Some Negative Results Concerning Prime Number Generators

Paul Pritchard

TR 83-542
February 1983

Department of Computer Science
Cornell University
Ithaca, New York 14853

Abstract. Programs due to Wirth and Misra for generating the prime numbers up to a specified limit are investigated. It is shown that Wirth's program is incorrect according to three increasingly weak criteria, and a composite number is exhibited that the program accepts as prime. This is the smallest known counter-example, and could not have been found by the usual method of program testing—the program would run for trillions of years on the fastest computer before reaching it! Closely related counter-examples are given to a conjecture of Misra concerning his program.

1. Introduction

Wirth [13] and Misra [8] presented programs for the problem of generating all (and only) the prime numbers up to a given limit.

The program presented by Wirth originated with Dijkstra [3]. It works as follows. First the only even prime 2 is accepted. Then each odd integer $x = 3, 5, 7, 9, \cdots$ is taken in turn and tested for primality. For each x, the odd primes $p = 3, 5, 7, 11, \cdots$ are taken in turn, until either $p > \sqrt{x}$, in which case x is accepted as prime, or $p \mid x$, in which case x is rejected as composite.

The program maintains two arrays. The first is an array p of all accepted primes, i.e. those less than x:

$$p[j] = p_j, \quad 1 \leq j \leq \pi(x-1),$$

where p_j denotes the j'th prime and $\pi(x)$ denotes the number of primes $\leq x$. The other array V is used to avoid dividing x by $p[j]$. The intention is that the assertion

$$V[j] = \textit{the least multiple of } p_j \textit{ not less than } x \tag{1}$$

should hold whenever x is to be divided by $p[j]$, $2 \leq j \leq \pi(\sqrt{x})$. For then

$$p_j \mid x \quad \equiv \quad V[j] = x \tag{2}$$

In Dijkstra's program $V[j]$ is initialized with $p[j]^2$ as soon as $x \geq p[j]^2$. (1) is guaranteed to hold before the division test by preceding the test with the loop

$$\textbf{while } V[j] < x \textbf{ do } V[j] := V[j] + p[j]$$

Wirth instead used

$$\textbf{if } V[j] < x \textbf{ then } V[j] := V[j] + p[j]$$

However this oversight does not give an obviously incorrect program—we have used it to generate the primes not greater than one million, which it does without error, and we could not decide the issue when we discovered it in 1979. This raises the possibility that the program is fortuitously correct, and that it may reveal new information, or at least provide evidence for plausible new conjectures, about the prime numbers.

Meanwhile, Misra [8] presented a different (and much more efficient) program to generate the primes up to N, and showed that a certain simplification in the coding could be achieved contingent on the truth of a statement that he conjectured to be true. The same coding simplification could also be used in the fastest known prime number generator, our *wheel sieve* of [10]. Misra's conjecture can be given a familiar setting: it amounts to the claim that in no pass of Eratosthenes' sieve are successive remaining numbers removed.

These questions concerning Wirth's program and Misra's conjecture turn out to be very closely related. In §2 we reformulate them in the setting of wheels [10], and show how two celebrated results in number theory together imply that Wirth's program is incorrect. In §3 we report on a successful search for counter-examples that provides explicit composite numbers accepted by Wirth's program and disposes of Misra's conjecture in the negative.

2. Wheels and their gaps

In [10] we used the notion of *wheels* to derive the most efficient known solution to the problem of generating the prime numbers up to some limit. We recall the key definitions:

$$R(m) : \{x \mid 1 \leq x \leq m \text{ and } \gcd(x,m)=1\}, \quad (m > 0)$$

$$\Pi_k : \prod_{j=1}^{k} p_j, \quad (k > 0)$$

$$W_k : R(\Pi_k), \quad (k > 0)$$

$R(m)$ is a reduced residue system (mod m) (see, e.g., [7]). W_k is the k'th wheel; it represents the pattern of numbers that are not divisible by any of the first k primes—these numbers repeat modulo Π_k.

