1992

On the Construction of Prime Desert n-Tuplets

Derek M. Marusarz '92
Illinois Wesleyan University

On the Construction of Prime Desert n-Tuplets

A Paper Presented For Senior
Research Honors in Mathematics

by

Derek M. Marusarz
Illinois Wesleyan University
May 14, 1992

On the Construction of Prime Desert *n*-Tuplets
by
Derek M. Marusarz

I. Introduction

In the summer of 1990, Professor George Polites received a summer research grant from the Lilly Foundation to continue his research in prime desert theory. Polites selected me to be his student research assistant under this grant. He had recently published an article on prime desert n-tuplets of length k in the American Mathematical Monthly [6] and that article formed the basis of our summer research activity. In the article a construction process for prime desert n-tuplets of length k was described and a necessary condition for their existence was established. My first assignment was to read the article and then use the information therein to construct some examples of prime desert n-tuplets of length k for various values of n and k. For this activity we used an existing primality testing program modified to fit our needs. Specifically, we taylored the Cohen-Lenstra version of the Adleman-Pomerance-Rumely Test, modified for version 7 of *UBASIC*, to search for the prime desert n-tuplets that we were hoping to construct. Interestingly enough, two "bugs" were discovered in this program!! We managed to correct them.

We wrote a few computer programs of our own. One that was particularly useful was a program that solved systems of linear congruences using the Chinese Remainder Theorem. I discovered a very efficient way of doing this.

I conducted a search of the literature on primality testing to find anything that might have been helpful and possibly implemented in our research [1], [2]. I was also looking at articles on primes in arithmetic sequences [5], [7] with the hope of finding something useful to us. Reading various journals such as The Journal of Number Theory and Mathematics of Computation proved to be quite difficult. For example, the primality testing algorithms were developed with highly advanced mathematics and were on the cutting edge of the field. Mathematical structures such as cyclotomic fields and Jacobi sums [8] were used quite heavily and, as a result, I had to familiarize myself as best as I could with these concepts. Complex analysis was used as well.

During that summer, several questions were discussed concerning the existence of prime desert n-tuplets of length k. As was stated above, a necessary condition on the value of k was given in Polites' paper. But what about the sufficiency condition? **Must** a prime desert n-tuplet of length k exist for any n and appropriate odd k? This question was one that occupied much of our time.

I considered whether or not the proof of Dirichlet's theorem could be manipulated to help us prove the existence of prime desert n-tuplets of length k. My reasoning here was that we are dealing with infinite sequences of the form $\{a + mi\}$, $\{a + (k+1) + mi\}$, $\{a + 2(k+1) + mi\}, \ldots, \{a + n(k+1) + mi\}$ and Dirichlet's theorem states that if you have a sequence of the form $\{a + bn\}$ where the greatest common divisor of a and b is 1, then there exists an infinite number of primes in the sequence. For our problem, we want to be guaranteed that there exists an $r \in Z^+$ such that $a + mr, a + (k+1) + mr, a + 2(k+1) + mr, \ldots, a + n(k+1) + mr$ are all prime; in other words, there exists an $r \in Z^+$ such that the above sequences "line up" with primes.

In number theory there is a very famous unsolved conjecture, called the twin prime conjecture, which states that there exists an infinite number of primes p such that p+2 is also prime. Probably the most famous theorem associated with twin primes is Brun's theorem. Brun's theorem states the following: Let p_n be the n^{th} prime such that p_n+2 is also prime. Then the series $\Sigma 1/p_n$ converges. The proof is based on a theorem found in any calculus book and goes like this: Find a convergent series Σa_n such that $1/p_n < a_n$. Then $\Sigma 1/p_n$ must converge [3]. The real difficulty lies in actually **finding** such a series that does what you want it to do. The series that Brun used was $\Sigma 1/(n \log^{3/2}(n+1))$. This result does not tell us, however, whether or not there is an infinite number of twin primes. If we were able to show that Σp_n diverged, **then** we would know that the number of twin primes is infinite. This led me to believe that I might have been able to produce a divergent series Σa_n with the property that $a_n < p_n$, thus showing that the series Σp_n diverges. I was unsuccessful in this regard.

If we were able to show that there is an infinite number of prime desert *n*-tuplets of length *k*, for any appropriate odd *k*, then we would have solved the twin prime conjecture because then we would know that there would be an infinite number of prime desert 1-tuplets of length $k = 1$, i.e., an infinite number of twin primes.

Research under the Lilly grant ended in August, 1990, but my research activity with the construction of prime desert *n*-tuplets did not. It continued to blossom into a senior honors research project.

II. Basic Concepts

Let *n* be a positive integer and let *k* be a positive odd integer. Then we have the following definitions :

Definition: A **prime desert of length *k*** is a set of *k* consecutive composite integers $c_1, c_2, \ldots, c_k$ such that $c_1 - 1$ and $c_k + 1$ are both prime.

Definition: A **prime desert *n*-tuplet of length *k*** is *n* consecutive prime deserts, each having the same length *k*.

Examples : The composite integers 8,9 and 10 form a prime desert of length $k = 3$

because 7 and 11 are both primes. The composite integers 90, 91, 92, 93, 94, 95 and 96 form a prime desert of length $k = 7$ because 89 and 97 are both primes. The integers 300661, 300667 and 300673 determine a prime desert twin or 2-tuplet of length $k = 5$ because 300661, 300667 and 300673 are primes and the integers between these numbers are composite. The integers 9843019, 9843049, 9843079, 9843109 and 9843139 determine a prime desert 4-tuplet of length $k = 29$.

As I mentioned in the introduction, I spent part of the summer of 1990 constructing various prime desert n-tuplets of length k. I will now discuss a procedure for doing this. When one chooses the value of n, the value for k is restricted by the necessary condition in Professor Polites' paper.

Definition: We say that **a ≡ b (mod m)** (read as "a is congruent to b modulo m") if a - b is divisible by m, where m is a positive integer greater than 1.

