]
\[ V_{2n-1} = V_n V_{n-1} - PQ^{n-1}, \]

(2.4)

\[ \Delta U_n = PV_n - 2QV_{n-1}, \]
\[ V_n = PU_n - 2QU_{n-1}, \]

(2.5)

\[ U_{m+n} = U_m U_{n+1} - QU_{m-1} U_n, \]
\[ \Delta U_{m+n} = V_m V_{n+1} - QV_{m-1} V_n, \]

(2.6)

\[ U_n(V_k(P, Q), Q^k) = U_{nk}(P, Q)/U_k(P, Q), \]
\[ V_n(V_k(P, Q), Q^k) = V_{nk}(P, Q). \]

(2.7)

These identities can all be verified by direct substitution from (2.1), using the simple facts that \( P = \alpha + \beta \) and \( Q = \alpha \beta \).

We also note that if \( (N, Q) = 1 \) and \( P'Q = P^2 - 2Q \) (mod \( N \)), then \( P' = \alpha/\beta + \beta/\alpha \) and \( Q' = \alpha/\beta \cdot \beta/\alpha = 1 \); hence,

\[ U_{2m}(P, Q) \equiv PQ^{m-1} U_m(P', 1) \quad (\text{mod } N). \]

(2.8)

Finally, we need the following

**Theorem** (see Lehmer [5]). If \( p \) is an odd prime, \( p \nmid Q \) and the Legendre symbol \((\Delta/p) = \epsilon\), then

\[ U_{(p-\epsilon)m}(P, Q) \equiv 0 \quad (\text{mod } p) \]
\[ V_{(p-\epsilon)m}(P, Q) \equiv 2Q^{m(1-\epsilon)/2} \quad (\text{mod } p). \]

3. The First Step of the Algorithm. Suppose that \( p \) is a prime divisor of \( N \) and

\[ p = \left( \prod_{i=1}^{k} q_i^{a_i} \right) - 1, \]

where \( q_i \) is again the \( i \)th prime and \( q_i^{a_i} \leq B_1. \) If \( R \) is defined as in (1.2), we have \( p + 1 \mid R. \) By the theorem of Section 2 we see that if \( (Q, N) = 1 \) and \((\Delta/p) = -1\), then \( p \mid U_R(P, Q), \) and therefore \( p \mid (U_R(P, Q), N). \)

To find \( U_R(P, Q), \) Guy and Conway seem to suggest that the first formulas of (2.2), (2.3), and (2.4) be used together with the second formula of (2.5) to obtain

\[ U_{2n-1} = U_n^2 - QU_{n-1}^2, \]
\[ U_{2n} = U_n(PU_n - 2QU_{n-1}), \]
\[ U_{2n+1} = PU_{2n} - QU_{2n-1}. \]

These formulas can be used in a power algorithm routine similar to that suggested by Lehmer [6] to find \( U_R(P, Q). \) The problem with this method is that \( R \) can be very large (for example, when \( B_1 = 10^5, R > 10^{43410} \)), and it is difficult to store its value in the computer. Also, if \( B_1 \) is increased to obtain a new \( R \) value, say \( R', \) we would...
have to start all over again at \( U(P, Q) \) and \( U_2(P, Q) \) to find \( U_R(P, Q) \) instead of continuing on from \( U_R(P, Q) \). These problems can be overcome by using a different technique.

If \( p \mid U_R(P, Q) \), then by (2.3) \( p \mid U_2(P, Q) \); thus, from (2.8) we have \( p \mid U_R(P', 1) \).

It follows that we lose no generality in assuming that \( Q = 1 \). Further, by the Theorem of Section 2, we also have
\[
V_{(p-\varepsilon)m}(P, 1) \equiv 2 \pmod{p};
\]
hence, if \( p \mid U_R(P, 1) \), then \( p \mid (V_R(P, 1) - 2) \). We will assume throughout the remainder of this paper that \( Q = 1 \) in our Lucas functions.

