MATHEMATICS OF COMPUTATION
VOLUME 56, NUMBER 194
APRIL 1991, PAGES 859-865

# AMICABLE PAIRS OF THE FORM $(i, 1)$

PATRICK COSTELLO

ABSTRACT. A method is given for finding amicable pairs of a certain type. When implemented, thirteen new amicable pairs were discovered. Using methods for finding new pairs from known pairs, the thirteen new pairs generated 64 other new pairs.

## 1. INTRODUCTION

An *amicable pair* is a pair of distinct positive integers $(a, m)$ where each integer is the sum of the proper divisors of the other. If we let $\sigma(x)$ denote the sum of all divisors of $x$, then saying that $(a, m)$ is an amicable pair is equivalent to saying $\sigma(a) = \sigma(m) = a + m$.

In the 1700's Leonhard Euler made a systematic study of many of the forms that amicable pairs have. He developed several methods for finding pairs and used his methods to discover 59 amicable pairs [3]. One particular form that Euler discovered was the following:

$$(es, ep),$$

where $s$ is the product of distinct primes not dividing the common factor $e$ and $p$ is a single prime not dividing $es$. Recently, pairs of this form have been labelled as type $(i, 1)$, where $i$ is the number of primes involved in $s$ and 1 refers to the fact that $p$ is a single prime [8]. It is not difficult to see that $i$ must in fact be greater than 1. A good question is, how big can $i$ get?

The only pairs known prior to Euler were three (2,1) pairs:

$$(220, 284) = (2^2 \cdot 5 \cdot 11, 2^2 \cdot 71) \quad \text{(Pythagoras)},$$
$$(17296, 18416) = (2^4 \cdot 23 \cdot 47, 2^4 \cdot 1151) \quad \text{(Fermat)},$$
$$(9363584, 9437056) = (2^7 \cdot 191 \cdot 383, 2^7 \cdot 73727) \quad \text{(Descartes)}.$$

Among Euler's pairs were thirteen $(2, 1)$ pairs, including the first known odd pairs. In 1946 Edward Escott [4] produced a list of 219 new pairs that contained seven $(3, 1)$ pairs. In 1968 Elvin Lee [5] published the list of all known pairs to that point, including six $(4, 1)$ pairs that he discovered. In 1982 Herman te Riele [7] found the first $(5, 1)$ pair.

Received September 6, 1989.
1980 *Mathematics Subject Classification* (1985 *Revision*). Primary 11A25.
*Key words and phrases.* Amicable pair.

It is the intent of this paper to show one method for discovering pairs of the type $(i, 1)$ and to list some results obtained through a computer implementation of this method.

## 2. The algorithm

Suppose we are searching for an amicable pair of the form $(es, ep)$, where $p$ is a prime, $s$ is a product of at least two distinct primes, $p$ does not divide $s$, and $p$ and $s$ are relatively prime to $e$. Since $\sigma$ is a multiplicative function, the condition that $\sigma(es) = \sigma(ep)$ implies that $\sigma(s) = \sigma(p) = p + 1$. Hence, $\sigma(s) - 1$ must be prime. The condition that $\sigma(es) = es + ep$ implies that

$$\frac{\sigma(e)}{e} = \frac{s+p}{\sigma(s)} = \frac{s+\sigma(s)-1}{\sigma(s)}.$$

These observations lead to the following two-step algorithm:

```
Step 1. Choose a range of $s$-values.
  For each $s$
      If $s$ is not prime then
          Calculate $\sigma(s)$
          If $\sigma(s) - 1$ is prime then
          Save $s$ and $(s + \sigma(s) - 1)/\sigma(s)$ in a list
Step 2. Choose a range of $e$-values.
  For each $e$
      Calculate $\sigma(e)/e$
      Search the list created in Step 1 for a match
      If $\sigma(e)/e = (s + \sigma(s) - 1)/\sigma(s)$ then
          If ($e$ and $s$ are relatively prime) and
              ($e$ and $\sigma(s) - 1$ are relatively prime) then
              Print that $es$, $e(\sigma(s) - 1)$ is an amicable pair.
```