We first show that Wirth's program **D'** does not exactly simulate Dijkstra's program **D**.

Theorem 1. In **D'**, (1) does not always hold before dividing x by p_j.
Proof. In [10] we showed that for each $k > 1$ there is a gap of exactly $2 \cdot p_{k-1}$ on W_k, which exceeds p_{k+1} for $k > 4$. Let a, b be successive members of W_k with $b-a = 2 \cdot p_{k-1}$. Since $\gcd(p_{k+1}, \Pi_k) = 1$, there is a unique number $a' \in W_{k+1}$ such that $a' \equiv a \pmod{\Pi_k}$ and $p_{k+1} \mid a'$. Now when $x = a'$, $V[k+1] \leq a'$, as the elements of V are never too big. But since every number properly between a' and $a' + 2 \cdot p_{k-1}$ is divisible by a prime $< p_{k+1}$, $V[k+1]$ will not be incremented until x reaches $a' + 2 \cdot p_{k-1}$. But then

$$V[k+1] \leq a' + p_{k+1} < a' + 2 \cdot p_{k-1} = x$$

for $k > 4$, and (1) fails with $j = k+1$. □

Theorem 1 shows that **D'** is incorrect in the sense that it does not work as intended. However the purpose of array V is to allow the division test to be replaced by an equality test, and the situation occurring in the proof causes no difficulties because $p_{k+1} \nmid a' + 2 \cdot p_{k-1}$. It may still be the case that (2) always holds before the test.

In [9] we presented a necessary and sufficient condition C for this weaker correctness criterion to be satisfied. In the language of wheels C takes a simple form:

C: Let n and m be successive members of W_k, with $n > 1$. Then

$$\left| W_k \cap ((n+j-1) \cdot p_{k+1}, m \cdot p_{k+1}) \right| \geq m-n-j, \quad 1 \leq j < m-n.$$

Here (a,b) denotes the set of integers in the open interval between a and b, and $|S|$ denotes the cardinality of set S. The appropriateness of C is not difficult to see: $n \cdot p_{k+1}$ and $m \cdot p_{k+1}$ are successive composite numbers with least prime factor p_{k+1}. Suppose the lesser of these is treated correctly by the program, so that $V[k+1] = n \cdot p_{k+1}$ when $x = n \cdot p_{k+1}$. As x takes on successive values, $V[k+1]$ is incremented by p_{k+1} precisely when both $x \in W_k$ and $V[k+1] < x$ are true. When x reaches $m \cdot p_{k+1}$, $V[k+1] \geq x - p_{k+1}$ must hold so that the final conditional increment makes $x = V[k+1]$ for the test. So $V[k+1]$ needs to be incremented $m-n-1$ times as x increases over $(n \cdot p_{k+1}, m \cdot p_{k+1})$. Now if C holds, $V[k+1]$ will be incremented when $x = \min\{ W_k \cap ((n+j-1) \cdot p_{k+1}, m \cdot p_{k+1})\}$, for $1 \leq j \leq m-n-1$, which is sufficiently often. If C does not hold it is clear that insufficient increments are done. Finally note that the requirement that $m,n \in W_k$ is not essential as the argument only depends on their residues $(\text{mod } \Pi_k)$.

But **D'** fails in this respect too:

Theorem 2. C is false for all sufficiently large k.
Proof. A result of Rankin/Schönhage [12] shows that the maximum gap g_k between successive members of W_k satisfies

$$g_k \geq \frac{(e^{\gamma} + o(1)) \cdot p_k \cdot \log p_k \cdot \log\log\log p_k}{(\log\log p_k)^2}$$

Let a,b be successive members of W_k with $b-a = g_k$. Since $\gcd(p_{k+1}, \Pi_k) = 1$, there is a unique number $b' \in W_{k+1}$ such that

$$b' \equiv b \ (\text{mod } \Pi_k) \quad \text{and} \quad p_{k+1} \mid b'. \tag{3}$$

