CONSTRUCTION OF BOOLEAN FUNCTIONS AND SOLUTIONS OF A SYSTEM OF LINEAR INEQUALITIES BY NSGA-II

Ph.D. THESIS

by

RAJNI GOYAL

DEPARTMENT OF MATHEMATICS
INDIAN INSTITUTE OF TECHNOLOGY ROORKEE
ROORKEE - 247 667 (INDIA)

JULY, 2013

CONSTRUCTION OF BOOLEAN FUNCTIONS AND SOLUTIONS OF A SYSTEM OF LINEAR INEQUALITIES BY NSGA-II

A THESIS

Submitted in partial fulfilment of the

requirements for the award of the degree

of

DOCTOR OF PHILOSOPHY

in

MATHEMATICS

by

RAJNI GOYAL

DEPARTMENT OF MATHEMATICS
INDIAN INSTITUTE OF TECHNOLOGY ROORKEE
ROORKEE - 247 667 (INDIA)

JULY, 2013

INDIAN INSTITUTE OF TECHNOLOGY ROORKEE
ROORKEE

CANDIDATE'S DECLARATION

I hereby certify that the work which is being presented in the thesis entitled "**CONSTRUCTION OF BOOLEAN FUNCTIONS AND SOLUTIONS OF A SYSTEM OF LINEAR INEQUALITIES BY NSGA-II"** in partial fulfilment of the requirements for the award of the Degree of Doctor of Philosophy and submitted in the Department of Mathematics of the Indian Institute of Technology Roorkee, Roorkee is an authentic record of my own work carried out during the period from August, 2008 to July, 2013 under the supervision of Dr. Shiv Prasad Yadav, Professor, Department of Mathematics, Indian Institute of Technology Roorkee, Roorkee.

The matter presented in this thesis has not been submitted by me for the award of any other degree of this or any other Institute.

(RAJNI GOYAL)

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

(Shiv Prasad Yadav)
Supervisor

Date: July 15, 2013

The Ph.D. Viva-Voce examination of **Ms**. **RAJNI GOYAL,** Research Scholar, has been held on…….

Signature of Supervisor Chairman, SRC External Examiner

Head of the Department/Chairman, ODC

Abstract

The research work presented in the thesis is study of methods developed to construct desired Boolean functions and solutions of system of linear inequalities.
The thesis consists of seven chapters. The chapter-wise summary of the thesis is as follows:

Chapter 1 is introductory in nature. In this chapter, we have defined the relevant supporting theory of Boolean functions. In particular, we have provided numerous definitions and theorems for various aspects of the theory. The necessary cryptographic properties which are used to analyze the strength of Boolean functions have been also defined and discussed, and inter-relations between pairs of selected properties are also discussed. Finally, we have presented a brief summary of major cryptanalytic attacks against Boolean functions and cipher systems.

In **Chapter 2**, we have developed a new evolutionary method to optimize the Boolean functions' properties by two objective optimization method. In this Chapter , we have taken balancedness, nonlinearity and resiliency, and developed an evolutionary method to construct Boolean functions having these properties at optimal level. We have got the desired functions and compared our results with previous results. Our results are as good as previous results.

In **Chapter 3** also, we have developed a new evolutionary method to optimize the Boolean functions' properties by two objective optimization method but here we have taken balancedness, nonlinearity and autocorrelation, and developed an evolutionary method to construct desired Boolean functions. We have got the desired functions and compared our results with previous results. Our results are at least as better as previous results.

In **Chapter 4**, we have developed a new method to optimize the Boolean functions' properties by three objective optimization method. In this chapter, we have taken balancedness, nonlinearity, resiliency and autocorrelation simultaneously to optimize these properties. We have got the desired functions and compared our results with previous results and found that our results are at least as better as available in the literature.

In **Chapter 5**, we have introduced the concept of biasedness in the proposed method and developed a new method based on biasedness to construct Boolean functions and got the desired results. In this chapter, we got the Boolean functions of 7 and 8 variables that could not be possible by the methods developed in Chapters 2, 3 and 4. We also compared our results with previous results and found that our results are at least as better as available in the literature.

In **Chapter 6**, we have developed a new method based on NSGA-II to solve a system of linear inequalities. This method is applicable for all types of inequalities. We have generated three examples of different types and solved them by the developed method. The developed method gives better spread of solutions. Consequently, our method is better than previous methods to solve the system of linear inequalities.

In **Chapter 7**, based on the study carried out in the thesis, conclusions are drawn and future scope of the research work is suggested.

List of Publications

Refereed Journals

1. Goyal R., *Solution of a set of linear inequalities by NSGA-II"*, OPSEARCH, 48(4), Oct-Dec 2011, 297-305, 2011.

2. Goyal R., Yadav S. P., Kishor A., *A new approach to solve a general system of linear inequalities based on NSGA-II"*, Int J Syst Assur Engg Manag, 3(1), 17-23, 2012.

3. Goyal R. and Yadav S. P., *An evolutionary approach to construct cryptographically strong Boolean functions"*, Int J Syst Assur Engg Manag, 3(1), pp.1-5, 2012.

4. Goyal R., Yadav S. P., *A multi-objective evolutionary approach to Construct desired Boolean functions"*, Manuscript Number: ASOC-D-12 00740R1, Applied Soft Computing ,Elsevier, 2012(Revised).

5. Goyal R., Yadav S. P. and Kishor A. *"A New Multiobjective Approach to Design Highly Non-linear Resilient Boolean Functions"*, Fundamenta Informaticae XXI (2001) 1001-1008 , 2011(communicated).

Conferences

1. Goyal R., Yadav S. P., and Kishor A. *Design of Boolean Functions Satisfying Multiple Criteria by NSGA-II,"* Proceedings of the International Conference on "Soft Computing for Problem Solving"- SocProc-2011 held on December 20-22, 2011 at The Institution of Engineers (India), IIT Roorkee Campus. 2011, AISC 130, pp. 461-468, 2011.

2. Goyal R. and Yadav S. P., *"An Evolutionary Multi-Objective Approach to Construct Balanced Boolean Functions Having High Nonlinearity and Low Autocorrelation"*, International Conference on Optimization Modelling and Applications (OPTIMA-2012) held at Delhi University during Nov 29 - Dec 01, 2012.

3. Goyal R. and Yadav S. P. (2012), *"An evolutionary multiobjective approach with biasedness to construct desired Boolean functions"*, International Symposium on Applied Optimization and Game- Theoretic Models (ISAOGTM-2013) held at Indian Statistical Institute, Delhi during Jan 09-11

Acknowledgements

I feel it's true that:

"If you strongly desire something with your heart, the whole world conspire to fulfill your desire."

This doctoral work would not have been possible without the support and encouragement of numerous people including my well wishers and my friends. It's a pleasent task to express my thanks to all those who contributed in many ways to the successfully completion of this study. During the period of my research work, I got support from many people whom I wish to acknowledge.

I, first of all, thank God for providing me the opportunity to pursue higher studies. I feel privileged to express my sincere regards and gratitude to my supervisor Dr. Shiv Prasad Yadav, Professor, Department of Mathematics, Indian Institute of Technology Roorkee, Roorkee for his valuable guidance, continuous encouragement throughout my research work. It has been a great experience and joy to work with Dr. Shiv Prasad Yadav. The critical comments rendered by him during the discussions are deeply appreciated.

I express my deep sense of gratitude to Dr. R. C. Mittal, Prof. and Head, Deptt. of Mathematics, IIT Roorkee, Dr. P.N Agarwal, Prof. and Chairman, DRC for providing the departmental facilities for carrying out my research work. Special thanks to my SRC members Dr. S. P. Sharma, Prof., Deptt. of Mathematics and Dr. R. S. Anand, Deptt. of Electrical Engg. for spending their valuable times during my research.

My sincere thanks to Dr. Amar kishor, ISI Kolkata for encouraging me and providing necessary help during my research. I am also thankful to Dr. Mohit Sharma, Karunesh Singh and Dr. Manigh Garg for moral support.

My special, sincere, heartfelt and indebtedness are due to my respected father Shri Beshashwar Goyal and my holly mother Smt. Suman lata for their sincere prayers, moral support and never ending blessings. I dedicate this thesis to them. I pay my heartily gratitude to my younger brother Girish Goyal and my elder sisters Deepa, Seema and

Anuradha. My special thanks to my guide's wife Smt. Gayatri Devi who loves me a lot and cares like my mother.

I express my heartfelt thanks and deepest regards to my friends Dr. Monika Goyal, Manu, Babita, Darshana and Nivedita whose support encouraged me always. I am thankful to my lab colleagues Sujeet Kumar Singh, Jolly Puri and Sandeep Mogha for the help they provided whenever required. My heartfelt thanks are to my friends Alka, Surendra, Neha, Smaita.

I am thankful to the Ministry of Human and Research Development, New Delhi, India for financial support to carry out this research work.

Roorkee (Rajni Goyal)

July 15th, 2013

Contents

List of Figures

List of Tables

Chapter 1

1.1 Introduction

In this world we are living in the age of information. The existing information is very important and is used in many forms like transaction(financial), documents(legal), plans and strategies(military), political etc. The protection of information is very important as its compromise may result in financial losses, exposure of commercial secretes or defence secretes and many more.

Cryptography is one of the most important tools used for information security. Security in cryptography is provided in many forms. Confidentiality, integrity and authentication are most important forms among them. Ensuring that information is kept private from unauthorized disclosure is known as confidentiality. By making sure that the information has not been modified since creation or storage is known as integrity. Authentication is the procedure of checking that the information are coming from the correct source.

The data storage system, authentication, key management system, cipher system, policies etc. are main components of security system. The overall strength of a security system depends on its individual components. In the same way the strength of a cipher system depends on its individual components.

Boolean functions are the most common and critical components of a cipher system. Boolean functions and S-Boxes(multidimensional Boolean functions) are highly suitable for receiving bits of linear feedback shift register as input in order to combine

them to produce the single keystream. So, they are often utilized in the keystream generation process of stream cipher.

There are different types of attacks like linear cryptanalysis [71], differential cryptanalysis [7], correlation attack [87] etc. To resist the cipher system from the attacks Boolean functions should have good combination of cryptographic properties such as balancedness, nonlinearity, resiliency, autocorrelation etc..

To understand the research work presented in this thesis, knowledge of Boolean functions and different attacks is required. So, a necessary background of theory of Boolean functions is provided in this chapter.

1.2 Theory of Boolean functions

The theory of Boolean functions is a wide area in itself. The comprehensive review of Boolean functions' theory is not required here. So, in this chapter we have given a brief review of the theory of Boolean functions.

1.2.1 Boolean function and its Properties:

We now discuss some important properties of Boolean functions. We also include specific Boolean function measures and different types of representations of Boolean functions. Some important Boolean functions' properties and how they contribute to provide security to the function are also discussed in this section.

Boolean Function:

Let $\mathbb{F}_2^n$ be the prime field of characteristic 2 and $\mathbb{F}_2 = \{0,1\}$. Then $\mathbb{F}_2^n$ is an n-dimensional vector space over $\mathbb{F}_2$. An element of $\mathbb{F}_2^n$ can be represented by a binary vector of length n.

Definition 1.1. *Any function* $f : \mathbb{F}_2^n \to \mathbb{F}_2$ *is called a Boolean function of n-variables. The set of all Boolean functions of n-variables is denoted by* $\mathcal{B}_n$*. The Hamming weight of a binary vector* $x = (x_1, \ldots, x_n) \in \mathbb{F}_2^n$ *is denoted by* $w_H(x)$ *and is defined to be the size of its support* $\{i \in N : x_i \neq 0\}$*, where* N=$\{1, 2, \ldots, n\}$*. The Hamming weight* $w_H(f)$ *of a Boolean function f on* $\mathbb{F}_2^n$ *is defined to be the size of its support* $\{x \in \mathbb{F}_2^n : f(x) \neq 0\}$*.*

Representation of Boolean functions:

(i)Truth Table Form(TTF): A Boolean function $\mathbb{F}_2^3 \to \mathbb{F}_2$ can be represented in TTF as given in Table 1.1.

$\mathbb{F}_2^3$	$\mathbb{F}_2$
x_1, x_2, x_3	f(x)
0, 0, 0	0
0, 0, 1	1
0, 1, 0	0
0, 1, 1	0
1, 0, 0	0
1, 0, 1	1
1, 1, 0	0
1, 1, 1	1

Table 1.1: Truth table representation of a Boolean function of 3-variables

Algebraic Normal Form(ANF): It is the n-variable polynomial representation of the Boolean function f(x) given by

$$\begin{aligned} f(x) &= \bigoplus_{I \in \mathcal{P}(\mathcal{N})} a_I \left(\prod_{i \in I} x_i\right) \\ &= \bigoplus_{I \in \mathcal{P}(\mathcal{N})} a_I x^I, \end{aligned}$$

where $\mathcal{P}(\mathcal{N})$ denotes the power set of $\mathcal{N}$=$\{1, 2, \ldots, n\}$. Since every bit in $\mathbb{F}_2$ equals its own square, therefore every coordinate x_i in the polynomial f(x) appears with exponent at most 1. This representation belongs to $\mathbb{F}_2[x_1, x_2, x_3, ..., x_n]/[x_1^2 \oplus x_1, x_2^2 \oplus x_2, ..., x_n^2 \oplus x_n]$. The ANF of every Boolean function always exists and is unique.

Example: f(x)=$x_1x_2x_3$+x_3x_1+x_3.

Conversion TTF to ANF and vice-versa: Let f(x) be a Boolean function whose TTF is given in Table 1.1.

The function f is the sum of the atomic functions f_1, f_2, f_3 whose TTF is given Table 1.2.

x_1, x_2, x_3	$f_1(x)$	$f_2(x)$	$f_3(x)$
0, 0, 0	0	0	0
0, 0, 1	1	0	0
0, 1, 0	0	0	0
0, 1, 1	0	0	0
1, 0, 0	0	0	0
1, 0, 1	0	1	0
1, 1, 0	0	0	0
1, 1, 1	0	0	1

Table 1.2: Truth table representation of Boolean function for 3-variables

The function $f_1(x)$ takes value 1 if and only if $1 \oplus x_1 = 1$, $1 \oplus x_2 = 1$ and $x_3 = 1$, i.e., if and only if $(1 \oplus x_1 = 1)(1 \oplus x_1 = 1)x_3$=1. Thus the ANF of f_1 can be obtained by expanding the product $(1 \oplus x_1 = 1)(1 \oplus x_1 = 1)x_3$. Similarly, the ANFs of f2 and f3 can be written. Hence, the ANF of f equals $(1 \oplus x_1 = 1)(1 \oplus x_1 = 1)x_3 \oplus x_3 \oplus x_1x_2x_3 = x_1x_2x_3 \oplus x_1x_3 \oplus x_3$.

Conversely, if we have Boolean function in ANF form, we can convert it in TTF as

follows:

Let a Boolean function f be given in the ANF as f(x)= $x_1x_2x_3 \oplus x_1x_3 \oplus x_3$. To convert this form to TTF, we put 1's in table according to the coordinates present in polynomial. In the first term of polynomial, we have all three coordinates. So, we put 1 for f(x) where all coordinates have value 1 in TTF. In the second term, we have x_1x_3. So, we put 1 for f(x) where these two coordinates have 1 in TTF. In the third term, we have x_3. So, we put 1 for f(x) where only x_3 is 1 in TTF. Remaining places in TTF should be occupied by 0's. So, the required TTF for f will be Table 1.1.

The degree of the ANF: The degree of ANF is denoted by *df* and is defined by $df = \max |I| : a_I \neq 0$, where $|I|$ denotes the size of I. It is also called the algebraic degree or nonlinear order of Boolean function.

Definition 1.2. *The Hamming distance between two functions f and g, denoted by hd(f,g), is defined as the number of truth table positions in which the functions f and g disagree, i.e.,*

$$hd(f, g) = |\{x : f(x) \neq g(x)\}| \tag{1.2.1}$$

where $|.|$ stands for the cardinality of the set.

Balancedness:

The Balancedness is very important property of Boolean functions. This is the basic property of Boolean functions. In most of cases we consider only balanced Boolean functions.

Definition 1.3. *A function is said to be balanced if the number of 0's is the same as the number of 1's in the output table. This property is called balancedness.*

Hamming weight for an n-variable balanced function is 2^{n-1}. Hence, an n-variable Boolean function not having weight 2^{n-1} can not be balanced. (We have used this criteria to get balanced Boolean function in our proposed method).
Balancedness is very important property to resist different types of attacks specially linear approximation(Section 1.3.2.). More deviation of a function from the balancedness or higher magnitude of function imbalancedness(deviation from 2^{n-1}), the more likelihood of a high probability linear approximation being obtained.

Functions Similarity:
By this property we compare two Boolean functions.

Definition 1.4. *The similarity between two functions f and g, denoted by s(f,g), is defined as the number of truth table positions in which the functions f and g agree, i.e.,*

$$s(f,g) = |\{x : f(x) = g(x)\}| \tag{1.2.2}$$

The relation between s(f,g) and hd(f,g) is given by

$$s(f,g) = 2^n - hd(f,g). \tag{1.2.3}$$

Linear Boolean Function: Linear Boolean function can be defined as:

Definition 1.5. *A linear Boolean function is denoted by $L_\lambda(x)$ and is defined by $L_\lambda(x) = \lambda_1.x_1 \oplus \lambda_2.x_2 \oplus \lambda_3.x_3 \oplus \ldots \oplus \lambda_n.x_n$, where λ, $x \in \mathbb{F}_2^n$, $\lambda_i.x_i$ denotes the bitwise AND of the ith bits of λ and x, and $\oplus$ denotes bitwise XOR. An affine Boolean function is denoted by $A_{\lambda,c}(x)$ and is defined by*

$$A_{\lambda,c}(x) = L_\lambda(x) \oplus c, \tag{1.2.4}$$

where c$\in \mathbb{F}_2$.

Affine Transformation:

Affine transformation of a Boolean function can be defined as:

Definition 1.6. *An affine transformation on the input of n-variable Boolean function f(x) can be defined as the resultant function g(x) given by as g(x) = f(Ax $\bigoplus$ a) $\bigoplus$ b, where $x \in \mathbb{F}_2^n$, A is an $n \times n$ invertible binary matrix, and a and $b \in \mathbb{F}_2$.* If a=0, it is called linear transformation.

Linear Structure:

Linear structure of a Boolean function can be defined as:

Definition 1.7. *The property that for an affine or a linear Boolean function f, the values f(x+s) and f(x), for every fixed s, are either always equal or always different is called the linear structure.*

Walsh Hadamard Transform:

The Walsh Hadamard Transform(WHT) provides another means of representing a Boolean function. It gives the measure of the correlation between a Boolean function and the set of all linear Boolean functions.

Definition 1.8. *For a given linear function L_λ specified by $\lambda \in \mathbb{F}_2^n$, the WHT of a func-*

tion f is denoted by $W_f(\lambda)$ and is defined by

$$W_f(\lambda) = \sum_{x \in \mathbb{F}_2^n} (-1)^{f(x)+\lambda.x}. \tag{1.2.5}$$

The vector representing the WHT of a function is referred to as its Walsh Hadamard Spectrum(WHS).