Let us see how this algorithm works on a small example. In Step 1 choose $s$ to be odd values ranging from 9 to 99. (See the beginning of §4 for a reason to choose just odd values.) The list of $s$- and $(s+\sigma(s)-1)/\sigma(s)$-values to save is: 15, 19/12; 21, 13/8; 33, 5/3; 35, 41/24; 51, 61/36; 55,7/4 ; 57, 17/10; 63, 83/52; 65, 37/21; 85, 16/9; 93, 55/32. When $e = 4$ in Step 2, we would search the list for $\sigma(4)/4 = 7/4$, which corresponds to $s = 55$. Since 4 and 55 are relatively prime, and 4 and 71 are relatively prime, we get the Pythagorean pair (220, 284). When $e = 819$ in Step 2, we get the Euler pair $(3^2 \cdot 7 \cdot 13 \cdot 5 \cdot 17, 3^2 \cdot 7 \cdot 13 \cdot 107)$.

Technically, this algorithm will find more than just $(i, 1)$ pairs because $s$ was not restricted to be a product of distinct primes. For example, $63 = 3^2 \cdot 7$ was an $s$-value that got put into the list, but 63 is not the product of distinct primes. If a pair $(e \cdot 63, e \cdot 103)$ were to be discovered, it would be labelled an irregular or exotic pair. Such a pair would be marked type X. There are very few known exotic pairs that would be produced from this algorithm, but they should not be discounted altogether. If $s$-values in Step 1 ranged from $5 \cdot 10^9$ to $6 \cdot 10^9$, we would save $s = 17^2 \cdot 59 \cdot 315461$. Letting $e = 2^3 \cdot 37$ in Step 2, we would obtain the known exotic pair

$$(2^3 \cdot 37 \cdot 17^2 \cdot 59 \cdot 315461, 2^3 \cdot 37 \cdot 5810810039).$$

## 3. Computational details

This algorithm requires that three functions be readily available. First, one needs a function that tests if the input to the function is prime or not. If the numbers to be tested for primality are not extremely large, a sophisticated primality test is not really necessary. Trial division by 2, 3, and every odd number after 3 until a divisor is found or the square root of the input is exceeded would be a suitable implementation of this function.

The second function needed is one to calculate the sum of all the divisors of the input. Suppose $x$ is the input. A simple way to compute $\sigma(x)$ is to initialize $s$ to be 0 and for each divisor $d$ between 1 and $\sqrt{x}$ to increment $s$ by $d + x/d$ (except when $d = \sqrt{x}$). Since this method requires $\sqrt{x}$ tests for division for every input $x$, and since the algorithm calls for submitting many values to this function, one might want a more efficient implementation.

The following method is a more efficient method for computing $\sigma(x)$ that takes advantage of the multiplicative nature of $\sigma$, i.e., if $x = p_1^{a_1} p_2^{a_2} \cdots p_r^{a_r}$, where the $p_i$ are distinct primes, then

$$\sigma(x) = \sigma(p_1^{a_1})\sigma(p_2^{a_2}) \cdots \sigma(p_r^{a_r}).$$

Initialize $s$ to be 1. Do trial division of $x$ by 2, 3, and odd numbers after 3. When a prime divisor $p$ is found, repeatedly divide $x$ by $p$ to determine the exact power of $p$ that divides $x$. At the same time sum up the powers of $p$ including $p^0$. This can be accomplished with the following process:

```
powersum ← 1
power ← 1
while $p$ divides $x$ do
    power ← power $*$ $p$
    powersum ← powersum+power
    $x \leftarrow x/p$
endloop
```

When the loop is exited, powersum holds the value $\sigma(p^a)$, where $a$ is the exact power of $p$ that divides $x$. Multiply this onto the variable $s$. Since $x$ has been reduced by dividing out $p^a$, the next divisor to be found in the trial divisions will be a prime. When $x$ becomes 1 or the trial divisor becomes greater than $\sqrt{x}$, we exit the function. In the case that $x$ becomes 1, $\sigma(x)$ is in $s$. In the case that the trial divisor is greater than $\sqrt{x}$, $\sigma(x)$ is $s$ times $(x + 1)$.