The first step of our \( p + 1 \) algorithm is now the following:

Let \( R = r_1r_2r_3 \cdots r_m \) as above and find \( P_0 \) such that \( (P_0^2 - 4, N) = 1 \). Define

\[
V_n(P) = V_n(P, 1), \quad U_n(P) = U_n(P, 1)
\]
and

\[
P_j \equiv V_j(P_{j-1}) \pmod{N} \quad (j = 1, 2, 3, \ldots, m).
\]

By the second formula of (2.7), we see that
\[
P_m \equiv V_R(P_0) \pmod{N}.
\]
We then calculate \( (P_m - 2, N) \).

To find \( V_r = V_r(P) \) from \( P \) we need only use the formulas

\[
\begin{aligned}
V_{2f-1} &\equiv V_fV_{f-1} - P, \\
V_{2f} &\equiv V_f^2 - 2, \\
V_{2f+1} &\equiv PV_f^2 - V_fV_{f-1} - P \pmod{N},
\end{aligned}
\]
(see the second formulas of (2.2), (2.3), and (2.4)).

Let
\[
r = \sum_{i=0}^{t} b_i 2^{t-i} \quad (b_i = 0, 1),
\]
\( f_0 = 1 \), and \( f_{k+1} = 2f_k + b_{k+1} \); then \( f_r = r \). Also, if \( V_0(P) = 2 \), \( V_1(P) = P \), then, to find the pair \((V_{j+i}, V_{j+i-1})\) from \((V_{j}, V_{j-1})\), we need only use the formula

\[
(V_{j+i}, V_{j+i-1}) = \begin{cases} 
(V_{2f_i}, V_{2f_i-1}) & \text{when } b_{k+1} = 0, \\
(V_{2f_i+1}, V_{2f_i}) & \text{when } b_{k+1} = 1,
\end{cases}
\]
together with (3.2).

4. The Second Step of the Algorithm. Suppose

\[
p = s \left( \prod_{i=1}^{k} q_i^{s_i} \right) - 1,
\]
where \( s \) is a prime, and \( B_1 < s \leq B_2 \). Define \( s_j \) and \( 2d_j \) as in Section 1. If \( (\Delta/p) = -1 \) and \( p \mid P_m - 2 \), then \( p \mid (U_s(P_m), N) \) by (2.7) and (3.1).

Let \( U[n] \equiv U_n(P_m), V[n] \equiv V_n(P_m) \pmod{N} \), and tabulate \( U[2d_j - 1], U[2d_j], U[2d_j + 1] \) for the distinct \( d_j \) by using

\[
U[0] = 0, \quad U[1] = 1 \quad \text{and} \quad U[n + 1] = P_mU[n] - U[n - 1].
\]
Put
\[ T[s_i] = \Delta U_{i}(P_m) = \Delta U_{i,R}(P_0)/U_R(P_0) \pmod{N}, \]
by the first formula of (2.7) and (3.1). From the second formula of (2.6), we have
\[
\begin{align*}
T[s_i] &\equiv P_m V[s_i] - 2V[s_i - 1], \\
T[s_i - 1] &\equiv 2V[s_i] - P_m V[s_i - 1] \pmod{N},
\end{align*}
\]
and from the second formula of (2.6) we get
\[
\begin{align*}
T[s_{i+1}] &\equiv T[s_i] U[2d_i + 1] - T[s_i - 1] U[2d_i], \\
T[s_{i+1} - 1] &\equiv T[s_i] U[2d_i] - T[s_i - 1] U[2d_i - 1] \pmod{N}.
\end{align*}
\]
Thus, to execute the second step of the algorithm we need only use (4.2) and (4.3) to obtain \( T[s_i] \), \( i = 1, 2, 3, \ldots \), and then evaluate
\[
H_t = \left( \prod_{i=0}^{c} T[s_{i+1}], N \right)
\]
for \( t = 1, c + 1, 2c + 1, \ldots, c[B_2/c] + 1 \). We must have \( p | H_i \) for some \( i \) if \( p \) satisfies (4.1) and \( \Delta/p = -1 \).