The Walsh Hadamard values in the spectrum of a Boolean function are constrained by Parseval's relationship. This is also known as Parseval's equation.

Theorem 1.1. Parseval's Equation [67]

For an n-variable Boolean function Parseval's relation is given by

$$\sum_{\lambda \in \mathbb{F}_2^n} [W_f(\lambda)]^2 = 2^{2n}. \tag{1.2.6}$$

Proof:

$$\sum_{\lambda \in \mathbb{F}_2^n} [W_f(\lambda)]^2 = \sum_{\lambda \in \mathbb{F}_2^n} (\sum_{x \in \mathbb{F}_2^n} (-1)^{f(x)+\lambda.x} \sum_{x \in \mathbb{F}_2^n} (-1)^{f(x)+\lambda.x}).$$

$$= \sum_{x \in \mathbb{F}_2^n} (-1)^{f(x)} (-1)^{f(x)} \sum_{\lambda \in \mathbb{F}_2^n} (-1)^{\lambda.x} (-1)^{\lambda.x}.$$

$$= 2^n 2^n$$

$$= 2^{n+n}$$

$$= 2^{2n}.$$

This value is constant for all n-variable Boolean functions. All Boolean functions

must satisfy Parseval's equation. But if a function satisfy this relationship, it may not be a Boolean function. (We will use this relationship to get high nonlinear Boolean function in our proposed method.)

Non-Linearity:

Nonlinearity is the one of the most important property of Boolean functions. Nonlinearity of an Boolean function f represents a measure of the dissimilarity between function f and n-variable affine function. All affine functions are considered cryptographically weak functions. It means they are low resistant to cryptanalysis attacks. So, more dissimilar f to affine functions will possess high nonlinearity and hence more resistant to cryptanalysis attacks. If a Boolean function has low nonlinearity, function will be low resistant to a particular type of attack(linear cryptanalysis Section 1.3.2.). Thus, to provide resistance to the linear cryptanalysis Boolean functions should have high nonlinearity.

Definition 1.9. *The nonlinearity Nl(f) of a Boolean function f is its minimum Hamming distance to all members of the set of affine functions. It is given by*

$$Nl(f) = (2^n - \max_{\lambda \in \mathbb{F}_2^n} |W_f(\lambda)|)/2. \tag{1.2.7}$$

Nonlinearity of a Boolean function is invariant under affine transformations. If f(x) is a Boolean function and A is an $n \times n$ invertible binary matrix, a and b $\in \mathbb{F}_2^n$ and c $\in \mathbb{F}_2$, then nonlinearity of $f(Ax + a) \oplus$ b.x $\oplus$ c will be equal to that of f(x).

Autocorrelation:

The autocorrelation of a function gives an indication of the imbalanceness of all first

order derivatives of a Boolean function and provides a measure of self similarity for Boolean function. The derivative of Boolean function f(x), taken with respect to a vector s, where x and $s \in \mathbb{F}_2^n$, is defined as $f(x) \bigoplus f(x+s)$. Similarly the derivative in polar form can be defined as $\widehat{f}(x)\widehat{f}(x+s)$.

Definition 1.10. *The autocorrelation of a function f is denoted by* $r_f(s)$ *and is defined by*

$$r_f(s) = \sum_{x \in \mathbb{F}_2^n} \hat{f}(x)\hat{f}(x+s) \tag{1.2.8}$$

where $\hat{f}(x) = (-1)^{f(x)}$.
$r_f(0)$ *has the maximum value and equals to* 2^n.

The autocorrelation function measures the directional derivative of a Boolean function for an input shift in the direction s over all $x \in \mathbb{F}_2^n$. Autocorrelation function sums the polarity form of the derivative of Boolean function. Autocorrelation of a Boolean function of n-variables is a real valued vector containing the 2^n values of $r_f(s)$. The range of these values is $[-2^n, 2^n]$. The autocorrelation measure of a Boolean function are invariant under affine transformation. A Boolean function f is considered to be good if r_f is small . To resist the differential cryptanalysis attack to the function, function should have optimal autocorrelation value.

The avalanche characteristic gives the measurement of the quality that how well input propagates throughout a process and affects the output uniformly. The autocorrelation measures the avalanche characteristics and determines the two indicators:
(i). absolute indicator and
(ii). sum-of-square indicator.

Cololllary 1.1 Let $|AC|_{max}$ denote the absolute indicator derived from the autocorrelation function $r_f(\lambda)$. Then

$$|AC|_{max} = max_{\lambda}|r_f(\lambda)| \tag{1.2.9}$$

with λ=$\{1, 2, \ldots, 2^n - 1\}$.

Cololllary 1.2 Let σ denote the sum-of-square indicator derived from the autocorrelation function $r_f(\lambda)$. Then

$$\sigma = \Sigma_{\lambda}[r_f(\lambda)]^2. \tag{1.2.10}$$

with λ=$\{1, 2, \ldots, 2^n - 1\}$.
The above two autocorrelation measures of Boolean function are invariant under affine transformations.

Avalanche Criterion:
Avalanche criterion of a Boolean function can be defined as:

Definition 1.11 *An n-variable Boolean function f(x) is said to satisfy the propagation criterion of degree k, denoted by PC(k), with respect to a non-zero vector* $\alpha \in \mathbb{F}_2^n$ *if*

$$\sum_x f(x) \bigoplus f(x + \alpha) = 2^{n-1}$$

$\forall$ α such that $1 \leq w_H(x) \leq k$.

Correlation Immunity:

The extent of independence between linear combination of input and output bits is termed as correlation immunity.

Definition 1.12. *A function $f \in \mathcal{B}_n$ is called the correlation immune of order m if $W_f(\alpha) = 0$ $\forall \alpha \in \mathbb{F}_2^n$ such that $1 \leq w_H(\alpha) \leq m$.*

The correlation immunity property is not invariant under affine(or linear) transformation. For a cryptographically strong Boolean function order of correlation immunity should be high. As we mentioned above correlation immunity gives an extent of independence between linear combination of input and output bits of a function. So, correlation attacks(Section 1.3.3.) exploit the dependence that may exist within combining functions of stream cipher(involve to produce keystream). Hence, higher is the order of correlation immunity, lesser is the chance of correlation attack.

Resiliency:

Resiliency of a boolean function can be defined as:

Definition 1.13. *A correlation immune function f of order m is called the m-resilient if f is balanced. The resiliency of m-resilient function f is denoted by Rs(f) and is defined to be m.*

1.2.2 Some special Boolean functions

There are some special Boolean functions. We will describe some of them with their properties.

Bent functions:

These functions were first developed by Rothaus [80]. These functions have maximum distance to the set of affine functions(maximum nonlinearity). That is why these functions are also called perfect nonlinear Boolean functions. These functions are not balanced and exist only for even n. These functions do not exhibit any order of correlation immunity, having the value of WHT always $2^{n/2}$ or $-2^{n/2}$. Beside having high nonlinearity, bent functions have one of another important criteria that they have minimum autocorrelation.

Definition 1.14. *A Boolean function is called bent if* $W_f(\lambda) = c\,2^{n/2}$*, where n is even(bent function exists only for even variables) and* $c = 1$ *or* -1*. The Bent functions have maximum nonlinearity and its value is* $(2^{n-1} - 2^{(n/2)-1})$*.*

Although bent functions have optimal(maximum) nonlinearity and have the lowest value of autocorrelation, still they are not good for practical use because they are not balanced. Furthermore, they have very low algebraic degree. So, they are good only for nonlinearity and autocorrelation points of view but not good for practical use because of their some other weakness discussed as above.

Semi-bent functions:

As discussed above bent functions are not good from cryptographic point of view because of unblessedness and low algebraic degree. So, some modified functions are given by Chee et al. [14] to overcome the weakness of bent functions. These functions are called semi-bent functions. These functions have the same properties as bent functions but are balanced. These functions are constructed by concatenating a bent function to the same bent function that has had an affine transformation applied to it and its output complemented.

If f(x) is a bent function of n(even) variables, then semi-bent function g(x) of n+1

variables can be given by

$$g(x) = f(x)||f(Ax \oplus b) \oplus 1,$$

where || stands for concatenations of two functions.

Multidimensional Boolean function or S-Boxes(Substitution Boxes):
Boolean functions are multiple input and single output functions. An S-Box is an extension of the theory of single output Boolean functions to the theory of multiple output Boolean function.

Definition 1.15. *Any function $f : \mathbb{F}_2^n \to \mathbb{F}_2^m$ is called $S - Box$. If m=1, f becomes a Boolean function.*

An $n \times m$ S-Box is a mapping from n input bits to m output bits.

Here we are giving some definition related to the S-Boxes

Definition 1.16. *The nonlinearity of an $n \times m$ S-box, denoted by $\mathrm{Nl}(S_{n,m})$, is defined as the minimum nonlinearity of each of its component output Boolean functions and their linear combinations. Let $S=(f_1, f_2, f_3, ..., f_m)$, where f_i(i=1,2,3,...,m) are n-variable Boolean functions. Let $g_j = \alpha_1^j f_1 + \alpha_2^j f_2 + \alpha_3^j f_3 + ... + \alpha_m^j f_m$, j=1,2,3,...,$2^m - 1$ and $G=\{g_1, g_2, \ldots, g_{2^m-1}\}$. Then the nonlinearity of S is given by*

$$Nl(S_{n,m}) = \min_G Nl(g_j). \tag{1.2.11}$$

Clearly, as n and m increase, the task of computing the nonlinearity value of an $n \times m$ S-box rapidly becomes computationally difficult.

Definition 1.17. Let $\hat{r}_{g_j}(\alpha)$ be the autocorrelation value of g_j for $\alpha \in \{1, 2, \ldots, 2^n - 1\}$. Then the maximum absolute autocorrelation value of S, denoted by $|AC(S_{n,m})|_{max}$, is defined by

$$|AC(S_{n,m})|_{max} = \max_{G} |\hat{r}_{g_j}(\alpha)|. \tag{1.2.12}$$

1.2.3 Relationship between cryptographic properties of Boolean functions:

A Boolean function is ideal if it has all its properties at optimal level. But in reality it is not possible to get such functions because all properties are inter-related. If we want to optimize some properties, it results in loss of the optimality of some other properties. We construct Boolean functions having the best trade-off among its properties. Now we will describe, in brief, the relationship among some properties of Boolean functions required for our research work.

Relationship between nonlinearity and autocorrelation:

We know that nonlinearity is the measurement of dissimilarity between a Boolean function and the nearest affine function in the set. The following theorem due to Wiener-Khintchine [84] exhibits the relationship between nonlinearity and autocorrelation.

Theorem 1.2. If f(x) is an n-variable Boolean function with WHT $W_f(\lambda)$ and autocorrelation $r_f(\lambda)$, then.

$$\sum_{\alpha} r_f(\alpha) L_\alpha(\lambda) = [W_f(\lambda)]^2 \tag{1.2.13}$$

where $L_\alpha(\lambda)$ is a linear function of λ characterized by α.

Proof:

$$\sum_{\alpha} r_f(\alpha) L_\alpha(\lambda) = \sum_{\alpha} [\sum_{x} \hat{f}(x) \hat{f}(x \oplus \lambda L_\alpha)(\lambda)]$$

$$= \sum_{\alpha} \sum_{x} (-1)^{f(x)} (-1)^{f(x \oplus \alpha)} (-1)^{\alpha . \lambda}$$

$$= \sum_{y} \sum_{x} (-1)^{f(x)} (-1)^{f(y)} (-1)^{(x \oplus y) . \lambda}$$

(on assuming $y = x \oplus \alpha$)

$$= \sum_{y} \sum_{x} (-1)^{f(x)} (-1)^{f(y)} L_\lambda(x) L_\lambda(y)$$

$$= \sum_{y} \sum_{x} (-1)^{f(x)} L_\lambda(x) (-1)^{f(y)} L_\lambda(y)$$

$$= [W_f(\lambda)]^2.$$

■

By Corollary 1.2, we conclude that the sum-of-square indicator σ will be high if $r_f(\lambda)$ will contain large magnitude. So, by Theorem 1.2, we can conclude that if σ is high, then nonlinearity will be low and vise-versa.
The same result is observed from the following relation:

$$Nl(f) \leq 2^{n-1} - \sqrt{(2^n + AC_{max})}/2 \tag{1.2.14}$$

where $AC_{max} = \max |r_f(\lambda)|$.
It is clear from the above relation that to get high nonlinearity, autocorrelation should be low and vise-versa.

Relationship between nonlinearity and correlation immunity:
Relationship between nonlinearity and autocorrelation can be explained by Parseval'theorem(Theorem 1.1). As discussed in definition 1.11, to get an m-order correlation immune Boolean function, the value of $W_f(\lambda)$ corresponding to the positions $w_H(x)$ must be zero. So, to get high correlation immune function, the more positions in WHS must be zero. Then by Parseval's relation some values in WHS should be high to get that equality, means low nonlinearity. Conversely, higher is the nonlinearity, low is the correlation immune function.

Relationship between autocorrelation and correlation immunity:
To understand the relation between autocorrelation and correlation immunity, consider following theorem([106]).

Theorem 1.3. If f(x) is an m-resilient Boolean function of n-variables with $2 \leq m \leq n$ and AC_{max} is the maximum absolute value of autocorrelation, then

$$ACmax \geq 2^{m-1} \sum_{i=0}^{\infty} 2^{i(m-1-n)} \tag{1.2.15}$$

Theorem 1.4. If f(x) is an m-resilient Boolean function of n-variables with $1 \leq m \leq n-1$ and AC_{max} is the maximum absolute value of autocorrelation, then

$$ACmax \geq 2^m \sum_{i=0}^{\infty} 2^{i(m-n)} \tag{1.2.16}$$

It is clear that Autocorrelation will increase as we increase the value of m. But the following theorem [106] gives the contradictory relation.

Theorem 1.5. If f(x)is an m-resilient Boolean function of n-variables satisfying *PC(k)*, then

$$m + k \leq n - 2. \tag{1.2.17}$$

It is obvious from the above relation that autocorrelation will decrease as m increases.

1.3 Some cryptanalytic attacks on cipher system:

We have given number of cryptographic properties of Boolean functions relevant to our research and relation between them. Here, we describe some cryptanalysis attacks and how the above properties provide resistance to these attacks.

1.3.1 Differential Cryptanalysis:

The technique of this attack is given by Biham and Shamir [7] and applied to block cipher in a chosen plaintext attack. On a block cipher system this attack involves the analysis of trend between plaintext input differences and corresponding output differences in the ciphertext. A differential attack generally seeks to exploit these trends to get information. The information may be key bits and some thing like this. This can

be explained as below:

Let for a Block cipher with block length I bits, A = A_1 A_2 $A_3...A_I$ and B=B_1 B_2 $B_3...B_I$ represent a plaintext and ciphertext block respectively. Let A^j and A^k be two I-bit blocks of plaintext with B^j and B^k their corresponding output blocks. Their input and output differences will be $\triangle_A = A^j \oplus A^k$ and $\triangle_B = B^j \oplus B^k$ respectively. A pair of corresponding input and output differences is called a differential. We want to minimize the correlation between input and output differences which in turn make accurate predictions of intermediate bits more difficult during the encryption process. In the cipher a series of differentials for consecutive rounds which satisfy $\Delta_B^k = \Delta_A^{k+1}$ for rounds 1 to l is called an l-round differential characteristic. This can be used to find the overall differential probability of cipher.

Differential probabilities are influenced by the cipher component within the rounds of the cipher. S-Boxes form a key component of block cipher with respect to their security as they give necessary sole source of nonlinearity of the system. An S-Box is a pair of two values, the first value presents the difference between two input values and the second for the difference between their corresponding output values. If the dimension of S-box is m×n, then there will be 2^{2n-1} - 2^{n-1} possible distinct pairs producing input differences from a possible 2^n distinct values. Tabulting the frequency of occurrence of all the resultant output differences, of which 2^m distinct values are possible, form the basis of the difference distribution table of the S-Boxes. Thus, the difference distribution table is a $2^n \times 2^m$ matrix containing the frequency of occurrence of all possible output differences given possible input differences. The largest value in the difference distribution table of an S-Box is usually written as δ and is called differential uniformity.

The value in each row of the pairing distribution table must sum to 2^n since an input difference exists for the pairing of every possible distinct input to the S-Box. Therefore, a flat difference distribution table, in which the frequency values are almost uniform, implies that the magnitude of the frequencies are small. An S-Box whose difference distribution table is flat provides little or no information about output difference which

may be exploited to reveal intermediate bits of cipher. Large frequency value in the difference distribution table can be used to form a differential characteristic with high probability.

In a typical cipher system, several rounds of processing occur with multiple S-Box look-ups. By combining S-Box differentials, a differential characteristic probability for the cipher system can be determined. In order for a cipher to successfully resist differential cryptanalysis, the differential characteristic probability should be small. Ciphers which contain a greater number of rounds are likely to be better able to achieve a low probability differential characteristic. The magnitude of S-Box differential will also affect the differential characteristic probability of overall system. The absence of any high values in the difference distribution table of the S-Box result in small S-Box differential probability and thus produces a differential characteristic with low probability.

Let D_S be a $2^n \times 2^m$ matrix representing the difference distribution table of an n×m S-Box. Let C_S be a $2^n \times 2^m$ matrix representing the autocorrelation matrix of S. In [105], it has been shown that for an n×m S-Box with $n \geq m$, the relationship between its difference distribution table and autocorrelation matrix is given by the matrix product $D_S\hat{L}$, where $\hat{L}$ is the polarity form of the linear matrix(Sylvester-Hadamard matrix [67]). A lower bound on the differential uniformity δ involving the maximum absolute values in the autocorrelation matrix of an S-Box is given in [105] as follows:

$$\delta \geq 2^{n-m} + 2^{-m} AC(S_{n,m})_{max} \tag{1.3.1}$$

where $AC(S_{n,m})_{max}$ is as defined in Definition 1.16. Noting that δ takes a value in the range $[2^{n-m},2^n]$, an $AC(S_{n,m})_{max}$ value of 0, exhibited by bent S-Box, results in the minimum δ value possible. Additionally, the presence of non-linear structure in the S-Box will consequently cause $AC(S_{n,m})_{max}$ to take the value 2^n. A further observation made in [105] is that a small δ implies values for $AC(S_{n,m})_{max}$. Hence, minimizing the

overall autocorrelation of S-Box helps to resist differential cryptanalysis through the minimization of their differential uniformity and in turn reducing the characteristic probability of the cipher.

It is clear from [105] that two upper bounds on the nonlinearity of an n×m S-Box are provided which relate it to the enumeration of non-zero entries in the difference distribution table of the S-Box and depend also on n and m. In essence, an increase in the number of nonzero entries in the table corresponds to an S-Box with potentially higher nonlinearity and vice-versa, i.e., a highly nonlinear S-Box forces a minimum number of nonzero entries in the difference distribution table, resulting less susceptibility to differential cryptanalysis.

It is shown that differential cryptanalysis [7, 8] breaks the Data Encryption Standard(DES). The susceptibility of DES is primarily due to the fact that the difference distribution tables of the DES S-Boxes exhibit clear non-uniformity, whilst resistance against differential cryptanalysis is characterized by a highly uniform difference table, as discussed above.