As an example of how efficient this implementation for calculating $\sigma(x)$ is, consider an input of $x = 378125$. Do trial divisions by 2, 3, and 5 and discover that 5 is a divisor. One stays in the loop described above five times for $p = 5$ since $5^5$ divides $x$. When the loop is exited, $s = \text{powersum} = 1 + 5 + 5^2 + 5^3 + 5^4 + 5^5 = 3906$ and $x = 121$. Do trial division by 7, 9, and 11 and discover that 11 is a divisor. One stays in the loop described above two times for $p = 11$ and upon leaving, $s = 3906 * (1 + 11 + 11^2) = 519498$. Since $x$ has become 1, one quits. Note that the last trial division done this way is by 11 versus doing trial divisions (as suggested in the simple implementation) by all integers less than $\sqrt{x} = 614.9$.

The third function the main algorithm needs is a function to compute the greatest common divisor of two integers. Step 2 requires this to determine when two numbers are relatively prime. This function is most efficiently implemented using the Euclidean Algorithm [6].

One can separate the two steps of the main algorithm into two separate computer programs. The program that implements Step 1 should save the list of $s$- and $(s + \sigma(s) - 1)/\sigma(s)$-values in a data file. This allows for various ranges of $e$-values to be compared with a single range of $s$-values.

Because of the internal computer inaccuracies involved in real number comparisons of the form "if $\sigma(e)/e = \text{listvalue}$ then ... ", it is advantageous to save the fractions $(s + \sigma(s) - 1)/\sigma(s)$ as a pair of integers representing the numerator and denominator. We use the greatest common divisor function to find the gcd of $s + \sigma(s) - 1$ and $\sigma(s)$, say $g$. Then we actually save the integer pair $((s+\sigma(s)-1)/g, \sigma(s)/g)$. In the program that implements Step 2, we similarly reduce $\sigma(e)/e$ to its reduced form, say $N/M$, and search the data file saved in Step 1 for a match with the integer pair $(N, M)$. In the case of a match, note that since $\sigma(s)$ is not really in the data file, it must be recomputed in order to test if $e$ and $\sigma(s) - 1$ are relatively prime.

It turns out that the data file saved in Step 1 is a large file for any reasonable range of $s$-values. Consequently, doing a linear search of the data file in Step 2 for each $e$-value would make the algorithm quite slow. One solution to this problem is the following idea. At the beginning of the Step 2 program read in the data file saved by Step 1 and store the values in a table by applying a hash function to the numerator and denominator pairs. Then as one goes through the $e$-values, a quick check for a match can be done by applying the same hash function to the numerator and denominator of the reduced form of $\sigma(e)/e$.

## 4. Restrictions and results

Restricting $s$-values in Step 1 to odd integers is based on the conjecture that there are no odd-even amicable pairs. (It has been shown that there are no odd-even (2,1) pairs [2].) For if $s$ is even and $\sigma(s) - 1$ is an odd prime, and $e$ and $s$ are relatively prime, then $e$ must be odd and so $(es, e(\sigma(s) - 1))$ is an odd-even pair.

