

A New Method of Generating Optimal Addition Chain Based on Graph

K. Mani ^a, M. Viswambari ^b

^a *Nehru Memorial College, Puthanampatti, Trichy, TamilNadu, India-621 007*

^b *Nehru Memorial College, Puthanampatti, Trichy, TamilNadu, India-621 007*

Abstract

In many number theoretic cryptographic algorithms, encryption and decryption is of the form $x^n \bmod p$, where n and p are integers. Exponentiation normally takes more time than any arithmetic operations. It may be performed by repeated multiplication which will reduce the computational time. To reduce the time further fewer multiplications are performed in computing the same exponentiation operation using addition chain. The problem of determining correct sequence of multiplications requires in performing modular exponentiation can be elegantly formulated using the concept of addition chains. There are several methods available in literature in generating the optimal addition chain. But novel graph based methods have been proposed in this paper to generate the optimal addition chain where the vertices of the graph represent the numbers used in the addition chain and edges represent the move from one number to another number in the addition chain. Method 1 termed as GBAPAC which generates all possible optimum addition chains for the given integer n by considering the edge weight of all possible numbers generated from every number in addition chain. Method 2 termed as GBMAC which generates the minimum number of optimum addition chains by considering mutually exclusive edges starting from every number. Further, the optimal addition chain generated for an integer using the proposed methods are verified with the conjectures which already existed in the literature with respect to addition chains.

Index Terms: Optimal Addition Chain, Graph, Conjectures, All Possible Addition Chain, Minimal Addition Chain.

© 2017 Published by MECS Publisher. Selection and/or peer review under responsibility of the Research Association of Modern Education and Computer Science

1. Introduction

An addition chain is a finite sequence of positive integers called elements, $I = a_0 \leq a_1 \leq a_2 \leq \dots \leq a_r = n$ with the property that for all $i > 0$ there exist a, j, k with $a_i = a_j + a_k$ and $r \geq i \geq j \geq k \geq 0$. This is called an addition chain

* Corresponding author. +917502334348
E-mail address: viswal391@gmail.com

of length r for the target n . An optimal addition chain is the one which has the shortest possible length denoted by $l(n)$ and it is a strictly increasing sequence as duplicate chain elements could be removed to shorten the chain [14].

In addition chain, the first number is always one, every subsequent number is obtained by adding two early numbers and n occurs at end of the chain. For the given exponent, it is possible to generate several addition chains, and the least length is better. If the shortest addition chain is found, then it will be useful to reduce the number of multiplications. Finding the optimal addition chain is very difficult and not necessarily unique. But it is enough to find optimal addition chain. Though, for the given integer, finding at least one of the shortest addition chains is an NP-hard problem. Based on the shortest addition chain, modular exponentiation is performed very fast. For example, when $n=170$, all possible optimum addition chains are

1-2-3-5-10-20-40-45-85-170	1-2-3-5-10-20-40-80-85-170	1-2-3-5-10-20-40-80-90-170
1-2-3-5-10-20-40-80-160-170	1-2-4-5-10-20-40-45-85-170	1-2-4-5-10-20-40-80-85-170
1-2-4-5-10-20-40-80-90-170	1-2-4-5-10-20-40-80-160-170	1-2-4-6-10-20-40-80-90-170
1-2-4-6-10-20-40-80-160-170	1-2-4-8-9-17-34-51-85-170	1-2-4-8-9-17-34-68-85-170
1-2-4-8-9-17-34-68-102-170	1-2-4-8-9-17-34-68-136-170	1-2-4-8-10-20-40-80-90-170
1-2-4-8-10-20-40-80-160-170	1-2-4-8-16-17-34-51-85-170	1-2-4-8-16-17-34-68-85-170
1-2-4-8-16-17-34-68-102-170	1-2-4-8-16-17-34-68-136-170	1-2-4-8-16-18-34-68-102-170
1-2-4-8-16-18-34-68-136-170	1-2-4-8-16-32-34-68-102-170	1-2-4-8-16-32-34-68-136-170

This is because the element 5 in the addition chains can be formed as ($5=2+3$, $5=4+1$), 10 can be formed as ($10=5+5$, $10=8+2$, $10=6+4$), 17 can be formed as ($17=8+9$, $17=16+1$). Then, 34 can be given as ($34=17+17$, $34=18+16$), 85 can be formed as ($85=45+40$, $85=80+5$, $85=51+34$, $85=68+17$). Finally, for 170 it can be formed as ($170=85+85$, $170=90+80$, $170=160+10$, $170=102+68$, $170=136+34$).

It is a known fact that larger the size of the field utilized, harder the problem of optimizing the computation of the field exponentiation. This is because a heuristic strategy is normally used to find the optimal addition chain for hard optimization problems. Since these problems have huge search spaces, they do not provide the guarantee on the quality of the solutions. Normally, a heuristic method starts from a non-optimal solution (partial solution) and iteration. After performing some iteration, it improves the solution until a reasonable valid solution could be achieved. Thus, to improve the partial solution which is considered at the initial stage, either deterministic or probabilistic search criteria is used [18].

Many methods already exist in the literature to generate the optimal addition chain. They are classified into two types viz.; deterministic and evolutionary algorithms. In deterministic, the optimal addition chain may not be obtained at all time. This is because everything is predetermined. Binary method, factor method, window method, sliding window method etc., are some examples of deterministic type. Evolutionary algorithms are inspired by the idea of either natural evolution or social behaviour of insects or birds. Though they may produce optimal addition chains for an integer, they are not obtained by a single run which eventually takes more time. Genetic Algorithm (GA), Artificial Immune System (AIS), Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO) etc., are some evolutionary algorithms.

The rest of the paper is organized as follows. Section 2 illustrates the work related to the addition chain. Section 3 describes some basic definitions commonly used in the literature of addition chain. The proposed methods for generating addition chain based on graphical representation and the working principle of the said methods are shown in section 4. Section 5 discusses the experimental results and their significance. Finally, section 6 ends with conclusion.

2. Related Work

This section shows various related work that were done in the available literature in generating the addition chain and their usefulness in the proposed work.

Edward G. Thurber [2] has described the computational aspects of generating minimal length of addition chains for integer n . In that, Search time for such chains was cut down using various pruning techniques. To increase the efficiency of the search further, the author introduced slant bounds. Also proved Scholz-Brauer Conjecture was true if n includes an l^0 -chain among its minimal chains. In [3], Noboru Kunihiro and et. al. have proposed two methods called run-length method and hybrid method to generate the addition chain. And the performance of these two methods are analysed and finally proved that the hybrid method is efficient than the other methods. The method reduces the addition chain length by 8% compared to other methods. This method works especially for numbers with large hamming weight.

Peter Tummeltshammer and James C. Hoe and Markus Puschel [4] have proposed a circuit based approach which combines the addition chains with constants. In this they have also proposed an algorithm which generates all the constants in the given set. They evaluated the quality of circuit using standard cell library. Also they compared the latency and efficiency of the addition chain based approach with the full multiplier. The method generates an algorithm which provides the multiplication logic using a multiplication circuit. Daniel J. Bernstein [6] has presented two new constructive upper bounds on the cost of two- dimensional addition chains. He also proposed a new binary chain to compute and used three additions per exponent bit in order to help protect against side-channel attacks. In [7] Younho Lee et. al., have proposed an algorithm to find addition/ subtraction chain and reduced the number of windows by subtraction which was based on small window method, and proved that the proposed algorithm found the shorter addition/ subtraction chain compared with previous algorithms.

In [8], Nareli cruz-cortes et.al. have explained the use of artificial immune system (AIS) for generating addition chain. They proposed a method for finding addition chains for hard exponents and the shortest addition chain for exponents of size less than 20 bits. The usage of probabilistic heuristic based on AIS search engine for large exponents was proposed. Raveen R. Goundar et al. [9] has proposed a new strategy to find an efficient doubling- free short addition-subtraction chain for an arbitrary integer by utilizing a precise golden ratio. They showed that Golden Ratio Addition Subtraction chain (GRASC) method has attained 12% to 28% reduction in average chain length compared to other methods.