So, we can conclude that to resist the cipher system to the differential cryptanalysis Boolean function(or S-box) should have high nonlinearity and low autocorrelation.

1.3.2 Linear Cryptanalysis:

Matsui [71] introduced linear Cryptanalysis. It is a form of plaintext attack which attempts to approximate the relationship between plaintext, ciphertext and key bits by forming a linear expression and evaluating the probability of linear expression accurately depicting the relationship. So, linear cryptanalysis attack aims to find the information about the key bits. We can explain the theory of linear cryptanalysis as follows.

Let X=$X_1X_2X_3...X_l$ and Y=$Y_1Y_2Y_3...Y_l$ be plaintext and ciphertext blocks respectively

of a block cipher of length l. Goal of linear cryptanalysis is to find a linear expression for some combination of input and output bits where,

$$\bigoplus_{i=1}^{l} \psi_i X_i = \bigoplus_{j=1}^{l} \tau_j Y_j \tag{1.3.2}$$

with ψ_i, $\tau_j \in \{0, 1\}$. The expression with the highest probability of being valid will be the best linear approximation, and the best affine approximation is the expression with the lowest probability of being valid. Let *p=P(X=Y)* be the probability related to the above expression. Cipher will be more resistant to linear and affine approximation if $p \approx 1/2$. So, the probability bias is given by $\mid p - 1/2 \mid$, the variation away from the expected probability for a random process. Any linear expression which seeks to relate the plaintext, ciphertext and key bits of a cipher must include the structure of the cipher and the components, including any S-boxes utilized in the rounds. To find a linear approximation to an $n \times m$ S-box, the linear relationships between inputs and outputs of the S-box may be calculated for all pairs of inputs and outputs. This is expressed in a $2^n \times 2^m$ matrix which is referred to as the linear approximation table. In this table, each entry $L_{X',Y'}$ can be defined as

$$L_{X',Y'} = 2^{n-1} - hd(X', Y'). \tag{1.3.3}$$

This value provides the signed probability bias, $L_{X',Y'}/2^n = p'$*=P(X'=Y')* - $1/2 \in$ [-1/2, 1/2]. If p'=0, then linear approximation to S-box is not possible. If $p' \to \pm 1/2$, then S-box can be easily approximated by a linear or affine function. Thus, the best linear approximation to an $n \times m$ S-box will be the linear expression of the form given in Equation 1.3.2. Generally, when we apply linear cryptanalysis to a block cipher system, it involves finding a linear approximation with a large signed probability at each stage, typically rounds, of the cipher. The probabilities of linear equations best approximating different stages of the encryption process are combined. The ability of this combination relies upon the assumption of independence of the linear approximation at each stage. To define the overall probability of a linear expression which necessar-

ily combines the probabilities of multiple linear approximations all holding under this assumption, Matsui used the Pilling-up lemma. Application of this lemma gives the probability for the cipher's overall linear approximation. The higher the probability calculated from the Pilling Lemma, the more likely the approximation will successfully retrieve relevant bits of the key given sufficient plaintext-ciphertext pairs. Further, at each stage of the process, greater the magnitude of the of the bias exhibited by the individual linear expression, higher the overall probability of approximating the cipher linearly. For linear approximation to a component s-box, bias values in its linear approximation table which are nonproportionally high will result in a more successful cryptanalytic attack on the cipher system.

A linear expression combines the multiple linear approximations. Matsui [71] defined the overall probability of the linear expression by using the Pilling-up Lemma. The overall linear approximation of the cipher is obtained by linking multiple linear expressions together. Its probability is obtained by using the Pilling-up Lemma. If this probability is higher, then the linear approximation will more successfully retrieve relevant bits of the key provided sufficient plaintext-ciphertext pairs are available. If the magnitude of the probability bias of the individual linear expression at each stage of the process is greater, the overall probability of approximating the cipher is higher. Bias values is linear approximation table of a component S-box which are disproportionably high will give more successful cryptanalytic attack on the cipher system. The bias values in the linear approximation table and the entries in the WHT matrix of all linear combinations of component Boolean functions of the S-box are closely related to each other by the relation [11]

$$bias = L_{X',Y'} = W_f(\lambda)/2 \tag{1.3.4}$$

Using Equation 1.2.7 in Equation 1.2.11, we get

$$Nl(S_{n,m}) = \min_{G} (2^n - \max_{\lambda \in \mathbb{F}_2^n} |W_{g_j}(\lambda)|)/2. \tag{1.3.5}$$

Putting the value of $W_f(\lambda)$ from Equation 1.3.4 in Equation 1.3.5, we get

$$Nl(S_{n,m}) = 2^{n-1} - \mid L_{X',Y'} \mid_{max} \tag{1.3.6}$$

where $\mid L_{X',Y'} \mid_{max}$ represents the maximum absolute value in the linear approximation table and $X' \neq 0$, $Y' \neq 0$. For large n and m the determination of the entire approximation table for an $n \times m$ S-box is infeasible. Nevertheless, the incorporation of highly nonlinear S-box into cipher system is desirable in order for the cipher to be resistant to linear cryptanalysis attacks. Matsui [71] showed that DES was breakable by linear cryptanalysis. This happened because of the existence of high magnitude values in the linear approximation tables of the DES S-boxes. The resistance against linear cryptanalysis requires low magnitude values in the linear approximation table. These values are obtained by the using of highly nonlinear S-box. Hence, high nonlinearity is an important property for the security of cipher systems and components.

1.3.3 Correlation Attacks

Siegenthaler [87] introduced the concept of a correlation attack in 1984. Since its inception a number of specific variants [87] such as

(i)Fast Correlation Attacks [68],

(ii) Divide and Conquer attacks [22],

(iii)Decimal Attacks [32] are developed. Still all these collectively are called correlation attacks. Modern stream ciphers use combination keystream generators such as those which comprise multiple linear feedback shift registers(LFSRs). These LFSRs linked together with a nonlinear combining function. Correlation attacks analyze the correlation between the keystream and a sequence of output bits from one or more LFSRs. The uncorrelatedness between the resulting keystream and some fixed subset of m input variables of the combining function from the individual LFSR determines the order of correlation immunity, m. In a stream cipher, often a cryptanalytic attack

targets the initial state of each of n individual LFSRs. This attack aims to find the most significant correlation between the output of the target LFSRs and the output of the combining function. This attack will be more inefficient and ineffective if length of LFSRs is longer or more LFSRs are targeted.

Meier and Staffelbach [68] developed a fast correlation attack. This uses a series of parity check equations which are determined from the feedback polynomial of LFSRs. The probability that all of the parity check equations hold for each bit in a keysyream, is calculated and the bit positions with the highest probabilities are then used to form a proposed candidate initial state. To get the state with perfect correlation small changes are made to this candidate. Greater efficiency is achieved as the parity check equations are able to be computed much quicker than the exhaustive search process of [86].

The best trade-off among the properties of a Boolean function can increase a stream cipher's resistance to correlation attacks. In particular, if a function has high nonlinearity, fast correlation attack becomes infeasible as a greater distance between function and the set of all affine functions that prevents a good linear approximation given by the parity check equations. Moreover, if function is correlation immune of order m, the output of function is not correlated to any fixed subset of m input variables. It will increase the resistance of a Boolean function to correlation attacks. To avoid the weakness of a Boolean function some other properties like balancedness, algebraic degree etc. also play an important role. If we want to prevent output bias and maintain high algebraic complexity, thus nonlinearity and correlation immunity should have optimal values.

■

Apart from construction of cryptographically strong Boolean functions, we have also developed the method to solve the set of linear inequalities. To solve a set of linear inequalities is a difficult task if system is nonconvex or number of variables is large. There are methods to solve the set of linear inequalities but most of them have some limitations. The method that we have developed is applicable for all types of prob-

lems(nonconvex, large variables size, large inequalities size) and works for all types of variables real as well as binary also. Apart from ability to solve all types of inequalities, our method gives multiple solutions in one iteration. In Chapter 6, we have explained our method and shown how our method is better than previous methods that have been used to solve the set of linear inequalities.

1.4 Heuristic Techniques

The focus of our work is to improve the cryptographic properties of cipher components and solution of system of linear inequalities. As the size of input space increases, it quickly becomes infeasible to exhaustive search space in order to analyze the properties exhibited by functions within the space. Thus to discover knowledge about functions(specially for large space), it is necessary to employ techniques to direct investigations to certain parts of the space which contain functions of interest(desired functions with multi criteria). The two main techniques which have been used for this purpose by researchers in the field are:

- Heuristic Techniques
- Algebraic Constructions.

  Heuristic techniques are driven by directed search algorithms typically searching in a localized area from a specified starting point. Their use is more frequent for searching in large spaces in order to find a large number of solutions which are satisfactory, but generally not optimal. That is why these techniques are usually applied to difficult combinatorial problems. Some well known heuristic Techniques are Simulated Techniques [53], Tabu Search [39], Genetic Algorithms [45] and Hill Climbing Techniques [71]. Verma and Ramesh [96] formulated scheduling of preventive maintenance as a constrained nonlinear multiobjective decision making problem and used an elitist GA to solve it. Verma et.al. [97] carried out a multiobjective TBPM optimization and got optimum results regarding some

parameters of a large engineering plant(LEP). Mishra and Jaiswal [72] considered a nondifferentiable multiobjective semi-infinite programming problem and established sufficient optimality conditions. Gupta et.al. [43] considered portfolio selection in a multiobjective decision making environment and employed real coded GA to solve it. Wang and Pham [98] studied a multiobjective maintenance optimization problem. NSGA-II is applied to solve the problem. The comparison results show that the optimization solutions is consistent between single-objective and multiobjective optimizations. Mohanty and Vijayaraghavan [73] suggested a method to construct multiobjective programming problem into a lexicographic goal programming problem by appropriately fixing priorities and goals. The conversion method uses the concept of conflict among objectives. Bector et.al. [3] have given an account of the fundamental principles of optimization theory with current research work. Bharti and Singh [9] gave computational algorithm to solve MOLPP using IF optimization method. Bharti et.al. [10] developed a new method for solving MOLPP in IF environment.

Algebraic Constructions rely on proven mathematical relationships holding for a generalized construction of functions. Whilst Algebraic Constructions have been shown to generally produce functions with the most optimum combinations of properties and they are not designed to produce a great number of such functions. Further, the existence of inherent weaknesses in functions produced by Algebraic Constructions is a valid concern. But as space size increases, these techniques are generally unable to generate optimal function. Let f be a Boolean function of n variables. If the number of input variables increases by one, the number of functions in the space increases by a factor of 2^{2^n} and the probability of searching optimal functions decreases. However, because Heuristic Techniques involve directed search methods, they have been shown to produce consistent results in finding functions with good properties, and unlike Algebraic Constructions, are able to produce a large number of such functions. For this reason, the

approach taken in this research has been primarily focussed on the application of Heuristic Techniques.

In this chapter we have described some existing techniques and also discussed our method that we have used in this thesis.

1.4.1 Overview of Existing Heuristic Techniques used

Genetic Algorithms and Hill Climbing Techniques have been used, for many years, as heuristic optimization techniques for non-cryptographic applications. Minsky [71] applied a form of hill climbing to develop artificial intelligence systems. Holland [45] introduced the concept of Genetic Algorithms to a study of cellular automata.

We now discuss Boolean functions' cryptographic applications, specifically in generating strong components for use in cipher systems to enhance their security.

1.5 Related Work by Other Researchers

We have described many properties of Boolean functions and their correlation in this chapter. Further, we are giving some literature review related to the properties of Boolean functions which we have used in this thesis.

1.5.1 Previous work Related to optimize the nonlinearity

High nonlinearity is one of the most essential properties required not only for the strong Boolean functions but also for the security of cipher system incorporating these components. Cryptanalytic attacks which exploit moderate nonlinearity values are linear cryptanalysis [71] and best affine approximation [25].

Research work that makes a related contribution to the area of obtaining highly nonlinear Boolean functions are given here. This work encompasses research that has been

conducted in the construction of such functions largely through algebraic means. A brief discussion of some selected examples of construction is now outlined.

Dobbertin [26] provides a proposition for the construction of highly nonlinear balanced Boolean functions from normal bent functions. The construction is based on the idea that turning a bent function into a balanced function will ensure high nonlinearity by minimizing the maximum absolute WHT value $W_f(\lambda)$. The substance of the proposition is the modification of the initial segment of a 2n-variable normal bent function which is constant. We replace it by a balanced function whose maximum absolute WHT value is small. To construct a 2n-variable balanced function f from a 2n-variable normal bent function g, an appropriate n-variable balanced function h is expected such that $WHT_{max}(f) = 2^n + WHT_{max}(h)$. Seberry et.al [84] presented construction methods for highly nonlinear n-variable Boolean functions, for even variables $n \geq 4$ (here n=4j or n=4j+2, $j \geq 1$) and a construction method involving the concatenation of all linear functions in F_2^{2j} and F_2^{2j+1} respectively was described. In each case, $L_0(x)$(first linear function) was replaced by a balanced Boolean function and it was then concatenated with the remaining linear functions. For n= 4j, $L_0(x)$ in F_2^{2j} was substituted for a balanced 2j-variable function representing the concatenation of the linear functions $L_{2^j}(x)$,...,$L_{2^{2j+1}-1}(x)$ in F_2^{j+1}. Similarly, Seberry et.al [84] also discussed concatenating linear functions when $n = 2^i$, $i \geq 2$ or $n = 2^u(2v + 1)$, $u \geq 1$, $v \geq 1$.

Maitra [60] proposed a construction method for balanced n($n > 3$ and odd) variable Boolean functions and found the low maximum absolute autocorrelation value and high nonlinearity by using this construction method. The component functions of this construction are taken to be bent. The construction involves the concatenation of two bent functions $b_1(x)$ and $b_2(x)$ in F_2^{n-1}, each produced by the $\oplus$ sum of bent subfunctions. If the hamming weights of the functions $b_1(x)$ and $b_2(x)$ are such that the resulting odd n-variable Boolean function f(x)= $b_1(x) \parallel b_2(x)$ is not balanced, then f(x)= $b_1(x) \parallel \overline{b_2(x)}$ is used to get achieve the balanced function, where $\overline{b_2(x)}$ represents complimenting function of $b_2(x)$. Youssef and Gong [103] discussed construction of Boolean functions with large distance to all bi-objective monomials. Gong and Khoo [51] suggested new

construction for Boolean functions.

1.5.2 Previous work Related to optimize the resiliency

Resiliency is an essential cryptographic property for a Boolean function. The Boolean functions are incorporated into those cipher systems whose most significant source of strength relies on little or no correlation between the combined input bits and the output bits of its component functions. The resilient Boolean functions have been most commonly used in stream ciphers. Cryptographically attacks on stream cipher typically focus on revealing the secret key by retrieving the initial states of linear feedback shift registers. This task is made easier if a high correlation exists between the input and output bits of the combining Boolean functions. The attacks exploiting this correlation are called the correlation attacks [47, 48, 68, 86]. We know that there exists a tradeoff between the correlation immunity and nonlinearity of a Boolean function. Now, we give a brief literature review of the work done to improve the resiliency.
Dawson et.al [23] proposed a method to construct n-variable balanced Boolean function to find non-zero order of correlation immunity coupled with high nonlinearity. This construction method combined heuristic method and an algebraic construction method. Wu and Dawson [101] gave a method for the construction of correlation immune Boolean functions. By using this method, they obtained 10-variable balanced Boolean functions having nonlinearity 480 and correlation immunity of order 1, and 10-variables balanced Boolean function having nonlinearity 464 and correalation immunity of order 2. Maity and Johansson [63] proposed a method to construct cryptographically important Boolean functions and got 8-variable and 10-variable 1-resilient Boolean functions having nonlinearity 116 and 488 respectively. Maity and Maitra [64] gave a small extension to the method proposed by Maity and Johnsson of [63]. By using this extension method, they constructed balanced, first order correlation immune functions having nonlinearity of 116. Roy [81] presented an interesting review which includes a summary of construction methods and bounds on cryptographic proper-

ties achievable for correlation immune functions. Fedorova and Tarannikov [31], Gong and Khoo [36], Khoo and Gong [51], Tarannikov [93, 94] worked on the construction of t-resilient Boolean functions. Gupta and Sarakar [42], Johansson and Pasalic [49], Kurosawa et.al. [56], Zhang and Zheng [104] worked on the construction of multiple output functions(S-boxes) which are correlation immune.

1.5.3 Previous word Related to optimize the Autocorrelation (propagation criteria *PC(k)*)

Seberry et.al. [85] presented methods for constructing highly non-linear balanced Boolean functions satisfying the strict avalanche criterion(SAC). Kavut and Yucel [50] developed a new method based on simulated annealing and hill climbing to construct Boolean function having low autocorrelation with other optimal properties. They got some Boolean functions of 8 and 9 variables. Gong and Khoo [36] introduced a notion of dual function to study Boolean functions. Using this notion, they constructed highly nonlinear resilient functions with better additive autocorrelation than the Maiorana-McFarland functions. Maitra [61] provided a construction method for unbalanced, first order correlation immune Boolean functions of even number of variables $n \geq 6$. He provided new lower bounds and related results on absolute indicator and sum of square indicator of autocorrelation values for low order of correlation immunity. These functions achieved the best nonlinearity of $2^{n-1} - 2^{n/2} + 2^{n/2-2}$.

Burnett et.al. [11] presented two heuristic optimization methods for generating n-variable Boolean functions. Method 1 generated many 8-variable balanced Boolean functions with nonlinearity 116 and maximum autocorrelation value 16, which had not previously been found directly by heuristic technique. Method 2 generated known optimal m-resilient Boolean functions with high nonlinearity, varying degrees of resiliency and maximum algebraic degree.

Guillot [41] presented an extension of the Maiorana-McFarland method [66] for building Boolean function with good cryptographic properties. He studied nonlinearity, resiliency and autocorrelation properties of Boolean functions. He could not get better result for nonlinearity. But he could get a better result for resiliency and autocorrelation.

Wei [100] constructed Boolean function having multiple cryptographic criteria(balancedness, nonlinearity, autocorrelation etc.) based on the use of linear error-correcting code. Izbenko [46] gave a modification of the hill climbing method to design balanced, highly nonlinear Boolean functions with high algebraic degree and low autocorrelation. The functions with the best known profiles have been designed. Tang et.al [91] gave a method to construct balanced Boolean functions of n-variables satisfying SAC, where $n \geq 10$ is an even integer. These functions have the highest nonlinearity and the best global avalanche characteristics(GAC) property compared with the known balanced Boolean functions with SAC property. Thavaneswaran et.al. [95] considered GARCH models and derived the theoretical autocorrelation functions for them.

1.6 Summary

The thesis consists of seven chapters. The chapter-wise summary of the thesis is as follows:

Chapter 1 is introductory in nature. In this chapter, we have defined the relevant supporting theory of Boolean functions. In particular, we have provided numerous definitions and theorems for various aspects of the theory. The necessary cryptographic properties which are used to analyze the strength of Boolean functions have been also defined and discussed, and inter-relations between pairs of selected properties are also discussed. Finally, we have presented a brief summary of major cryptanalytic attacks

against Boolean functions and cipher systems.