5. Implementation and Results. One of the difficulties in implementing the \( p + 1 \) algorithm of Sections 3 and 4 is the possibility that \( p \) in (4.1) is such that \( (P_0^2 - 4/p) = +1 \) for the selected value of \( P_0 \). There is no way of knowing beforehand that this will not occur. If we assume that the values of \( P_0 \) such that \( (\Delta/p) = -1 \) are randomly distributed, the probability that \( (\Delta/p) = 1 \) is the same as the probability that \( (\Delta/p) = -1 \), i.e., \( \frac{1}{2} \). Thus the probability that \( (\Delta_i/p) = 1 \), \( i = 1, 2, 3, \ldots, n - 1 \), for each of \( n \) trials at a \( P_0 \) value and \( (\Delta_n/p) = -1 \) for the \( n \)th trial is \( \left( \frac{1}{2} \right)^n \). (We assume that the \( P_0 \) values selected are independent.) It follows that the probability that we will find some \( \Delta_i \) such that \( (\Delta_i/p) = -1 \) after at most \( n \) trials at a \( P_0 \) value is
\[
\sum_{i=1}^{n} \left( \frac{1}{2} \right)^i = 1 - \left( \frac{1}{2} \right)^n.
\]
Thus, if \( N \) has a prime factor \( p \) which satisfies (4.1), and we use the algorithm of Section 3 with three trials at a \( P_0 \) value, we would expect to find that \( p | (P_m - 2, N) \) for seven of every eight such \( N \) tested. The referee has pointed out that, instead of making three guesses at \( P_0 \), one could make many guesses to obtain \( \Delta_1, \Delta_2, \Delta_3, \ldots \) as possible values of \( \Delta \) for which \( (\Delta/p) = -1 \). One could then, by time sharing, test each of these \( \Delta_k \) values a fraction \( \beta_k \) of the time, where, of course, \( \sum_{i=1}^{\infty} \beta_i = 1 \). Let \( T_0 \) (a function of the largest prime factor of \( p + 1 \)) be the time required by the algorithm if we were able to choose a \( \Delta \) for which \( (\Delta/p) = -1 \). Then the time-sharing algorithm succeeds with probability 1 in an expected time
\[
T = T_0 \sum_{k=1}^{\infty} \frac{1}{2^k \beta_k}.
\]
We naturally wish to select the $\beta_k$'s in such a way that $T$ is minimized. We note that by Cauchy's inequality

$$\left( \sum_{k=1}^{\infty} \frac{1}{2^k \beta_k} \right) \left( \sum_{k=1}^{\infty} \beta_k \right) \geq \left( \sum_{k=1}^{\infty} \left( \frac{1}{2^k \beta_k} \right)^{1/2} \right)^2;$$

hence,

$$\sum_{k=1}^{\infty} \frac{1}{2^k \beta_k} \geq \left( \sum_{k=1}^{\infty} 2^{-k/2} \right)^2 = \left( \sqrt{2} + 1 \right)^2.$$

Thus, an optimal choice of $\beta_k$ is $\beta_k = 2^{-k/2}(\sqrt{2} - 1)$. Compared to taking 3 equally weighted trials, this method is slower when both succeed (ratio $3 + 2\sqrt{2} : 3$) but it succeeds with probability 1 instead of 7/8.

Both the $p - 1$ and the $p + 1$ methods were implemented on an AMDAHL 470-V7 computer and run with $c$ in (1.3) and (4.4) put equal to 14 and $B_1 = 10^5$, $B_2 = 2 \times 10^5$. Since the $p + 1$ method is much slower (two times slower for Step 1 and about four times slower for Step 2) than the $p - 1$ method, we always ran the $p - 1$ method first on any given $N$ value.