Now consider C with $m = b'/p_{k+1}$. C requires

$$\left| W_{k+1} \cap (b' - 2 \cdot p_{k+1}, b') \right| \geq 1,$$

but this is impossible for k sufficiently large as $2 \cdot p_{k+1} = o(g_k)$. □

Theorem 2 constructs a situation in which x is not recognized as a multiple of its smallest prime factor, say p_{j_1}. But it is possible that x will nevertheless be recognized as a multiple of another prime factor p_{j_2} with $1 \le j_1 < j_2 \le \pi(\sqrt{x})$, and that the program never accepts a composite number. We proceed to dispose of this last possibility.

Theorem 3. D' accepts composite numbers.
Proof. We extend the proof of theorem 2. Consider the sequence of numbers

$$b', \quad b' + \Pi_{k+1}, \quad b' + 2{\cdot}\Pi_{k+1}, \quad \cdots$$

For each number b'' in the sequence, (3) holds with b'' for b'. So if b''/p_{k+1} is prime, b'' will be incorrectly accepted as prime. The numbers b''/p_{k+1} form an arithmetic progression with first term b'/p_{k+1} and common difference Π_k. But $\gcd(b'/p_{k+1}, \Pi_k) = 1$, so by Dirichlet's famous theorem (see [6] for a proof) there are infinitely many prime members of this arithmetic progression. □

In [3], Dijkstra noted that since x is odd, $V[j]$ could be taken as the least *odd* multiple of p_j not less than x and incremented by $2{\cdot}p[j]$. The Rankin/Schönhage result shows that **D'** would still fail even if this doubled increment were used.

Misra's conjecture is equivalent to the assertion that no successive members of W_k differ by a multiple of p_{k+1}. As we pointed out in [10], in view of the Rankin/Schönhage result it seems unlikely that this conjecture is true.

We proceed to find counter-examples to Misra's conjecture, and in the process obtain composite numbers accepted by Wirth's program.

3. A search for counter-examples

To refute Misra's conjecture we need only find successive numbers $a, b \in W_k$ with $b-a = 2{\cdot}p_{k+1}$. Such numbers would also eventually lead to a composite number accepted by Wirth's program by way of the constructions in the proofs of theorems 2 and 3.

It is known (see [1]) that $g_k > 2{\cdot}p_{k-1}$ only for $p_k = 23{,}37$ and for $p_k \ge 43$, i.e. for $k = 9{,}12$ and for $k \ge 14$. Computing g_k seems to be a very difficult task. The following easily verified criteria are helpful for small k:

(i) If $2{\cdot}p_{k+1} > g_k$ then $g_{k+1} =$ the maximum sum of two successive gaps on W_k.

(ii) If $2{\cdot}p_{k+1} > g_{k+1}$ then $g_{k+2} =$ the maximum sum of three successive gaps on W_k.

By computing W_7 and using these criteria, we found that $g_9 = 40 < 2{\cdot}p_{10} = 58$. This suggests that a considerably larger value than $k = 9$ is needed for a counter-example. But computing W_k is a time-consuming task, as there are $\phi(\Pi_k) = \prod_{j=1}^{k}(p_j-1)$ members. It is obvious that we cannot afford to compute entire wheels.

Fortunately it is not necessary to compute the *maximum* gap on W_k—we need only try to find a single *large* gap of size $2{\cdot}p_{k+1}$. We can do this by playing the following game.

> Start with the set of integers between 1 and $2 \cdot p_{k+1}-1$ inclusive. For each $j=1,2,3,\ldots,k$, choose a residue class (mod p_j) and remove all numbers belonging to that residue class. The residue class can be freely chosen subject to the constraint that it does not contain 0 or $2 \cdot p_{k+1}$. The game is won if all members of the set are removed.

The Chinese remainder theorem guarantees that if we win the game then there is a gap of exactly $2 \cdot p_{k+1}$ on W_k.