***Theorem** (Polites) :* Let p_i denote the i^{th} odd prime and suppose k is odd. If $p > p_i$ then a necessary condition for the existence of prime desert (p_i-1)-tuplets of length k beginning at p+1 is

$$k \equiv (2 \cdot 3 \cdot 5 \cdots p_i - 1) \bmod (2 \cdot 3 \cdot 5 \cdots p_i).$$

The same restriction on k applies to prime desert k-tuplets for n such that $p_i-1 < n \le p_{i+1}-2$. Thus if $k \equiv (2 \cdot 3 \cdot 5 \cdots p_i - 1) \bmod (2 \cdot 3 \cdot 5 \cdots p_i)$ but $k \not\equiv (2 \cdot 3 \cdot 5 \cdots p_{i+1} - 1) \bmod (2 \cdot 3 \cdot 5 \cdots p_{i+1})$, then a prime desert (p_i-1)-tuplet may be extended to at most a prime desert $(p_{i+1} - 2)$-tuplet.

So, in order to have a prime desert twin (or triplet), k must be congruent to 5 modulo 6 ($k \equiv 5 \pmod 6$); in order to have a prime desert 4-tuplet (or 5-tuplet), k must be congruent to 29 modulo 30 ($k \equiv 29 \pmod{30}$), and so on. If we wish to construct a prime desert n-tuplet of length k, say a prime desert 5-tuplet of length k, our choice for k is restricted. There is no prime desert 5-tuplet of length 21 since 21 is not congruent to 29 modulo 30.

The first step in the construction process is called "filling in the slots." Here we guarantee that the numbers between the numbers to be prime are composite. Let's see how this process works in finding a prime desert triplet (3-tuplet) of length $k = 11$. Consider the following "slot chart":

p		
p + 2	p + 14	p + 26
p + 4	p + 16	p + 28
p + 6	p + 18	p + 30
p + 8	p + 20	p + 32
p + 10	p + 22	p + 34
p + 12	p + 24	p + 36

Note that the numbers of the form p + (odd integer) are not listed because they are already composite (each is divisible by 2 since p is odd). The numbers we want to be prime are p, p+12, p+24 and p+36. The numbers between these 4 numbers are to be composite.

We make some assumptions about p. Suppose that 3 does not divide p ($3 \nmid p$). Then 3 must divide one of p+1 or p+2 ($3|(p+1)$ or $3|(p+2)$). We make one of these choices, say $3|(p+1)$, or, equivalently, $3|(p+4)$. This leads to the following slot chart where the occurrence of the number 3 to the right of p+4 means that p+4 is divisible by 3.

p					
p + 2		p + 14		p + 26	
p + 4	3	p + 16	3	p + 28	3
p + 6		p + 18		p + 30	
p + 8		p + 20		p + 32	
p + 10	3	p + 22	3	p + 34	3
p + 12		p + 24		p + 36	

Notice that $3|(p+1)$ implies that $3|((p+1) + 3m)$ for all $m \in Z^+$. Notice too that $3|(p+1)$ implies that 3 does not divide p+12, p+24 or p+36, the numbers we want to be prime. This is something about which one must be careful for it is entirely possible that when we use any given prime to fill a slot, we will end up filling one of the slots we want to be prime.

We now move on to the prime 5 and make the assumption that $5 \nmid p$. Then 5 must divide one of p+1, p+2, p+3 or p+4. Let's say that $5|(p+3)$. Then $5|((p+3) + 5m)$ for all $m \in Z^+$ and the slot chart above becomes

p					
p + 2		p + 14		p + 26	
p + 4	3	p + 16	3	p + 28	3,5
p + 6		p + 18	5	p + 30	
p + 8	5	p + 20		p + 32	
p + 10	3	p + 22	3	p + 34	3
p + 12		p + 24		p + 36	

Next we consider the prime 7 and suppose $7 \nmid p$. Then 7 must divide one of p+1, p+2, p+3, p+4, p+5 or p+6. Say $7|(p+2)$. Then $7|((p+2) + 7m)$ for $m \in Z^+$ and we have the following slot chart:

p					
p + 2	7	p + 14		p + 26	
p + 4	3	p + 16	3,7	p + 28	3,5
p + 6		p + 18	5	p + 30	7
p + 8	5	p + 20		p + 32	
p + 10	3	p + 22	3	p + 34	3
p + 12		p + 24		p + 36	

This process is continued with consecutive primes until all the slots are filled. For our example, the final slot chart appears below where we have assumed that 11|(p+9), 13|(p+6), 17|(p+14) and 19|(p+7).

p					
p + 2	7	p + 14	17	p + 26	19
p + 4	3	p + 16	3,7	p + 28	3,5
p + 6	13	p + 18	5	p + 30	7
p + 8	5	p + 20	11	p + 32	13
p + 10	3	p + 22	3	p + 34	3
p + 12		p + 24		p + 36	

We have made statements such as 3|(p+1), 5|(p+3), 7|(p+2), etc., and these can be translated into congruence language. This is the second step in the construction process. For example, if 3|(p+1), then $p+1 \equiv 0 \pmod{3}$, which implies that $p \equiv 2 \pmod{3}$; if 5|(p+3), then $p+3 \equiv 0$ (mod5) or $p \equiv 2 \pmod{5}$, and so on. Thus what we have is a system of congruences in the variable p. Since we used primes for the moduli, this system of congruences has a solution by the Chinese Remainder Theorem (CRT).

Definition: We say that **gcd(a,b) = c** (read as "the greatest common divisor of a and b is c") if c|a (c divides a) and c|b (c divides b) and for all d such that d|a and d|b, d|c.

***Theorem** (CRT) :* The system of n linear congruences

$x \equiv c_1 \pmod{m_1}$

$x \equiv c_2 \pmod{m_2}$

.

.

.

$x \equiv c_n \pmod{m_n}$

has a unique solution modulo $m = m_1 \cdot m_2 \cdots m_n$ if $\gcd(m_i, m_j) = 1$ for $i \neq j$, $1 \leq i \leq n$, $1 \leq j \leq n$ [4].

The third step in the construction process is to solve the system of congruences obtained in step two using the Chinese Remainder Theorem. For our example above, the resulting system of congruences is

$p \equiv 2 \pmod{3}$

$p \equiv 2 \pmod{5}$

$p \equiv 5 \pmod{7}$

$p \equiv 2 \pmod{11}$

$p \equiv 7 \pmod{13}$

$p \equiv 3 \pmod{17}$

$p \equiv 12 \pmod{19}$

and the solution to the system is

$p \equiv 3970727 \pmod{4849845}$.