Our programs were run on a total of 497 numbers. From an early version of a table of Brillhart et al. [2] (the most recent version includes the factors found here) we obtained 323 of these 497 numbers. These are factors of integers of the form $b^n - 1$ with $b = 2, 3, 5, 7, 10, 11, 12$. They were obtained by first dividing the main algebraic, including aurifeuillian, factor of $b^n - 1$ by any of its algebraic divisors and then trial dividing by all primes up to $2^{35}$. Those remaining composite factors which were between 42 and 60 digits made up the 323 numbers referred to above. From an as yet unpublished table of factors of Fibonacci numbers, John Brillhart provided the author with the remaining 174 integers. Eighty-four of these are factors of the Fibonacci numbers $U_n (n = 1, 2, 3, \ldots, 1000)$, where $U_{m+1} = U_m + U_{m-1}$ and $U_0 = U_1 = 1$, and 90 are factors of the Lucas numbers $V_n (n = 1, 2, 3, \ldots, 500)$, where $V_{m+1} = V_m + V_{m-1}$ and $V_0 = 2, V_1 = 1$. These numbers are between 41 and 80 digits in length and were known to have no divisors less than $2^{32}$ and no algebraic factors.

\begin{table}[h]
\centering
\begin{tabular}{ccc}
\hline
$b$ & $m$ & $b, mL$ \\
\hline
2 & $4k - 2$ & $2^{2k-1} - 2^k + 1$ \\
3 & $6k - 3$ & $3^{2k-1} - 3^k + 1$ \\
5 & $10k - 5$ & $5^{4k-2} - 5^{2k-1} + 3 \cdot 5^{2k-1} - 5^k + 1$ \\
6 & $12k - 6$ & $6^{4k-2} - 6^{2k-1} + 3 \cdot 6^{2k-1} - 6^k + 1$ \\
7 & $14k - 7$ & $7^{4k-3} - 7^{2k-2} + 3 \cdot 7^{2k-2} - 7^k + 1$ \\
10 & $20k - 10$ & $10^{8k-4} - 10^{4k-3} + 5 \cdot 10^{4k-3}$ \\
& & $-2 \cdot 10^{2k-2} + 7 \cdot 10^{4k-2}$ \\
& & $-2 \cdot 10^{3k-1} + 5 \cdot 10^{2k-1}$ \\
& & $-10^k + 1$ \\
11 & $22k - 11$ & $11^{11k-5} - 11^{4k-4} + 5 \cdot 11^{4k-4}$ \\
& & $-11^{7k-3} - 11^{4k-3} + 11^{3k-2}$ \\
& & $-11^{4k-2} - 11^{3k-1} + 5 \cdot 11^{2k-1}$ \\
& & $-11^k + 1$ \\
12 & $6k - 3$ & $12^{2k-1} - 2^{2k-1} 3^k + 1$ \\
\hline
\end{tabular}
\end{table}

License or copyright restrictions may apply to redistribution; see https://www.ams.org/journal-terms-of-use
The results of running the programs on these numbers are given in Tables 2, 3, and 4. As is done in [2] we use the notation \( b, m - \) and \( b, m + \) to denote the numbers \( b^m - 1 \) and \( b^m + 1 \). The notation \( b, mL \) and \( b, mM \) for aurifeuillians is more complicated. We give their values (taken from [2]) in Table 1.

Note that \( b, mL \) and \( b, mM \) are factors of \( b^m + 1 \) for \( b = 2, 3, 6, 7, 10, 11, 12 \) and \( 5, mL \) and \( 5, mM \) are factors of \( 5^m - 1 \).

In the first column of Tables 2, 3, and 4, we give the number which the composite integer \( N \) divides. In the second column, we give the number of decimal digits in \( N \). In the third and fourth columns, we give the prime factors of \( N \) found by the computer program. A factor followed by an ‘E’ is one which was found by using Step 2 of the appropriate algorithm. An asterisk (*) in the first column is used to denote the fact that once the prime factors found in columns 3 and 4 had been divided into \( N \), the remaining cofactor of \( N \) is prime; hence, we have a complete factorization of the number in column 1. Primality of these numbers was established by using the program described in Williams and Judd [9]. Two asterisks (**) in the first column indicate that this cofactor of \( N \), while composite, was subsequently factored by M. Wunderlich using the continued fraction method of Morrison and Brillhart [7]. It should be noted that the factors found here for \( 10,65 - \) and \( 10,69 - \) were found independently by G. J. Stevens in South Australia. He also used the \( p - 1 \) method.