A program was written in Pascal to implement Step 1 and run on a VAX 11/785 minicomputer with all odd $s$-values ranging from 9 to $10^5$. While not requiring much computer time, the program used up a lot of disk space to store the list of $s$- and $(s + \sigma(s) - 1)/\sigma(s)$-values. Just within this particular range there are 10360 different $s$-values to save. In order to discover new amicable pairs one needs to find $es$-values greater than $10^{10}$ because all amicable pairs with smaller integer less than $10^{10}$ are known [8]. Consequently, we wanted to let $s$-values get fairly large (at least to $10^7$), but file space limitations (as well as main memory limits on the size of the hash table in Step 2) prohibit saving all $s$-values less than $10^7$. It was decided to save only the $s$-values less than $10^7$ that contained a numerator $< 50000$ when $(s + \sigma(s) - 1)/\sigma(s)$ was reduced.

It took approximately 16 hours of CPU time on the VAX to create the data file that contains all odd $s$-values satisfying these restrictions.

A second program was written in Pascal to implement Step 2 and run piecemeal on $e$-values ranging from 4 to $5 \cdot 10^7$. The program utilized the idea of applying a hash function to each numerator-denominator pair from the data file of Step 1 and storing $s$ and the pair in a table. The program also checks for the possibility of several $s$-values having the same numerator-denominator pair when a match with $\sigma(e)/e$ is found. For example, when $s$ is 205, 25705, and 35905, one gets the numerator-denominator pair (38, 21). Then $e = 5733$ actually matches with three $s$-values and one obtains three different amicable pairs.

This second program was very fast when the range of $e$-values was small, but slowed down considerably on larger values. To do just the final range of $e$-values chosen, $4.5 \cdot 10^7 \leq e \leq 5 \cdot 10^7$, the program required nearly 14 hours of CPU time on the VAX. While 49 previously known pairs were obtained by this second program, twelve new (3,1) pairs and one new (4,1) pair were discovered. Their factorizations are listed in Table 1. Notice that pairs 7 and 9 and pairs 10 and 11 demonstrate the fact that several different $e$-values can also match up with a single $s$-value.

In addition, Table 1 lists the number of "daughter" pairs generated by each pair. These are new amicable pairs that are found by applying a few clever tricks to a known pair and arriving at new pairs that have much in common with the known pair. 60 of the daughter pairs were generated using the ideas described in [9]. After sending the "mother" and "daughter" pairs to H. te Riele, he was able to find the additional four "daughter" pairs using some of his breeding programs.

Pairs 5, 6, and 12 also generate Thabit rules as described by W. Borho in [1]. This is an additional method for generating new amicable pairs from known pairs that depends heavily on primality testing of large numbers. When H. te Riele checked the Thabit rules by testing primality of all terms less than $10^{100}$, he found that no new pairs were generated .