Once our system of congruences has been solved, we can use the resulting congruence to search for the primes. (This is the final step in the construction process.) Suppose that $p \equiv a \pmod{m}$ is the solution to our system. Then we can re-write this in the following form: $p = a + mi$ where $i \in Z$. This, in turn, leads to the arithmetic sequence $\{a + mi\}$, and we need to find an $r \in Z^+$ such that $a + mr$, $a + (k+1) + mr, \ldots, a + n(k+1) + mr$ are all prime, that is, such that p, $p+(k+1)$, $p+2(k+1), \ldots, p+n(k+1)$ are all prime. Now we know by Dirichlet's theorem that in each of the sequences $\{a + mi\}$, $\{a + (k+1) + mi\}, \ldots, \{a + n(k+1) + mi\}$ there is an infinite number of primes because $\gcd(a + s(k+1), m) = 1$ for all $s \in \{1,2,\ldots,n\}$. In our example, we have that $p \equiv 3970727 \bmod (4849845)$. Therefore, we can rewrite this in the form $p = 3970727 + 4849845i$ where $i \in Z$. We now must find an $r \in Z^+$ such that $3970727 + 4849845r$, $3970739 + 4849845r$, $3970751 + 4849845r$ and $3970763 + 4849845r$ are all prime, and again, by Dirichlet's theorem, we know that in the sequences $\{3970727 + 4849845i\}$, $\{3970739 + 4849845i\}$, $\{3970751 + 4849845i\}$ and $\{3970763 + 4849845i\}$ there exists an infinite number of primes because $\gcd(3970727 + 12s, 4849845) = 1$ for all $s \in \{1,2,3\}$.

***<u>Theorem</u>* (Dirichlet) :** There exists an infinite number of primes in the sequence

$$\{a + bn\} \text{ iff } \gcd(a,b) = 1 .$$

The way in which we search for the primes is search for a value of i, say r, such that $a + mr$ is prime. When we find that r, we add $(k+1)$ to $a + mr$ and check to see if $a + mr + (k+1)$ is prime. If it is, we add $(k+1)$ to $a + mr + (k+1)$ and check to see if $a + mr + 2(k+1)$ is prime. This is continued until we find that $a + mr + n(k+1)$ is prime. If at some stage we add $(k+1)$ and the resulting number is not prime, then we move on to the next value of i such that $a + mi$ is prime. In our example, we find that $r = 464$ is the first value of i such that $a + mi$ is prime. That is, $3970727 + 4849845 \cdot 464 = 2254298807$ is prime. Adding $k+1 = 12$ to 2254298807 also gives a prime. $2254298807 + 2 \cdot 12 = 2254298831$ is also prime and so is

2254298807 + 3·12 = 2254298843. Therefore, the numbers p = 2254298807, p+12 = 2254298819, p+24 = 2254298831 and p+36 = 2254298843 determine a prime desert triplet (3-tuplet) of length $k = 11$.

For another example, let's attempt to find a prime desert 5-tuplet of length $k = 29$. First, we must construct an appropriate slot chart. Beginning with the chart

p				
p+2	p+32	p+62	p+92	p+122
p+4	p+34	p+64	p+94	p+124
p+6	p+36	p+66	p+96	p+126
p+8	p+38	p+68	p+98	p+128
p+10	p+40	p+70	p+100	p+130
p+12	p+42	p+72	p+102	p+132
p+14	p+44	p+74	p+104	p+134
p+16	p+46	p+76	p+106	p+136
p+18	p+48	p+78	p+108	p+138
p+20	p+50	p+80	p+110	p+140
p+22	p+52	p+82	p+112	p+142
p+24	p+54	p+84	p+114	p+144
p+26	p+56	p+86	p+116	p+146
p+28	p+58	p+88	p+118	p+148
p+30	p+60	p+90	p+120	p+150

and assuming 3|(p+1), 5|(p+2), 7|(p+5), 11|(p+3), 13|(p+11), 17|(p+6), 19|(p+8), 23|(p+20), 29|(p+28), 31|(p+16), 37|(p+1), 41|(p+3), 43|(p+12), 47|(p+10), 53|(p+8), 59|(p+57), 61|(p+48), 67|(p+18), 71|(p+63) and 73|(p+56), we obtain the slot chart p

p+2	5	p+32	5	p+62	5	p+92	5	p+122	5,19
p+4	3	p+34	3	p+64	3	p+94	3	p+124	3,7,11
p+6	17	p+36	11	p+66	23	p+96	7	p+126	41
p+8	19,53	p+38	37	p+68	7	p+98	43	p+128	13
p+10	3,47	p+40	3,7,17	p+70	3	p+100	3	p+130	3
p+12	5,7,43	p+42	5	p+72	5	p+102	5,11,13	p+132	5
p+14	11	p+44	41	p+74	17	p+104	47	p+134	71
p+16	3,31	p+46	3,19	p+76	3,13	p+106	3	p+136	3
p+18	67	p+48	61	p+78	31	p+108	17	p+138	7
p+20	23	p+50	13	p+80	11	p+110	7	p+140	31
p+22	3,5	p+52	3,5	p+82	3,5,7	p+112	3,5,23,37	p+142	3,5,17
p+24	13	p+54	7	p+84	19	p+114	53	p+144	29
p+26	7	p+56	73	p+86	29	p+116	59	p+146	11
p+28	3,29	p+58	3,11	p+88	3	p+118	3	p+148	3
p+30		p+60		p+90		p+120		p+150	

The corresponding system of congruences is

$p \equiv 2 \pmod{3}$

$p \equiv 3 \pmod{5}$

$p \equiv 2 \pmod{7}$

$p \equiv 8 \pmod{11}$

$p \equiv 2 \pmod{13}$

$p \equiv 11 \pmod{17}$

$p \equiv 11 \pmod{19}$

$p \equiv 3 \pmod{23}$

$p \equiv 1 \pmod{29}$

$p \equiv 15 \pmod{31}$

$p \equiv 36 \pmod{37}$

$p \equiv 38 \pmod{41}$

$p \equiv 31 \pmod{43}$

$p \equiv 37 \pmod{47}$

$p \equiv 45 \pmod{53}$

$p \equiv 2 \pmod{59}$

$p \equiv 13 \pmod{61}$

$p \equiv 49 \pmod{67}$

$p \equiv 8 \pmod{71}$

$p \equiv 17 \pmod{73}$

and this system has the following solution:

$p \equiv 10506472643279344167703921838 \pmod{20364840299624512075310661735}$.