Raveen R. Goundar et. al. [10] has proposed an efficient SPA resistant elliptic curve scalar multiplication algorithm over odd prime fields. They also proposed an explicit algorithm short addition-subtraction chain by utilizing golden ratio and named it as Golden Ratio Addition Chain (GRAC). The proposed method has attained 3% to 18% reduction in the average chain length. In [11] Fabien et al., have proposed a method to modify a key generation using small Euclidean addition chain which secure against side-channel attacks. Two different ways to generate Euclidean addition chains was proposed. One was analysis of the size and another was distribution of obtained keys. A new scheme in the context of fixed base point scalar multiplication was proposed.

Maurice Mignotte and Amadou Tall [12] have presented the binary method which is optimal for any integer of hamming weight 1 or 2 and also showed that there are exactly four types of addition chains possible for those kind of integers. They proved that binary method is not optimal for integers of hamming weight 3. In [13], Saul Dominguez-Isidro presented Evolutionary Programming (EP) to minimize the length of addition chains. Also, four experiments were designed to test the performance of EP algorithm. This requires less evaluation per run with respect to some state-of-the-art nature-inspired algorithms. Neill Michael Clift [14] described a new algorithm for calculating optimal addition chain. For this, pre-computed values are not needed. This algorithm was faster to calculate ranges of optimal addition chain. In [15], Amadou Tall has proposed Lucas addition-subtraction chains. They also proved that Lucas addition chain gives minimal addition chains for all even integers.

M. A. Mohamed and K. A. Mohd Atan [16] have described a new method called composition method in which it is based on the generalization of decomposition method in which the optimal nearer addition chain is almost obtained. In composition method a single rule is used whereas in decomposition method, it uses dedicated rule for each prime from decomposed n . Amadou Tall and Ali Yassin [17] presented a new way of computing the shortest addition chains using generalized continued fractions and Euclidean algorithm. Yara

Elias and Pierre McKenzie [19] have established the results for arbitrary fixed g and adapted methods for constructing g - addition chains when $g=2$ to the case $g>2$.

In [20] MA Mohamed and MR MD SAID, have proposed an idea of non-adjacent form into decomposition method at prime layer and named the new hybrid method as signed decomposition method (SDM). And proposed SDM produces shorter addition subtraction chain than older methods. Sajal Chakroborty and Babul Hasan [21] have proposed a technique for scenario based multi-period stochastic programming problems. They developed the technique on decomposition based pricing method. A model was also developed by collecting data from super market and analyzed the profit.

3. Basic Definitions

This section describes the basic definitions related to addition chains which are useful in understanding the proposed methodology.

3.1. Basic steps in addition chain

The construction [14] of each element of an addition chain is called a step. For an addition chain $l = a_0 \leq a_1 \leq \dots \leq a_r = n$, the following steps are involved.

Doubling step: $a_i = 2a_{i-1}, i > 0$

Non-doubling step: $a_i = a_j + a_k, i > j > k \geq 0$

The steps of the form $a_i = 2a_j, j \leq i - 2$ are defined as non-doubling steps.

Big step: $\lambda(a_i) = \lambda(a_{i-1}) + 1$

Small step: $\lambda(a_i) = \lambda(a_{i-1})$

Thus, the length of the addition chain $l(n)$ can be split into two components as

$$l(n) = \lambda(n) + S(n)$$

It is noted that, not all doubling steps are big steps but big steps are always doubling [14]. Because $\lambda(n)$ is fixed for a given positive integer, finding optimal addition chains amounts to minimizing the number of small steps across all possible chains. Once the addition chain is generated it must be proved or disproved with various conjectures which already exist in the literature.

As Knuth observed [1], either $\lambda(ai) = \lambda(ai-1)$ or $\lambda(ai) = \lambda(ai-1) + 1$. In the former case, step i is called a *small step* and is called a *big step* otherwise. There are exactly $\lambda(n)$ big steps in any chain for n . The number of steps, r , in an addition chain for n can be expressed as $r = \lambda(n) + N(n)$, where $N(n)$ denotes the number of small steps in the chain. It should be noted that $N(n)$ is chain dependent. Minimizing $N(n)$ will result in a minimal length addition chain for n . If $j = i - 1$, then step i is called a *star step*. An addition chain that consists entirely of star steps is called a *star chain*. If $j = k = i - 1$, then step i is called a *doubling* [2].

3.2. Conjectures in addition chain

The various existing conjectures in the addition chain proposed in the literature [12] are

- For any integer $n > 2^k$ with $k \in \mathbb{N}$, if $l(n) = k + 1$, then $n = 2^k + 2^j$ for some $j \leq k$.
- $l(2n) = l(n) + 1$.
- $l(2n) \geq l(n)$.
- If p is a prime, show that, $n = 2^p - 1$ is also prime (Mersenne prime.), then $l(n) = \max \{l(m); m \leq n\}, 2 \leq p \leq 7$.
- $l(2^n - 1) \leq n - 1 + l(n)$.

- $l(n) \leq \log_2(n) + S_2(n) - 1$.

4. Proposed Methodology

A graph based addition chain has been proposed in this paper. It is noted that a graph denoted as $G = (V, E)$ consists of the set of vertices $V = \{v_1, v_2, \dots, v_n\}$ and the set of edges $E = \{e_1, e_2, \dots, e_n\}$. But in the proposed graph based addition chain V represents the set of intermediate numbers which are being used in the addition chain. E represents the edges to connect two numbers in the addition chain. The weight of edge denoted as $w(e)$ is a non-negative integer. Initially, $w(e)$ is assigned 1 and it is incremented by 1 when the same edge is used in generating the addition chain. Without loss of generality, let $v_1=1, v_2=2, v_j=m, 3 \leq j < l$, where m is a non-negative integer except 1, 2 and n , and $v_l=n$, where l is the last vertex in which addition chain is to be terminated.

A multi-digraph G is a finite non-empty set of objects called vertices denoted by V with a multi-set of ordered vertex pairs called arcs denoted by E . Duplicate elements are allowed in a multi-set. An edge goes from a vertex $u \in V$ to a vertex $v \in V$ if $(u, v) \in E$ [14]. A directed multi-digraph (V, E) consists of vertices V and edges E and a function $f: E \rightarrow V \times V = \{(u, v) | u, v \in V\}$. In the proposed graph based addition chain, to generate the addition chain acyclic multi digraph is used. This is because minimum two numbers can be generated from a particular number by adding the current number to the previous number or doubling the current number itself.

Formally for an addition chain A of length r we have a multi-digraph $G_A = (V, E, \alpha, \omega)$ where V is the set of vertices, E is the set of edges and α, ω are mappings that take an edge to its start and end vertex respectively. This gives $V = \{v_i : 0 \leq i \leq r\}$ and $E = \{(v\gamma(i), v_i), (v\delta(i), v_i) : 1 \leq i \leq r\}$, $\alpha, \omega : E \rightarrow V, (v_1, v_2) \alpha \rightarrow v_1, (v_1, v_2) \omega \rightarrow v_2$. We label each vertex with its numerical value from the addition chain. In a graph with directed edges, the in-degree of a vertex v , denoted as $deg^-(v)$, is the number of edges with v as their terminal vertex. The out-degree of a vertex v , denoted as $deg^+(v)$, is the number of edges with v as their initial vertex. Thus, for each vertex, the d^+ and d^- are minimum two except the first. The vertex would have represented an element calculated in the chain that was not used in the construction of the target.

4.1. Proposed Method 1: Graph Based All Possible Addition Chain (GBAPAC)

In the proposed GBAPAC method, the first two numbers are always 1 and 2 and the last number is always n , where n is the number for which addition chain is to be formed. To generate the next number in the addition chain from 2, the possibilities are 3 and 4. They are obtained by addition and doubling steps respectively and their corresponding edge weight of 2-3, 2-4 is 1. As 3 and 4 are generated simultaneously from 2, any one of them is considered as next number in the addition chain.

