Improvement of Long Binary Sequence Merit Factors using Modified Legendre Algorithms

B. Suribabu Naick
Asst Prof, Dept of ECE
GITAM University
Visakhapatnam
Andhra Pradesh

P. Rajesh Kumar
Head of Department
Dept. of ECE
Andhra University
Visakhapatnam
Andhra Pradesh

ABSTRACT

Low autocorrelation binary sequence (LABS) detection is a classic problem in the literature. We use these sequences in many real-life applications. The detection of these sequences involves many problems. In the literature, various methods have been developed to approach the LABS issue. Based on the length of the sequence, an appropriate method can be selected and implemented. For short length sequences, linear search is possible and as the length increases we can implement various stochastic optimization algorithms. In our case that is for long binary sequences, we can use construction methods. Kristiansen and Parker [1] in their work have shown that Legendre sequences with periodic rotation can achieve a merit factor of 6.34.We have applied these Legendre sequences to steepest descent and prime step algorithms with some modifications. We call these techniques as modified Legendre algorithms. Using these improved methods we were able to achieve a merit factor of 6.4245 for long binary sequences.

Keywords

Legendre sequences, prime step algorithm, steepest descent algorithm.

1. INTRODUCTION

We use Low autocorrelation binary sequences (LABS) in radar pulse compression techniques, various communication methods and physics like spin icing glasses and also in the field of chemistry. In LABS, the complexity lies not in the application part, but in the detection/generation part. The first problem is the presence of huge search space; second is that, no analytical method can be used directly to get the global optimum (Best solution). The third problem epistasis, while in search for a global optimum, a small change in one parameter will result in an enormous difference in the outcome. Due to this problem, the time required to solve this issue increases exponentially. When the length of the binary sequence is small (<48~50), linear search can be implemented as the search space is small. With medium length sequences (50-200) various stochastic and Mimetic algorithms have been implemented. These algorithms will not do an exhaustive search but instead try to estimate the location of global optima in the vast search space. As the sequence length increases (beyond 200), even the stochastic methods become incapable to deal with this problem. At this point, the only possible approach would be to generate these sequences directly. Legendre sequences serve as the perfect example of this type of construction. Research has proven that enhancing the Legendre sequences will further improve the merit factor values to a greater degree. There are numerous methods used to generate long binary Sequences, apart from Legendre sequences; others include twin prime, three and four prime Jacobi sequences.

2. LITERATURE

Linear search is applicable only for short length binary sequences. Golay [4] presented the work of Lindner, who performed an exhaustive search for N≤32.Mertens showed a parallel branch and bound algorithm for short length sequences up to a range of N=48[5].This algorithm was later improvised by Bauke [6], and he has achieved the merit factors up to a range of N=60.The linear search failed to produce better results for long length sequences. Due to the lack of scalability of these methods, long sequences cannot be dealt with limited time and computing power.

Stochastic methods were introduced to solve this problem [10][12]. In the beginning, even these methods failed to produce proper outputs. Simulated Annealing and Ant colony optimization methods failed miserably. But later, Prestwich [7] used a CLS algorithm that utilized constraint programming to obtain results up to a length of N=48. After that, data used a Tabu search algorithm to get results up to a range of N=48.

To obtain extended binary sequences, we have to rely on various construction methods. One of them is Legendre sequences. If N is the length of the binary sequence, we want to obtain the merit factors as N tends to infinity. There is an observation by Turyn that, a quarter rotated periodic Legendre sequence tends to a merit factor value of 6.0 as N tends to infinity [2]. These same phenomena were proved to be true for modified Jacobi sequences of length (pq). Recent improvements include the achievement of merit factor observations up to a range of 3000 by two students [3].

Matthew Parker and Kristiansen [1] used low complexity search to obtain a merit factor of 6.3421, they have extended the Legendre sequences. We have analyzed their method and made lots of improvements. We have applied N/4 Rotated periodically extended Legendre sequences to steepest descent and prime step algorithms with some modifications. We call these techniques as modified Legendre algorithms. We have achieved a merit factor of 6.4245 with these algorithms.

3. BACKGROUND

3.1 Autocorrelation, Energy, Merit Factor

Correlation function is a statistical tool that is used to measure the similarity between two sequences. Only it answers the question "to what level is sequence one similar to sequence two?" Autocorrelation is the correlation of a sequence with itself with the presence of some time lag.

Assume that A is a binary sequence with length N. We will represent the sequence by $a_1a_2a_3\ldots a_N$ with ai ∈ {-1, 1} for $1 \le i \le N$. The aperiodic autocorrelation of elements in the sequence A is

$$c_m = \sum_{i=1}^{N} a_i \, a_{i+m} \qquad (-N < m < N) \qquad 1$$

The cross-correlation can be defined, in the same way. If A and B are two different sequences, the cross-correlation between A and B is X=A*B

$$x_m = \sum_{i=1}^{N} a_i\, b_{i+m} \quad (-N < m < N) \qquad 2$$

Low autocorrelation binary sequence problem for the length I, is represented as LABS (I).

The solution for LABS (I) involves, finding a binary sequence of length I, which bears the minimum energy. Assume that I=3, we already knew the solutions for this range. We have four solutions for range=3.Thesolutionsare{1,-1,-1},{-1,1,1},{-1,-1,1},{1,1,-1}.The energies of all these sequences are same.