The Rankin/Schönhage proof suggests a strategy for playing the game. The proof goes as follows. The sequence $p_1, p_2, \ldots, p_k$ is divided into three sections. The primes p_j in the middle section remove the numbers congruent to 0 (mod p_j). Then the primes in the first section are taken in increasing order, and for each prime a residue class is deleted that contains a maximum number of remaining numbers. Finally, the primes in the third section are shown to outnumber the remaining numbers, so that each of these primes can be used to remove a single number.

We adapted this proof as follows. An interactive program was written that "asked for" three parameters, viz. k and two numbers between 1 and k that defined the three sections of the first k primes. Subject to the constraints of our game (which rule out the precise Rankin/Schönhage construction) the primes in the middle and then first sections were taken in order and each used to remove an optimal number of remaining numbers. In case of more than one largest residue class, a random choice was made, permitting many trials with the same parameters. The game was won if the primes in the third section were at least as numerous as the numbers remaining after using the other primes.

Many games were won, including one only with $k = 19$ (the smallest successful value). The outcome of this game is shown in table 1 below as a sequence of $p_{k+1}-1 = 70$ numbers. The j'th number in this sequence is the smallest prime that removes the number $2 \cdot j$. The prime 2 removes all odd numbers. The four circled primes are those that remove only one number. These are the primes in the third section; they may be permuted freely.

3	7	5	3	37	23	3	5	7	3
13	47	3	11	17	3	43	5	3	19
41	3	5	13	3	31	29	3	23	7
3	17	5	3	67	11	3	5	19	3
59	37	3	7	53	3	11	5	3	13
7	3	5	61	3	29	31	3	47	43
3	41	5	3	7	17	3	5	11	3

Table 1: a won game of gap construction with $k = 19$.

It follows immediately that Misra's conjecture is false. To each of the $4! = 24$ permutations of the circled primes there corresponds two gaps on W_{19}: one for the least prime factors taken in the order shown in table 1, and one for the reverse order. By using the Chinese remainder theorem we found all 24 pairs $a', b' \in W_{20}$ with least prime factor of $b' = 71$, $a' = b' - 2 \cdot 71$ and least prime factor of $a' + 2 \cdot j$ as specified by the j'th number in table 1 for all j, $1 \leq j \leq 70$. The other 24 pairs are easily found since they have the form $\Pi_{20} - b'$, $\Pi_{20} - a'$ where a', b' range over the first set of pairs. The smallest of these 48 counter-examples to Misra's conjecture—that no successive remaining numbers a', b' are removed in a pass of Eratosthenes' sieve—is

$$a' = 78962232457704773453418\,19, \quad b' = 7896223245770477345341961.$$

To find a composite number accepted by **D'**, we need only find a prime value of $b'/71 + n \cdot \Pi_{19}$ for one of the 48 values of b' and some $n \geq 0$. Five of the numbers $b'/71$ are base-3 pseudoprimes, i.e., satisfy Fermat's little theorem, that $x^{n-1} \equiv 1 \pmod{n}$ if n is prime and $\gcd(x, n) = 1$, with $x = 3$. To prove that these numbers p are prime, as we suspect to be the case, it is sufficient to find an integer x such that

$$x^{p-1} \equiv 1 \pmod{p}$$

and, for all prime divisors q of $p - 1$,

$$x^{(p-1)/q} \not\equiv 1 \pmod{p}$$

Then x is a primitive root (mod p). See e.g. [5, p. 375].

The process is repeated if necessary for the large prime factors of $p-1$. We established the primality of the smallest probable prime $p = b'/71$ using the method of Brent/Pollard [2] to carry out the factorizations involved. The details given below in table 2 enable a quick check that $b'/71$ is indeed prime, assuming the primality of numbers $< 10^7$ is easy to check.