Suppose 3 is selected as next number, the other possible numbers from 3 are 4, 5, 6, where 4, 5 are obtained by addition step and 6 is by doubling step. As three numbers are generated from 3, the weight of edge 2-3 is 4 (1+ 3=4) but the edge weight for 3-4, 3-5, 3-6 is 1, since those edges are newly generated. Similarly, if 4 is taken as the next number in addition chain the weight of edge 2-4 is 4 (1+3) with the possibilities from 4 are 5, 6 and 8. Correspondingly the edge weight for 4-5, 4-6, 4-8 is 1. Likewise, taking 5 as next number, 5 can be generated from both 3 and 4. If 5 is obtained from 3, then the other possible numbers from 5 are 6, 7, 8 and 10. At the same time edge weight for 2-3 is 6 and 3-5 is 5 but the edge for 5-6, 5-7, 5-8, 5-10 is 1 because they are newly created edges. On the other hand, if 5 is obtained from 4, then the other possibilities are 6, 8, 9 and 10 and their corresponding edge weight is 1. But the edge weight of 2-4 and 4-5 is incremented by one every time when a new possibility is generated. Thus, the edge weight for 2-4 is 6 and 4-5 is 5. The process is terminated when n is reached.

In general, let $i=1; v_i=1$ and $v_{i+1}=2$. The corresponding edge weight $w(e_i(v_i, v_{i+1}))=1$. To generate the next number, $i=i+1; v_i \leftarrow v_{i+1}$, where v_{i+1} is computed as $v_{i+1} \leftarrow \{v_i+v_j, 1 \leq j \leq i\}$, $w(e_i(v_i, v_{i+1}))=1$, $w(e_i(v_i, v_{i-1}))=w(e_i(v_{i-1}, v_i)) + d^+(v_i)$. As v_{i+1} has sometime more than one possibility depending on j , at a time any one value of

v_{i+1} will be taken as the next number randomly. From that, other possible numbers are generated using addition and doubling. The edge weight will be increased based on the numbers which occur previously in the addition chain. Similar process can also be performed for other possibilities of v_{i+1} . The process is repeated till it reaches n and the optimal addition chain is one which has maximum edge weight between numbers starting from 1 to n or the length of the addition chain for the given integer n is accepted as input. The proposed GBAPAC method is shown in algorithm 1.

Algorithm 1 GBAPAC(n)

```

//This algorithm is used to find the optimal addition chain for the given integer n
//opac – optimal addition chain, lopac – length of opac
Input n, w
Output opac (n), lopac (n)
Begin
Step 1: Initialization of required variables
     $i \leftarrow 1; c_1 \leftarrow 1; c_2 \leftarrow 2;$ 
Step 2: Outputting the first edge of addition chain
    Print  $C_1$  ‘-’  $C_2$ 
Step 3: Switching to the next number of addition chain
    
$$C_3 \leftarrow \begin{cases} C_1 + C_2 \\ C_2 + C_2 \end{cases}$$

     $e_1(i) \leftarrow c_1 // c_2$ 
     $w_1(i) \leftarrow 1; e_2(i) \leftarrow c_2 // c_3; w_2(i) \leftarrow 1$ 
     $opac(i) \leftarrow e_1(i); lopac(i) \leftarrow 1$ 
     $pac(i) \leftarrow e_1(i) // e_2(i)$ 
     $lpac(i) \leftarrow w_1(i) + w_2(i)$ 
Step 4: Repeat the following till the optimal addition chain is reached
do
{
     $i++$ 
    if ( $i \geq 2$ )
    {
         $t \leftarrow c_3$ 
        
$$C_3 \leftarrow \begin{cases} C_3 + C_1 \\ C_3 + C_2 \\ C_3 + C_3 \end{cases}$$

         $c_1 \leftarrow c_2; c_2 \leftarrow t; e_1(i) \leftarrow c_1 // c_2$ 
         $w_1(i) \leftarrow w_1(i-1) + d^+(v_i) *$ 
         $e_2(i) \leftarrow c_2 // c_3; w_2(i) \leftarrow 1$ 
         $opac(i) \leftarrow pac(i-1); lopac(i) \leftarrow w_1(i)$ 
         $opac(i) \leftarrow opac // C_3; lpac(i) \leftarrow lpac(i) + 1$ 
         $i++;$ 
    }
}
while ( $lopac \leq w \ \&\& \ c_3 == n$ )

```

Algorithm 2 GBMAC(n)

All the steps involved in algorithm 1 GBAPAC are also used in algorithm 2 GBMAC except $w_1(i) \leftarrow w_1(i-1)+1$

4.1.1. The working principle of GBAPAC algorithm

The working principle of the proposed GBAPAC algorithm is shown in Table 1. In order to understand the proposed method, the following notations are used.

{ac- Addition chain; c_1 - previous number in ac; c_2 -current number in ac; c_3 -next number in ac $e(c_i, c_j)$: edge between i^{th} and j^{th} number; $w(e)$:weight of edge e ; $l(ac)$ - length of addition chain; opac -optimal addition chain; $l(opac)$ -length of opac; apac: possible addition chain; $l(apac)$ -length of apac}.

Table 1. Working Principle of GBAPAC

c_1	c_2	c_3	$e_1(c_1,c_2)$	$w(e_1)$	$e_2(c_1,c_2)$	$w(e_2)$	opac (c_2)	$l(opac (c_2))$	apac(e_2)	$l(apac(e_2))$
1	2	3	1-2	1	2-3	1	1-2	1	1-2-3	2
		4	1-2	1	2-4	1	1-2	1	1-2-4	2
2	3	4	2-3	2	3-4	1	1-2-3	2	1-2-3-4	3
		5	2-3	2	3-5	1	1-2-3	2	1-2-3-5	3
		6	2-3	2	3-6	1	1-2-3	2	1-2-3-6	3
	4	5	2-4	3	4-5	1	1-2-4	2	1-2-4-5	3
		6	2-4	3	4-6	1	1-2-4	2	1-2-4-6	3
		8	2-4	3	4-8	1	1-2-4	2	1-2-4-8	3
3	4	7	3-4	2	4-7	1	1-2-3-4	3	1-2-3-4-7	4
4	5	6	4-5	2	5-6	1	1-2-4-5	3	1-2-4-5-6	4
		7	4-5	2	5-7	1	1-2-4-5	3	1-2-4-5-7	4
		9	4-5	2	5-9	1	1-2-4-5	3	1-2-4-5-9	4
		10	4-5	2	5-10	1	1-2-4-5	3	1-2-4-5-10	4
	6	7	4-6	2	6-7	1	1-2-4-6	3	1-2-4-6-7	4
		8	4-6	2	6-8	1	1-2-4-6	3	1-2-4-6-8	4
		10	4-6	2	6-10	1	1-2-4-6	3	1-2-4-6-10	4
		12	4-6	2	6-12	1	1-2-4-6	3	1-2-4-6-12	4
4	5	8	4-5	2	5-8	1	1-2-3-4-5	4	1-2-3-4-8	5
	6	9	4-6	2	6-9	1	1-2-3-4-6	4	1-2-3-4-6-9	5
		11	4-6	2	6-11	1	1-2-3-4-5-6	5	1-2-3-4-5-6-11	6
	7	8	4-7	2	7-8	1	1-2-3-4-7	4	1-2-3-4-7-8	5
		9	4-7	2	7-9	1	1-2-3-4-7	4	1-2-3-4-7-9	5
		10	4-7	2	7-10	1	1-2-3-4-7	4	1-2-3-4-7-10	5
		11	4-7	2	7-11	1	1-2-3-4-7	4	1-2-3-4-7-11	5
		14	4-7	2	7-14	1	1-2-3-4-7	4	1-2-3-4-7-14	5
	8	9	4-8	2	8-9	1	1-2-4-8	4	1-2-4-8-9	4
		10	4-8	2	8-10	1	1-2-4-8	4	1-2-4-8-10	4
		12	4-8	2	8-12	1	1-2-4-8	4	1-2-4-8-12	4
		16	4-8	2	8-16	1	1-2-4-8	4	1-2-4-8-16	4
		11	4-8	2	8-11	1	1-2-3-4-8	4	1-2-3-4-8-11	5

4.1.2. Proposed Method 1 – An Example

To generate the addition chain for an integer $n=12$. The length of the addition chain given as input is 4. As the addition chain always starts with 1, i.e., $v_1=1$ and the next number should be 2 i.e., $v_2=2$ which is obtained either by adding 1 to itself or doubling it. Since 1 and 2 are must in generating the addition chain for 12, the edge weight is not considered. It is shown in fig.1.