In **Chapter 2**, we have developed a new evolutionary method to optimize the Boolean functions' properties by two objective optimization method. In this Chapter , we have taken balancedness, nonlinearity and resiliency, and developed an evolutionary method to construct Boolean functions having these properties at optimal level. We have got the desired functions and compared our results with previous results. Our results are as good as previous results.

In **Chapter 3** also, we have developed a new evolutionary method to optimize the Boolean functions' properties by two objective optimization method but here we have taken balancedness, nonlinearity and autocorrelation, and developed an evolutionary method to construct desired Boolean functions. We have got the desired functions and compared our results with previous results. Our results are at least as better as previous results.

In **Chapter 4**, we have developed a new method to optimize the Boolean functions' properties by three objective optimization method. In this chapter, we have taken balancedness, nonlinearity, resiliency and autocorrelation simultaneously to optimize these properties. We have got the desired functions and compared our results with previous results and found that our results are at least as better as available in the literature.

In **Chapter 5**, we have introduced the concept of biasedness in the proposed method and developed a new method based on biasedness to construct Boolean func-

tions and got the desired results. In this chapter, we got the Boolean functions of 7 and 8 variables that could not be possible by the methods developed in Chapters 2, 3 and 4. We also compared our results with previous results and found that our results are at least as better as available in the literature.

In **Chapter 6**, we have developed a new method based on NSGA-II to solve a system of linear inequalities. This method is applicable for all types of inequalities. We have generated three examples of different types and solved them by the developed method. The developed method gives better spread of solutions. Consequently, our method is better than previous methods to solve the system of linear inequalities.

In **Chapter 7**, based on the study carried out in the thesis, conclusions are drawn and future scope of the research work is suggested.

Chapter 2

A new Evolutionary multiobjective Approach to Construct Balanced Boolean Functions based on Two Objectives-Nonlinearity and Resiliency

Boolean functions are the basic components in cryptography. Many desirable properties such as nonlinearity, balancedness, autocorrelation etc. are known for Cryptographically strong Boolean functions . It is difficult task to get optimal trade-offs among such properties. Nowadays, the design of strong cryptographic Boolean functions is a multi-objective problem. In this chapter, we focus on non-linearity, resiliency related criteria and explore a multiobjective evolutionary approach aiming to find balanced Boolean functions having the best trade off between nonlinearity and resiliency for 4, 5 and 6 variables. We show that the multiobjective approach is an efficient alternative to single objective optimization approaches presented so far [1]. In this chapter, we show how non-dominating sorting genetic algorithm($NSGA-II$) can be used to construct

Boolean functions with profiles of cryptographically relevant properties.

2.1 Introduction

Boolean functions are the basic components in cryptography. In this chapter, we have outlined a number of cryptographic properties of Boolean functions. Among them high nonlinearity, balancedness, high algebraic degree, high resiliency and low autocorrelation are important from cryptographic point of view. These properties contribute to the security system and provide resistance to them. There are different types of cryptanalytic attacks against Boolean functions like differential cryptanalysis [7], correlation attack [68], linear cryptanalysis [71], etc. Boolean functions are the most vulnerable to attack if exploitable weaknesses in their components exist. For this reason, it is important to be able to ensure that any Boolean function exhibits the appropriate combination and measures of robust cryptographic properties. To resist differential cryptanalysis Boolean functions should have high nonlinearity and low autocorrelation, to resist linear cryptanalysis nonlinearity should be high, to resist correlation attack correlation immunity should be high [12] etc. The Boolean functions having the best trade-offs between nonlinearity and resiliency will resist the correlation attack. So, In this chapter, we propose an evolutionary multiobjective method to construct balanced Boolean functions having the best trade-offs between nonlinearity and resiliency.

It is well known that a function cannot at the same time be balanced and can have maximum algebraic degree, maximum distance to linear functions and maximum distance to linear structure. If we increase the resiliency of a Boolean function, it decreases the nonlinearity of the function. Similarly, if we decrease the autocorrelation, it decreases the nonlinearity. The trade-offs between some of these criteria have been shown in [1,62,70,83]. Some works have been sought to combine algebraic construction with deterministic computer search methods [62,83] and some authors have attempted using search heuristics such as genetic algorithms, hill climbers, and simulated anneal-

ing to generate Boolean functions [18, 69, 70]. A clear trade-off has been shown for correlation immunity, algebraic degree and nonlinearity [19]. By using meta-heuristic search (particularly hill climbing, genetic algorithms and simulated annealing), Burnett et. al. [50], Kavut and Yucel [11], and Izbenko et. al. [46] designed Boolean functions. Pasalic [78] investigated the possibilities of an iterative concatenation method towards construction of Boolean functions resistant to algebraic cryptanalysis. Beelen and Leander [4] constructed Boolean functions by using concatenation from codes over $\mathbb{F}_q$ containing a first-order generalized ReedMuller code. However, generating Boolean functions purely by constructive algebraic methods becomes increasingly difficult as the number of criteria to be satisfied is augmented. The more criteria to be taken into account, the more difficult is to generate Boolean functions satisfying those properties purely by constructive algebraic means. Previous researches for generating highly non-linear balanced functions by search heuristics have used single objective optimization approaches by considering either the non-linearity or autocorrelation. These methods are called direct-single objective approaches. Clark and Jacob [18], and Clark et al. [19] proposed a two-stage optimization approach and obtained the best functions of small number of variables. In the first stage, an annealing-based method is used to evolve functions restricting the spread of Walsh values. In the second stage, hill climbing with respect to nonlinearity or autocorrelation is applied to the best function obtained in the first stage. This method also focuses on single objective optimization, although at the end of the run other properties of the function are also measured. Better results in terms of the individual objectives have been achieved by the two-stage optimization approach compared to the direct single objective methods. The two-stage approach is an important improvement towards the automatic generation of highly non-linear Boolean functions with cryptographic application. However, it suffers from some drawbacks and limitations. First, the objective function used during the first stage introduces two parameters and needs to be fine tuned in order to produce good results and second, the method offers very limited function diversity. Aguirre et. al. [1] focused on nonlinearity related criteria and used an evolutionary

multi-objective approach, namely multiobjective random bit climber(moRBC), to find balanced Boolean functions of similar characteristics satisfying multiple criteria. They showed that the multiobjective approach is an efficient alternative to single objective optimization approaches. Keeping this in the view, in this chapter, we have developed a new evolutionary multiobjective approach and constructed cryptographically strong Boolean functions, that is, the balanced Boolean functions having the best trade-off between nonlinearity and resiliency.

Section-wise the remaining chapter is arranged as follows: Section 2.2 gives a brief description of NSGA-II. In Section 2.3, we have developed an evolutionary multiobjective approach to construct the desired Boolean functions. Section 2.4 gives results and discussions of our work. In Section 2.5, we have concluded our work.

2.2 A Brief Description of NSGA-II

There are many simple evolutionary algorithms like simple genetic algorithm, Simulated annealing, Hill climbing etc. in literature. All these methods are single objective optimization methods. But in real life problems we come across to the multiobjective optimization problems(MOOPs) and want to get optimal trade-off among objectives. So, only single objective optimization method is not sufficient. Nondominated sorting genetic algorithm(NSGA) developed by Srinivas and Deb. [89] is an extension of the Genetic Algorithm for multiobjective optimization. Multiobjective evolutionary algorithms(MOEAs) which use non-dominated sorting and sharing have been mainly criticized for their (i) $O(kN^3)$ computational complexity (where k is the number of objectives and N is the population size), (ii) non-elitism approach, and (iii) the need for specifying a sharing parameter. Deb et. al. [21] suggested a non-dominated sorting based MOEA known as the Non-dominated Sorting GA-II(NSGA-II), which alleviates

all the above three difficulties. Specifically, a fast non-dominated sorting approach with $O(kN^2)$ computational complexity is presented.

NSGA-II was proposed to resolve the weaknesses of NSGA, specially its non-elitist nature. Although several elitist MOEA exist, few are widely used, one of them is NSGA-II. It maintains the solutions of the best front found in a generation into the next generation (elitism). The introduction of the controlled elitism operator in the NSGA-II algorithm produces a better equilibrium between exploitation and exploration.

In NSGA-II, for each solution one has to determine how many solutions dominate it and how many solutions to whom it dominates. The NSGA-II estimates the density of solutions surrounding a particular solution in the population by computing the average distance of two points on either side of this solution along each of the objectives of the problem. This value is called the **crowding distance**. During selection, the NSGA-II uses a crowded-comparison operator which takes into consideration both the nondomination rank of an individual solution in the population and its crowding distance (i.e., nondominated solutions are preferred over dominated solutions, but between two solutions with the same nondomination rank, the one that resides in the less crowded region is preferred). Instead, the elitist mechanism of the NSGA-II consists of combining the best parents with the best offspring obtained (i.e., a selection). Due to its clever mechanisms, the NSGA-II is much more efficient (computationally speaking) than its predecessor NSGA, and its performance is so good that it has become very popular in the last few years. As we know simple evolutionary algorithm is extended to maintain a diverse set of solutions with the emphasis on moving towards a true Pareto-optimal region. The non-dominated sorting GA (NSGA) proposed by Srinivas and deb [89], is one of the first such algorithms. It is based on several layers of classification of the individuals. Non-dominated individuals get a certain dummy fitness value and then are removed from the population. This process is repeated until the entire population has been ranked. It is a very effective algorithm but it has been criticized for its computational complexity, lack of elitism and its requirement for specifying sharing

parameters in the algorithm. Based on these issues, a modified version of the NSGA, named NSGA-II [21] is developed. Two distinct entities are calculated in the NSGA-II to validate the quality of a given solution. The first is a **domination-count** where the number of solutions that dominate a given solution are tracked. The second keeps track of how many sets of solutions a given solution dominates. In the process, all the solutions in the first non-dominated front will have their domination count zero. The next step is to select each solution in which the nondomination count is set to zero and visit all other solutions in the solution set and reduce the domination count by one. In doing so, if the domination count of any other solution becomes zero, this solution is grouped in a separate list. This list is flagged as the second nondominated front. This process is then continued with each member of the second list until the next non-dominated front is identified. The process is continued until all fronts are identified. Based on the non-domination count given to a solution, a non-domination level will be assigned. Those solutions that have higher nondomination levels are flagged as non-optimal and will never be visited again. One of the key requirements of a successful solution method is ensuring that a good representative sample from all possible solutions is chosen. Introduction of a density estimation process and a crowded-comparison operator has helped NSGA-II to address the above need. The crowding-distance computation requires sorting of a given population according to each objective function value in ascending order of magnitude. Once this is done, the two boundary solutions with the largest and smallest objective values are assigned distance values of infinity. All other solutions lying in between these two solutions are then assigned a distance value calculated by the absolute normalized distance between each pair of adjacent solutions. After each population member is assigned a crowding-distance value, a crowded-comparison operator is used to compare each solution with the others. This operator considers two attributes associated with every solution which is the nondomination rank and the crowding-distance. Every solution is rated with others based on the non-domination rank. Solutions with lower ranks are deemed better in this attribute. Once solutions that belong to the best front are chosen based on the non-domination rank, the solution

that is located in a lesser-crowded region is considered better and forms the basis of the NSGA-II algorithm.

NSGA-II, doveloped by Deb et.al. [21] is a generational MOEA that aims at approximating the Pareto optimal fronts for a given problem, while keeping high diversity in its result set.
It works on three main modules:
1. Non-dominated Sorting
2. Crowding distance assignment
3. Crowded comparison operator.

1. Non-dominated Sorting: At a certain generation t, it partitions the population P_t into different fronts F_i with index i indicating the non-domination rank shared by all solutions in such a front. The first front F_1 is the actual non-dominated front, i.e., it consists of all non-dominated solutions in the population P_t at a certain generation t. The second front F_2 consists of all solutions that are non-dominated in the set $P_t - F_1$, i.e., each member of F_2 is dominated by at least one member of F_1 as shown in Figure. 2.1 Generally, front F_k comprises all solutions that are non-dominated in the set $P_t - \bigcup_i F_i$ (i varies from 1 to $k-1$).

2. Crowding distance assignment: To get an estimate of the density of solutions surrounding a particular solution i in the population, we take the average distance of two solutions on either side of the solution i along each objective. This quantity i_d serves as an estimate of the perimeter of the cuboid formed by using the nearest neighbors as vertices. It is called the **Crowding distance**(Figure. 2.2). The crowding distance assignment calculates a crowding distance value for each individual within a certain front F_i as the difference between objective function values in the nearest neighbors on each side of the individual, then summed up over all objectives. The

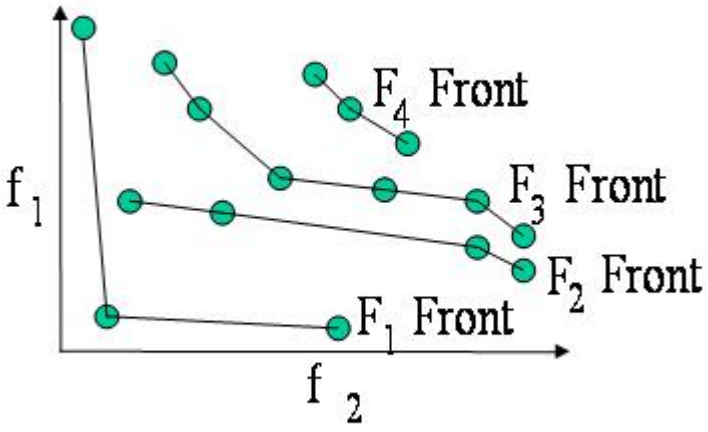

Figure 2.1: Nondominated Sorting of a Population

crowding distance values for extreme solutions (i.e., solutions with the smallest and largest function values occurring within the front) are assigned an infinite value, which is motivated by the pursuit of diversity and which effectively preserves them into the next generation. The front in which they are contained should be partially discarded when a new population P_{t+1} is formed.

3. Crowded comparison operator $\prec_n$: It guides the selection process by defining an ordering on P_t. Each solution i in the population has two attributes:

1. non-domination rank(i_r);
2. crowding distance(i_d).

   The crowded comparison operator $\prec_n$ can be defined as

   i $\prec_n$ j if $i_r < j_r$ or $i_r = j_r$ and $i_d > j_d$.

Between two solutions with different non-domination ranks, the solution having the lower rank is preferred. If both solutions belong to the same front(having the

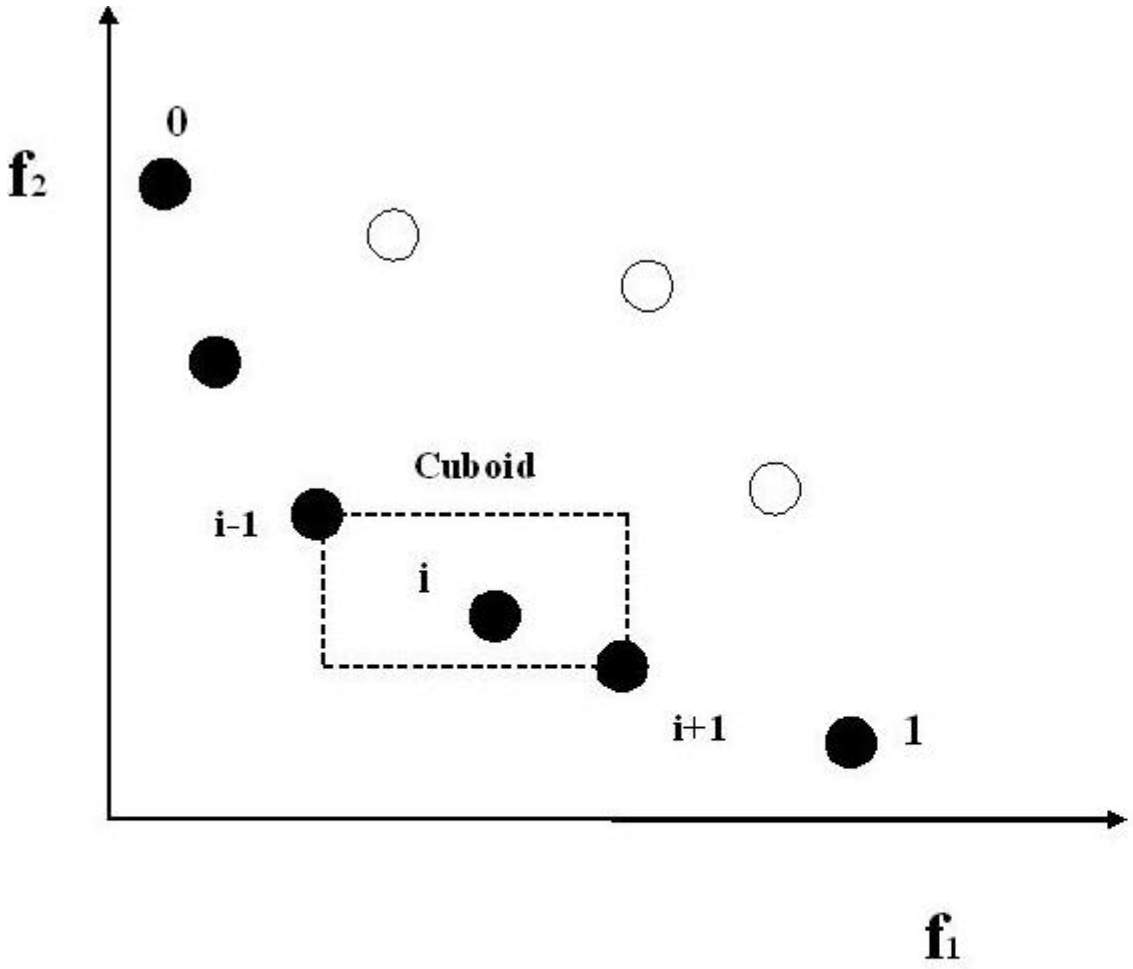

Figure 2.2: Description of Crowding Distance

same rank), we prefer the solution that is located in the lesser crowded region (i.e., with higher crowding distance value). The crowding comparison operator guides the selection process at the various stages of algorithm toward uniformly spread-out Pareto-optimal fronts.

2.2.1 Algorithm of NSGA-II :

Algorithmically, NSGA-II is explained below:

step 1. Combine parent and offspring populations P_t and Q_t and create $R_t = P_t \bigcup Q_t$. Perform a non-dominated sorting on R_t and identify different fronts F_i, $i = 1, 2, ..., etc.$

step 2. Set new population $P_{t+1} = \phi$. Set a counter $i = 1$. Until $|P_{t+1}| + |F_i| < N$(N is population size), perform $P_{t+1} = P_{t+1} \bigcup F_i$ and set $i = i + 1$.

step 3. Perform the Crowding sort procedure(i.e., assign crowding distance and apply crowded comparison operator) and include the widely spread $N - |P_{t+1}|$ solutions by using the crowding distance values in the sorted F_i to P_{t+1}.

step 4. Create offspring population Q_{t+1} from P_{t+1} by using the crowded tournament selection, crossover and mutation operators.

The flow chart depicting the NSGA-II algorithms in shown in Figure 2.3.

2.2.2 Constraint Handling in NSGA-II

Consider a constrained MOOP. The constraints divide the search space into two regions - feasible and infeasible regions. In MOOP, all Pareto-optimal solutions must be

feasible solutions. The constrained problems in NSGA-II can be handled on the basis of dominance. A solution i is said to **constrained-dominate** a solution j if any of the following conditions is true:

1 . Solution i is feasible and solution j is not.