The most important property of LABS sequences is symmetry. Even if the obtained result is complemented or reversed, the energy value will remain the same. This feature is crucial in obtaining remaining global optima. If we know one solution we can complement it to get another solution.

Golay [13] for the first introduced the concept of merit factors. The merit factor indicates the autocorrelation side lobe energy of a given sequence

$$F(A) = \frac{N^2}{2\sum_{m=1}^{N-1} c_m{}^2} \qquad 3$$

3.2 Binary Sequence – Element Flipping

We have applied steepest descent and prime step algorithms to Legendre sequences. The whole procedure will be carried out in sequential steps to reduce the execution time. The first step is to generate the Legendre sequences. Before applying these sequences to algorithms, we have to perform an intermediate step. This action is called one-element or multi-element flipping. In order to understand why we are doing this step, we have first to understand how steepest descent and prime step algorithms work. These two algorithms help us to determine the sequence that has the least autocorrelation energy. The algorithm has to identify those elements in the sequence which upon flipping (1 to -1 or -1 to 1) will give the least energy. There will be one element flips or multi-element flips

3.2.1 One – Element Flips

In this case, only one element is flipped at a time, and we iterate the whole process for all the elements. The flip with the least energy is the solution. To do this, we will calculate the energy of primary Legendre sequence and then we have to take the energies of sequences for all the possible flips. Then we will calculate the difference between primary sequence energy and current flipped sequence energy. After that, we will compare all the differences to see which flip got the least energy. Let us say that with one element flip the difference in energy is $\delta 1$ .For all the flips, the difference energies will be saved in the vector Δ.

So $\Delta = [\delta_1, \delta_2, \delta_3, \delta_4 \ldots, \delta_N]$. Here, δ_j represents the complete difference in auto-correlation energy between primary and flipped sequence (produced by flipping element j). We compare all the values in Δ. While doing this lot of time wastage would be there. So instead of computing Δ in this procedure we have designed a better approach, by expressing Δ in its correlation and autocorrelation terms and applying Fast Fourier transform techniques (FFT) to it.

Energy of the sequence is given by

$$E = 2\sum_{m=1}^{N-1} c_m^2 \qquad 4$$

If one element in A is changed by flipping then the change in energy is given by

$$\delta_j = 2\sum_{m=1}^{N-1} d_m^2 - E \qquad 5$$

Here D= $[d_1, d_2 \ldots d_N]$ are the auto-correlation energies of the flipped sequence A. The auto-correlation side-lobe energies differ whenever a_i or a_{i+m} in (1) change. That is when i=j or i+m=j

We can simplify the equation (6)

$$d_m = c_m - 2a_j\left[a_{j+m} + a_{j-m}\right] \qquad 6$$

$$d_m = \sum_{i=1}^{N}(a_i\, a_{i+m}) - 2a_j\, a_{j+m} - 2a_{j-m}\, a_j \qquad 7$$

We have to define two terms to simplify the notation (7) further

$$S_{j,m} = a_{j+m} + a_{j-m} \qquad 8$$

$$P_{j,m} = a_{j+m}\, a_{j-m} \qquad 9$$

Based on the above assumptions the modified energies will become

$$d_m = c_m - 2a_j\, S_{j,m} \qquad 10$$

And the side lobe autocorrelation energy m is given by

$$d_m^2 = c_m^2 - 4a_j S_{j,m} c_m + 4a_j^2 S_{j,m,}^2 \qquad 11$$

The total change in energy of autocorrelation side lobes with an element flip j is given by

$$\delta_j = 2\sum_{m=1}^{N-1} \left(-4a_j S_{j,m} c_m + 4a_j^2 S_{j,m,}^2\right) \qquad 12$$

$$= 8\sum_{m=1}^{N-1} \left(-a_j S_{j,m} c_m + S_{j,m,}^2\right) \qquad 13$$

The above procedure will take a lot of time to compute the δj for all values of j. We have to simplify this process to execute the steps in less time. We have to remember that

$$S_{j,m}^2 = a_{j+m}^2 + a_{j-m}^2 + 2P_{j,m} \qquad 14$$

So from 13 we get

$$\delta_j = 8\sum_{m=1}^{N-1} \left(-a_j S_{j,m} c_m + 2P_{j,m}\right) + \sum_{m=1}^{N-1} \left(a_{j+m}^2 + a_{j-m}^2\right) \qquad 15$$

$$= 8\sum_{m=1}^{N-1} \left(-a_j S_{j,m} c_m + 2P_{j,m}\right) + 8(N-1) \qquad 16$$

Now we can rewrite the change in energy as

$$\delta_j = -8a_j \sum_{m=1}^{N-1} S_{j,m} c_m + 8\sum_{m=-N+1}^{N-1} P_{j,m} + 8\,(N-2) \qquad 17$$

Observe the equation 17 keenly. We can relate this equation to correlations pair.

$$\sum_{m=1}^{N-1} S_{j,m} c_m \backslash= \sum_{m=1}^{N-1} (a_{j+m} + a_{j-m}) c_m \qquad 18$$

$$= \sum_{m=1}^{N-1} (C_m a_{j+m} + C_m a_{N+1-j+m}^r) \qquad 19$$

Here r indicates the reverse of the corresponding sequence. Equation (19) is now in the form of cross-correlation (2)

$$\sum_{m=1}^{N-1} S_{j,m}c_m = (C*A)_j + (C*A^r)_{N-j+1} \qquad 20$$

Here the alphabet C represents the autocorrelation sidelobe energies of A*A. The second term in (7) is also in the form of correlation.