### Table 2

<table>
<thead>
<tr>
<th>N divides</th>
<th>D</th>
<th>Factor(s) found by ( p-1 ) method</th>
<th>Factor(s) found by ( p+1 ) method</th>
</tr>
</thead>
<tbody>
<tr>
<td>** 2, 173+</td>
<td>46</td>
<td>47635010587</td>
<td>32800634261E</td>
</tr>
<tr>
<td>2, 197+</td>
<td>59</td>
<td>197002597249</td>
<td></td>
</tr>
<tr>
<td>** 2, 209+</td>
<td>54</td>
<td>94803416684681</td>
<td></td>
</tr>
<tr>
<td>2, 235+</td>
<td>56</td>
<td></td>
<td></td>
</tr>
<tr>
<td>** 2, 265-</td>
<td>52</td>
<td>197748738449921</td>
<td>6215074747201E</td>
</tr>
<tr>
<td>2, 291+</td>
<td>54</td>
<td>5636963037465601E</td>
<td></td>
</tr>
<tr>
<td>** 2, 297+</td>
<td>44</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2, 298M</td>
<td>42</td>
<td>14641916303149E</td>
<td></td>
</tr>
<tr>
<td>** 2, 309+</td>
<td>44</td>
<td>2400744386937</td>
<td></td>
</tr>
<tr>
<td>** 2, 351-</td>
<td>52</td>
<td>57189086913727</td>
<td></td>
</tr>
<tr>
<td>** 2, 363-</td>
<td>56</td>
<td>75824014993</td>
<td></td>
</tr>
<tr>
<td>2, 394L</td>
<td>46</td>
<td>152874915601</td>
<td></td>
</tr>
<tr>
<td>2, 410L</td>
<td>49</td>
<td>61213422340181</td>
<td></td>
</tr>
<tr>
<td>** 2, 418L</td>
<td>54</td>
<td>8857714771093</td>
<td></td>
</tr>
<tr>
<td>** 2, 422M</td>
<td>49</td>
<td>2291059412513</td>
<td></td>
</tr>
<tr>
<td>2, 425L</td>
<td>59</td>
<td>84948746297, 6211454306149</td>
<td></td>
</tr>
<tr>
<td>** 2, 458M</td>
<td>50</td>
<td>44185520789894155033573E</td>
<td></td>
</tr>
<tr>
<td>2, 470M</td>
<td>55</td>
<td>87255998201</td>
<td></td>
</tr>
<tr>
<td>** 2, 480+</td>
<td>58</td>
<td>137603804161</td>
<td></td>
</tr>
<tr>
<td>2, 482M</td>
<td>59</td>
<td>76119208744309</td>
<td></td>
</tr>
<tr>
<td>** 2, 558M</td>
<td>54</td>
<td>779844757597</td>
<td></td>
</tr>
<tr>
<td>** 2, 602M</td>
<td>55</td>
<td></td>
<td>236344687097</td>
</tr>
<tr>
<td>2, 610M</td>
<td>55</td>
<td>1621474400951381</td>
<td>87251820842149E</td>
</tr>
<tr>
<td>** 2, 624M</td>
<td>50</td>
<td></td>
<td></td>
</tr>
<tr>
<td>** 2, 654M</td>
<td>54</td>
<td>1193312900149</td>
<td></td>
</tr>
<tr>
<td>2, 750M</td>
<td>48</td>
<td>16806919432501</td>
<td></td>
</tr>
<tr>
<td>** 2, 774M</td>
<td>59</td>
<td>16512828061649</td>
<td></td>
</tr>
<tr>
<td>** 2, 870L</td>
<td>55</td>
<td>4431960464101E</td>
<td></td>
</tr>
<tr>
<td>** 3, 134+</td>
<td>50</td>
<td></td>
<td>719571227339189</td>
</tr>
<tr>
<td>** 3, 136+</td>
<td>46</td>
<td>2670091735108484737</td>
<td></td>
</tr>
</tbody>
</table>
Factors were found for slightly over one quarter (134) of the 497 numbers tested. Most of these factors were found by the \( p - 1 \) method (112 vs. 32). This is what we would expect since (i) the \( p - 1 \) method was used first and (ii) for the numbers in Table 2 there is a built-in bias toward the success of the \( p - 1 \) test. This is because any prime divisor of \( b^m - 1 \) which does not divide any algebraic factor of \( b^m - 1 \) must be of the form \( km + 1 \).
Table 3