$b' = 60135101134368910576192061$

$p = b'/71 = 846973255413646627833691$

prime factorization of $p - 1$: $2 \cdot 3^2 \cdot 5 \cdot 47 \cdot 65119 \cdot 1922567 \cdot 1599337511$

least positive primitive root (mod p) = 2

subsidiary information:

$1599337511 - 1 = 2 \cdot 5 \cdot 159933751$

least positive primitive root (mod 1599337511) = 7

$159933751 - 1 = 2 \cdot 3 \cdot 5^4 \cdot 42649$

least positive primitive root (mod 159933751) = 3

Table 2: a proof of primality for $b'/71$

D' incorrectly accepts the composite number b' above as prime. This is the smallest counter-example that we know of. (We played the game of gap construction without the restrictions needed for a gap of *exactly* $2 \cdot p_{k+1}$, but did not find another example with $k \leq 19$.) Since only a short time was spent playing this game, it is very doubtful that this is the smallest counter-example. Nevertheless, we expect that the smallest counter-example will be a similarly large number: certainly one that could not be found by running the program and testing each accepted number for primality. (We have shown in [9] that to examine the numbers up to N with program **D** takes $\Theta(N^{1.5}/\log^3 N)$ arithmetic operations, whence program **D'** takes $\Omega(N^{1.5}/\log^3 N)$ arithmetic operations.)

4. Final remarks

The normal method of program testing could not discover our counter-example to the correctness of Wirth's program. It might be argued that this is of no import anyway, because the program would probably have to run for aeons (literally) before accepting a composite number. A similar remark might be made with respect to the coding variation in Misra's program that depends on his false conjecture.

We would take exception to such opinions (which are admittedly from a man of straw) for two main reasons. Firstly, there is no guarantee that the smallest counter-example is not in practical range, and it would seem difficult to prove that a particular counter-example was smallest unless it *was* in practical range. Secondly, and more importantly, the point of Wirth's program was not to be an efficient generator of the prime numbers. Indeed, it is far less efficient than the best known practical methods [11]. The program was instead presented as an exercise in systematic program development, and it is in this context that we have examined it. We have pursued the consequences of a slipup which anyone might have made, because we believe our programs, and especially our exemplary programs, must be correct and be seen to be correct—failing which they should be seen to be incorrect.

References

[1] Brauer, A. Question concerning the maximum term in the diatomic series—a reply. *Amer. Math. Monthly 40*, 7 (1933), 409-410.

[2] Brent, R. P. An improved Monte Carlo factorization algorithm. *BIT 20* (1980), 176-184.

[3] Dijkstra, E. W. Notes on structured programming. In Dahl, O. -J., Hoare, C. A. R. and Dijkstra, E. W.: *Structured Programming*. Academic Press, New York, 1972, 1-82.

[4] Hardy, G. H. and Wright, E. M. *An Introduction to the Theory of Numbers*. 5th Ed., Oxford University Press, Oxford, England, 1979.

[5] Knuth, D. E. *The Art of Computer Programming, vol. 2: Seminumerical Algorithms*. 2nd Ed., Addison-Wesley, Reading, Massachusetts, 1981.

[6] Le Veque, W. J. *Topics in Number Theory, vol. 2*. Addison-Wesley, Reading, Massachusetts, 1955.

[7] Le Veque, W. J. *Fundamentals of Number Theory*. Addison-Wesley, Reading, Massachusetts, 1977.

[8] Misra, J. An exercise in program explanation. *ACM Trans. Program. Lang. Syst. 3* (1981), 104-109.

[9] Pritchard, P. On the prime example of programming. In *Language Design and Programming Methodology: Lecture Notes in Computer Science 79*, Springer-Verlag, Berlin Heidelberg and New York, 1980, 85-94.

[10] Pritchard, P. Explaining the wheel sieve. *Acta Informatica 17* (1982), 477-485.

[11] Pritchard, P. Fast compact prime number sieves (among others). *J. Algorithms*, to appear.

[12] Rankin, R. A. The difference between consecutive prime numbers V. *Proc. Edinburgh Math. Soc. (2) 13* (1962/63), 331-332.

[13] Wirth, N. *Systematic Programming: An Introduction*. Prentice-Hall, Englewood Cliffs, New Jersey, 1973.