$$\sum_{m=-N+1}^{N-1} P_{j,m} = \sum_{m=-N+1}^{N-1} a_{j+m}a_{j-m} \qquad 21$$

$$= \sum_{m=-N+1+j}^{N-1+j} a_m a^r_{N+1-2j+m} \qquad 22$$

We already know that $1 \leq am \leq N$, so we have to change the limits of the summation to make the sum congruent to cross-correlation (2).We have to do this in such a way that it will not affect the aggregate value

$$\sum_{m=-N+1}^{N-1} P_{j,m} = \sum_{m=1}^{N} a_m a^r_{N+1-2j+m} \qquad 23$$

$$= (A*A^r)_{N+1-2j} \qquad 24$$

We have to mix the two summations to get the δj value

$$\delta_j = -8a_j(C*A)_j + (C*A^r)_{N-j+1} + 8(A*A^r)_{N+1-2j} + 8\,(N-2) \qquad 25$$

If we want to compute the above equation directly, lot of time is required. To simplify the process, we make use of Fourier transform [9]. FFT is a perfect way to compute this sum.

$$\mathrm{FFT}\,(A*B) = \mathrm{FFT}\,(A)\ \mathrm{FFT}\,(B^r) \qquad 26$$

When we apply inverse Fourier transform to above equation we get

$$A*B = F^{-1}(\mathrm{FFT}\,(A)\ \mathrm{FFT}\,(B^r)) \qquad 27$$

Here F-1 indicates the inverse Fourier transform function.

Equation (A.25) consists of multiple correlation pairs. So the Fourier transform implementation involves multiple FFTS. The minimum length of each FFT is 2N-1.If N is of smaller length, equation (A.13) is a better approach than (A.27).This change will not create any problems in our calculations because in our experiments we only consider very long sequences.

3.2.2 *Multi – Element Flips*

So far, we have explained the mechanism of single element flips calculations. Now we want to explain multiple elements flip calculations. In multiple element flips, more than one element flips will be allowed to give the solution, and the whole process repeats for several iterations. We want to describe the procedure for calculations here

Assume that s be a set that consists of all element flips. Let f be another set that consists of all the distances between the flipped elements. Let g_m be the pairwise sum of all the products of all flipped elements. The indices all these elements differ by m. The value of g_m will be zero when m∉f..

$$g_m = \sum_{j\in s,(j+m)\in s} a_j a_{j+m}, \ \text{if } (m\in f) \qquad 28$$

$$g_m = 0, \qquad \text{otherwise}$$

Now the changed autocorrelation side lobe energies will be described as

$$d_m = c_m - 2\sum_{n\in s} a_n S_{n,m} + 4g_m \qquad 29$$

We can rewrite Equation (29) as

$$d_m^2 = -4c_m \sum_{n\in S} a_n S_{n,m} + 4\left(\sum_{n\in S} a_n S_{n,m}\right)^2 + 8g_m c_m \qquad 30$$

$$-16g_m \sum_{n\in S} a_n S_{n,m} + 16g_m^2$$

The total energy of sidelobes is given by

$$\delta_S = 2\sum_{m=1}^{N-1}(d_m^2 - c_m^2) \qquad 31$$

$$= 2\sum_{m=1}^{N-1} n - 4c_m \sum_{m\in S} a_n S_{n,m} + 4\left(\sum_{n\in S} a_n S_{n,m}\right) * \left(\sum_{n\in S} a_n S_{n,m}\right) a$$

$$+2\sum_{m\in f}\left(8g_m c_m 16g_m \ \sum_{n\in S} a_n S_{n,m}\ 16g_m^2\right). \qquad 32$$

Some part of this equation replicates single element changes.

We will use (13) to transform this equation. So we get

$$\delta_S = \sum_{n\in S}\delta_n + 8\sum_{m=1}^{N-1}\sum_{n\in S}\sum_{p\in S,p\neq n} a_n a_p S_{n,m}S_{p,m} \qquad 33$$

$$+16\sum_{m\in f}\left(g_m c_m - 2g_m\sum_{n\in S} a_n S_{n,m} + 2g_m^2\right).$$

We will arrange the triple summations according the requirements of correlations structures.

$$\delta_S = \sum_{n\in S}\delta_n + 8\sum_{n\in S}\sum_{p\in S,p\neq n} a_n a_p \sum_{m=1}^{N-1} S_{n,m}S_{p,m} \qquad 34$$

$$+16\sum_{m\in f}(\delta_m c_m - 2g_m\sum_{n\in S} a_n S_{n,m} + 2g_m^2)$$

Now we have to expand the term $S_{n,\,m}\ S_{p,\,m}$

$$\sum_{m=1}^{N-1} S_{n,m}S_{p,m} = \sum_{m=1}^{N-1}(a_{n+m} + a_{n-m})\left(a_{p+m} + a_{p-m}\right) \qquad 35$$

$$= \sum_{m=1}^{N-1}(a_{n+m}a_{p+m} + a_{n+m}a_{p-m} + a_{n-m}a_{p+m} + a_{n-m}a_{p-m}) \qquad 36$$