<table>
<thead>
<tr>
<th>N</th>
<th>D</th>
<th>Factor(s) found by p-1 method</th>
<th>Factor(s) found by p+1 method</th>
</tr>
</thead>
<tbody>
<tr>
<td>*U_247</td>
<td>46</td>
<td>5307027867738937</td>
<td>409100738617</td>
</tr>
<tr>
<td>*U_307</td>
<td>55</td>
<td>8554266443577</td>
<td>7901346123803597</td>
</tr>
<tr>
<td>*U_313</td>
<td>59</td>
<td>5449038756620509</td>
<td></td>
</tr>
<tr>
<td>*U_323</td>
<td>52</td>
<td>6567762529, 1196762644057</td>
<td></td>
</tr>
<tr>
<td>*U_343</td>
<td>57</td>
<td>758275080626801</td>
<td></td>
</tr>
<tr>
<td>U_361</td>
<td>72</td>
<td>56486966761</td>
<td>43397676601</td>
</tr>
<tr>
<td>U_367</td>
<td>48</td>
<td></td>
<td></td>
</tr>
<tr>
<td>U_377</td>
<td>71</td>
<td>361575655741</td>
<td></td>
</tr>
<tr>
<td>*U_387</td>
<td>45</td>
<td>42136290591640129</td>
<td>14279673833</td>
</tr>
<tr>
<td>*U_403</td>
<td>65</td>
<td>972663078773E</td>
<td></td>
</tr>
<tr>
<td>U_411</td>
<td>51</td>
<td>65688564527041</td>
<td></td>
</tr>
<tr>
<td>*U_421</td>
<td>75</td>
<td>36768087721</td>
<td></td>
</tr>
<tr>
<td>U_455</td>
<td>55</td>
<td>2043118036369</td>
<td></td>
</tr>
<tr>
<td>U_459</td>
<td>57</td>
<td>6936488411701, 59666387254501</td>
<td>1795220677069</td>
</tr>
<tr>
<td>*U_465</td>
<td>42</td>
<td></td>
<td></td>
</tr>
<tr>
<td>*U_483</td>
<td>56</td>
<td></td>
<td></td>
</tr>
<tr>
<td>U_485</td>
<td>74</td>
<td>16892304192301, 511715857773521</td>
<td>1250839826281</td>
</tr>
<tr>
<td>*U_495</td>
<td>51</td>
<td>10069148777</td>
<td></td>
</tr>
<tr>
<td>U_507</td>
<td>63</td>
<td></td>
<td>2192843129417</td>
</tr>
<tr>
<td>*U_531</td>
<td>63</td>
<td></td>
<td></td>
</tr>
<tr>
<td>*U_549</td>
<td>67</td>
<td>5883010433, 80256319951861</td>
<td>10424083697</td>
</tr>
<tr>
<td>U_555</td>
<td>53</td>
<td>49649320649221</td>
<td></td>
</tr>
<tr>
<td>U_567</td>
<td>68</td>
<td>49114912141, 3936504300121</td>
<td></td>
</tr>
<tr>
<td>U_591</td>
<td>79</td>
<td>22221540969737</td>
<td></td>
</tr>
<tr>
<td>U_595</td>
<td>73</td>
<td>8310112721, 9022425301</td>
<td></td>
</tr>
<tr>
<td>U_633</td>
<td>73</td>
<td>41773163881</td>
<td></td>
</tr>
<tr>
<td>*U_675</td>
<td>70</td>
<td>664155895901</td>
<td></td>
</tr>
<tr>
<td>U_765</td>
<td>77</td>
<td>72208475461</td>
<td></td>
</tr>
</tbody>
</table>

The real problem with the $p + 1$ test is the fact that it is quite slow. For our program we found that it was about nine times slower (when used three times for three different trials at $P_0$ value) than the $p - 1$ test. Thus, one should probably use a higher bound for $B_1$ or $B_2$ for the $p - 1$ test than for the $p + 1$ test. We remark here, however, that if we had increased the max($B_1$, $B_2$) to $10^7$, the $p - 1$ test would very likely have found nine of the 32 factors found here by the $p + 1$ test. This is because each of the remaining numbers $p$ is such that a prime which exceeds $10^7$ divides $p - 1$. 