2 . Both solutions i and j are infeasible, but solution i has a smaller overall constrained violation.

3 . Both solutions i and j are feasible and solution i dominates solution j.

When comparing two feasible solutions, the solution which dominates the other solution is considered a better solution. On the other hand, if both solutions are infeasible, the solution having a lesser number of constrained violations is a better solution.

The flow chart depicting the NSGA-II algorithms is shown in Figure 2.3.

2.3 The Proposed Method for Construction of Balanced Boolean Functions Having the Best Trade-offs Between Nonlinearity and Resiliency

In this section, we will describe the proposed method to construct the balanced Boolean functions having the best trade-offs between nonlinearity and resiliency. The proposed method consists of:
(i) Formulation of MOOP,
(ii) Application of NSGA-II.

(i) **Formulation of MOOP :** We will use the criteria of nonlinearity, balancedness

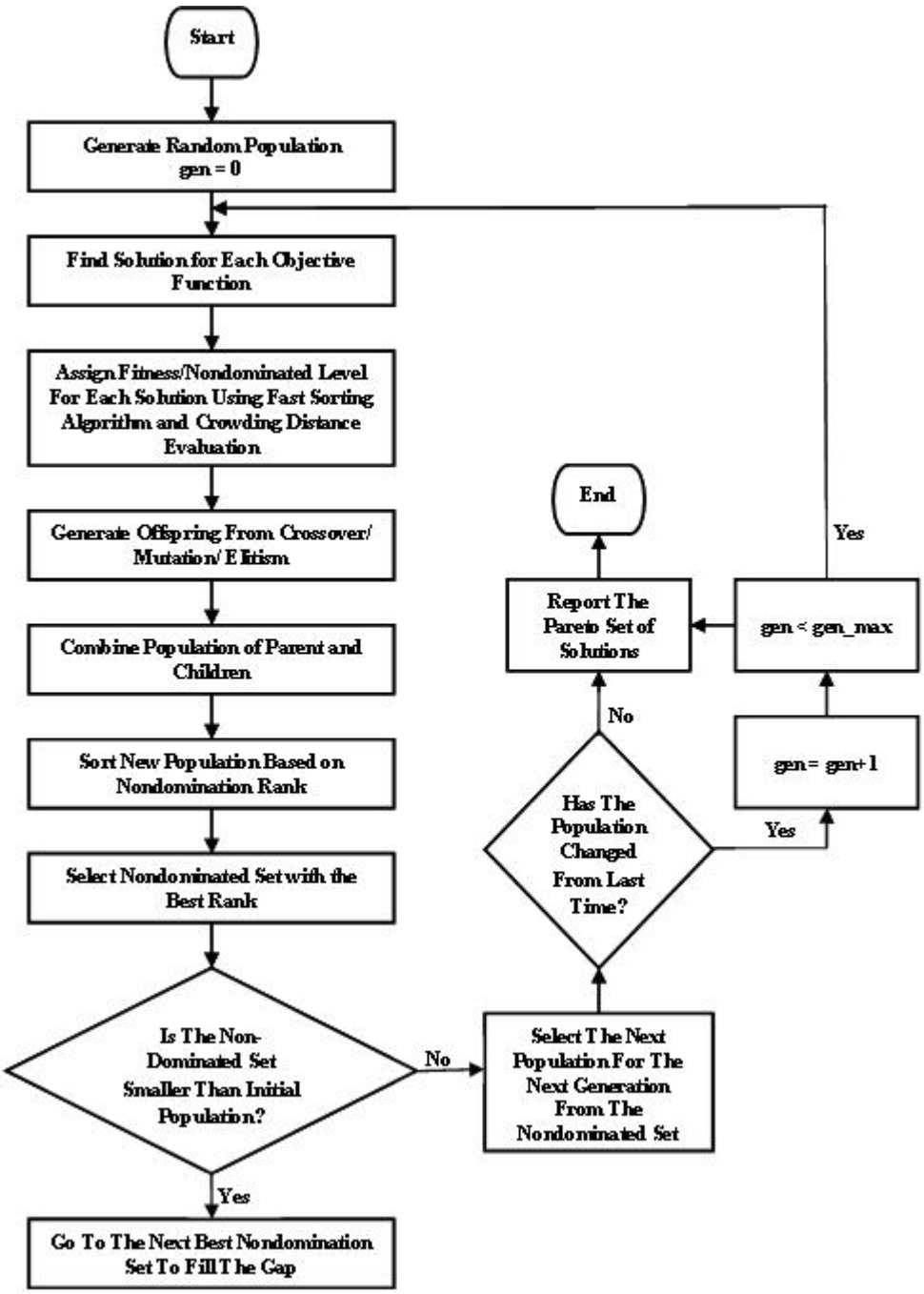

Figure 2.3: Flow chart of NSGA-II

and resiliency to construct the desired Boolean functions.

Now we describe the evaluation criteria used to evolve Boolean functions. The first

evaluation criterion used is the non-linearity Nl and the second is resiliency .

We know that the bent functions have the maximum nonlinearity and for bent functions, $|W_f(\lambda)| = 2^{n/2}$ for all $\lambda \in \mathbf{F}_2^n$. For balanced Boolean functions this ideal bound cannot be achieved but it does suggest that an objective function that seeks to minimize the spread of Walsh Hadamard values is well motivated. So, we take

$$f_1 = \sum_{\lambda \in \mathbf{F}_2^n} ||W_f(\lambda)| - 2^{n/2}|^R \tag{2.3.1}$$

as the first objective function, where R $\in [3, 10]$ is a constant.

If a function has resiliency m, then $W_f(\lambda)$ will be zero for all those $\lambda \in \mathbf{F}_2^n$ for which $w_H(\lambda) \leq m$. Hence, we take

$$f_2 = \sum_{\lambda \in \mathbf{F}_2^n} |W_f(\lambda)| \tag{2.3.2}$$

as the second objective function, where $w_H(\lambda) \leq m$ for $\lambda \in \mathbf{F}_2^n$.

Now,

$$f_1 = \sum_{\lambda \in \mathbf{F}_2^n} ||W_f(\lambda)| - 2^{n/2}|^R$$

for $w(\lambda) \geq m$,

(where $W_f(\lambda) = \sum_{x \in \mathbf{F}_2^n} (-1)^{f(x)+\lambda.x}$ and n is the number of variables.)

So,

$$f_1 = \sum_{\lambda \in \mathbf{F}_2^n} || \sum_{x \in \mathbf{F}_2^n} (-1)^{f(x)+\lambda.x}| - 2^{n/2}|^R,$$

$$= \sum_{\lambda \in \mathbf{F}_2^n} || \sum_{x \in \mathbf{F}_2^n} (-1)^{f(x)}.(-1)^{\lambda.x}| - 2^{n/2}|^R,$$

$$= \sum_{\lambda \in \mathbf{F}_2^n} || \sum_{x \in \mathbf{F}_2^n} (1 - 2.f(x)).a_{\lambda.x}| - 2^{n/2}|^R,$$

$$= \sum_{\lambda \in \mathbf{F}_2^n} ||M_\lambda - \sum_{x \in \mathbf{F}_2^n} 2f(x).a_{\lambda.x}| - 2^{n/2}|^R \tag{2.3.3}$$

where

$$M_\lambda = \sum_{x \in \mathbf{F}_2^n} a_{\lambda.x} \tag{2.3.4}$$

and

$$a_{\lambda.x} = (-1)^{\lambda.x} \tag{2.3.5}$$

$f(x)'s$ are binary variables(Boolean functions that have to be found. We have taken R = 3 for all values of m.

Similarly,

$$f_2 = \sum_\lambda |W_f(\lambda)|$$

$$= \sum_\lambda | \sum_{x \in \mathbf{F}_2^n} (-1)^{f(x)+\lambda.x}|,$$

$$= \sum_\lambda | \sum_{x \in \mathbf{F}_2^n} (-1)^{f(x)}.(-1)^{\lambda.x}|,$$

$$= \sum_\lambda | \sum_{x \in \mathbf{F}_2^n} (1 - 2.f(x)).a_{\lambda.x}|,$$

$$= \sum_\lambda |M_\lambda - \sum_{x \in \mathbf{F}_2^n} 2f(x).a_{\lambda.x}| \tag{2.3.6}$$

for $w_H(\lambda) \leq m$.

So, our problem as an MOOP is given below:

$$\left.\begin{array}{l} minimize F = (f_1, f_2) \\ subject \quad to \\ \sum_{x \in \mathbf{F}_2^n} f(x) = 2^{n-1}. \end{array}\right\} \tag{2.3.7}$$

($\sum_{x \in \mathbf{F}_2^n} f(x)$ should be equal to 2^{n-1} for balanced function).

(ii) **Application of NSGA-II**: Now we apply NSGA-II to the MOOP given in (2.3.7) and get the results(Boolean functions). The obtained results are given in Table 2.1 and discussed in Section 2.4. Comparison of the results is given in Table 2.2. The parameters that we have used in NSGA-II are given in Tables 2.3, 2.4, 2.5 and 2.6.

(4,1,3,12)	0 1 1 1 0 0 0 1 1 0 0 0 1 1 1 0
(5,1,3,12)	1 1 0 1 1 0 0 1 0 0 1 1 1 0 0 0 0 0 0 0 1 1 1 0 1 0 1 1 0 1 0 1
(5,2,3,12)	1 0 1 0 0 0 1 1 0 1 0 1 1 1 0 0 0 1 0 1 1 1 0 0 1 0 1 0 0 0 1 1
(6,1,4,24)	1 0 1 1 1 1 0 1 0 1 0 0 1 0 0 1 0 0 0 1 0 0 1 1 0 1 0 0 1 0 1 1 1 1 1 0 0 0 0 0 1 0 1 1 1 0

Table 2.1: Results obtained by the proposed method

Previous Results	Results by the proposed method
(4,1,3,12)	(4,1,3,12)
(5,1,3,12)	(5,1,3,12)
(5,2,3,12)	(5,2,2,8)
(6,1,4,24)	(6,1,4,24)

Table 2.2: Comparison of results

number of generation	200
population size	100
probability of crossover	0.7
probability of mutation	0.1
seed random number	0.9876
number of bits for binary variables (have been taken equal bits for all variables)	1
number of objective functions	2
number of constraints	2

Table 2.3: Parameters used in NSGA-II for 4 variables.

number of generation	500
population size	200
probability of crossover	0.9
probability of mutation	0.01
seed random number	0.9876
number of bits for binary variables (have been taken equal bits for all variables)	1
number of objective functions	2
number of constraints	2

Table 2.4: Parameters used in NSGA-II for 5 variables with resiliency 1.

number of generation	1000
population size	500
probability of crossover	0.7
probability of mutation	0.1
seed random number	0.9876
number of bits for binary variables (have been taken equal bits for all variables)	1
number of objective functions	2
number of constraints	2

Table 2.5: Parameters used in NSGA-II for 5 variables with resiliency 2

number of generation	1000
population size	500
probability of crossover	0.7
probability of mutation	0.1
seed random number	0.9876
number of bits for binary variables (have been taken equal bits for all variables)	1
number of objective functions	2
number of constraints	2

Table 2.6: Parameters used in NSGA-II for 6 variables.

2.4 Results and Discussion

By applying the proposed method on the basis of the criteria balancedness, nonlinearity and resiliency, we got the desired Boolean functions. These functions are balanced and have the best trade-offs between nonlinearity and resiliency. Applying our method, we have constructed such functions of 4, 5 and 6 variables as shown in Table 2.1. The parameters taken to construct functions of 4, 5 and 6 variables are listed in Tables 2.3, 2.4, 2.5 and 2.6. Table 2.2 shows that our method gives at least as better results as are given in literature [1, 18, 19].

2.5 Conclusion.

In this chapter, we have developed a new evolutionary multiobjective approach to construct desired Boolean functions. By applying our method, we got at least as better results(Table 2.1) as obtained in literature [1, 18, 19]. Thus our approach to construct desired Boolean functions is at least as better as the approaches available in the literature.

Chapter 3

A new Evolutionary multiobjective Approach to Construct Balanced Boolean Functions based on Two Objectives-Nonlinearity and Autocorrelation

In Chapter 2, we have proposed an evolutionary multiobjective approach to construct balanced Boolean functions based on two objectives having the best trade-offs between nonlinearity and resiliency. Now, in this chapter, we will construct balanced Boolean functions by the proposed evolutionary multiobjective approach based on two objectives having the best trade-offs between nonlinearity and autocorrelation. So,we focus on non-linearity and autocorrelation related criteria and explore a multiobjective evolutionary approach aiming to find balanced Boolean functions having the best trade-offs between nonlinearity and autocorrelation for 4, 5 and 6 variables. We show that the multiobjective approach is an efficient alternative to single objective optimization approaches presented so far [1].

3.1 Introduction

In Chapter 2, we have constructed Boolean functions having the best trade-offs between nonlinearity and resiliency. Boolean functions having the best trade-offs between nonlinearity and resiliency will be resistant to the cryptanalysis like linear cryptanalysis and correlation attack. But to resist the Boolean functions to the Differential cryptanalysis, Boolean functions should have optimal(low) value of autocorrelation. The Boolean functions having the best trade-offs between nonlinearity and autocorrelation will be resistant to the linear cryptanalysis and differential cryptanalysis. So, in this chapter, we propose an evolutionary multiobjective approach to construct balanced Boolean functions having the best trade-off between nonlinearity and autocorrelation. Many of the best such functions of small number of variables have been obtained by single objective two stage optimization method [18, 19]. Agguire et.al. [1] found such functions by two stage optimization method. Previous works aiming to generate highly non-linear balanced functions by search heuristics have used single objective optimization approaches targeting directly either the non-linearity with respect to the set of affine functions or Boolean functions with linear structures [18].

All the optimization works developed so far search the space of Boolean functions for particular properties. Aguirre et. al. [1] developed an evolutionary multiobjective approach, called multiobjective random bit climber(moRBC) and found balanced Boolean functions of similar characteristics satisfying multiple criteria. They showed that the multiobjective approach is an efficient alternative to single objective optimization approaches. Keeping this in the view, in this chapter, we have developed a new evolutionary multiobjective approach and constructed cryptographically strong Boolean functions based on two objectives nonlinearity and autocorrelation.

Section-wise the remaining chapter is arranged as follows: In Section 3.2, we have developed an evolutionary multiobjective approach to construct the desired Boolean functions. Section 3.3 gives results and discussions of our work. In Section 3.4, we

have concluded our work.

3.2 Proposed method for Construction of Balanced Boolean Functions Having the Best Trade-offs Between Nonlinearity and Autocorrelation

In this section, we will describe the proposed method to construct the balanced Boolean functions having best trade-offs between nonlinearity and autocorelation. The proposed method consists of
(i) Formulation of MOOP, (ii) Application of NSGA-II.

(i) **Formulation of MOOP :** We will use the criteria of nonlinearity, balancedness and autocorrelation to construct the desired Boolean functions.
We know that the bent functions have the maximum nonlinearity and for bent functions $|W_f(\lambda)| = 2^{n/2}$ for all $\lambda \in \mathbf{F}_2^n$. For balanced Boolean functions this ideal bound cannot be achieved but it does suggest that a objective function that seeks to minimize the spread of Walsh Hadamard values is well motivated. So, we take

$$f_1 = \sum_{\lambda \in \mathbf{F}_2^n} ||W_f(\lambda)| - 2^{n/2}|^R \tag{3.2.1}$$

as the first objective function, where R $\in [3, 10]$ is a constant.

The autocorrelation of a function f is given by

$$r_f(\lambda) = \sum_{x \in \mathbf{F}_2^n} (-1)^{f(x)+f(x+\lambda)} \tag{3.2.2}$$

and $r_f(0)$ is maximum.

So, we take

$$f_2 = \max_{\lambda \in \mathbf{F}_2^n} |r_f(\lambda)|, \tag{3.2.3}$$

as the second objective function, where $\lambda \in \mathbf{F}_2^n$ and $\lambda \neq zero$.

Now, for all $\lambda \in \mathbf{F}_2^n$,

$$f_1 = \sum_{\lambda \in \mathbf{F}_2^n} ||W_f(\lambda)| - 2^{n/2}|^R,$$

$$= \sum_{\lambda \in \mathbf{F}_2^n} ||\sum_{x \in \mathbf{F}_2^n} (-1)^{f(x)+\lambda.x}| - 2^{n/2}|^R,$$

$$= \sum_{\lambda \in \mathbf{F}_2^n} ||\sum_{x \in \mathbf{F}_2^n} (-1)^{f(x)}.(-1)^{\lambda.x}| - 2^{n/2}|^R,$$

$$= \sum_{\lambda \in \mathbf{F}_2^n} ||\sum_{x \in \mathbf{F}_2^n} (1 - 2.f(x)).a_{\lambda.x}| - 2^{n/2}|^R,$$

$$f_1 = \sum_{\lambda \in \mathbf{F}_2^n} ||M_\lambda - \sum_{x \in \mathbf{F}_2^n} 2f(x).a_{\lambda.x}| - 2^{n/2}|^R \tag{3.2.4}$$

where

$$M_\lambda = \sum_{x \in \mathbf{F}_2^n} a_{\lambda.x} \tag{3.2.5}$$

$$a_{\lambda.x} = (-1)^{\lambda.x} \tag{3.2.6}$$

$f(x)'s$ are binary variables(Boolean function) to be determined. We have taken R = 3 for all values of m.

Similarly, for all $\lambda \in \mathbf{F}_2^n$,

$$f_2 = \max_{\lambda} |\sum_{x \in \mathbf{F}_2^n} (-1)^{f(x)+f(x+\lambda)}| \tag{3.2.7}$$

So, our problem as an MOOP is given below:

$$\left.\begin{array}{l} \min F = (f_1, f_2) \\ subject \quad to \\ \sum_{x \in \mathbf{F}_2^n} f(x) = 2^{n-1} \end{array}\right\} \tag{3.2.8}$$

($\sum_{x \in \mathbf{F}_2^n} f(x)$ should be equal to 2^{n-1} for balanced function).

(ii) **Application of NSGA-II**: Now we apply NSGA-II to the MOOP given in (3.2.8) and get the results(Boolean functions). The obtained results are given in Table 3.1 and discussed in Section 3.4. The parameters that we have used in NSGA-II are given in Tables 3.2, 3.3 and 3.4,

No. of variables	Previous results	Results by the proposed method
4	Nl(f) = 4 and $r_f = 8$	Nl(f) = 4 and $r_f = 4$.
5	Nl(f) = 12 and $r_f = 8$	Nl(f) = 12 and $r_f = 8$.
6	Nl(f) = 26 and $r_f = 16$	Nl(f) = 26 and $r_f = 8$.