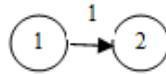


Fig.1. GBAPAC for 2

After switching to v_2 , the next number in the addition chain is $v_3 = \{3, 4\}$ where 3 and 4 are obtained either by adding 1 to 2 or doubling 2 itself respectively. Correspondingly, $(w(e(v_2, v_3))=1)$ It is shown in fig. 2.

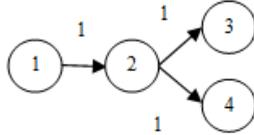


Fig.2. GBAPAC from 2

Suppose the next number in the addition chain generated is $v_3=3$, from v_3 the other numbers $v_4 = \{4, 5, 6\}$ are generated using addition or doubling steps and $w(e(v_3, \{v_4\})) = 1$. But at the same time $w(e(v_2, v_3)) = w(e(v_2, v_3)) + d^+(v_3) = 1 + 3 = 4$. It is noted that the optimal addition chain is found for every starting number at each stage. Thus, the optimal addition chain for 3 is 1-2-3, because 3 is the starting number and it has the maximum edge weight from the previous number 2. It is shown in fig. 3.

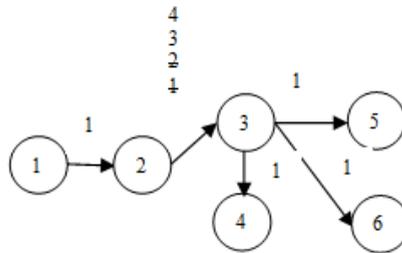


Fig.3. GBAPAC from 3

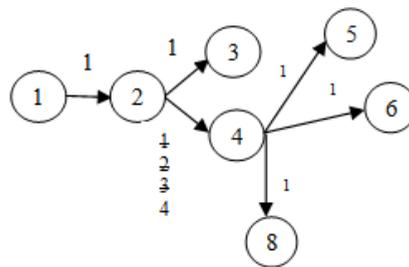


Fig.4. GBAPAC from 4 where the Previous Number is 2.

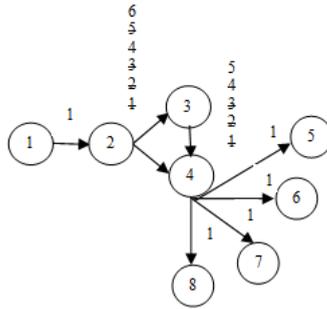


Fig.5. GBAPAC from 4 where the Previous Number is 3

On the other hand if 4 is taken as the starting number in the addition chain $v_3=4$ then the other numbers generated from 4 are $v_4 = \{5, 6, 8\}$ and they are obtained either by addition or doubling steps and $w(e(v_3, \{v_4\})) = 1$. But at the same time $w(e(v_2, v_3)) = w(e(v_2, v_4)) + d^+(v_3) = 1 + 3 = 4$. It is shown in fig. 4. If 3 and 4 is connected then $v_3=3$ and $v_4 = \{4, 5, 6\}$. Thus, $w(e(v_2, v_3)) = e(v_2, v_3) + d^+(v_3) = 1 + 3 = 4$. It is shown in fig.5. As 4 is the starting number, it is obtained in two different ways 1-2-3-4 and 1-2-4, the addition chain 1-2-4 is selected as optimum addition chain for 4 because its length is 2. But the addition chain 1-2-3-4 is not considered as the optimum addition chain even though the edge 2-3 has maximum weight.

Let the next number taken in the addition chain is $v_4=5$ where 5 is obtained either from $v_3=3$ or 4. Correspondingly the optimum addition chain for 3, 4 is 1-2-3, 1-2-4 respectively. From v_4 the other numbers generated are $v_5 = \{6, 7, 8, 10\}$ or $v_5 = \{6, 7, 9, 10\}$ are generated using addition or doubling steps if $v_3=3$ or 4 respectively. Then $w(e(v_4, \{v_5\})) = 1$, $w(e(v_3, v_4)) = (e(v_3, v_4)) + d^+(v_5) = 1 + 4 = 5$ and $w(e(v_2, v_3)) = w(e(v_2, v_3)) + w(e(v_3, v_4)) = 2 + 4 = 6$. As 5 is the starting number at this stage, there are two different optimal addition chains for 5 viz., 1-2-3-5 or 1-2-4-5. It is shown in fig.6 and fig.7 respectively.

Let the next number taken in the addition chain is $v_5=6$ where 6 is obtained either from $v_3=3$ or $v_4=5$. Correspondingly the optimum addition chain using 3, 4 is 1-2-3, 1-2-4, $\{1-2-3-5, 1-2-4-5\}$ respectively. From v_5 the other numbers generated are $v_5 = \{7, 9, 10, 12\}$ or $v_5 = \{7, 9, 10, 11, 12\}, \{7, 8, 10, 11, 12\}$. They are generated using addition or doubling steps if $v_3=3$ or $v_4=5$ respectively. Then $w(e(v_5, \{v_6\})) = 1$, $w(e(v_3, v_4)) = w(e(v_3, v_4)) + 4 = 5$ if $v_3=3$, $w(e(v_3, v_4)) = w(e(v_3, v_4)) + w(e(v_4, v_5)) + 5$, $w(e(v_2, v_3)) = w(e(v_2, v_3)) + w(e(v_3, v_4))$ if $v_4=6$, as 6 is the starting number at this stage, there are two different optimal addition chain for 6 viz., 1-2-3-6 or 1-2-4-6. Further, 12 is obtained from doubling of 6, the optimal addition chain for 12 is 1-2-3-6-12 and its length is 4 which is same as the length obtained as input and hence the process is stopped. It is shown in fig.8. The other possible optimal addition chain for 12 is 1-2-4-8-12 and it is traced in similar manner. It is shown in fig.9.

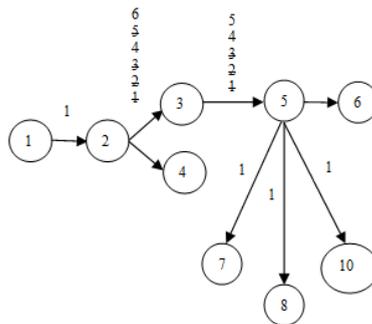


Fig.6. GBAPAC from 3 where the Previous Number is 2

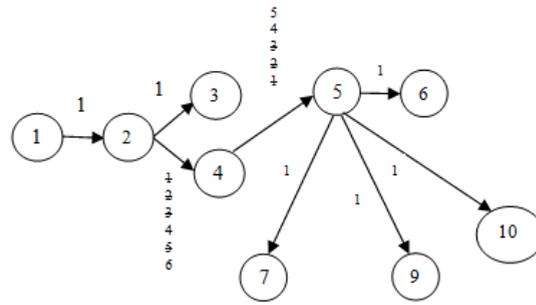


Fig.7. GBAPAC from 4 where the Previous Number is 2

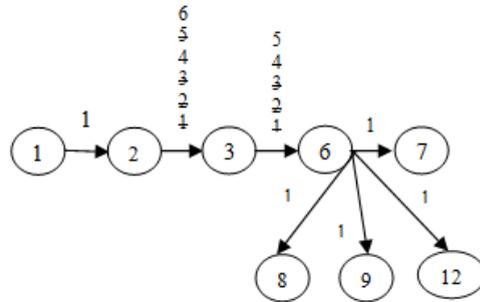


Fig.8. GBAPAC from 6 where the Previous Number is 3

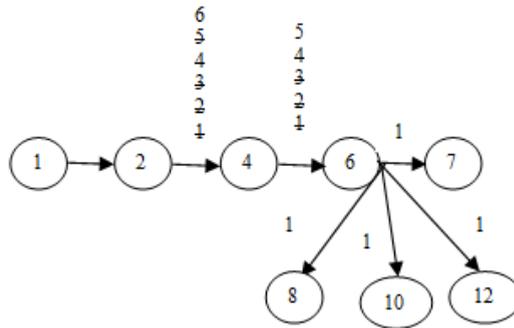


Fig.9. GBAPAC from 6 where the Previous Number is 4

4.2. Proposed Method 2: Graph Based Minimal Addition Chain (GBMAC)

The main difference between GBAPAC and GBMAC is that, in GBMAC not all possible numbers are generated from the particular number in forming addition chain. This is because they are mutually exclusive. That is, only one number is generated by doubling step and the rest of the numbers are generated using addition step. As any one of the step is taken in generating the next number from the current number, the edge weight of current number and previous numbers is incremented by 1 where the previous numbers are numbers from which addition chain is obtained for the current number. It is noted that, the edge weight is not increased