This leads to the sequence {a + mi} where a = 10506472643279344167703921838 and m = 20364840299624512075310661735. The first value of i that gives us the six primes we are looking for is 933,911 and the prime desert 5-tuplet of length $k = 29$ is determined by the following 6 primes:

1901895887553527097610962311 5517423

19018958875535270976109623115517453

19018958875535270976109623115517483

19018958875535270976109623115517513

19018958875535270976109623115517543

19018958875535270976109623115517573.

Suppose, in our first example above, that we had "filled the slots" in a somewhat different manner. That is, suppose we had made the assumption that $3|(p+2)$ rather than $3|(p+1)$ or that $7|(p+4)$ rather than $7|(p+2)$. Would we have needed the **same** set of consecutive primes that were used before to fill all of the slots? Consider the following example of a prime desert triplet (3-tuplet) of length $k = 11$ where we use a larger set of primes. In **this** example, we assume that $3|(p+2)$, $5|(p+3)$, $7|(p+4)$, $11|(p+6)$, $13|(p+4)$, $17|(p+10)$, $19|(p+15)$, $23|(p+16)$ and $29|(p+22)$. The following slot chart results:

p					
p + 2	3	p + 14	3	p + 26	3
p + 4	7,13	p + 16	23	p + 28	3,5,11
p + 6	11	p + 18	5,7	p + 30	13
p + 8	3,5	p + 20	3	p + 32	3,7
p + 10	17	p + 22	29	p + 34	19
p + 12		p + 24		p + 36	

The resulting system of congruences is

$p \equiv 1 \pmod{3}$

$p \equiv 2 \pmod{5}$

$p \equiv 3 \pmod{7}$

$p \equiv 5 \pmod{11}$

$p \equiv 9 \pmod{13}$

$p \equiv 7 \pmod{17}$

$p \equiv 4 \pmod{19}$

$p \equiv 7 \pmod{23}$

$p \equiv 7 \pmod{29}$

which has the solution

$$p \equiv 2427112957 \pmod{3234846615}.$$

Our triplet in this example is determined by the following primes:

487654105207

487654105219

487654105231

487654105243.

So, we were able to find another triplet, but not as efficiently as before. We used **more** primes this time, resulting in more congruences, resulting in **larger** numbers to check for primality, and this is a problem because the larger the numbers to be checked for primality, the more CPU time required. In our next example we will show another prime desert 5-tuplet of length $k = 29$ where **less** primes are used than before, resulting in fewer congruences, resulting in smaller numbers to check for primality. Compare the assumptions made in the following construction process with those made in our earlier example. (See page 10.)

Suppose 3|(p+1), 5|(p+2), 7|(p+5), 11|(p+3), 13|(p+11), 17|(p+6), 19|(p+18), 23|(p+20), 29|(p+28), 31|(p+16), 37|(p+8), 41|(p+3), 43|(p+48), 47|(p+38), 53|(p+31), 59|(p+39), 61|(p+43), 67|(p+47) and 71|(p+45). This leads to the following slot chart for constructing a prime desert 5-tuplet of length $k = 29$. (Notice here that only the smallest prime that can fill a slot is listed in that slot.)

p									
p+2	5	p+32	5	p+62	5	p+92	5	p+122	5
p+4	3	p+34	3	p+64	3	p+94	3	p+124	3
p+6	17	p+36	11	p+66	23	p+96	7	p+126	41
p+8	37	p+38	47	p+68	7	p+98	59	p+128	13

p+10	3	p+40	3	p+70	3	p+100	3	p+130	3
p+12	5	p+42	5	p+72	5	p+102	5	p+132	5
p+14	11	p+44	41	p+74	17	p+104	61	p+134	43
p+16	3	p+46	3	p+76	3	p+106	3	p+136	3
p+18	19	p+48	43	p+78	31	p+108	17	p+138	7
p+20	23	p+50	13	p+80	11	p+110	7	p+140	31
p+22	3	p+52	3	p+82	3	p+112	3	p+142	3
p+24	13	p+54	7	p+84	53	p+114	67	p+144	29
p+26	7	p+56	19	p+86	29	p+116	71	p+146	11
p+28	3	p+58	3	p+88	3	p+118	3	p+148	3
p+30		p+60		p+90		p+120		p+150	

The solution to the system of congruences (that results from the above assumptions) is $p \equiv 16517196004197368635754783 \pmod{27897041506334948048370769 5}$ and a prime desert 5-tuplet from this solution is determined by the following primes:

213935732539317200392792458065603

213935732539317200392792458065633

213935732539317200392792458065663

213935732539317200392792458065693

213935732539317200392792458065723

213935732539317200392792458065753.

Notice that we cut the number of primes needed to complete the slot chart by one, thus making the solution and modulus resulting from solving the system of congruences much smaller. A very good question at this point would be: is there a "best way" to fill in the slots for a given n and k? How does one find it? This is the main problem that I have been working on and will deal with in the next section.

III. Main Conjecture

As was seen at the end of the last section, when given an n and k to construct a prime desert n-tuplet of length k our fill in the slots procedure can be carried out in several ways using various sets of consecutive primes. If one were to fill the slots for a 3-tuplet of length $k = 23$ using the primes 3,5,7,11,…,43, there are 45,015,704,984,250 ways to do it (this number is the product of all of the possible slots that can be filled by each prime from 3 to 43), but when using the primes 3,5,7,11,…,41 there are only 1,154,248,845,750 different ways. Clearly, one should try to find a way of filling the slots in the most efficient manner. This was the main problem on which I worked and I have come to some conclusions on it.

As I thought about the problem I wondered if we shouldn't alter the construction scheme presented in Professor Polites' paper. For example, why not drop the condition that slots be filled with consecutive primes beginning with the prime 3 and simply choose primes at random to fill our slots? Then, as before, we would be led to a system of linear congruences whose solution would lead to a Dirichlet sequence where we could look for a prime p that gives us our desired prime desert n-tuplet of length k. A problem with this is that we are looking for a **systematic** scheme that can be implemented in the same way each time we try to construct a prime desert n-tuplet of length k. At the same time, we want to keep the size of our numbers as small as possible to allow us to test their primality as quickly as possible. (In general, the more digits a number has the longer it takes to see if it is a prime.) To this end we should try to determine the largest prime that would be needed to fill all of our slots. But how would one do this if primes are chosen randomly?