$$= \sum_{m=-N+1}^{N-1}(a_{n+m}a_{p+m} + a_{n+m}a_{p-m} - 2a_n a_p) \qquad 37$$

$$= \sum_{m=-N+1+n}^{N-1+n}(a_m a_{p-n+m}) + \qquad 38$$

$$\sum_{m=-N+1}^{N-1}(a_{n+m}a^r_{N+1-p+_m} - 2a_n a_p)$$

$$= c_{p-n} + (A*A^r)_{N+1-p-n} - 2a_n a_p \qquad 39$$

With the help of (39), we can simplify δ_S

$$\delta_s = \sum_{n\in S}\delta_n + 8\sum_{n\in s}\sum_{p\in s,p\neq n}(a_n a_p c_{n-p} - 2) \qquad 40$$

$$+ 8\sum_{n\in s}\sum_{p\in s,p\neq n} a_n a_p\ (A*A^r)_{N+1-p-n}$$

$$+ 16\sum_{m\in f}\left(g_m c_m - 2g_m\sum_{n\in s} a_n S_{n,m} + 2g_m^2\right).$$

$$8\sum_{n\in s}\sum_{p\in s,p\neq n} a_n a_p c_{n-p} = \sum_{m\in f} 16\,g_m c_m \qquad 41$$

$$8\sum_{n\in s}\sum_{p\in s,p\neq n} -2 = -16N_s(N_s - 1) \qquad 42$$

Here NS represents the number of flipped elements. We will use 41 and 42 on 40 to give

$$\delta_s = \sum_{n\in s}\delta_n - 16N_s(N_s-1) + \qquad 43$$

$$8\sum_{n\in s}\sum_{p\in s,p\neq n} a_n a_p\ (A * A^r)_{N+1-p-n} \quad + \ 32$$

$$\sum_{n\in s}\delta_n - \ 16N_s(N_s-1)16\sum_{n\in s}\sum_{p\in s,p>n} a_n a_p\ (A * A^r)_{N+1-p-n}$$

$$+\ 32\sum_{m\in f}\left(g_m c_m - g_m\sum_{n\in s} a_n S_{n,m} + 2\,g_m^2\right)$$

The speed of (43) depends on the number of flipped elements. If Δ is calculated first then δS does not depend upon the length of the sequence N. The time required to implement 43 is very high. So in order to execute the calculation faster 43 is approximated by the following equation.

$$\approx -32\ \textstyle\sum_{m\in f}\left(g_m \sum_{n\in s} a_n S_{n,m}\right) \ \approx\ -32\ N_s^2 \qquad 44$$

If the N value increases, the accuracy of (A.44) decreases. In our experience, our inaccuracies have been minimum.

3.2.3 Reduction of Delta Complexity

Our algorithms implement an iterative method in which single or multi-element flips will occur. Once the flip occurs, the Δ will become useless. We should calculate new Δ for each and every iteration, which makes the program execution very complicated. We found a method to overcome this problem.

We will flip an element k in the sequence. Indicate the modified sequence by a dot. The modified sequence is $\dot{A}$.

$$\dot{a}_i = \begin{cases} -a_i, & i = k \\ a_i, & \text{otherwise} \end{cases} \qquad 45$$

In the similar notation we can write $\dot{\Delta} = [\dot{\delta}_i, \dot{\delta}_2, \ldots \dot{\delta}_N]$. $\dot{\Delta}$ is the new vector for the changed energy. When j = k, we get $\dot{\delta}_j = -\delta_k$.When $j \neq k$, we get $\delta_{j,k} = \delta_k + \delta_j$ .When we apply this change in (A.43) we get

$$\dot{\delta}_j = \delta_{j,k} - \delta_k = \delta_j + 16a_j a_k\left((A * A^r)_{N+1-j-k} + 2c_{|k-j|}\right) \qquad 46$$
$$-32a_j a_{2j-k} - 32a_k\, a_{2k-j} - 64$$

Equation 46 represents the autocorrelation $A*A^r$ and the sidelobe energies C. If we assume that Δ is calculated earlier using (25), then A and A^r need not be calculated again. It is done already at the time of (25).When m = N + 1 - 2k, the modified elements in A and A^r overlap. So $A*A^r$ will not have any change.

$$A * A^r = \sum_{i=1}^{N} a_i a_{N-i+1-m} - 4a_k a_{N+1-m-k} \qquad 47$$
$$= (A * A^r)_m - 4a_k a_{N+1-m-k}$$

We can calculate sidelobe energies without any problem.

They are given by

$$\dot{c}_m = c_m - 2a_k\,(a_{k+m} + a_{k-m}) \qquad 48$$

With the help of (46), we reduced the complexity of Δ calculation. Thus, it will update Δ value continuously

3.3 Construction of Legendre Sequences

3.3.1 Legendre symbol

A lot of algorithms are present in the literature with which we can solve quadratic equations. But the classical methods are restricted to a region C. No algorithms were present in order to solve the quadratic equations over the finite field regions. The complexity is very high in finite fields. Let us take an example. For an integer x, and an odd prime number p, previously it was not even possible to know if there exists any solution for z2≡x(mod p).To solve this kind of typical problems, Legendre introduced his notation[14] to prove the quadratic reciprocity law.