License or copyright restrictions may apply to redistribution; see https://www.ams.org/journal-terms-of-use
<table>
<thead>
<tr>
<th>N</th>
<th>D</th>
<th>Factor(s) found by p-1 method</th>
<th>Factor(s) found by p+1 method</th>
</tr>
</thead>
<tbody>
<tr>
<td>*V254</td>
<td>50</td>
<td>52206665291</td>
<td>347366417511089201</td>
</tr>
<tr>
<td>*V271</td>
<td>46</td>
<td>256205941501</td>
<td></td>
</tr>
<tr>
<td>*V283</td>
<td>56</td>
<td>14323688738249</td>
<td></td>
</tr>
<tr>
<td>V299</td>
<td>46</td>
<td>70963651961</td>
<td></td>
</tr>
<tr>
<td>V302</td>
<td>63</td>
<td>12441241017224321</td>
<td></td>
</tr>
<tr>
<td>V304</td>
<td>45</td>
<td>54184296181</td>
<td></td>
</tr>
<tr>
<td>V331</td>
<td>64</td>
<td>404112157123</td>
<td></td>
</tr>
<tr>
<td>V338</td>
<td>57</td>
<td>68520477202692467E</td>
<td></td>
</tr>
<tr>
<td>*V346</td>
<td>72</td>
<td>389132418765256896001</td>
<td></td>
</tr>
<tr>
<td>*V352</td>
<td>67</td>
<td>316590102769</td>
<td></td>
</tr>
<tr>
<td>V358</td>
<td>75</td>
<td>19997474011</td>
<td></td>
</tr>
<tr>
<td>V367</td>
<td>71</td>
<td>26024651929</td>
<td>18736753266019E</td>
</tr>
<tr>
<td>*V369</td>
<td>48</td>
<td>3827019260681</td>
<td></td>
</tr>
<tr>
<td>V374</td>
<td>59</td>
<td>1769526527</td>
<td></td>
</tr>
<tr>
<td>*V384</td>
<td>42</td>
<td>54975368761</td>
<td></td>
</tr>
<tr>
<td>*V390</td>
<td>41</td>
<td>64690797641</td>
<td></td>
</tr>
<tr>
<td>V406</td>
<td>67</td>
<td>33637840386809</td>
<td></td>
</tr>
<tr>
<td>*V413</td>
<td>67</td>
<td>722601651307</td>
<td></td>
</tr>
<tr>
<td>*V418</td>
<td>59</td>
<td>316722762859</td>
<td></td>
</tr>
<tr>
<td>V419</td>
<td>79</td>
<td>1006118006507</td>
<td></td>
</tr>
<tr>
<td>V426</td>
<td>55</td>
<td>386610981607</td>
<td></td>
</tr>
<tr>
<td>*V428</td>
<td>74</td>
<td>28677143808961</td>
<td></td>
</tr>
<tr>
<td>*V460</td>
<td>69</td>
<td>10812035185331</td>
<td></td>
</tr>
<tr>
<td>*V469</td>
<td>70</td>
<td>49721203549E</td>
<td>6430515046741</td>
</tr>
<tr>
<td>*V477</td>
<td>66</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

6. Acknowledgements. The author wishes to thank John Brillhart and Sam Wagstaff for making available to him the tables referred to in Section 5.

Department of Computer Science
University of Manitoba
Winnipeg, Manitoba, Canada R3T 2N2

2. John Brillhart, D. H. Lehmer, J. L. Selfridge, Bryant Tuckerman & S. S. Wagstaff, Jr., Factorizations of $b^n \pm 1$, $b = 2, 3, 5, 7, 10, 11, 12$ Up to High Powers. (To appear.)