considerably as in GBAPAC because all possible edges are not taken into account. From the current number either the number obtained by doubling step or any one of the number which is obtained by the addition step is taken. The process is repeated till the optimal addition chain is found.

In general, let $i=1$; $v_i = 1$ and $v_{i+1} = 2$. The corresponding edge weight $w(e_i(v_i, v_{i+1}))=1$. To move to the next number, $i = i+1$; $v_i \leftarrow v_i+1$, $v_i \leftarrow v_{i+1}$, where v_{i+1} is computed as $v_{i+1} \leftarrow \{v_i+v_j, 1 \leq j \leq i\}$, $w(e_i(v_i, v_{i+1}))=1$ or $v_{i+1}=2(v_i)$. As v_{i+1} has the number which are obtained either addition or doubling steps but only one number is taken as they are mutually exclusive and hence $w(e_i(v_i, v_{i+1}))= w(e_i(v_{i-1}, v_i))+ 1$. Similar process can also be performed for other possibilities of v_{i+1} but the edge weight of current and previous numbers are incremented by 1. The process is repeated till it reaches n and the length of the addition chain $l(n)$ is found. This $l(n)$ is compared with w_1 which is accepted as input and it is considered as the optimal weight. The optimal addition chain is one which has maximum edge weight between the numbers starting from 1 to n and $l(n)$ should not exceed w_1 . The proposed algorithm is shown in algorithm 2.

4.2.1. The working principle of GBAPAC algorithm

The working principle of the proposed GBMAC algorithm in generating the addition chain for the integer 12 is shown in Table 2. It is seen from the Table 3 that if 2 is the current number, the number 3 is obtained by addition step (2+1) and 4 is obtained from doubling step (2+2). Since they are mutually exclusive at this stage (i.e. 3 and 4 cannot be the 3rd number in any addition chain) any one of the number is taken. In this case 4 is taken as the next number. From 4, the numbers 5, 6 are obtained using addition step and 8 is obtained using doubling step. Even though 5 and 6 are obtained using addition step, they are also mutually exclusive and any one of the number is taken in the next stage for further processing. In general, only one number is considered in generating the next number $w(e(v_i, v_j))= w(e(v_i, v_j))+1$ where $i=2, \dots, j$, and $i \neq j$.

Table 2. Working Principle of GBMAC

C ₁	C ₂	C ₃	e ₁ (C ₁ ,C ₂)	w(e ₁)	e ₂ (C ₁ ,C ₂)	w(e ₂)	opac (C ₂)	l(opac (C ₂))	apac(e ₂)	l(apac(e ₂))
1	2	3	1-2	1	2-3	1	1-2	1	1-2-3	2
		4	1-2	1	2-4	1	1-2	1	1-2-4	2
2	3	6	2-3	2	3-6	1	1-2-3	2	1-2-3-6	3
	4	6	2-4	2	4-6	1	1-2-4	2	1-2-4-6	3
		8	2-4	2	4-8	1	1-2-4	2	1-2-4-8	3
4	6	12	4-6	2	6-12	1	1-2-4-6	3	1-2-4-6-12	4
	8	12	4-8	2	8-12	1	1-2-4-8	3	1-2-4-8-12	4

4.2.2. Proposed Method 2 – An Example

It is noted that to generate the addition chain for $n=12$, the graph shown in fig.2 using GBAPAC is similar to GBMAC in generating numbers 3 and 4. As 3 and 4 are mutually exclusive, 3 and 4 are obtained by addition and doubling steps respectively from the same number. The next number chosen in addition chain is either 3 or 4.

Let the next number be taken in addition chain is 3. To obtain the next numbers from 3, the numbers 4 and 5 are obtained using addition step and 6 is obtained using doubling step correspondingly the edge weight of 3-4, 3-5 and 3-6 are 1. Suppose 4 or 5 or 6 are taken as the next number from 3, the previous edge weight of 2-3 is 3. It is shown in fig. 10.

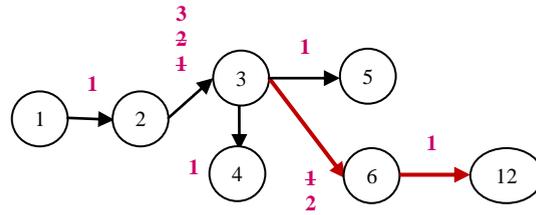


Fig.10. Graph with all Possible Addition Chains from 3.

Let the next number be taken in the addition chain is 4. To obtain the next numbers from 4, the numbers 5 and 6 are obtained using addition step and 8 is obtained using doubling step correspondingly the edge weight of 4-5, 4-6 and 4-8 are 1. Suppose 5 or 6 or 8 are taken as the next number from 4, the previous edge weight of 2-4 is 3. It is shown in fig. 11.

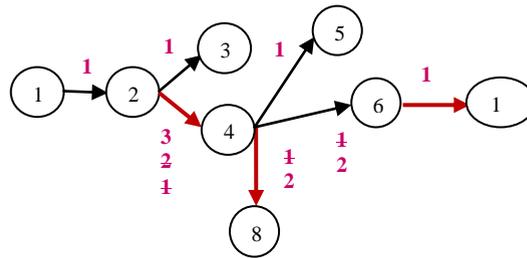


Fig.11. Graph with all Possible Addition Chains from 4.

5. Experimental Results

The proposed GBAPAC and GBMAC methods are implemented in VC++. The total number of addition chains generated using two different methods GBAPAC and GBMAC up to the range of integers n are shown in Table 3. It is observed from Table 3 that after the range above 200, the total number of addition chains generated by GBMAC method is less when the same is compared with GBAPAC and their corresponding graphical representation is shown in fig 12.

Table 3. Total Number of Addition Chains up to 1024 using Both Methods

Range of Integers up to n		100	200	300	400	500	600	700	800	900	1000	1024
Total Number of Addition Chains	GBAPAC	8157	43209	72731	70422	129033	144535	171066	234095	212026	267149	126884
	GBMAC	8157	43239	66398	55762	103061	112535	137029	193414	169565	220025	101525

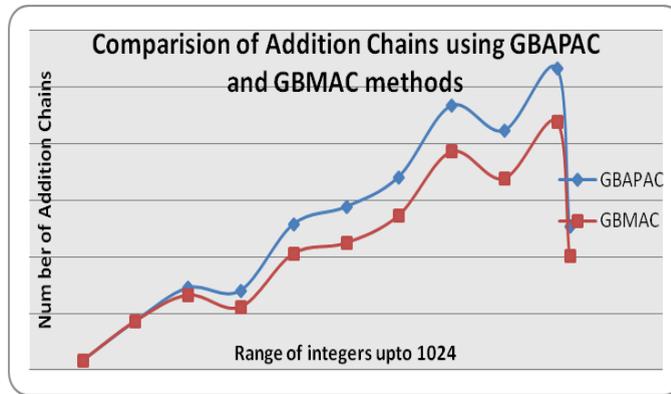


Fig.12. Comparison of Addition Chains using GBAPAC and GBMAC Methods

Table 4. Comparison of Number of Addition Chains and Their Respective Length for Some Integers Generated using GBAPAC and GBMAC