$$\left(\frac{a}{p}\right) = \begin{cases} 1 \text{ only if a is quadratic residue} - \text{modulo p and a} \neq 0 \ (\text{mod p}) \\ -1 \text{ if a is a quadratic nonresidue} - \text{ modulo p} \\ 0 \text{ if a} = 0\ (\text{mod(p)}) \end{cases} \qquad 49$$

An alternate definition of Legendre symbol is given by

$$\left(\frac{a}{p}\right) \equiv a^{\frac{(p-1)}{2}}\ (\text{mod p) and } \left(\frac{a}{p}\right) \in \{-1,0,1\} \qquad 50$$

3.3.2 Legendre Sequence

Now we have an idea of Legendre symbol. We will use the Legendre symbol's notation to define a Legendre sequence

We can define the Legendre sequence as

$$\left(\frac{j}{N}\right) = \begin{cases} 0, & \text{only if } \ j = 0 \\ +1, & \text{only if j is a quadratic} - \text{ residue (mod N)} \\ -1, & \text{other} - \text{wise} \end{cases} \qquad 51$$

Here N is the length of the sequence, and it must be an odd prime. The Legendre symbol assigns zero value to its first element. But in the present sequence, for convenience purpose, we will assign that value to +1.The correct sequence definition is

$$a_j = \begin{cases} 1, & \text{if} \quad j = 0 \\ \left(\frac{j}{N}\right), & \text{if } 0 < j < N \end{cases} \qquad 52$$

3.3.3 Rotated Legendre Sequences

Generally speaking, the merit factors of Legendre sequences lie in a range of 1.5.But when we rotate the Legendre sequences, the merit factor jumps to a value of six. Rotation is the method in which we will detach a part of the Legendre sequence from one side and will attach it to the other end. Golay has introduced an equation describing this method

$$1/_F = \left(2/_3\right) - 4f + 8f^2, \qquad f \leq \frac{1}{2} \qquad 53$$

Here f is the fraction of a rotation

3.3.4 Extended Legendre Sequences

Kirilusha proved that when the rotated Legendre sequences undergo periodic extension, the merit factor of these sequences reaches beyond a value of six. Borwein [8]

Concluded that 0.25 is not the most efficient fraction of the rotation. He recommended a value of 0.2211 for rotation and a value of 0.0578 to extend the sequence. He achieved a merit factor value of 6.3421 using his method. He gave an equation accurately to determine the long Legendre sequence merit factors. The equation is

$$\lim_{n\to\infty} F(X_r;(X_r)^t) = \frac{6(1+t)^2}{-8t^3 + 18t^2 + 1 + 48\left(r - \frac{1}{4} + \frac{t}{2}\right)^2} \qquad 54$$

We need not go into the details of the equation because it does not have a proof.

Kris and parker [1] extended the Legendre sequences with the help of directed search. They almost got the replicable results of that of Borwein. Schmidt [11] in their research proved that m-sequences with periodic extension have a merit factor of 3.34.Equation.53 has been proved without Ergodicity postulate by Jensen and Hoholdt[15].

3.4 Modified Legendre Algorithms

We have developed two algorithms for Legendre sequences. We achieved satisfactory results with these algorithms. The two algorithms are Prime step algorithm and steepest descent algorithm. We are inspired from Baden's research [16].

3.4.1 Prime step algorithm

It is a non-iterative algorithm, and the computation is very simple. The algorithm improves the merit factor by flipping all the elements in the isolated sequence. We will consider only those sequences that have a negative Δ. Consider a new sequence D produced by inverting all the elements in sequence A which has a negative Δ. The formula describes the prime step algorithm.

$$d_j = \begin{cases} a_j & \delta_j \geqq 0 \\ -a_j & \delta_j < 0 \end{cases} \qquad 55$$

The prime step algorithm first calculates Equation (25) and then applies (55) to it with no iterations.

3.4.2 Steep descent algorithm (modified version of steepest descent algorithm)

When there is no analytic procedure to obtain the minimum value of a given function, we will go for iterative technique for achieving approximate solution. Newton has defined a method to achieve the minimal value, but it is not reliable. To solve these kinds of problems, we go to steepest descent method. The first step in the algorithm is to calculate the gradient of the given function at a given point. In order find the local optimum, we should proceed stepwise towards negative gradient of given function. We will begin at a place x0 and slowly we will proceed from xi to xi+1 by taking minimal values along the straight line from xl towards the course of –ve gradient.

Consider an example: If we apply this algorithm to a random function f(x), the structure of the iteration will be

$$x_i = x_{i-1} - \varepsilon f'(x_{i-1}) \qquad 56$$

We applied steepest descent algorithm to LABS problem. Results were satisfactory, but still we tried to improve them. For very long sequences, the inability of steepest descent method to create multiple flips in a single iteration became a limitation. We were not able to get the merit factors in quick time. The algorithm flips the elements one at a time, and the flip that gives the highest merit factor will be the solution. Without multiple element flips and a better iterative procedure, it is almost impossible to say that this method has the best approach. The steepest descent algorithm is very simple, but it has its limitations when applied to long Legendre sequences

We modified the steepest descent algorithm and named it as steep descent algorithm. This algorithm allows multiple element flips in each iteration. So the number of iterations will be less, which improves the speed of the algorithm. In every iteration, all the elements in the sequence with negative Δ value will be flipped. If the merit factor value improves with this iteration, Δ is again calculated for the new sequence, and the iterations will go on. If the value of the merit factor is not increasing then, we will consider the element subsets with – Δ (subsets range from 10%,20% ….so on).Algorithm selects the subset that provides the best merit factor implements it, and the process of iteration will continue. When additional flips do not produce any improvement in merit factors, the iteration procedure will terminate. This termination will also occur when large number of iterations is over.