TABLE 1
*New amicable pairs of type* $(i, 1)$

| type | pairs | number of daughter pairs |
|---|---|---|
| (3, 1) | $3 \cdot 5^2 \cdot 11 \cdot 31 \cdot 7 \cdot 67 \cdot 2749$<br>$3 \cdot 5^2 \cdot 11 \cdot 31 \cdot 1495999$ | 1 |
| | $3^4 \cdot 5 \cdot 11^3 \cdot 41 \cdot 431 \cdot 439$<br>$3^4 \cdot 5 \cdot 11^3 \cdot 7983359$ | 13 |
| | $3^3 \cdot 5^2 \cdot 31^2 \cdot 17 \cdot 29 \cdot 61$<br>$3^3 \cdot 5^2 \cdot 31^2 \cdot 33479$ | 3 |
| | $3^6 \cdot 5 \cdot 13 \cdot 17 \cdot 149 \cdot 1637$<br>$3^6 \cdot 5 \cdot 13 \cdot 4422599$ | 1 |
| | $3^2 \cdot 5 \cdot 13 \cdot 463 \cdot 11 \cdot 19 \cdot 6481$<br>$3^2 \cdot 5 \cdot 13 \cdot 463 \cdot 1555679$ | 1 |
| | $3^2 \cdot 5^2 \cdot 13 \cdot 199 \cdot 17 \cdot 269 \cdot 397$<br>$3^2 \cdot 5^2 \cdot 13 \cdot 199 \cdot 1934279$ | 17 |
| | $3^4 \cdot 7 \cdot 11^2 \cdot 23 \cdot 43 \cdot 53 \cdot 919$<br>$3^4 \cdot 7 \cdot 11^2 \cdot 23 \cdot 2185919$ | 1 |
| | $3^4 \cdot 5 \cdot 11 \cdot 503 \cdot 41 \cdot 59 \cdot 2011$<br>$3^4 \cdot 5 \cdot 11 \cdot 503 \cdot 5070239$ | 5 |
| | $3^5 \cdot 7^2 \cdot 13 \cdot 23 \cdot 43 \cdot 53 \cdot 919$<br>$3^5 \cdot 7^2 \cdot 13 \cdot 23 \cdot 2185919$ | 1 |
| | $3^3 \cdot 5 \cdot 19 \cdot 37 \cdot 41 \cdot 73 \cdot 83 \cdot 163$<br>$3^3 \cdot 5 \cdot 19 \cdot 37 \cdot 41 \cdot 1019423$ | 6 |
| | $3^2 \cdot 7 \cdot 13 \cdot 19 \cdot 37 \cdot 41 \cdot 73 \cdot 83 \cdot 163$<br>$3^2 \cdot 7 \cdot 13 \cdot 19 \cdot 37 \cdot 41 \cdot 1019423$ | 6 |
| | $3 \cdot 5 \cdot 7^2 \cdot 419 \cdot 11 \cdot 17 \cdot 11731$<br>$3 \cdot 5 \cdot 7^2 \cdot 419 \cdot 2534111$ | 4 |
| (4, 1) | $3^4 \cdot 5 \cdot 113 \cdot 7 \cdot 17 \cdot 47 \cdot 1129$<br>$3^4 \cdot 5 \cdot 113 \cdot 7810559$ | 5 |

## 5. CONCLUSION AND FUTURE WORK

The nice thing about this approach to discovering new amicable pairs is that new pairs can be discovered with single-precision arithmetic on 32-bit computers. This approach, when restricted to $e$- and $s$-values less than $10^{10}$, requires no multiple-precision software and can produce new amicable pairs up to about 18 digits long. This approach also lends itself quite readily to parallel processing. If one makes the Step 1 data file available to several processors (or machines), one can have each processor (or machine) work on a different range of $e$-values.

## Acknowledgment

I want to thank M. Garcia for his suggestion that I pursue this very profitable line of investigation.

## Bibliography

1. W. Borho, *On Thabit ibn Kurrah's formula for amicable numbers*, Math. Comp. **26** (1972), 571–578.
2. P. J. Costello, *Amicable pairs of Euler's first form*, J. Recreational Math. **10** (1978), 183–189.
3. L. Euler, *De numeris amicabilibus*, Leonardi Euleri Opera Omnia, Ser. I, Vol. 2, Teubner, Leipzig and Berlin, 1915, pp. 59–61; 86–162.
4. E. B. Escott, *Amicable numbers*, Scripta Math. **12** (1946), 61–72.
5. E. J. Lee, *Amicable numbers and the bilinear diophantine equation*, Math. Comp. **22** (1968), 181–187.
6. K. H. Rosen, *Elementary number theory*, 2nd ed., Addison-Wesley, Reading, Mass., 1988.
7. H. J. J. te Riele, *Table of* 1869 *New Amicable Pairs Generated from* 1575 *Mother Pairs*, Report NN 27/82, Mathematical Centre, Oct. 1982.
8. ——, *Computation of all the amicable pairs below* $10^{10}$, Math. Comp. **47** (1986), 361–368.
9. ——, *On generating new amicable pairs from given amicable pairs*, Math. Comp. **42** (1984), 219–223.

Department of Mathematics, Statistics, and Computer Science, Eastern Kentucky University, Richmond, Kentucky 40475-3133