Number	Length		No. of. Chains		Number	Length		No. of. Chains	
	M1	M2	M1	M2		M1	M2	M1	M2
1023	13	13	1072	870	991	13	13	450	360
1021	13	13	934	750	990	12	12	96	80
1020	12	12	240	187	989	13	13	376	304
1015	13	13	898	712	984	12	12	664	536
1007	13	13	137	174	983	13	13	489	388
1003	13	13	844	682	980	12	12	336	260
1000	12	12	656	532	979	13	13	639	511
997	13	13	586	467	978	12	12	24	17
996	12	12	48	39	976	12	12	861	692
995	13	13	750	594	975	12	12	24	20

M1- GBAPAC

M2- GBMAC

Table 4 shows the number of addition chains generated for some specific numbers using GBAPAC and GBMAC methods. It is observed from Table 4 that number of addition chains generated using GBMAC is somewhat minimum when it is compared with GBAPAC for some integers. Table 5 shows the length of the optimal addition chain for the integers upto 1024. It is noted that the optimal addition chain and its length for the integer upto 1024 generated by both GBAPAC and GBMAC methods. They are exactly same as the optimal addition chain and its length produced by EP where EP is one of the latest addition chain algorithm. It is evident from Table 5 that the proposed methods also produce the same total number of optimal addition chain length for the integers upto 1024 as in EP. Beyond that it is not possible to reduce the total length which is shown in Table 5.

Table 6 shows the comparison of addition chain generated by the proposed methods and the existing methods for some hard exponents where the hard exponents are exponent for which the addition chains are difficult to be generated. It is observed from Table 6 that for some integer the proposed methods reduce the length of the addition chain, maintaining the same length when it is compared with GA, PSO and EP. Further, it is noticed from Table 7 that the proposed method never increases the addition chain length (upto 1024 the total length is 11115). Also the proposed addition chain methods also satisfy the conjectures illustrated in section 3.2.

Table 5. Addition Chain Length of Exponents upto 1024

Length	Solutions	Total
1	{2}	1
2	{3,4}	4
3	{5,6,8}	9
4	{7,9,10, 12,16}	20
5	{11, 13, 14, 15, 17, 18, 20, 24, 32}	45
6	{19, 21, 22, 23, 25, 26, 27,28,30,33,34,36,40,48,64}	90
7	{29, 31, 35, 37, 38, 39, 41, 42, 43, 44,45,46,49,50,51,52,54,56,60,65,66,68,72,80,96,128}	182
8	{47,53,55,57,58,59,61,62,63,67, 69,70,73,74,75,76,77,78,81,82,83,84,85,86,88,90,92,97,98, 99,100,102,104,108,112,120,129,130,132,136,144,160,192,256}	352
9	{71,79,87,89,91,93,94,95,101,103,105,106,107,109,110,111,113,114,115,116,117,118,119,121,122,123,124, 125,126,131,133,134,135,137,138,140,145,146,147,148,149,150,152,153,154,156,161,162,163,164,165,166, 168,170,172,176,180,184,193,194,195,196,198,200,204,208,216,224,240,257,258,260,264,272,288,320,384, 512}	702
10	{127,139,141,142,143,151,155,157,158,159,167,169,171,173,174,175,177,178,179,181,182,183,185,186,187 ,188,189,190,197,199,201,202,203,205,206,207,209,210,211,212,213,214,215,217,218,219,220,221,222,225, 226,227,228,229,230,231,232,233,234,236,238,241,242,243,244,245,246,248,249,250,252,255,259,261,262, 265,266,268,270,273,274,276,280,281,289,290,291,292,293,294,296,297,298,300,304,306,308,312,321,322, 323,324,325,326,328,330,332,336,340,344,352,360,368,385,386,387,388,390,392,396,400,408,416,432,448, 480,513,514,516,520,528,544,576,640,768,1024}	1360
11	{191,223,235,237,239,247,251,253,254,263,267,269,271,275,277,278,279,282,283,284,285,286,287,295,299 ,301,302,303,305,307,309,310,311,313,314,315,316,317,318,319,327,329,331,333,334,335,337,338,339,341, 342,343,345,346,347,348,349,350,351,353,354,355,356,357,358,359,361,362,363,364,365,366,367,369,370, 371,372,373,374,375,376,377,378,380,381,382,389,391,393,394,395,397,398,399,401,402,403,404,405,406, 409,410,411,412,413,414,415,417,418,419,420,421,422,423,424,425,426,428,429,430,433,434,435,436,437, 438,440,441,442,444,447,449,450,451,452,453,454,455,456,457,458,459,460,462,464,466,468,472,476,481, 482,483,484,485,486,488,489,490,492,495,496,498,500,504,510,515,517,518,521,522,524,529,530,532,536, 540,545,546,548,552,553,560,561,562,577,578,579,580,581,582,584,585,586,588,592,594,596,600,608,612, 616,624,641,642,643,644,645,646,648,650,652,656,660,664,672,680,688,704,720,736,769,770,771,772,774, 776,780,784,792,800,816,832,864,896,960}	2585
12	{379,383,407,427,431,439,443,445,446,461,463,465,467,469,470,471,473,474,475,477,478,479,487,491,493 ,494,497,499,501,502,503,505,506,507,508,509,511,519,523,525,526,527,531,533,534,535,537,538,539,541, 542,543,547,549,550,551,554,555,556,557,558,559,563,564,565,566,567,568,569,570,571,572,573,574,575, 583,587,589,590,591,593,595,597,598,599,601,602,603,604,605,606,609,610,611,613,614,615,617,618,619, 620,621,622,623,625,626,627,628,629,630,631,632,633,634,636,637,638,639,647,649,651,653,654,655,657, 658,659,661,662,663,665,666,667,668,669,670,673,674,675,676,677,678,679,681,682,683,684,685,686,687, 689,690,691,692,693,694,696,697,698,699,700,702,705,706,707,708,709,710,711,712,713,714,715,716,717, 718,721,722,723,724,725,726,728,729,730,731,732,734,735,737,738,739,740,741,742,744,745,746,747,748, 749,750,752,754,756,759,760,762,764,765,773,775,777,778,779,781,782,783,785,786,787,788,790,793,794, 795,796,798,801,802,803,804,805,806,808,809,810,812,813,815,817,818,819,820,822,824,825,826,828,830, 833,834,835,836,837,838,839,840,841,842,843,844,845,846,848,849,850,852,856,858,860,865,866,867,868, 869,870,872,873,874,876,879,880,882,884,888,891,894,897,898,899,900,901,902,903,904,905,906,908,910, 912,914,916,918,920,924,928,932,936,944,952,961,962,963,964,965,966,968,969,970,972,975,976,979,980, 984,990,992,996,1000,1008,1020}	3984
13	{617,635,671,695,701,703,719,727,733,743,751,753,755,757,758,761,763,766,767,789,791,797,799,807,811 ,814,821,823,827,829,831,847,851,853,854,855,857,859,861,862,863,871,875,877,878,881,883,885,886,887, 890,892,893,895,907,909,911,913,915,917,919,921,922,923,925,926,927,929,930,931,933,934,935,937,938, 939,940,941,942,943,945,946,947,948,949,950,951,953,954,955,956,957,958,959,967,971,973,974,977,979, 981,982,983,985,986,987,988,989,991,993,994,995,997,998,999,1001,1002,1003,1004,1005,1006,1007,100 8,1009,1010,1011,1012,1013,1014,1015,1016,1017,1018,1019,1021,1022,1023}	1781
Overall Total		11115

Table 6. Addition Chain for Some Hard Exponents as in [5, 8, 18] and Generated by the Proposed Method