We have tried several variations of steep descent method. We varied the number of element flips in each iteration, and we observed the results. For almost all the variations, the merit factor values were identical. The changes are not discussed here as they are not important. It became apparent to us that the steep descent algorithm is far more efficient than steepest descent algorithm. We named, both the prime step algorithm and steep descent algorithm as modified Legendre algorithms because with the application of these algorithms the merit factors of Legendre sequences has reached above 6.4

4. RESULTS

To get the results, we applied Legendre sequences (both normal and extended) to both of the modified algorithms. We got significant improvements in merit factors for various sequences. We conclude that, on a relative basis, steep descent algorithm performed well than prime step algorithm. In results, first we will discuss the Prime step results and then the Steep descent results. Later, we will compare the results of Steep descent and prime Step algorithms using two statistical methods. Finally, we will discuss the highest merit factors obtained in our experiments.

4.1 Prime Step Algorithm Results

There was a significant improvement in merit factor values when we applied the prime step algorithm to the Legendre sequences. We have selected only Legendre sequences because even without the application of prime step algorithm, they are capable of producing good results by themselves with the periodical extension. Once we have applied the algorithm, it further improved merit factor values.

Table 1

Sequence Length	Legendre -Merit factor	Periodically extended - merit factor	Prime step -Merit factor
467	1.5021	6.0096	6.4039
1669	1.4987	5.9498	6.3235
2309	1.499	6.0039	6.3235
3461	1.4994	6.0098	6.3314
4973	1.4994	5.9838	6.3401
5821	1.4996	5.99	6.3511
6221	1.4996	5.9811	6.3275
7741	1.4997	5.9856	6.3378
7829	1.4997	5.9957	6.3243
17597	1.4999	5.9921	6.3343

Table 1 shows the merit factor values for some of the Legendre sequences. The generated Legendre sequences tend to a merit factor range of 1.5(approximately).When we rotated the sequence by N/4 and extended it periodically, the merit factors tend to a range of 5.9 to 6.0.But when we applied prime step algorithm to it the merit factor range improved significantly to (6.3-6.4).

We have conducted experiments for odd primes up to a length of 40000.We have observed the same phenomenon in all these sequences. We tried to represent all those merit factors in a graph, but the graph lacks clarity. So we sampled a sequence from each of hundredth series. So for a length of 40000 we got 400 sequences that would efficiently determine the performance of our results

Fig.1 represents the comparison graph for 400 elements; they are samples from 40000 elements. We can conclude from the figure that the merit factors of Prime step Legendre completely dominated the merit factors of normal Legendre sequences. We can also prove this in the case of extended Legendre sequences. But Prime step is not our best algorithm, Steep descent is. So we will show the comparison in steep descent algorithm case.

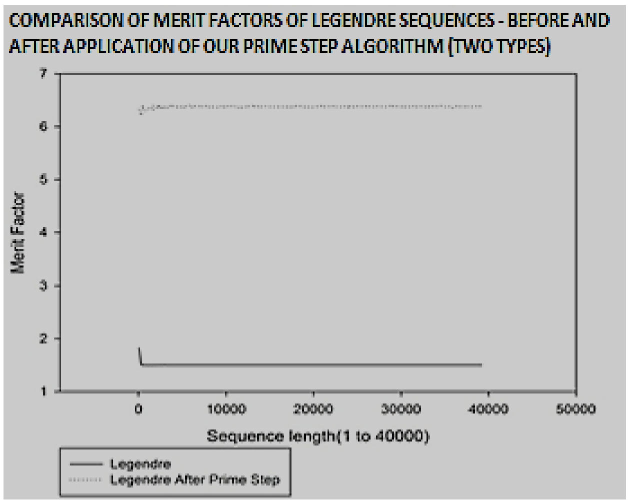

Fig.1

4.2 Steep Descent Algorithm Results

So far in our experiments this algorithm provided the best results. At shorter lengths, the merit factor values for the prime step and steep descent algorithms are almost identical. But with an increase in sequence length, we observed a significant difference in both the values. Steep descent provides better results for long length sequences.

Table 2

Sequence length	Prime step – Merit Factors	Steep Descent – merit factors
127	6.1117	6.1117
131	5.8726	5.8726
167	6.0063	6.0063
179	6.1716	6.1716
16573	6.3450	6.3459
16633	6.3766	6.3811
16661	6.3778	6.3813
16699	6.3504	6.3512
16829	6.3617	6.3647

In Table 2 we want to show the difference between the two algorithms for short and long length sequences. So we took

the first four column elements from a range below 1000 and the next five column elements from a range of 16000.We can clearly see the increase in merit factor for long length sequences in case of Steep descent legend merit factors. Whereas for short length sequences the merit factors for both the algorithms remains the same.

Similar to Prime step algorithm case, we want to compare the results of Legendre sequences with Steep descent Legendre sequences .Figure 3 describes an exact representation of that. We can clearly observe the domination of steep descent merit factors over Legendre merit factors. In Table 3 we have represented some of our steep descents Legendre results. Our merit factor reached around 6.4245.We also compared the steep descent results with extended Legendre results in Figure 2.Clearly steep descent merit factors dominate the extended Legendre merit factors.