Table 3.1: Comparison of results

number of generation	200
population size	100
probability of crossover	0.7
probability of mutation	0.1
seed random number	0.9876
number of bits for binary variables (have been taken equal bits for all variables)	1
number of objective functions	2
number of constraints	2

Table 3.2: Parameters used in NSGA-II for 4 variables.

number of generation	500
population size	200
probability of crossover	0.9
probability of mutation	0.01
seed random number	0.9876
number of bits for binary variables (have been taken equal bits for all variables)	1
number of objective functions	2
number of constraints	2

Table 3.3: Parameters used in NSGA-II for 5 variables.

number of generation	1000
population size	500
probability of crossover	0.7
probability of mutation	0.1
seed random number	0.9876
number of bits for binary variables (have been taken equal bits for all variables)	1
number of objective functions	2
number of constraints	2

Table 3.4: Parameters used in NSGA-II for 6 variables.

3.3 Results and Discussion

By applying the proposed method on the basis of the criteria balancedness, nonlinearity and autocorelation, we got the desired Boolean functions. These functions are balanced and have the best trade-offs between nonlinearity and autocorrelation. Applying our method, we have constructed such functions of 4, 5 and 6 variables as shown in Table 3.1. The parameters taken to construct functions of 4, 5 and 6 variables are listed in Tables 3.2, 3.3, 3.4. Table 3.1 shows that our method gives at least as better results as are available in the literature [1, 18, 19].

3.4 Conclusion.

In this chapter, we have developed a new evolutionary multiobjective approach to construct desired Boolean functions. By applying our method, we got at least as better results(Table 3.1) as given in literature [1, 18, 19]. Thus our approach to construct desired Boolean functions is at least as better as the approaches available in the liter-

ature.

Chapter 4

A New Evolutionary Multiobjective Approach to Construct Balanced Boolean Functions Based on Three Objectives-Nonlinearity, Resiliency and Autocorrelation

In Chapters 2 and 3, we developed evolutionary multiobjective approaches based on two objectives nonlinearity and resiliency, and nonlinearity and autocorrelation respectively and constructed the balanced Boolean functions having the best trade-offs between nonlinearity and resiliency, and nonlinearity and autocorrelation. In this chapter, we propose to develop an evolutionary multiobjective approach based on three objectives nonlinearity, resiliency and autocorrelation, and construct balanced Boolean functions having the best trade-offs among nonlinearity, autocorrelation and resiliency for 4, 5 and 6 variables.

4.1 Introduction

Nonlinearity, balancedness, resiliency, autocorrelation etc. are the main properties of Boolean functions. There are some common cryptanalysis like linear cryptanalysis, differential cryptanalysis, correlation attack etc on cipher system. To resist a Boolean function from them, it should have optimal properties.

In the Chapters 2 and 3, we have constructed Boolean functions having the best trade-offs between nonlinearity and resiliency, and nonlinearity and autocorrelation respectively. Boolean functions having the best trade-offs between nonlinearity and resiliency will be resistant to the cryptanalysis like linear cryptanalysis and correlation attack, and Boolean functions having the best trade-off between nonlinearity and autocorrelation will be resistant to the linear cryptanalysis and differential cryptanalysis. But to resist all above attacks simultaneously Boolean functions should be balanced and have the best trade-offs among nonlinearity, resiliency and autocorrelation. Nonlinearity and resiliency are very important properties to resist cryptanalysis like linear cryptanalysis and correlation attack. But to resist the attack like differential cryptanalysis Boolean functions should have high nonlinearity and low autocorrelation. So, in this chapter we propose to construct balanced Boolean functions having the best trade-off among nonlinearity, resiliency and autocorrelation. Many of the best such functions of small number of variables have been obtained by single objective two stage optimization method [18, 19]. Agguire et.al. [1] found such functions for small number of variables by two stage optimization method. Previous works aiming to generate highly non-linear balanced functions by search heuristics have used single objective optimization approaches, targeting directly either the non-linearity with respect to the set of affine functions or the non-linearity with respect to the set of Boolean functions with linear structures [18]. We call these methods direct-single objective approaches. Clark and Jacob [18], Clark et al. [19] have proposed a two-stage optimization approach. In the first stage, an annealing-based method is developed to evolve Boolean

functions restricting the spread of Walsh values. In the second stage, hill climbing with respect to nonlinearity or autocorrelation is applied to the best function obtained in the first stage. This method also focuses on single objective optimization, although at the end of the run other properties of the function are also measured. Best results in terms of individual objectives have been achieved by the two-stage optimization approach than by the direct single objective methods. The two-stage approach is an important improvement towards the automatic generation of highly non-linear Boolean functions with cryptographic applications. However, it suffers from some drawbacks and limitations. First, the objective function used during the first stage introduces two parameters and needs to be fine tuned in order to produce good results. Second, the method offers very limited function diversity. Third, the method is not scalable in the sense that it does not support the addition of criteria that could be related to non-linearity or to other cryptographic characteristics. Boolean functions are the basic components in cryptography. In Chapter 1, we have outlined a number of cryptographic properties of Boolean functions. Among them high nonlinearity, balancedness, high algebraic degree and low autocorrelation are important from cryptographic point of view. These properties contribute to the security system and provide resistance to it. There are different types of cryptanalytic attacks against Boolean functions like differential cryptanalysis [7], linear cryptanalysis [71], correlation attack [86,87] etc. Boolean functions are most vulnerable to attack if exploitable weaknesses in their components exist. For this reason, it is important to be able to ensure that any Boolean function exhibits the appropriate combination and measures of robust cryptographic properties. To resist differential cryptanalysis Boolean function should have high nonlinearity and low autocorrelation, to resist linear cryptanalysis nonlinearity should be high, to resist correlation attack correlation immunity should be high and etc. [12].

The trade-offs between some of these criteria have been studied by some authors [62, 70, 83]. Some works have been done to combine algebraic construction with

deterministic computer search methods [62, 83] and some authors have attempted using search heuristics such as genetic algorithms, hill climbering, and simulated annealing to generate Boolean functions [18, 69, 70]. A clear trade-off has been shown for correlation immunity, algebraic degree and nonlinearity [19]. The design of suitable functions has received significant attention from cryptographers for decades. Metaheuristic search(particularly hill climbing, genetic algorithms and simulated annealing) has emerged as a potentially very powerful tool for the design of such functions. However, generating Boolean functions purely by constructive algebraic methods becomes increasingly difficult as the number of criteria to be satisfied is augmented. The more criteria to be taken into account, the more difficult is to generate Boolean functions satisfying those properties purely by constructive algebraic means. Many of the best functions of small numbers of variables have been obtained by single objective two stage optimization method [19]. All the optimization works so far have searched the space of Boolean functions with particular properties. Aguirre et. al. [1] have developed a multiobjective random bit climber(moRBC) algorithm. This algorithm uses elitism and bias selection by Pareto dominance to search on non-linearity criteria and generate several balanced functions of similar characteristic by using moRBC algorithms. They observed that, overall, the performance by the multi-objective approach seems promising in the sense that the multiobjective approach is more efficient than a two-stage optimization requiring much fewer runs and evaluations to generate functions with high non-linearity approach. Keeping this in the view, we have developed an evolutionary multiobjective approach and constructed balanced Boolean functions having the best trade-off among nonlinearity, autocorrelation and resiliency for 4, 5, and 6 variables.

Section-wise the remaining chapter is arranged as follows: In Section 4.2, we have developed an evolutionary multiobjective approach to construct the desired Boolean functions. Section 4.3 gives results and discussions of our work. In Section 4.4, we

have concluded our work.

4.2 The Proposed Method for Construction of Boolean Functions

In this section, we will describe the proposed method to construct the desired Boolean functions. The proposed method consists of
(i) Formulation of MOOP,
(ii) Application of NSGA-II.

(i) **Formulation of MOOP :** We will use the criteria of nonlinearity, resiliency and autocorrelation to construct the desired Boolean functions.

We know that the bent functions have the maximum nonlinearity and for bent functions, $|W_f(\lambda)| = 2^{n/2}$ for all $\lambda \in \mathbf{F}_2^n$. For balanced Boolean functions this ideal bound cannot be achieved but it does suggest that an objective function that seeks to minimize the spread of Walsh Hadamard values is well motivated. So, we take

$$f_1 = \sum_{\lambda} ||W_f(\lambda)| - 2^{n/2}|^R \tag{4.2.1}$$

as the first objective function, where R $\in [3, 10]$ is a constant.

If a function has resiliency m, then $W_f(\lambda)$ will be zero for all those $\lambda \in \mathbf{F}_2^n$ for which $w_H(\lambda) \leq m$. Hence, we take

$$f_2 = \sum_{\lambda} |W_f(\lambda)| \tag{4.2.2}$$

as the second objective function, where $w_H(\lambda) \leq m$ for $\lambda \in \mathbf{F}_2^n$.

The autocorrelation of a function f is given by

$$r_f(\lambda) = \sum_{x \in \mathbf{F}_2^n} (-1)^{f(x)+f(x+\lambda)}$$

and $r_f(0)$ is maximum,

So, we take

$$f_3 = \max_{\lambda} |r_f(\lambda)| \tag{4.2.3}$$

as the third objective function, where $\lambda \in \mathbf{F}_2^n$ and $\lambda \neq zero$.

Now, for all those $\lambda \in \mathbf{F}_2^n$ for which $w_H(\lambda) > m$,

$$\begin{aligned} f_1 &= \sum_{\lambda} ||W_f(\lambda)| - 2^{n/2}|^R, \\ &= \sum_{\lambda} |\sum_{x \in \mathbf{F}_2^n} (-1)^{f(x)+\lambda.x}| - 2^{n/2}|^R, \\ &= \sum_{\lambda} |\sum_{x \in \mathbf{F}_2^n} (-1)^{f(x)}.(-1)^{\lambda.x}| - 2^{n/2}|^R, \\ &= \sum_{\lambda} |\sum_{x \in \mathbf{F}_2^n} (1 - 2.f(x)).a_{\lambda.x}| - 2^{n/2}|^R, \\ &= \sum_{\lambda} ||M_\lambda - \sum_{x \in \mathbf{F}_2^n} 2f(x).a_{\lambda.x}| - 2^{n/2}|^R \end{aligned} \tag{4.2.4}$$

where

$$M_\lambda = \sum_{x \in \mathbf{F}_2^n} a_{\lambda.x} \tag{4.2.5}$$

$$a_{\lambda.x} = (-1)^{\lambda.x} \tag{4.2.6}$$

$f(x)'s$ are binary variables(Boolean function) to be determined. Let us take R = 3 for all values of m.

For all those $\lambda \in \mathbf{F}_2^n$ for which $w_H(\lambda) \leq m$,

$$f_2 = \sum_\lambda |W_f(\lambda)|$$

$$= \sum_\lambda |\sum_{x \in \mathbf{F}_2^n} (-1)^{f(x)+\lambda.x}|,$$

$$= \sum_\lambda |\sum_{x \in \mathbf{F}_2^n} (-1)^{f(x)}.(-1)^{\lambda.x}|,$$

$$= \sum_\lambda |\sum_{x \in \mathbf{F}_2^n} (1 - 2.f(x)).a_{\lambda.x}|,$$

$$= \sum_\lambda |\sum_{x \in \mathbf{F}_2^n} 2f(x).a_{\lambda.x} - M_\lambda| \tag{4.2.7}$$

and for all $\lambda \in \mathbf{F}_2^n$,

$$f_3 = \max_\lambda |r_f(\lambda)|$$

$$= \max_\lambda |\sum_{x \in \mathbf{F}_2^n} (-1)^{f(x)+f(x+\lambda)}| \tag{4.2.8}$$

So, our problem as an MOOP is given below:

$$\left.\begin{array}{l} Min(f_1, f_2, f_3), \\ s.t. \\ \sum_{x\in\mathbb{F}_2^n} f(x) = 2^{n-1} \end{array}\right\} \tag{4.2.9}$$

($\sum_{x\in\mathbf{F}_2^n} f(x)$ should be equal to 2^{n-1} for balanced function).

(ii) **Application of NSGA-II**: Now we apply NSGA-II to the MOOP given in(3.2.9) and get the results(Boolean functions). The obtained results are described in Section 3.3. The parameters that we have used in NSGA-II are given in Tables 3.2, 3.3 and 3.4.

4.3 Results and Discussion

After applying the method developed in Section 4.2, we got the desired Boolean functions(Table 4.1). These functions are balanced and have the best trade-off among nonlinearity, resiliency and autocorrelation. Applying our method, we have constructed such functions of 4, 5 and 6 variables as shown in Table 4.1. The parameters taken to construct functions of 4, 5 and 6 variables are listed in Table 4.2, Table 4.3 and Table 4.4 respectively. Table 4.1 shows that our method gives at least as better results as are given in literature [1, 18, 19].

4.4 Conclusion.

In this chapter, we have developed a method to construct desired Boolean functions of 4, 5 and 6 variables. By applying our method, we got at least as better results (Table

No. of variables	Previous results	Results by the proposed method
4	Nl(f) = 4, $r_f = 8$ and resiliency=1	Nl(f) = 4 and $r_f = 4$ and resiliency=
5	Nl(f) = 12 and $r_f = 8$ and resiliency=1	Nl(f) = 12 and $r_f = 8$ and resiliency=
6	Nl(f) = 26 and $r_f = 16$ and resiliency=1	Nl(f) = 26 and $r_f = 8$ and resiliency=

Table 4.1: Comparison of results

number of generation	200
population size	200
probability of crossover	0.7
probability of mutation	0.1
seed random number	0.9876
number of bits for binary variables (have been taken equal bits for all variables)	1
number of objective functions	3
number of constraints	2

Table 4.2: Parameters used in NSGA-II for 4 variables.

number of generation	1000
population size	500
probability of crossover	0.7
probability of mutation	0.1
seed random number	0.9876
number of bits for binary variables (have been taken equal bits for all variables)	1
number of objective functions	3
number of constraints	2

Table 4.3: Parameters used in NSGA-II for 5 variables.

4.1) in comparison with the results given in literature [1, 18, 19]. Thus our method to construct desired Boolean functions is at least as better as the methods available in the literature.

number of generation	2000
population size	1000
probability of crossover	0.7
probability of mutation	0.1
seed random number	0.9876
number of bits for binary variables (have been taken equal bits for all variables)	1
number of objective functions	3
number of constraints	2

Table 4.4: Parameters used in NSGA-II for 6 variables.

Chapter 5

An Evolutionary Multiobjective Approach with Biasedness to Construct Desired Boolean Functions

Many desirable properties are known for Cryptographically strong Boolean functions. It is difficult task to get optimal trade-off among such properties. In this chapter, we have focused on nonlinearity, balancedness and autocorrelation, and explored an evolutionary multiobjective approach with biasedness to construct balanced Boolean functions having the best trade-offs among them. By including biasedness, we can get desired set of solutions instead of all solutions. Biasedness diverts the solutions towards the desired region. So, we get only the solutions that are desired.

5.1 Introduction

Many real-world optimization problems are naturally posed as multi-objective optimization problems. They have been suitably converted into single-objective optimiza-

tion problems and solved. The basic difficulty arises due to the nature of the optimality conditions for multiple objectives. In the presence of multiple and conflicting objectives, the resulting optimization problem gives rise to a set of optimal solutions, instead of one optimal solution. Multiple optimal solutions exist because no one solution can be optimal for multiple conflicting objectives. Let us illustrate this concept through an example. If cost and reliability are two objectives in a design optimization, it is clear that a minimum cost solution is usually not maximally reliable and a maximally reliable solution is not often the cheapest. In such a scenario, none of these two extreme solutions (the cheapest and the most reliable solutions) can be declared as an absolute optimum corresponding to both objectives of design. In the parlance of multi-criteria decision-making, both these solutions are optimal in some sense or they are Pareto-optimal. In fact, there exist many other solutions in the search space which are also Pareto-optimal. Since none of these solutions can be said to be an absolute optimum, the onus on the part of the user is then to first find as many such solutions as possible. Once multiple such solutions are found, usually, a higher-level decision-making strategy is adopted to choose a solution from the set of obtained Pareto-optimal solutions.

Multi-objective evolutionary algorithms(MOEAs) have found increasing attention due to the ability to find multiple Pareto-optimal solutions in single simulation run. But often, users need to impose a particular order of priority to objectives. In this chapter, we describe the technique to identify a preferred or a compromised solution, and finally suggest a biased sharing technique [20] which can be used during the optimization phase to find a biased distribution of Pareto-optimal solutions in the region of interest. By including biasedness we can get desired set of solutions instead of all solutions. Bieasedness diverts the solutions towards the desired region. So, we get only the desired solutions. If the Pareto optimal solutions are too many, their analysis to reach the final decision is quite a challenging and burdensome process for the decision maker(DM). In addition, in a particular problem, the user may not be interested in the complete Pareto set, instead, the user may be interested in a certain region of the Pareto set. Such a biasedness can arise if all objectives are not of equal importance to the user.

Finding a preferred distribution in the region of interest is more practical and less subjective than finding one biased solution in the region of interest. Keeping this in the view, we have developed an evolutionary approach with biasedness and constructed cryptographically strong Boolean functions, that is, the balanced Boolean functions having the best trade-offs between nonlinearity and autocorrelation.

Section-wise chapter is arranged as follows: Section 5.2 gives a brief description of biased sharing technique and reason of using it. In Section 5.3, we have developed an evolutionary multiobjective approach with biasedness and constructed the desired Boolean functions. Section 5.4 gives results and discussions of our work. In Section 5.5, we have concluded our work.

5.2 Biasedness sharing technique

Here, we discuss a sharing approach [20] which uses a biased distance metric. In calculating the distance metric in the fitness-space sharing, the following normalized distance metric d(i,j) between two solutions i and j is suggested as

$$d(i,j) = [\sum_{k=1}^{m}(f_k^{(i)} - f_k^{(j)})^2/(f_k^{(max)} - f_k^{(min)})^2]^{1/2} \tag{5.2.1}$$

This distance metric is nothing but the normalized Euclidian distance between two objective vectors. In the proposed biased sharing approach, an unequal weightage is given to each objective in computing the Eucledian distance. For example, if $\omega_k \in (0,1)$ is the weight assigned to the kth objective function, then the normalized ϖ_k is calculated as follows:

$$\varpi_k = (1-\omega_k)/\sum_{r=1}^{m}(1-\omega_r) \tag{5.2.2}$$

k=1,2,3,...m.

and modified distance metric is computed as follows:

$$d(i,j) = (\sum_{k=1}^{m} \varpi_k (f_k^{(i)} - f_k^{(j)})^2 / (f_k^{(max)} - f_k^{(min)})^2)^{1/2} \tag{5.2.3}$$

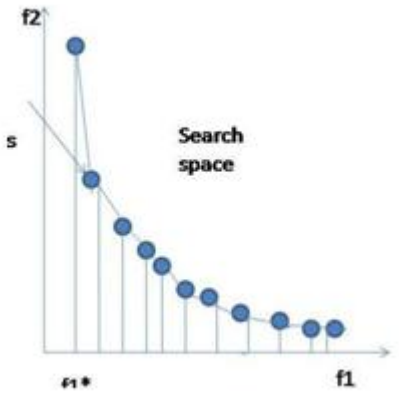

Figure 5.1: Illustration of biasedness sharing

The fitness-based sharing can then be used with this distance metric.
The highest priority objective always gets the highest weightage. For convex Pareto-optimal regions, a higher weight ω for an objective function will produce more dense solutions near the individual optimum. Figure 5.1 explains this fact. For a two-objective optimization problem, if an extreme case of $\omega_1 = 0$ and $\omega_2 = 1$ is used, the corresponding ϖ is as follows: $\varpi_1 = 0$ and $\varpi_2 = 1$. Since the effect of f_2 is absent in calculating the distance metric, equal number of solutions are expected to be created in each equal partition of the f_1 search in the space. Thus, the density of Pareto optimal solutions in the partition closer to the individual best (here f_1^*) will be less. For non-convex Pareto-optimal region, $\varpi_k = \omega_k / \max_{i=1}^{M} \omega_k$.
We introduce weights into multiobjective optimization if we solve it by single objective optimization technique. But if we apply multiobjective optimization technique, there is no need to convert multiobjectives into single objective. In this case we construct con-

straint(s) to give weightage to the particular objective(s) to support the corresponding objective(s).