Basically, what I was looking for was a formula that would predict the largest prime in the set of consecutive primes necessary to fill the slots in the most efficient way, that is, the "smallest largest" prime necessary in the set. These kinds of formulas are few and far between and to get a formula that produced primes and was useful for my particular problem was a very difficult task. Therefore, I turned my attention to something that proved to be more fruitful. I thought about how the primes get distributed on the slot chart and noted that the smaller the prime, the more slots it fills. This seemed useful and it in fact was the basis for my conjecture about how one ought to go about filling the slots in the most efficient manner. My conjecture is the following:

Conjecture ***(Marusarz)*:** If one fills the slots in the following manner, then **no** other slot filling scheme will exist that entails using a smaller set of consecutive primes. Scheme: start with the prime 3 and fill in all the possible slots for **each** possible case that applies to the prime 3. These cases are $3|(p+1)$ and $3|(p+2)$. Take the cases for 3 that yield the highest number of slots filled and use each of these with each of the possible cases that apply to the prime 5 (which will be among $5|(p+1)$, $5|(p+2)$, $5|(p+3)$ and $5|(p+4)$). Take the combined cases for 3 and 5 that yield the highest number of slots filled and use them with all possible cases that apply to the prime 7 (which will be among $7|(p+1),\ldots,7|(p+6)$). Continue this process until all of the slots are filled.

Since implementing my conjecture "by hand" would be tedious work (to put it mildly!), I wrote (with help) a computer program in *PASCAL* to fill slots according to my conjecture. I had had limited experience with computer programming so this was no easy task. The program was written in version 4.0 of *TurboPascal* and was run in a PC emulator called Soft PC on a NeXT computer.

The main program calls several procedures. The first procedure, called "GETPRIMES", initializes the array **Prime** which holds the primes to be used for slot filling and the matrix **Possi** which determines and stores all possible and impossible slot choices for each prime in the array Prime. (As was mentioned earlier, we must be sure that the slots represented by p, p+(k+1), p+2(k+1),...,p+n(k+1) are **not** filled by **any** of the primes being used.) Next a linked list, OurList, is set up to hold all slot filling schemes under consideration. A WHILE loop goes through the list of primes and sends each one to a procedure called "PROCESS." In this procedure the possible slots that can be filled by the prime currently being "processed" are retrieved from the matrix Possi. For each prime/possibility pair that the procedure determines, it sends the pair to a procedure called "FILLTHESLOTS." This procedure then fills the slot chart with this prime/possibility pair and assigns a count (based on how many slots were filled) and decides, based on this count, whether or not it will be kept as one of the cases to be combined with the next prime. For example, when constructing a prime desert 3-tuplet of length $k = 23$, the program will go through the possiblities for the prime 3 and find that $3|(p+1)$ and $3|(p+2)$ yield the same maximum number of slots filled. Thus it keeps those cases and combines them with all possible cases for the prime 5. It then finds that $3|(p+1)$ and $5|(p+1)$, or $3|(p+2)$ and $5|(p+1)$ yield the same maximum number of slots filled. These cases are kept and combined with all possible cases for the prime 7, and so on.