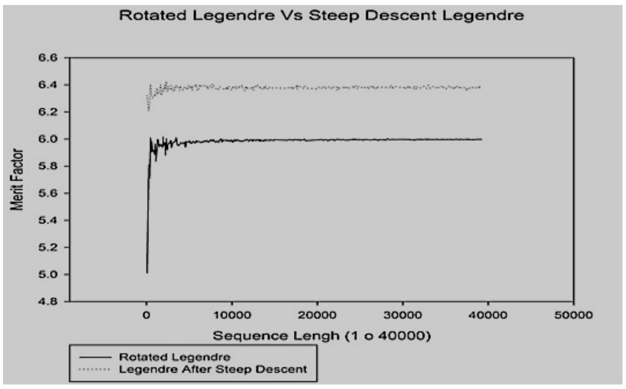

Fig.2

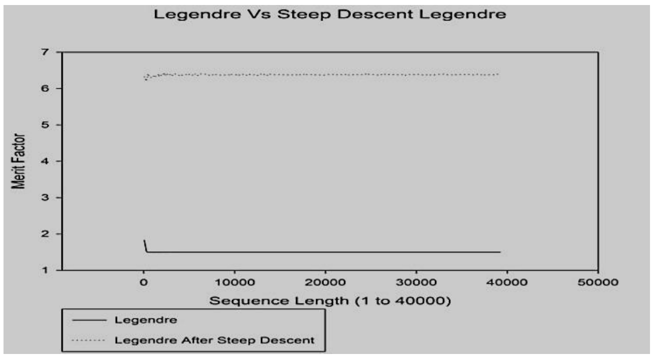

Fig.3

4.3 Steep Descent VS Prime Step Results

The results show that the steep descent results improve over

Table 3

Sequence Length	Legendre -Merit factor	Periodically extended - merit factor	Steep Descent- Merit Factor
467	1.5021	6.0096	6.4039
1669	1.4987	5.9498	6.4008

2141	1.499	5.9756	6.3981
2309	1.499	6.0039	6.4245
3461	1.4994	6.0098	6.3979
5821	1.4996	5.99	6.4017
6221	1.4996	5.9811	6.4033
7741	1.4997	5.9856	6.4078
12941	1.4998	5.997	6.3988

Prime step results for long sequences. We compared the two results in Figure 3.There is no clarity in visual representation. So we should adopt another method that would determine the results clearly. We used two statistical methods to compare the results. The results show that the steep descent results improve over Prime step results for long sequences. We compared the two results in Figure 3.There is no clarity in visual representation. So we should adopt another method that would determine the results clearly. We used two statistical methods to compare the results.

1) Mann-Whitney Rank Sum test

2)Kruskal-Wallis one-way Analysis of Variance on Ranks

4.3.1 Mann-Whitney Rank Sum test

We compared the results of Prime step Legendre and Steep descent Legendre algorithms using this tool. This statistical test ranks the merit factors for each and every condition between two groups (in our case Prime step Legendre and Steep descent Legendre) and then it will analyze how different the two rank sums are. If there exists a systematic change between the two selected groups, then most of the high ranks will represent one group(Steep descent).Then most of the low ranks will represent another group(Prime step).Due to this, the rank totals will change for each cluster. The statistic “U” represents the difference between the two ranks. We can clearly observe the difference in median value, and the 25% and the 75% values.

Mann-Whitney U Statistic = 53183.500

Table 4

GROUP	Number of samples	Median	25%	75%
Prime step Legendre	400	6.378	6.372	6.383
Steep Descent Legendre	400	6.380	6.374	6.386

With test results in Table 4, we can conclude that that the steep descent algorithm performs well than Prime step algorithms in almost all the instances.

4.3.2 Kruskal-Wallis one-way Analysis of Variance on Ranks

We compared the results of Prime step Legendre, Steep descent Legendre algorithms with original Legendre sequence using this tool. It is an extension of Mann-Whitney Rank Sum test for three different groups. We can conclude from the results that prime step and steepest descent algorithms performed well over Legendre sequences.

Table 5

GROUP	Number of samples	Median	25%	75%
Legendre	400	1.500	1.500	1.500
Prime step Legendre	400	6.378	6.372	6.383
Steep Descent Legendre	400	6.380	6.374	6.386

Table 6

Comparison between	Difference of ranks	q	P<0.05
Legendre vs. steep descent Legendre	194624.500	34.013	yes
Prime step Legendre vs steep descent Legendre	17537.000	3.065	No
Legendre vs. Prime step Legendre	177087.500	30.948	yes

In Table 6,”Difference of ranks is very important. High-rank differences indicate that the second group has performed exceedingly well over the first group.

4.4 Best Merit Factors

In table7, we include our top ten merit factors with their corresponding sequence lengths. In figure 4 we compared the top ten merit factors of steepest descent to that of prime step algorithm. Some of the bubbles overlap, so we cannot see all the ten results.