5.2.1 Why biasedness sharing technique

In Chapter 2, we took the same objective functions and developed the method without using the concept of biasedness. We got the desired Boolean functions of 4, 5 and 6 variables. The developed method(Section 4 [37]) failed for variables more than 6. So, in this chapter, we include biasedness in the proposed method. By using the proposed method, we will construct the desired Boolean functions of 7 and 8 variables.

5.3 The Proposed Method

In this section, we will describe the proposed method to construct the desired Boolean functions. The proposed method consists of (i) Formulation of MOOP with biasedness and (ii) Application of NSGA-II.

(i) **Formulation of MOOP with biasedness :** We will use the criteria of nonlinearity, balancedness and autocorrelation to construct the desired Boolean functions. Nonlinearity of a Boolean function is given by

$$Nl = 2^{(n-1)} - (1/2) \max_{\lambda} W_f(\lambda).$$

We desire Nl to take the value R(say). So, we take

$$f_1 = R - Nl \tag{5.3.1}$$

as the first objective function. The maximum value of Nl is 56 for 7 variables and 120 for 8 variables. That is why, in the present discussion, we have taken R=56 for 7 variables and 120 for 8 variables.

The autocorrelation of a function f is given by

$$r_f(\lambda) = \sum_{x \in \mathbb{F}_2^n} (-1)^{f(x)+f(x+\lambda)},$$

and $r_f(0)$ is maximum.

So, we take

$$f_2 = \max_{\lambda} |r_f(\lambda)| \tag{5.3.2}$$

as the second objective function, where $\lambda \in \mathbb{F}_2^n$ and $\lambda \neq zero$.

Now, $\forall \lambda \in \mathbb{F}_2^n$,

$$f_1 = R - Nl$$

$$= R - (2^{(n-1)} - (1/2) \max_{\lambda} W_f(\lambda))$$

$$= R - (2^{(n-1)} - (1/2) \max_{\lambda} \sum_{x \in \mathbb{F}_2^n} (-1)^{f(x)+\lambda.x}) \tag{5.3.3}$$

Similarly, $\forall \lambda \in \mathbb{F}_2^n$,

$$f_2 = \max_{\lambda} |\sum_{x \in \mathbb{F}_2^n} (-1)^{f(x)+f(x+\lambda)}|. \tag{5.3.4}$$

So, our problem as an MOOP with biasedness is given below:

$$\left.\begin{array}{l} \min F = (f_1, f_2) \\ subject \quad to \\ \sum_{x \in \mathbb{F}_2^n} f(x) = 2^{n-1}, \\ Nl = R. \end{array}\right\} \tag{5.3.5}$$

$\sum_{x \in \mathbb{F}_2^n} f(x)$ should be equal to 2^{n-1} for balanced function. To give more weightage to the first objective, we take second constraint as $Nl = R$. (The concept of biasedness sharing technique in MOOP)

(ii) **Application of NSGA-II**: Now we apply NSGA-II to the MOOP with biasedness given in (5.3.5) and get the results(Boolean functions). The obtained results are described in Section 5.4. The parameters that we have used in NSGA-II are given

in Tables 5.2 and 5.3.

5.4 Results and Discussion

By applying the method developed in Section 5.3, we got the desired Boolean functions. These functions are balanced and have the best trade-offs between nonlinearity and autocorrelation. Applying our method, we have constructed such functions of 7 and 8 variables as shown in Table 5.1. The parameters taken to construct functions are listed in Table 5.2 and Table 5.3 respectively. Table 5.1 shows that our method gives at least as better results as are known in literature [1, 18, 19].

No. of variables	Previous results	Results by proposed method
7	[1]Nl(f) = 54, $r_f = 16$ [1]Nl(f) = 56, $r_f = 16$ [18, 19]Nl(f) = 56, $r_f = 16$	Nl(f) = 54 and $r_f = 12$.
8	[1]Nl(f) = 112 and $r_f = 24$ [1, 18, 19]Nl(f) = 116 and $r_f = 24$	Nl(f) = 116 and $r_f = 20$.

Table 5.1: Comparison of results

number of generation	3000
population size	2000
probability of crossover	0.7
probability of mutation	0.1
seed random number	0.9876
number of bits for binary variables (have been taken equal bits for all variables)	1
number of objective functions	2
number of constraints	3

Table 5.2: Parameters used in NSGA-II for 7 variables.

number of generation	6000
population size	3000
probability of crossover	0..7
probability of mutation	0.01
seed random number	0.9876
number of bits for binary variables (have been taken equal bits for all variables)	1
number of objective functions	2
number of constraints	3

Table 5.3: Parameters used in NSGA-II for 8 variables.

5.5 Conclusion.

In this chapter, we have developed an evolutionary multiobjective method with biasedness to construct the desired Boolean functions. By applying our method, we got at least as better results(Table 5.1) in comparison with the results given in literature [1, 18, 19]. Thus our method to construct the desired Boolean functions is at least as better as the methods available in the literature.

Chapter 6

A new approach to solve a general system of linear inequalities based on NSGA-II

A large variety of real life problems in practice are formulated as integer optimization problems. A system of linear inequalities can be solved by conjugate gradient algorithm for linear constraints. But this algorithm is applicable only to inequalities of the type $\sum_{j=1}^{n} a_{ij}x_j > 0$, i=1,2,3,...,m.

The construction of exact algorithms designed to solve integer problems has been considerably improved during the last 50 years. But very often they can not be applied to solve practical problems of middle and large size because of their excessive runtimes and memory requirements. Some theoretical and algorithmic investigations are devoted to combinatorial or binary problems. In general, the solution of the integer problem remains considerably harder. The hybrid methods are promising tools, since they combine the best features of different methods (exact techniques or metaheuristics) in a complementary mode. Since obtaining a good feasible solution in reasonable time is completely satisfactory for many practical problems, the development of heuristic algorithms having polynomial computational complexity, is still a problem of the

present day. Many large size real problems can not be solved by exact algorithms due to their exponential computational complexity. In such case approximate polynomial time algorithms are better to use.

A set of inequalities can be solved by graphical method or simplex method. But these methods have some drawbacks. Graphical method is conveniently applicable to a system containing two variables; while simplex method gives one solution at a time. So, in this chapter, a new approach to solve a general system of linear inequalities based on NSGA-II is developed. NSGA-II is a fast and elitist approach. The main advantage of using NSGA-II is that it provides multiple solutions(Pareto-optimal fronts) in one single simulation run.

6.1 Introduction

A system of linear inequalities occurs in many real life problems. A wide variety of real life problems in logistics, economics, social science and politics can be reformulated as linear inequalities. The combinatorial problems like the Knapsack-capital budgeting problem, warehouse location problem, traveling salesman problem, decreasing cost and machinery selection problem, network and graph problems such as maximum flow problems, set covering problems, matching problems, weighted matching problems, spanning tree problems and many scheduling problems can also be solved as linear inequalities [15, 44, 58, 77]. To find the solutions for this class of problems requires use of considerable computational resources. The development of efficient hybrid methods, combining in a suitable way the best features of different approaches(exact or approximate) is the actual direction, in which many researcher devote their efforts to solve successfully various hard practical problems.

A system of linear inequalities can be solved by converting it into optimization problems [35]. There are mainly two types of methods to solve linear inequalities.

1. Exact methods [35]

2. Approximate methods [20, 21, 39, 53]

Exact methods: There are many exact methods in literature to solve linear inequalities. The development of exact optimization methods for linear inequalities specially linear integer problems during the last 50 years has been very successful. There are mainly three approaches for solving linear integer problems.

(a) Cutting-planes algorithms based on polyhedral combinatorics,

(b) Enumeration approaches: Branch and Bound(BB), Branch and cut(BC) and branch and Price(BP) methods,

(c). Relaxation and Decomposition(RD) techniques.

Ben-Isreal and Charnes [6] proposed a primal cutting-plane algorithm and Young [102] proposed a finitely convergent primal cutting-plane algorithm for general integer programs. Later on Glover [38] and Young [102] gave simplified versions. Because of poor computational experience, this line of research has been very inactive. An exception is a primal cutting-plane algorithm for the traveling salesman problem. Although this algorithm has been moderately successful, it seems to be inferior to a fractional cutting-plane algorithm for the traveling salesman problem. Another strategy for cutting-plane algorithms is to maintain integrality and dual feasibility, and then to use cuts to obtain primal feasibility.

Enumeration approaches are known under different names. The most popular of them are BB, implicit enumeration, and divide and conquer [75]. The explicit enumeration is the simplest approach for solving a pure integer programming problem by means of enumeration of all possibilities which are finite in number. However, due to the combinatorial explosion of number of these possibilities resulting from the parameter **size** only instances having relative small size could be solved by such an approach within a reasonable computational time limit. The BB method developed by Land and Doing [57] consist of branches and bounds. The **branching** refers to the enumeration part of the solution technique and bounding refers to the fathoming of possible solutions

by comparison to a known upper or lower bound on the solution value. The commercial BB codes usually relax the problem by dropping the integrality conditions and solve the resultant continuous linear programming problem over the constraint system. The bounds obtained from the LP-relaxations are often weak which may cause standard BB algorithms to fail in practice. Later on some methods like BC [44], BP [2], Column generation concept, LP relaxation, Combinatorial relaxation [27, 28] and Lagrangian relaxation [5, 40] were developed to improve BB algorithms.

Dahl [24] has given Fourier-Motzkin Elimination method to solve the linear inequalites. Fourier-Motzkin elimination is a classical method for solving linear inequalities in which one variable is eliminated in each iteration. Robert Orsi et. al. gave Newton-Like Method for solving Rank constrained linear matrix inequalities [76]. Local quadratic convergence of the Newton-like algorithm is not a priori guaranteed. Korovin et.al [55] have given a method to solve the set of linear inequalities over the rational and real numbers and presented experimental evaluation. The method and heuristics are evaluated against various benchmarks and compared to other methods, such as the Fourier-Motzkin elimination method and the simplex method.

All the above methods to solve the linear inequalities is applicable only for small size. As the number of variables increases, their complexity increases very high and less chance to get optimal solutions. For problem with large variables Approximate methods are good.

Approximate methods: A huge number of approximate algorithms has been created for the solution of large real life LIP optimization problems without any guarantee for optimality of the final solution [88]. Many local search based metaheuristics have been developed to avoid the trap of local optimality and to find a global optimal solution [79]. It was proven that they are highly useful in practice. The development of approximate algorithms is important in the sense that they terminate their performance using a polynomial number of standard mathematical operations.

Linear inequalities can also be solved by graphical method or simplex method [90].

But graphical method is conveniently applicable to a system containing two variables and simplex method gives one solution at a time. Also, the simplex method may give only one solution out of many solutions to the system. Nagaraja and krishna [74] developed a conjugate gradient algorithm for linear constraints to solve a set of linear inequalities. But Warmack and Gonzaloz [99] developed an algorithm for the optimal solutions of consistent and inconsistent linear inequalities. The algorithm is developed as a non- enumerative search procedure. The set of inequalities considered are $\sum_{j=1}^{n} c_{ij}x_j > 0$, $i = 1, 2, 3...m$. Clark and Gonzalez [17] developed an algorithm to find the optimal solutions of a system of linear inequalities and applied to pattern recognition. Forsgren and Murray [34] discussed Newton method of the line search type for large scale minimization subject to linear inequality constraints. They have also given the convergence analysis.

A number of multiobjective evolutionary algorithms(MOEAs) like MOGA-III, SPEA2, NSGA, NPGA and MOMGA([30], [33], [92], [21], [29], [89]) have been suggested to solve multi-objective optimization problems(MOOPs). NSGA-II, in most of the problems, finds much better convergence near the true Pareto-optimal front compared to other elitist MOEAs like PAES [54] and SPEA [29] that pay special attention to creating diverse Pareto-optimal fronts. Simulation results of constrained NSGA-II show much better performance of NSGA-II [21].

So, in this chapter, we have developed a methodology to solve a general system of linear inequalities based on NSGA-II. The developed method gives better spread of solutions. The presence of multi-objectives in problem gives rise to a set of efficient solutions(Pareto-optimal solutions) instead of single optimal solution. In the absence of any further information one of these Pareto-optimal solutions cannot be said to be better than the other [21].

The rest of the chapter is organized as follows: In Section 6.2, we have described a general system of linear inequalities(with examples in Subsection 6.2.1) and the

algorithm developed to solve it based on NSGA-II. Section 6.3 gives results with its discussion and Section 6.4 gives conclusion of our work.

6.2 The method developed to solve a general system of linear inequalities

Let a general the system of linear inequalities to be solved be

$$a_i \leq \sum_j c_{ij} x_j \leq b_i \tag{6.2.1}$$

where i= 1,2,3,...,m; j= 1,2,3,...,n; $x_j \in \mathbf{B}$, a bounded set(continuous or discrete), c_{ij} are real numbers, and all a_i and b_i are nonzero real numbers.

step 1. To solve (6.2.1), let us define

$$f_i = \sum_j c_{ij} x_j \tag{6.2.2}$$

$$G_i = \frac{f_i - a_i}{a_i} \tag{6.2.3}$$

and

$$H_i = \frac{b_i - f_i}{b_i} \tag{6.2.4}$$

step 2. Now, we formulate the MOOP as given below:

$$\left.\begin{array}{l} Min(G_1, G_2, G_3, ..., G_m, H_1, H_2, H_3, ..., H_m), \\ s.t. \\ f_i - a_i \geq 0 \\ b_i - f_i \geq 0 \\ x_j \in \mathbf{B} \\ i = 1, 2, 3, ..., m, j = 1, 2, 3, ..., n. \end{array}\right\} \tag{6.2.5}$$

step 3. Now, we apply NSGA-II to the MOOP given in (6.2.5).

6.2.1 Illustration

We illustrate the method developed in Section 6.2 by the following numerical examples.

Example 1: Let us apply the method developed in Section 6.2 to the following system of linear inequalities to find all possible solutions.

$$\left.\begin{array}{lll} 14 & \leq 3x_1 + 15x_2 + 0x_3 + 0x_4 + 0x_5 + 0x_6 + 0x_7 + 0x_8 + 0x_9 + 0x_{10} + 15x_{11} & \leq 20 \\ 10 & \leq 1x_1 + 1x_2 + 2x_3 + 4x_4 + 2x_5 + 2x_6 + 4x_7 + 4x_8 + 6x_9 + 4x_{10} + 3x_{11} & \leq 16 \\ 9 & \leq 0x_1 + 2x_2 + 4x_3 + 1x_4 + 5x_5 + 3x_6 + 0x_7 + 4x_8 + 4x_9 + 5x_{10} + 5x_{11} & \leq 20 \\ 15 & \leq 0x_1 + 4x_2 + 1x_3 + 2x_4 + 6x_5 + 4x_6 + 5x_7 + 1x_8 + 5x_9 + 2x_{10} + 3x_{11} & \leq 18 \\ 12 & \leq 0x_1 + 2x_2 + 5x_3 + 6x_4 + 4x_5 + 2x_6 + 3x_7 + 3x_8 + 3x_9 + 0x_{10} + 5x_{11} & \leq 20 \\ 13 & \leq 0x_1 + 2x_2 + 3x_3 + 4x_4 + 2x_5 + 4x_6 + 5x_7 + 1x_8 + 1x_9 + 6x_{10} + 5x_{11} & \leq 24 \\ 10 & \leq 0x_1 + 4x_2 + 0x_3 + 5x_4 + 3x_5 + 5x_6 + 2x_7 + 6x_8 + 2x_9 + 3x_{10} + 3x_{11} & \leq 20 \\ 15 & \leq 0x_1 + 4x_2 + 4x_3 + 1x_4 + 3x_5 + 1x_6 + 6x_7 + 6x_8 + 2x_9 + 3x_{10} + 3x_{11} & \leq 21 \\ 13 & \leq 0x_1 + 6x_2 + 4x_3 + 5x_4 + 3x_5 + 1x_6 + 2x_7 + 2x_8 + 4x_9 + 5x_{10} + 1x_{11} & \leq 20 \\ 9 & \leq 0x_1 + 4x_2 + 5x_3 + 2x_4 + 0x_5 + 6x_6 + 3x_7 + 3x_8 + 5x_9 + 2x_{10} + 3x_{11} & \leq 20 \end{array}\right\} \tag{6.2.6}$$

where $x_1, x_2, x_3, ..., x_{11} \in \mathbf{F_2}$ and $\mathbf{F_2} = \{0, 1\}$.

To solve the above system of inequalities by the method developed in Section 6.2,

we proceed as follows.

Let us define

$$
\begin{aligned}
f_1 &= 3x_1 + 15x_2 + 0x_3 + 0x_4 + 0x_5 + 0x_6 + 0x_7 + 0x_8 + 0x_9 + 0x_{10} + 15x_{11}, \\
f_2 &= x_1 + 1x_2 + 2x_3 + 4x_4 + 2x_5 + 2x_6 + 4x_7 + 4x_8 + 6x_9 + 4x_{10} + 3x_{11}, \\
f_3 &= 0x_1 + 2x_2 + 4x_3 + 1x_4 + 5x_5 + 3x_6 + 0x_7 + 4x_8 + 4x_9 + 5x_{10} + 5x_{11}, \\
f_4 &= 0x_1 + 4x_2 + 1x_3 + 2x_4 + 6x_5 + 4x_6 + 5x_7 + 1x_8 + 5x_9 + 2x_{10} + 3x_{11}, \\
f_5 &= 0x_1 + 2x_2 + 5x_3 + 6x_4 + 4x_5 + 2x_6 + 3x_7 + 3x_8 + 3x_9 + 0x_{10} + 5x_{11}, \\
f_6 &= 0x_1 + 2x_2 + 3x_3 + 4x_4 + 2x_5 + 4x_6 + 5x_7 + 1x_8 + 1x_9 + 6x_{10} + 5x_{11}, \\
f_7 &= 0x_1 + 4x_2 + 0x_3 + 5x_4 + 3x_5 + 5x_6 + 2x_7 + 6x_8 + 2x_9 + 3x_{10} + 3x_{11}, \\
f_8 &= 0x_1 + 4x_2 + 4x_3 + 1x_4 + 3x_5 + 1x_6 + 6x_7 + 6x_8 + 2x_9 + 3x_{10} + 3x_{11}, \\
f_9 &= 0x_1 + 6x_2 + 4x_3 + 5x_4 + 3x_5 + 1x_6 + 2x_7 + 2x_8 + 4x_9 + 5x_{10} + 1x_{11}, \\
f_{10} &= 0x_1 + 4x_2 + 5x_3 + 2x_4 + 0x_5 + 6x_6 + 3x_7 + 3x_8 + 5x_9 + 2x_{10} + 3x_{11}.
\end{aligned}
$$

We have

a_1=14, a_2=10, a_3=9, a_4=15, a_5=12, a_6=13, a_7=10, a_8=15, a_9=13, a_{10}=9, b_1=20, b_2=16, b_3=20, b_4=18, b_5=20, b_6=24, b_7=20, b_8=21, b_9=20, b_{10}=20.