A copy of the program that will determine the last prime needed (according to my conjecture) for a prime desert n-tuplet of length k where $n = 3$ and $k = 11$ appears below.

```
PROGRAM Slots (Output);
{This program determines the last prime necessary to fill all of the slots for a prime
desert n-tuplet of length k. It also finds all sets of congruences to be satisfied by the
prime p that will eventually lead to the n-tuplet.}
CONST
N = 3; {The number of consecutive deserts}
K = 11; {The length of each desert}
S = 36; { The number of slots: S = N*(K+1)}
C = 20; {The number of primes in the array called "Prime."}
D = 72; {D = Prime[C]-1}
TYPE
EntryPointer = ^Entry;
SlotArray = ARRAY[1..S] OF BOOLEAN;
CongArray = ARRAY[1..C] OF INTEGER;
PrimeArray = ARRAY[1..C] OF INTEGER;
PossArray = ARRAY[1..C, 1..D] OF BOOLEAN;
Entry = RECORD
        Slot : SlotArray;
        {1..N*(K+1)}{Keeps track of all slots currently filled}
        Cong : CongArray;
        {A list of congruences currently assigned for the primes in Prime.}
        Count : Integer;
        {The number of slots filled so far}
        Next : EntryPointer
        {The next entry on the list}
       END;
OurListType = RECORD
             start : EntryPointer;
             ends : EntryPointer
        END;
VAR
 x : EntryPointer;
 OurList : OurListType;
 {This linked list holds all entries that are consistent with the algorithm.}
 Finished : BOOLEAN;
 Prime : PrimeArray;
 Max,w,q,I,J,list : INTEGER;
 Possi : PossArray;

PROCEDURE Delete; {This procedure deletes the leading entry from OurList.}
VAR z : entrypointer;
BEGIN
 z := OurList.start;
 OurList.start := OurList.start^.next;
 q := q-1;
 DISPOSE(z);
END;
```

```
PROCEDURE AddToList(x : EntryPointer); {This adds x^ to the end of OurList.}
BEGIN
 OurList.ends^.next := x;
 OurList.ends := x;
 x^.Next := NIL;
 q := q+1;
END;

PROCEDURE Initialize(x : EntryPointer);
{This procedure generates the first entry for OurList by assigning the unique
 congruence  for the prime 2. The array Cong is set to zero to indicate that no
 congruence has been assigned for any odd prime.}
VAR
 i : INTEGER;
BEGIN
 FOR i := 1 TO  ( N*(K+1) DIV 2) DO
 BEGIN
        x^.slot[2*i] := False;
        x^.slot[2*i - 1] := True
  END;
  x^.count := N*(K+1) DIV 2 ;
  x^.Next := NIL;
  FOR i := 1 TO C DO
 x^.Cong[i] := 0;
END;

PROCEDURE FillTheSlots(w : INTEGER; J : INTEGER);
{This procedure makes a copy of each entry on the OurList, adds the congruence
  p + j ≡ 0 (mod Prime[w]) to this entry. Entries for which the count is maximum are
  appended to the list.}
VAR
 i,d : INTEGER;
 x,y,g : EntryPointer;
 BEGIN
   NEW(x);
   x^.slot := OurList.start^.slot;
   x^.cong := Ourlist.start^.cong;
   x^.count := Ourlist.start^.count;
   x^.next := nil;
   x^.Cong[w] := j;
   i := j;
   WHILE( i <= s ) do
       BEGIN
       IF x^.slot[i] = FALSE THEN
           BEGIN
            x^.slot[i] := TRUE;
            x^.count := x^.count + 1
           END;
       i := i+prime[w]
```

```
   END;{WHILE}
   IF (x^.count = max) THEN {x^ matches the current best count and is appended to
     AddToList(x)                    OurList.}
     ELSE
   IF (x^.count > max) THEN {The count for x^ exceeds that of all entries which
                                     have been assigned a congruence mod Prime[w], so
                                     all of these are deleted before appending x^.}
 BEGIN
      y := OurList.start

WHILE (NOT(y^.next = NIL) AND (y^.next^.Cong[w] = 0)) DO
             y := y^.next;
             OurList.ends := y;
             g := y^.next;
             y^.next := NIL;
      WHILE NOT(g = NIL) DO

 BEGIN
   y := g;
   g := g^.next;
   DISPOSE(y);
   q := q-1;
     END;
             AddToList(x);
             max := x^.count;
          END ELSE DISPOSE(x);
 END;

PROCEDURE Process(w : INTEGER);
{This procedure takes all leading entries on OurList which have not been assigned
  a congruence modulo Prime[w], and assigns all possible congruences for this prime,
  calling FillTheSlots. When all possible congruences have been assigned to the entry
  it is removed from the list.}
VAR
 j,i : INTEGER;
 x,y,z : EntryPointer;
BEGIN
 WHILE (OurList.start^.Cong[w] = 0) DO
     BEGIN
        FOR j := 1 TO (Prime[w] - 1) DO
            IF Possi[w,j] THEN FillTheSlots(w,j);
        Delete;
     END; {WHILE}
END;

PROCEDURE GetPrimes;
{This procedure intializes the array Prime to hold the first C odd primes, and the
  matrix Possi which holds the possible congruences for these primes.}
VAR
```

```
 i,j,v,t,b : INTEGER;
BEGIN
  Prime[1] := 3;
  Prime[2] := 5;
  Prime[3] := 7;
  Prime[4] := 11;
  Prime[5] := 13;
  Prime[6] := 17;
  Prime[7] := 19;
  Prime[8] := 23;
  Prime[9] := 29;
  Prime[10] := 31;
  Prime[11] := 37;
  Prime[12] := 41;
  Prime[13] := 43;
  Prime[14] := 47;
  Prime[15] := 53;
  Prime[16] := 59;
  Prime[17] := 61;
  Prime[18] := 67;
  Prime[19] := 71;
  Prime[20] := 73
  FOR i := 1 TO C DO
    FOR j := 1 TO (Prime[C]-1) DO
      Possi[i,j] := True;
  FOR i := 1 TO C DO
    FOR j := 1 to Prime[i]-1 DO

      FOR v := 0 TO (n*(k+1)-j) DIV Prime[i] DO
        FOR t := 1 TO N DO BEGIN
          b := j + v*Prime[i];
          IF (b = t*(K+1)) THEN
            Possi[i,j] := FALSE;
        END;
END;

PROCEDURE WriteGoodResults;
{Outputs results to screen if program is successful.}
VAR
  y : entrypointer;
  i,j : INTEGER;
BEGIN
y := ourlist.start;
REPEAT
BEGIN
   writeln;
   FOR i := 1 to C DO
   BEGIN
      WRITE(y^.cong[i]);
      WRITE(' ');
```

```
  END;
    WRITELN(' max =',max,' nk= ',n*k);
    WRITELN('count = ',y^.count);
   y := y^.next;
 END;
 UNTIL y = nil;
END;

BEGIN {Main Program}
 max := 0;
 GetPrimes;
 NEW(x);
 Initialize(x);
 OurList.start := NIL;
 OurLIst.ends := NIL;
 OurList.start := x;
 OurList.ends := x;
 q := 1;
 Finished := FALSE;
 w := 0;
 WHILE ((NOT Finished) AND (w < c)) DO
    BEGIN
    w := w+1;
    Process(w);
    If max >= N*K THEN Finished := TRUE;
 END; {WHILE}
 IF Finished THEN WriteGoodResults ELSE
 WRITELN('You have used all primes in the array Prime.');
END.
```

The list OurList in my program can grow to extreme length during the execution of the program, resulting in a severe drain on a computer's memory and a possible "crashing" of the program. I experienced this when attempting to determine the last prime needed for the construction of prime desert n-tuplets of length k for relatively small values of n and k. However, the use of a *PASCAL* compiler provided needed additional memory and I was then able to look at more extensive examples.

Below are two examples illustrating the output of various "runs" of my program.

Example: For a prime desert 2-tuplet of length $k = 5$, 3|(p+2), 5|(p+4) and 7|(p+3) is one scheme to fill the slots and 3|(p+1), 5|(p+3) and 7|(p+2) is another. These are the only ways that the computer found to fill the slots according to my conjecture. My conjecture would then say that there is no scheme to fill the slots that is better than these two ways; that is, it is impossible to fill all of the slots with a smaller set of consecutive primes than the set {3,5,7}.

Example: For a prime desert 3-tuplet of length $k = 11$, the computer came up with the following eight schemes :

3|(p+1), 5|(p+3), 7|(p+2), 11|(p+4), 13|(p+6), 17|(p+3) and 19|(p+14)

3|(p+1), 5|(p+3), 7|(p+2), 11|(p+4), 13|(p+6), 17|(p+14) and 19|(p+1)

3|(p+1), 5|(p+3), 7|(p+2), 11|(p+9), 13|(p+6), 17|(p+9) and 19|(p+14)

3|(p+1), 5|(p+3), 7|(p+2), 11|(p+9), 13|(p+6), 17|(p+14) and 19|(p+7)

3|(p+2), 5|(p+3), 7|(p+6), 11|(p+5), 13|(p+4), 17|(p+5) and 19|(p+10)

3|(p+2), 5|(p+3), 7|(p+6), 11|(p+5), 13|(p+4), 17|(p+10) and 19|(p+3)

3|(p+2), 5|(p+3), 7|(p+6), 11|(p+10), 13|(p+4), 17|(p+5) and 19|(p+16)

3|(p+2), 5|(p+3), 7|(p+6), 11|(p+10), 13|(p+4), 17|(p+16) and 19|(p+3).