Table 7

Sequence length	Prime step merit factors	Steep descent merit factors
2309	6.4205	6.4245
7741	6.4036	6.4078
467	6.4039	6.4039
6221	6.4033	6.4033
5821	6.4017	6.4017

1669	6.4008	6.4008
12941	6.3956	6.3988
17597	6.3978	6.3986
2141	6.3939	6.3981
3461	6.397	6.3979

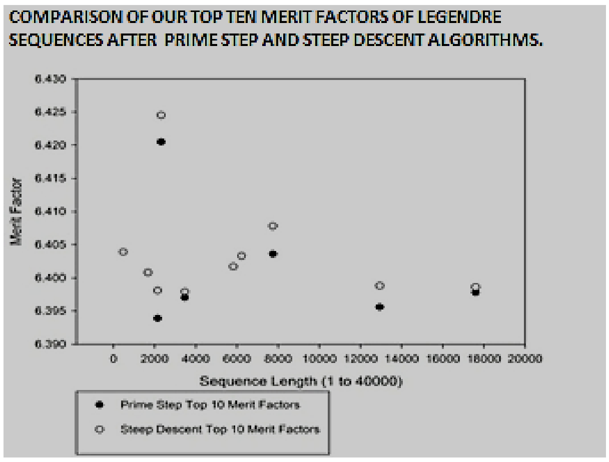

Fig.4

5. CONCLUSION

Extended Legendre sequences exhibit good merit factor range of 6.0. Kristiansen [1] used direct search method on extended Legendre sequences to obtain a merit factor of 6.34.We applied these extended sequences to Prime step and steepest descent algorithms. There is a significant improvement in merit factor range. We got the highest merit factor value of 6.4245.We have calculated the merit factors of Legendre sequences up to a length of 40000.We want to increase the length to a value of one million to observe the behavior of Extended Legendre sequences under the influence of enhanced algorithms. If Kristiansen had utilized the inner values in their direct search, they might achieve the same results that we have got. A lot of research is going on various classes of sequences other than Legendre sequences. Jacobi sequences, Modified Jacobi sequences, m-sequences are the others. In future, we want to apply our algorithms to these sequences to observe the asymptotic merit factor behavior. These sequences by themselves exhibit good merit factor values. The major limitation of LABS problem is computational power. If we can access supercomputers instead of normal ones, we can achieve very interesting results and we can do the research on very long length sequences (say 100 million).

6. REFERENCES

[1] R.A Kristiansen and M.G.Parker," Binary Sequences With merit factor ≥ 6.3",IEEE Trans.Theory,vol.50,no,12,pp,3385-3389,Dec.2004.

[2] M. J. E. Golay, "The merit factor of Legendre sequences,"IEEE Trans. Inf. Theory, vol. IT-29, no. 6, pp. 934–936, Nov. 1983.

[3] A. Kirilusha and G. Narayanaswamy, "Construction of New Asymptotic Classes of Binary Sequences based on Existing Asymptotic Classes," Tech. Rep. Dept. Math. Comput. Sci., Univ. of Richmond, Richmond, VA, 1999.

[4] M. J. E. Golay, The merit factor of long low autocorrelation binary sequences.,IEEE Transactions on Information Theory 28 (3) (1982) 543–549.

[5] S. Mertens, Exhaustive search for low-autocorrelation binary sequences, Journal of Physics A: Mathematical and General 29 (1996) 473–481.

[6] S.Mertens, H.Bauke,Ground statesof the Bernasconi model with open boundary conditions, website available at http://www-e.uni-agdeburg.de/mertens/research/labs/open.dat (accessed January2007).

[7] S. Prestwich, A hybrid local search for low autocorrelation binary sequences,Technical Report TR-00-01, Department of Computer science,National University of Ireland, Cork, Ireland (2000).

[8] P. Borwein, K.-K. S. Choi, and J. Jedwab, "Binary sequences with merit factor greater than 6.34," IEEE Trans. Inf. Theory, vol. 50, no. 12, pp. 3234–3249, Dec. 2004.

[9] R. N. Brace well, "The Fourier Transform and its Applications", 2nd edition. New York: McGraw-Hill, 1986.

[10] J. E. Gallardo, C. Cotta, and A. J. Fernandez, "Finding low autocorrelation binary sequences with Memetic algorithms,"Appl. Soft Computer.,vol. 9, no. 4, pp. 1252–1262, 2009

[11] J. Jedwab and K.-U. Schmidt, "Appended -Sequences with Merit Factor Greater than 3.34", 2010, submitted for publication

[12] J. Jedwab, "A Survey of the Merit Factor Problem for Binary Sequences," Tech. Rep. Dept. Mathematics, Simon Fraser University, Burnaby, BC, Canada, 2004.

[13] M. Golay, "A class of finite binary sequences with alternate auto-correlation values equal to zero (corresp.)," IEEE Trans. Inf. Theory, vol.IT-18, no. 3, pp. 449–450, May 1972.

[14] J. M. Jensen, H. E. Jensen, and T. Høholdt, "The merit factor of binary sequences related to difference sets," IEEE Trans. Inf. Theory, vol. 37,no. 3, pp. 617–626, May 1991.

[15] T. Hoholdt and H. E. Jensen, "Determination of the merit factor of Legendre sequences," IEEE Trans. Inf. Theory, vol. 34, no. 1, pp. 161–164, Jan. 1988.

[16] John Michael Baden, "Efficient Optimization of the Merit Factor of Long Binary Sequences", IEEE transactions on information theory, vol. 57, no. 12, December 2011.