Number n	Addition Chain Proposed in [5, 8, 18]	L(n)	Addition Chain by the Proposed Method	L(n)
39	1 - 2 - 3 - 4 - 7 - 11 - 18 - 25 - 32 - 39	9	1 - 2 - 3 - 5 - 8 - 13 - 26 - 39	7
95	1 - 2 - 4 - 5 - 7 - 14 - 21 - 42 - 84 - 91 - 95	10	1 - 2 - 3 - 4 - 7 - 11 - 22 - 44 - 51 - 95	9
197	1 - 2 - 3 - 4 - 5 - 6 - 12 - 24 - 48 - 49 - 98 - 196 - 197	12	1 - 2 - 3 - 5 - 6 - 12 - 24 - 48 - 96 - 101 - 197	10
221	1 - 2 - 3 - 4 - 5 - 6 - 7 - 12 - 24 - 48 - 55 - 110 - 220 - 221	13	1 - 2 - 3 - 5 - 8 - 13 - 26 - 52 - 104 - 117 - 221	10
255	1 - 2 - 4 - 8 - 16 - 17 - 34 - 68 - 85 - 170 - 255	10	1 - 2 - 3 - 5 - 10 - 15 - 30 - 60 - 120 - 135 - 255	10
343	1 - 2 - 4 - 6 - 8 - 15 - 17 - 30 - 47 - 94 - 109 - 117 - 234 - 343	13	1 - 2 - 3 - 4 - 7 - 14 - 21 - 42 - 84 - 168 - 175 - 343	11
349	1 - 2 - 4 - 8 - 9 - 17 - 34 - 68 - 136 - 272 - 340 - 349	11	1 - 2 - 3 - 5 - 8 - 13 - 21 - 42 - 84 - 168 - 181 - 349	11
445	1 - 2 - 3 - 6 - 7 - 12 - 24 - 42 - 55 - 110 - 111 - 222 - 444 - 445	13	1 - 2 - 3 - 4 - 7 - 11 - 22 - 44 - 45 - 89 - 178 - 267 - 445	12
22531	1 - 2 - 4 - 8 - 16 - 32 - 64 - 128 - 256 - 512 - 1024 - 1025 - 2048 - 2050 - 4096 - 4097 - 8192 - 16384 - 20481 - 22531	19	1 - 2 - 3 - 5 - 8 - 11 - 22 - 44 - 88 - 176 - 352 - 704 - 1408 - 2816 - 5632 - 11264 - 22528 - 22531	17
30578	1 - 2 - 4 - 8 - 16 - 32 - 64 - 128 - 256 - 257 - 514 - 1028 - 2056 - 4112 - 4368 - 4369 - 8736 - 8738 - 17472 - 21840 - 30578	20	1 - 2 - 3 - 5 - 7 - 14 - 28 - 56 - 112 - 119 - 238 - 476 - 952 - 1904 - 1911 - 3822 - 7644 - 15288 - 30576 - 30578	19
6271	1 - 2 - 3 - 6 - 12 - 24 - 48 - 96 - 192 - 384 - 768 - 1536 - 3072 - 6144 - 6240 - 6264 - 6270 - 6271	17	1 - 2 - 3 - 5 - 7 - 12 - 24 - 48 - 96 - 97 - 194 - 388 - 776 - 783 - 1566 - 3132 - 6264 - 6271	17
11231	1 - 2 - 3 - 6 - 12 - 24 - 25 - 50 - 100 - 200 - 400 - 800 - 1600 - 3200 - 6400 - 9600 - 11200 - 11225 - 11231	18	1 - 2 - 3 - 5 - 7 - 12 - 24 - 27 - 39 - 42 - 43 - 86 - 172 - 344 - 688 - 1376 - 1403 - 2806 - 5612 - 11224 - 11231	18
18287	1 - 2 - 3 - 6 - 9 - 15 - 30 - 45 - 47 - 94 - 188 - 190 - 380 - 760 - 1520 - 3040 - 6080 - 12160 - 18240 - 18287	19	1 - 2 - 3 - 6 - 7 - 13 - 26 - 52 - 65 - 71 - 142 - 284 - 568 - 1136 - 2272 - 2285 - 4570 - 9140 - 18280 - 18287	19
34303	1 - 2 - 3 - 6 - 12 - 14 - 28 - 56 - 112 - 224 - 448 - 504 - 1008 - 2016 - 4032 - 8064 - 16128 - 32256 - 34272 - 34300 - 34303	20	1 - 2 - 3 - 4 - 7 - 14 - 28 - 31 - 62 - 66 - 132 - 133 - 266 - 532 - 1064 - 2128 - 4256 - 4287 - 8574 - 17148 - 34296 - 34303	20
110591	1 - 1 - 2 - 4 - 5 - 10 - 20 - 40 - 80 - 160 - 320 - 640 - 1280 - 2560 - 2570 - 5140 - 7710 - 12850 - 25700 - 51400 - 102800 - 110510 - 110590 - 110591	22	1 - 2 - 3 - 5 - 7 - 12 - 24 - 48 - 53 - 106 - 212 - 424 - 431 - 862 - 1724 - 3448 - 3455 - 6910 - 13820 - 27640 - 27647 - 55294 - 110588 - 110591	22
4169527	1 - 2 - 3 - 6 - 12 - 24 - 48 - 96 - 192 - 384 - 768 - 1536 - 2304 - 4608 - 9216 - 18432 - 36864 - 73728 - 147456 - 294912 - 589824 - 589825 - 1179650 - 1769475 - 3538950 - 4128775 - 4165639 - 414167943 - 4169479 - 4169527	29	1 - 2 - 3 - 5 - 10 - 13 - 26 - 31 - 62 - 124 - 127 - 254 - 508 - 1016 - 2032 - 4064 - 8128 - 16256 - 16287 - 32574 - 65148 - 130296 - 260592 - 521184 - 1042368 - 1042381 - 2084762 - 4169524 - 4169527	28
2211837	1 - 2 - 3 - 6 - 9 - 15 - 30 - 60 - 120 - 126 - 252 - 504 - 1008 - 2016 - 4032 - 8062 - 16128 - 16143 - 32286 - 64572 - 129144 - 258288 - 516576 - 1033152 - 2066304 - 2195448 - 2211591 - 2211717 - 2211837	28	1 - 2 - 4 - 8 - 9 - 18 - 36 - 54 - 63 - 67 - 134 - 268 - 536 - 1072 - 2144 - 4288 - 8576 - 8639 - 17278 - 34556 - 69112 - 138224 - 276448 - 552896 - 552959 - 1105918 - 2211836 - 2211837	27
75064310	1 - 2 - 3 - 4 - 5 - 7 - 8 - 14 - 15 - 16 - 30 - 31 - 32 - 64 - 128 - 143 - 286 - 572 - 1144 - 2288 - 4576 - 4581 - 9162 - 9163 - 18326 - 36652 - 73304 - 146608 - 293216 - 586432 - 1172864 - 2345728 - 2345759 - 4691518 - 9383036 - 18766072 - 18766077 - 37532154 - 37532155 - 75064310	39	1 - 2 - 3 - 5 - 10 - 11 - 21 - 42 - 63 - 126 - 131 - 142 - 143 - 286 - 572 - 1144 - 2288 - 4576 - 9152 - 9163 - 18326 - 36652 - 73304 - 146608 - 293216 - 586432 - 1172864 - 2345728 - 4691456 - 4691519 - 9383038 - 18766076 - 37532152 - 37532155 - 75064310	34

Table 7. Optimal Number of Addition Chains Generated by Various Methods Existing [13] in the Literature and the Proposed Graph Based Methods

E	Opt.	AIS	GA	PSO	EP	GBAPAC & GBMAC
[1,512]	4924	4924 (+)	4924	-	4924	4924
[1,1000]	10808	10813 (+)	10809 (+)	-	10808	10808
[1,1024]	11115	11120 (+)	-	11120 (+)	11115	11115

AIS- Artificial Immune System, GA- Genetic Algorithm, PSO- Particle Swarm Optimization, EP- Evolutionary Programming, GBAPAC- Graph Based All Possible Addition Chain, GBMAC- Graph Based Minimum Addition Chain

The addition chain generated by the proposed methods is useful in proving the conjectures as illustrated in section 3.2.