This again means , according to my conjecture, that there is no slot filling scheme that will fill all of the slots using a smaller set of consecutive primes than {3,5,7,11,13,17,19}. Notice that this is precisely the example I worked on in the previous section and in that example I used the fourth scheme listed above. Notice, too, that of the eight cases four occur for 3|(p+1) and four occur for 3|(p+2). This "symmetry" of output has appeared in every example tested.

Recall that we discussed previously two examples of prime desert 5-tuplets of length $k = 29$. For the first example the set {3,5,7,11,...,73} of consecutive primes was used to complete the slot chart, and for the second we used the consecutive primes 3,5,7,11,...,71. When using my computer program to fill the slots, only the primes 3,5,7,11,...,67 were needed, and, if we assume that 3|(p+1), 5|(p+4), 7|(p+5), 11|(p+9), 13|(p+11), 17|(p+4), 19|(p+2), 23|(p+11), 29|(p+8), 31|(p+5), 37|(p+11), 41|(p+9), 43|(p+6), 47|(p+5), 53|(p+3), 59|(p+3), 61|(p+18) and 67|(p+32), we obtain the completed slot chart below where only the smallest prime that can fill a slot is listed in that slot.

p									
p+2	19	p+32	67	p+62	59	p+92	43	p+122	37
p+4	3	p+34	3	p+64	3	p+94	3	p+124	3
p+6	43	p+36	31	p+66	29	p+96	7	p+126	23
p+8	29	p+38	17	p+68	7	p+98	31	p+128	13
p+10	3	p+40	3	p+70	3	p+100	3	p+130	3
p+12	7	p+42	11	p+72	17	p+102	13	p+132	41
p+14	5	p+44	5	p+74	5	p+104	5	p+134	5
p+16	3	p+46	3	p+76	3	p+106	3	p+136	3
p+18	61	p+48	37	p+78	19	p+108	11	p+138	7
p+20	11	p+50	13	p+80	23	p+110	7	p+140	17
p+22	3	p+52	3	p+82	3	p+112	3	p+142	3
p+24	5	p+54	5	p+84	5	p+114	5	p+144	5
p+26	7	p+56	53	p+86	11	p+116	19	p+146	47
p+28	3	p+58	3	p+88	3	p+118	3	p+148	3
p+30		p+60		p+90		p+120		p+150	

This slot filling scheme leads to the system of congruences

$$p \equiv 2 \pmod{3}$$

$$p \equiv 1 \pmod{5}$$

$$p \equiv 2 \pmod{7}$$

$$p \equiv 2 \pmod{11}$$

$$p \equiv 2 \pmod{13}$$

$$p \equiv 13 \pmod{17}$$

$$p \equiv 17 \pmod{19}$$

$$p \equiv 12 \pmod{23}$$

$$p \equiv 21 \pmod{29}$$

$$p \equiv 26 \pmod{31}$$

$$p \equiv 26 \pmod{37}$$

$$p \equiv 32 \pmod{41}$$

$$p \equiv 37 \pmod{43}$$

$$p \equiv 42 \pmod{47}$$

$$p \equiv 50 \pmod{53}$$

$$p \equiv 56 \pmod{59}$$

$$p \equiv 43 \pmod{61}$$

$$p \equiv 35 \pmod{67}$$

which has the solution

$$p \equiv 16141438454296219383719 6 \pmod{392916077554013352793954 5}.$$

The corresponding Dirichlet sequence leads to our prime desert 5-tuplet of length k = 29 when we reach the 3,133,441st term in the sequence. It is determined by the following six primes:

1231179363108363608488260966 1541

1231179363108363608488260966 1571

1231179363108363608488260966 1601

1231179363108363608488260966 1631

1231179363108363608488260966 1661

1231179363108363108488260966 1691.

So, we now know that it is possible to construct a prime desert 5-tuplet of length $k = 29$ using only the consecutive primes 3,5,7,11,...,67. We also know, if my conjecture is true, that no other slot filling scheme using consecutive primes beginning with 3 will do unless we go all the way out to the prime 67.

We close this section with three more examples where the results were obtained based on output from my computer program.

Example: This desert 3-tuplet of length $k = 35$ was constructed assuming that

3|(p+1), 5|(p+4), 7|(p+6), 11|(p+1), 13|(p+8), 17|(p+13), 19|(p+7), 23|(p+4), 29|(p+2), 31|(p+18), 37|(p+1), 41|(p+1), 43|(p+6), 47|(p+19), 53|(p+15) and 59|(p+32). The resulting system of congruences is

$p \equiv 2 \pmod 3$

$p \equiv 1 \pmod 5$

$p \equiv 1 \pmod 7$

$p \equiv 10 \pmod{11}$

$p \equiv 5 \pmod{13}$

$p \equiv 4 \pmod{17}$

$p \equiv 12 \pmod{19}$

$p \equiv 19 \pmod{23}$

$p \equiv 27 \pmod{29}$

$p \equiv 13 \pmod{31}$

$p \equiv 36 \pmod{37}$

$p \equiv 40 \pmod{41}$

$p \equiv 37 \pmod{43}$

$p \equiv 28 \pmod{47}$

$p \equiv 38 \pmod{53}$

$p \equiv 27 \pmod{59}$

and the solution to the system is

$$p \equiv 90070271372335653395 6 \pmod{9613801750771063195 35}.$$

The primes that determine this triplet of length $k = 35$ are found at the 14,979th term in the Dirichlet sequence determined by the above solution and are the numbers

1440141434519369891684872 1

1440141434519369891684875 7

1440141434519369891684879 3

1440141434519369891684882 9.