- For any integer $n > 2^k$ with $k \in \mathbb{N}$, if $l(n) = k+1$, then $n = 2^k + 2^j$ for some $j \leq k$. $l(2n) = l(n)+1$.
let $n = 68$, $l(n)$ is 7. Let $k=6$. This is because $2^k = 2^6$; $2^6 = 64$; $n > 2^k$; $68 > 64$; $l(n) = k+1 = 7$. Then 68 can be split as, $68 = 2^6 + 2^2$ where $j=2$; $2 \leq 7$.
- $l(2n) \geq l(n)$
Let $n=100$. Then $l(n)=8$; $l(2n) = l(200)=9$. Thus $l(2n) > l(n)$
- If p is a prime, show that, $n=2^p-1$ is also prime (Mersenne prime.), then $l(n) = \max \{l(m); m \leq n\}$, $2 \leq p \leq 7$,
Let $p=5$; $n=2^5-1 = 32-1 = 31$; $n=31$; $l(n) = \max \{l(m); m \leq n\}$;
- $l(31) = \max \{l(3), l(4), l(5), \dots, l(31)\}$
 $= \max \{2, 2, 3, 3, 4, 3, 4, 4, 5, 4, 5, 5, 5, 4, 5, 5, 6, 5, 6, 6, 6, 5, 6, 6, 6, 6, 7, 6\} = l(31) = 7$
- $l(2^n - 1) \leq n - 1 + l(n)$.
Let $n=9$, $l(2^9 - 1) = l(2^9 - 1) = l(511) = 12 = n - 1 + l(n) = 9 - 1 + 4$. Thus, $l(2^n - 1) \leq n - 1 + l(n)$
- $l(n) \leq \log_2(n) + S_2(n)$, where $S_2(n)$ is the Hamming weight of n

6. Conclusion

Graph based generation of optimal addition chain methods GBAPAC and GBMAC have been thought of and implemented successfully using two different methods. The first method generates all possible optimal addition chains for the given integer n whereas, the second method generates some restricted optimal addition chain. It is observed from the tables that the optimal addition chain produced by both methods have equal length which are exactly equal to the length of addition chain for an integer n available in literature. It is evident from the table that, the total length of addition chain up to 1024 is 11115 till now. Further, it is observed that the lengths of addition chain for some hard exponents have been reduced. The proposed graph based addition chain methods prove the conjectures like Scholz-Brauer. The idea used in the proposed method is unique, innovative and non-existing in the literature. The generated addition chain using the proposed methods may be incorporated in modular exponentiation which plays a vital role in many public key cryptographic algorithms like RSA, ElGamal, etc., so that the encryption and decryption time of the said algorithms may substantially be reduced. Further, in performing scalar point multiplication, the said methods may be used in reducing repeated addition operation which will eventually reduce the encryption and decryption time too.

References

- [1] D. Knuth, Art of Computer Programming—Semi Numerical Algorithms, Vol. 2, Addison-Wesley, Third Edition, 1998.

- [2] Edward G. Thurber, "Efficient Generation of Minimal Length Addition Chains", society for industrial and applied mathematics, SIAM J. Comput., Vol. 28, no. 4, pp. 1247-1263, March 1999.
- [3] Noboru Kunihiro et al., "New Methods for Generating Short Addition Chains", IEICE TRANS. Fundamentals, Vol. E83- A, No.1, January 2000.
- [4] Peter Tummelshammer and James C. Hoe and Markus Puschel, "Multiple Constant Multiplication by Time Multiplexed Mapping of Addition Chains", DAC'04, June 2004.
- [5] Nareli Cruz-Cortés, et al., "Finding Optimal Addition Chains Using a Genetic Algorithm Approach", Springer-Verlag, 2005, pp. 208-215.
- [6] Daniel J. Bernstein, "Differential addition chains", available at: <https://cr.yp.to/ecdh/diffchain-20060219.pdf><https://cr.yp.to/ecdh/diffchain>, February 2006.
- [7] Younho Lee et al., "Expansion of Sliding Window Method for Finding Shorter Addition/Subtraction-Chains", International Journal of Network Security, Vol.2, No.1, PP.34–40, Jan. 2006.
- [8] Nareli Cruz- Cortes et al., "An Artificial Immune System Heuristic for Generating Short Addition Chains", IEEE Transactions on Evolutionary Computation, 2007.
- [9] Raveen R. Goundar et al., "New Strategy for Doubling-Free Short Addition-Subtraction Chain", Applied Mathematics & Information Sciences, 2(2) (2008), 123–133.
- [10] Raveen R. Goundar et al., "SPA Resistant Scalar Multiplication using Golden Ratio Addition Chain Method", IAENG International Journal of Applied Mathematics, 38:2, IJAM, June 2008.
- [11] Fabien Herbaut et al., "Random Euclidean Addition Chain Generation and Its Application to Point Multiplication", INDOCRYPT 2010, Springer-Verlag Berlin Heidelberg 2010.
- [12] Maurice MIGNOTTE and Amadou TALL, "A Note on Addition Chains", International Journal of Algebra, Vol. 5, 2011, no. 6, 269 – 274.
- [13] Dominguez- Isidro and E. Mezura-Montes, "An Evolutionary Programming Algorithm to Find Minimal Addition Chains", I Congreso Internacional de Ingenieria Electronica, Instrumentacion Y Computacion, 22 al 24 de Junio del, July 2011.
- [14] Neill Michael Clift, "Calculating optimal addition chains", available at: Springerlink.com, September 2011.
- [15] Amadou TALL, "A generalization of the Lucas addition chains", available at: <https://eprint.iacr.org/2011/378.pdf>, 2011.
- [16] M. A. Mohamed and K. A. Mohd Atan, "Rule Based Representation of Integer for a New Addition Chain Method, Applied Mathematical Sciences, Vol. 6, 2012, no. 30, 1497 – 1503.
- [17] Amadou Tall and Ali Yassin Sanghare, "Efficient computation of addition-subtraction chains using generalized continued Fractions, International Journal of Applied Mathematical Research, IJAMR, 2013.
- [18] Mr. K. Mani, "Generation of Addition Chain using Deterministic Division Based Method", International Journal of Computer Science & Engineering Technology (IJCSET), ISSN: 2229-3345 Vol. 4 No. 05 May 2013.
- [19] Yara Elias and Pierre McKenzie, "On Generalized Addition Chains", Integers 14, available at: <https://www.emis.de/journals/Integers/papers/o16/o16.pdf>, March 2014.
- [20] MA Mohamed and MR MD SAID, "A Hybrid Addition Chain Method for Faster Scalar Multiplication", WSEAS Transactions on Communications, E-ISSN: 2224-2864, Volume 14, 2015.
- [21] Sajal Chakroborty, M. Babul Hasan, "A Proposed Technique for Solving Scenario Based Multi-Period Stochastic Optimization Problems with Computer Application", International Journal of Mathematical Sciences and Computing (IJMSC), Vol.2, No.4, pp.12-23, 2016.

Authors' Profiles

Mani. K received his MCA and M.Tech. from the Bharathidasan University, Trichy, India in Computer Applications and Advanced Information Technology respectively. Since 1989, he has been with the Department of Computer Science at the Nehru Memorial College, affiliated to Bharathidasan University where he is currently working as an Associate Professor. He completed his PhD in Cryptography with primary emphasis on evolution of framework for enhancing the security and optimizing the run time in cryptographic algorithms. He published and presented around 25 research papers at international journals and conferences.



Viswambari. M received her MSc and M.Phil from Bharathidasan University, Trichy, India. Currently, she is pursuing her Ph.D in Cryptography, Bharathidasan University, Trichy. Her research interest is on Cryptography, Network Security.

How to cite this paper: K. Mani, M. Viswambari, "A New Method of Generating Optimal Addition Chain Based on Graph", International Journal of Mathematical Sciences and Computing(IJMSC), Vol.3, No.2, pp. 37-54, 2017.DOI: 10.5815/ijmsc.2017.02.04