The slot chart for this example appears below and on the next page.

p					
p+2	29	p+38	37	p+74	5
p+4	3	p+40	3	p+76	3
p+6	7	p+42	41	p+78	11
p+8	13	p+44	5	p+80	31
p+10	3	p+46	3	p+82	3
p+12	11	p+48	7	p+84	5

p+14	5	p+50	23	p+86	13
p+16	3	p+52	3	p+88	3
p+18	31	p+54	5	p+90	7
p+20	7	p+56	11	p+92	43
p+22	3	p+58	3	p+94	3
p+24	5	p+60	13	p+96	23
p+26	19	p+62	7	p+98	17
p+28	3	p+64	3	p+100	3
p+30	17	p+66	47	p+102	19
p+32	59	p+68	53	p+104	5
p+34	3	p+70	3	p+106	3
p+36		p+72		p+108	

Example **:** This desert 5-tuplet of length k = 59 was constructed assuming that

3|(p+1), 5|(p+2), 7|(p+3), 11|(p+2), 13|(p+10), 17|(p+6), 19|(p+18), 23|(p+20), 29|(p+26), 31|(p+13), 37|(p+5), 41|(p+14), 43|(p+9), 47|(p+10), 53|(p+4), 59|(p+8), 61|(p+50), 67|(p+54), 71|(p+15), 73|(p+48), 79|(p+78), 83|(p+5), 89|(p+8), 97|(p+1), 101|(p+27), 103|(p+30), 107|(p+37), 109|(p+37) and 113|(p+61).

The solution to the system of congruences that results is p ≡ a (mod m)

where a = 6061508053833487987316418698802803674169224763 and

m = 15805027320208803894072603145771831246637343495

and the six primes that determine the 5-tuplet are:

167875063566493570286063006632024280096036801131960643

167875063566493570286063006632024280096036801131960703

167875063566493570286063006632024280096036801131960763

1678750635664935702860630066320242800960368011319608 23

1678750635664935702860630066320242800960368011319608 83

1678750635664935702860630066320242800960368011319609 43

<u>Example</u> *:* According to my conjecture, the last prime q needed to fill all of the slots for a prime desert of length $k = 59$ is $q = 37$, for a prime desert twin of length $k = 59$ is $q = 59$, for a prime desert triplet of length $k = 59$ is $q = 71$, for a prime desert 4-tuplet of length $k = 59$ is $q = 103$, and for a prime desert 5-tuplet of length $k = 59$ is $q = 113$. The last prime q needed for a prime desert of length $k = 89$ is $q = 43$, for a prime desert twin of length $k = 89$ is $q = 73$, for a prime desert triplet of length $k = 89$ is $q = 107$, for a prime desert 4-tuplet of length $k = 89$ is $q = 137$, and for a prime desert 5-tuplet of length $k = 89$ is $q = 163$.

IV. Concluding Remarks

The last example above suggests an interesting question. If one were to find the last prime needed to construct a desert, a twin, a triplet, a quadruplet, a quintuplet, etc., all of the same length, is there a pattern or relationship for these primes and if so, what is it? Is there a formula that gives the last prime needed for any n and appropriate odd k?

The observant reader knows full well that to be absolutely certain that my conjecture will hold true for any particular prime desert n-tuplet of length k that is to be constructed, **every** possible slot chart would have to be considered to see if even **one** of them uses fewer consecutive primes than those obtained by my computer program. I ran my program to find the last prime needed for a prime desert twin of length $k = 11$ and found it to be 13. Then I ran an altered version of my program to check **all** possible slot filling schemes using the primes 3,5,7 and 11 (there are 128 such cases). None of these filled **all** of the slots. Thus my conjecture is true for a prime desert n-tuplet of length k when $n = 2$ and $k = 11$. I did the same thing for a prime desert triplet of length $k = 11$. My program found that 19 is the last prime needed to fill all of the slots and after checking each of the 4,914 possible slot filling schemes using the primes 3,5,7,11,13 and 17, my altered program found that none of them filled **all** of the slots. Thus my conjecture is true for a prime desert n-tuplet of length k when $n = 3$ and $k = 11$. Obviously, as the values of n and/or k get larger, the total number of slot filling cases becomes so numerous that it no longer is feasible to check all of them.

In the example on page 30 it was noted that in order to fill all of the slots for a prime desert 4-tuplet of length 59, the last prime needed is $q = 103$. Actually, I could not be certain of this because the computer was reaching its maximum memory capacity and my program was about to "crash." In attempting to verify the result by modifying the program and doing the calculations in four separate stages, I found that all of the slots can be filled by the primes 3,5,7,11,...,83. One way to do this is to assume 3|(p+1), 5|(p+3), 7|(p+6), 11|(p+8), 13|(p+1), 17|(p+16), 19|(p+4), 23|(p+3), 29|(p+15), 31|(p+21), 37|(p+2), 41|(p+40), 43|(p+24), 47|(p+32), 53|(p+33), 59|(p+57), 61|(p+12), 67|(p+35), 71|(p+56), 73|(p+36), 79|(p+54) and 83|(p+34). In other words, $\boldsymbol{q = 83}$ seems to be the last prime needed, **not** $q = 103$. Thus I found an example showing my conjecture to be false and it now appears that what can be said is that my conjecture gives a "near optimum" slot filling scheme for determining a prime desert n-tuplet of length k. The question first asked on page 15 must be asked again: is there a "best way" to fill in the slots for a given n and k?

In closing, I would like to give thanks to my project advisor, Professor George Polites, for without him this paper would not have been written. Also, I give special thanks to my academic advisor, Professor J. Philip Kavanagh, for his invaluable help with my *PASCAL* program. Thanks, as well, to my committee members who sacrificed their time to read and critique my paper.

References

1. Adleman, L.M., Pomerance, C. and Rumely, R.S., "On Distinguishing Prime Numbers from Composite Numbers", Annals. of Mathematics, No. 2, Vol. 117, pp. 176 - 201, 1983.

2. Cohen, H., and Lenstra, Jr., H.W., "Primality Testing and Jacobi Sums", Mathematics of Computation, Vol. 42, pp. 297 - 330, 1984.

3. Landau, Edmund, Elementary Number Theory (Elementare Zahlentheorie) , Chelsea Publishing Company, New York, pp. 120-121, 1958.

4. Long, Calvin T., Elementary Introduction to Number Theory, 3rd Ed., Prentice-Hall, Inc., New Jersey, 1987.

5. Mozzochi, C. J., "On the Difference Between Consecutive Primes, Journal of Number Theory, Vol. 24, pp. 181 - 187, 1986.

6. Polites, George W., "Prime Desert *n*-Tuplets", American Mathematical Monthly, Vol. 95, No. 2, pp. 98 - 104, 1988.

7. Pritchard, P. A., "Long Arithmetic Progressions of Primes, Some Old, Some New" Mathematics of Computation, Vol. 45, pp. 263 - 267, 1985.

8. Weil, A., "Jacobi Sums as Grössencharaktere", Transactions of the American Mathematical Society, Vol. 273, pp. 487 - 495, 1982.