The MOOP formulation is as given below

$$\left.\begin{array}{l} Min(G_1, G_2, G_3, ..., G_{10}, H_1, H_2, H_3, ..., H_{10},), \\ s.t. \\ f_i - a_i \geq 0, \\ b_i - f_i \geq 0, \\ x_j \in \mathbf{F_2}, \\ i = 1, 2, 3, ..., 10, j = 1, 2, 3, ..., 11. \end{array}\right\} \quad (6.2.7)$$

Now we set the parameters to be used in NSGA-II. List of parameters is given in Table 6.1.

number of generation	500
population size	100
probability of crossover	0.8
probability of mutation	0.1
seed random number	0.9876
number of bits for binary variables (have been taken equal bits for all variables)	1
number of objective functions	20
number of constraints	20

Table 6.1: Parameters used in NSGA-II for Example 1

Now we apply NSGA-II to the MOOP(6.2.7). For this we set the parameters to be used in NSGA-II as given in Table 6.1. We get the solutions of inequalities (6.2.6) through the MOOP(6.2.7). These solutions are given in Table 6.3 and their Pareto fronts are given in Figure 6.3

Example 2: In Example 1, we have considered a rectangular system of 10 inequalities

in 11 variables. Let us now consider a square system of 11 inequalities in 11 variables.

$$\left.\begin{array}{lll}
15 & \leq 3x_1 + 15x_2 + 0x_3 + 0x_4 + 0x_5 + 0x_6 + 0x_7 + 0x_8 + 0x_9 + 0x_{10} + 15x_{11} & \leq 19 \\
14 & \leq 1x_1 + 1x_2 + 2x_3 + 4x_4 + 2x_5 + 2x_6 + 4x_7 + 4x_8 + 6x_9 + 4x_{10} + 3x_{11} & \leq 18 \\
14 & \leq 0x_1 + 2x_2 + 4x_3 + 1x_4 + 5x_5 + 3x_6 + 0x_7 + 4x_8 + 4x_9 + 5x_{10} + 5x_{11} & \leq 19 \\
16 & \leq 0x_1 + 4x_2 + 1x_3 + 2x_4 + 6x_5 + 4x_6 + 5x_7 + 1x_8 + 5x_9 + 2x_{10} + 3x_{11} & \leq 19 \\
1 & \leq 0x_1 + 2x_2 + 5x_3 + 6x_4 + 4x_5 + 2x_6 + 3x_7 + 3x_8 + 3x_9 + 0x_{10} + 5x_{11} & \leq 15 \\
13 & \leq 0x_1 + 2x_2 + 3x_3 + 4x_4 + 2x_5 + 4x_6 + 5x_7 + 1x_8 + 1x_9 + 6x_{10} + 5x_{11} & \leq 13.9 \\
10 & \leq 0x_1 + 4x_2 + 0x_3 + 5x_4 + 3x_5 + 5x_6 + 2x_7 + 6x_8 + 2x_9 + 3x_{10} + 3x_{11} & \leq 20 \\
15 & \leq 0x_1 + 4x_2 + 4x_3 + 1x_4 + 3x_5 + 1x_6 + 6x_7 + 6x_8 + 2x_9 + 3x_{10} + 3x_{11} & \leq 21 \\
13 & \leq 0x_1 + 6x_2 + 4x_3 + 5x_4 + 3x_5 + 1x_6 + 2x_7 + 2x_8 + 4x_9 + 5x_{10} + 1x_{11} & \leq 20 \\
9 & \leq 0x_1 + 4x_2 + 5x_3 + 2x_4 + 0x_5 + 6x_6 + 3x_7 + 3x_8 + 5x_9 + 2x_{10} + 3x_{11} & \leq 20 \\
14 & \leq 1x_1 + 3x_2 + 5x_3 + 3x_4 + 5x_5 + 5x_6 + 3x_7 + 3x_8 + 1x_9 + 3x_{10} + 1x_{11} & \leq 20.
\end{array}\right\} \quad (6.2.8)$$

x_1, x_2, x_3,..., $x_{11} \in \mathbf{F_2}$ and $\mathbf{F_2} = \{0, 1\}$.

After applying the developed method, we find that the system is inconsistent.

Example 3: In this example, let us take the same system of inequalities as taken in Example 2 but $x_j \in [-1, 1]$. The parameters to be used in NSGA-II for this example are given in Table 6.2. After applying the developed method, we get the solutions as given in Table 6.4.

6.3 Results and Discussion

After applying the developed method discussed in Section 6.2, we got the solutions of Examples 1 and 3 as given in Tables 6.3 and 6.4 respectively. From Tables 6.3 and 6.4, we observe that there are hundred solutions(based on population size) but some are repeated. Example 2 has no solution. On comparing Examples 2 and 3, we see that

the system of inequalities in Example 2 is inconsistent because x_1, x_2, x_3,..., $x_{11} \in \mathbf{F_2}$, while in Example 3 the variables $x_i \in [-1, 1]$. Observe that In Examples 1 and 2 all the variables are binary taking values 0 or 1, while in Example 3 all the variables are real taking values in [-1,1]

6.4 Conclusion

In this chapter, we have developed a method to solve a general system of linear inequalities based on NSGA-II. We have illustrated the technique by three numerical examples. Examples 1 and 3 are consistent, while Example 2 is inconsistent. Example 1 deals with a rectangular system consisting of 10 inequalities in 11 variables. Examples 2 and 3 deal with a square system consisting of 11 inequalities in 11 variables. The main advantage of using the method developed in this chapter is that multiple solutions, if exist, of a problem are obtained in one single simulation run.

number of generation	500
population size	200
probability of crossover	0.8
probability of mutation	0.1
seed random number	0.9876
number of bits for binary variables (have been taken equal bits for all variables)	1
number of objective functions	20
number of constraints	20

Table 6.2: Parameters used in NSGA-II for Example 3

1	1 1 1 0 0 1 1 0 0 1 0	35	0 1 1 1 1 0 1 0 0 0 0	69	0 1 0 1 1 0 1 1 0 0 0
2	1 1 0 1 1 0 1 1 0 0 0	36	1 1 1 0 1 1 0 0 0 1 0	70	0 1 1 1 1 0 1 0 0 0 0
3	0 1 1 1 1 0 1 0 0 0 0	37	1 1 1 0 0 1 1 0 0 1 0	71	1 1 1 1 1 0 1 0 0 0 0
4	1 1 0 1 1 0 1 1 0 0 0	38	1 1 1 0 1 0 1 0 0 1 0	72	1 1 1 0 1 0 1 0 0 1 0
5	1 1 1 1 0 1 1 0 0 0 0	39	1 1 1 0 0 1 1 0 0 1 0	73	0 1 1 0 1 1 0 0 0 1 0
6	1 1 1 0 1 1 0 0 0 1 0	40	0 1 1 0 1 1 0 0 0 1 0	74	0 1 1 1 0 1 1 0 0 0 0
7	0 1 1 0 1 1 0 0 0 1 0	41	0 1 1 0 1 1 0 0 0 1 0	75	0 1 1 0 0 1 1 0 0 1 0
8	1 1 0 1 1 0 1 1 0 0 0	42	0 1 1 0 1 1 0 0 0 1 0	76	0 1 1 0 0 1 1 0 0 1 0
9	1 1 0 1 1 0 1 1 0 0 0	43	0 1 1 1 0 1 1 0 0 0 0	77	0 1 1 1 0 1 1 0 0 0 0
10	1 1 1 0 0 1 1 0 0 1 0	44	1 1 1 1 0 1 1 0 0 0 0	78	0 1 1 0 0 1 1 0 0 1 0
11	0 1 1 0 1 1 0 0 0 1 0	45	0 1 0 1 1 0 1 1 0 0 0	79	1 1 0 1 1 0 1 1 0 0 0
12	1 1 1 0 1 0 1 0 0 1 0	46	1 1 1 0 1 0 1 0 0 1 0	80	1 1 1 0 0 1 1 0 0 1 0
13	0 1 1 1 0 1 1 0 0 0 0	47	0 1 1 0 1 1 0 0 0 1 0	81	1 1 1 1 1 0 1 0 0 0 0
14	0 1 1 0 1 0 1 0 0 1 0	48	1 1 1 0 1 0 1 0 0 1 0	82	1 1 1 1 1 0 1 0 0 0 0
15	1 1 1 1 1 0 1 0 0 0 0	49	1 1 0 1 1 0 1 1 0 0 0	83	0 1 0 1 1 0 1 1 0 0 0
16	1 1 1 0 1 0 1 0 0 1 0	50	0 1 1 1 0 1 1 0 0 0 0	84	0 1 1 0 1 1 0 0 0 1 0
17	0 1 0 1 1 0 1 1 0 0 0	51	1 1 0 1 1 0 1 1 0 0 0	85	0 1 0 1 1 0 1 1 0 0 0
18	1 1 1 1 1 0 1 0 0 0 0	52	1 1 1 1 1 0 1 0 0 0 0	86	1 1 1 0 1 1 0 0 0 1 0
19	1 1 1 1 0 1 1 0 0 0 0	53	0 1 1 1 0 1 1 0 0 0 0	87	0 1 1 0 0 1 1 0 0 1 0
20	1 1 1 0 1 0 1 0 0 1 0	54	0 1 1 0 0 1 1 0 0 1 0	88	0 1 1 1 0 1 1 0 0 0 0
21	0 1 0 1 1 0 1 1 0 0 0	55	0 1 1 0 0 1 1 0 0 1 0	89	1 1 1 1 1 0 1 0 0 0 0
22	1 1 1 0 1 1 0 0 0 1 0	56	1 1 1 0 1 0 1 0 0 1 0	90	0 1 1 0 1 1 0 0 0 1 0
23	0 1 0 1 1 0 1 1 0 0 0	57	1 1 0 1 1 0 1 1 0 0 0	91	0 1 1 0 0 1 1 0 0 1 0
24	1 1 1 1 0 1 1 0 0 0 0	58	0 1 1 0 1 1 0 0 0 1 0	92	0 1 1 0 1 1 0 0 0 1 0
25	1 1 1 0 1 0 1 0 0 1 0	59	0 1 1 0 0 1 1 0 0 1 0	93	1 1 1 0 1 1 0 0 0 1 0
26	0 1 1 0 1 1 0 0 0 1 0	60	0 1 1 0 0 1 1 0 0 1 0	94	1 1 1 1 0 1 1 0 0 0 0
27	1 1 0 1 1 0 1 1 0 0 0	61	1 1 1 0 1 1 0 0 0 1 0	95	0 1 1 0 1 1 0 0 0 1 0
28	1 1 1 0 1 1 0 0 0 1 0	62	0 1 1 1 0 1 1 0 0 0 0	96	1 1 1 1 1 0 1 0 0 0 0
29	0 1 1 1 0 1 1 0 0 0 0	63	1 1 1 0 1 0 1 0 0 1 0	97	1 1 1 0 1 1 0 0 0 1 0
30	0 1 1 1 0 1 1 0 0 0 0	64	1 1 1 1 1 0 1 0 0 0 0	98	0 1 1 1 1 0 1 0 0 0 0
31	0 1 1 0 1 0 1 0 0 1 0	65	0 1 1 0 0 1 1 0 0 1 0	99	0 1 1 0 0 1 1 0 0 1 0

32	0 1 1 1 0 1 1 0 0 0 0	66	1 1 1 0 1 1 0 0 0 1 0	100	0 1 1 1 0 1 1 0 0 0 0
33	0 1 1 1 0 1 1 0 0 0 0	67	1 1 1 0 1 1 0 0 0 1 0		
34	0 1 1 0 1 1 0 0 0 1 0	68	0 1 1 0 1 0 1 0 0 1 0		

Table 6.3: Solutions of Example 1.

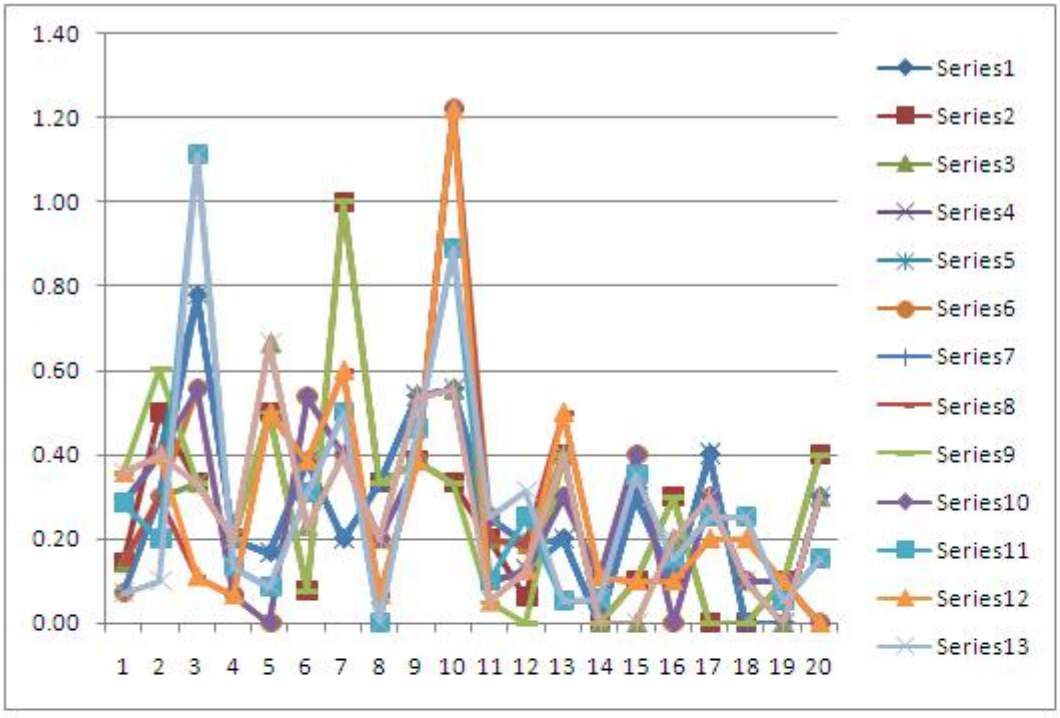

Figure 6.1: Pareto fronts of solutions of Example 1(here on the horizontal axis i stands for the ith objection function f_i, i= 1,2,3, ... ,20).

1	-.451, .726, .737, .190, .995, .228, -.414, -.239, .537, .631, .894
2	-.197, .705, .748, -.163, .122, .628, -.411, .776, .800, .615, .8.55
3	-.513, .670, .863, .649, .411, .226, -.450, -.862, .901, .504, .917
4	-.513, .670, .863, .649, .411, .226, -.450, -.862, .901, .500, .917
5	-.476, .721, .878, -.559, .491, .921, -.991, .977, .940, .996, .888
6	-.720, .670, .696, .166, .704, .500, -.564, -.302, .795, .532, .927
7	-.304, .575, .541, -.124, .318, .981, -.997, .948, .878, .947, .870
8	-.252, .714, .844, -.589, .841, .476, -.359, .10, .81, .606, .841
9	-.346, .746, .897, -.943, .731, .462, -.522, .139, .791, .562, .856
10	-.363, .657, .951, .157, .614, .170, -.374, -.944, .835, .622, .947
11	-.327, .664, .8.58, .411, .4.17, .230, -.396, -.972, .860, .645, .880
12	-.126, .729, .844, -.510, .995, .801, -.242, .120, .100, .470, .802
13	-.586, .698, .961, .293, .947, .855, -.848, -.851, .493, .496, .919
14	-.622, .689, .551, .642, .402, .924, -.997, -.876, .893, .567, .901
15	.259, .585, .925, -.564, .981, .965, -.986, .981, .497, .960, .793
16	-.147, .748, .888, .217, .609, .379, -.994, .180, .100, .977, .801
17	.411, .572, .909, -.499, .990, .467, -.606, .976, .497, .960, .793
18	-.703, .744, .696, .166, .699, .500, -.568, -.247, .484, .619, .924
19	-.568, .720, .176, -.319, .833, .100, -.803, .100, .744, .670, .901
20	.260, .584, .921, -.706, .999, .504, -.606, .987, .454, .960, .793
21	-.556, .720, .866, .554, .406, .205, -.303, -.930, .750, .495, .916
22	-.495, .713, .750, -.641, .712, .436, -.473, .747, .999, .891, .869
23	.168, .657, .943, -.730, .545, .465, -.881, .269, .940, .914, .804
24	.168, .657, .967, .919, .545, .465, -.881, .269, .100, .914, .804
25	-.588, .689, .609, .679, .516, .852, -.867, -.932, .748, .465, .931
26	-.737, .753, .796, .139, .702, .500, -.608, -.247, .910, .619, .924
27	-.648, .580, .988, .379, .673, .851, -.935, -.864, .861, .578, .943
28	-.438, .675, .910, .574, .406, .184, -.319, -.868, .804, .495, .917
29	-.727, .739, .502, .642, .910, .885, -.997, -.910, .906, .566, .913

30	-.629, .720, .187, -.122, .669, .100, -.783, .100, .740, .595, .902
31	-.145, .695, .863, -.50, .967, .790, -.896, .126, .985, .990, .840
32	-.145, .753, .849, -.564, .934, .904, -.607, .999, .573, .654, .792
33	-.727, .689, .551, .642, .402, .924, -.997, -.876, .893, .567, .917
34	.168, .649, .931, .919, .545, .501, -.881, .258, .942, .914, .779
35	-.145, .695, .863, -.493, .969, .767, -.981, .871, .809, .974, .840
36	-.389, .681, .906, -.560 .994, .733, -.411, .105, .996, .470, .915
37	-.440, .685, .248, -.895, .958, .987, -.789, .712, .821, .631, .928
38	-.222, .584, .702, -.154 .349, .987, -.986, .965, .785, .890, .870
39	-.538, .695, .878 -.559 .491 .887 -.991 .878 .989 .996 .888
40	-.187, .753, .952, -.564, .934, .380, -.610, .998, .570, .963, .793
41	-.323, .648, .911, .167, .849, .184, -.410, -.624 .965 .580, .908
42	-.187, .753, .888, -.564, .934 .379, -.610, .999, .570, .963, .793
43	-.491, .659, .922, .177, .669, .181, -.400, -.635, .907 .580, .895
44	-.265, .736, .844, -.534, .966, .465, -.271, -.169, .906, .623, .813
45	-.438, .676, .920, .574, .406, .184, -.322, -.954, .804, .495, .917
46	.339, .584, .860, -.553, .982, .588, -.606, .814, .497, .967, .793
47	-.717, .658, .689, .642, .539, .934, -.995, -.897, .883, .567, .925
48	.168, .657, .943, .919, .545, .465, -.881, .269, .940, .914, .804
49	-.297, .585, .541, -.101, .348, .986, -.986, .959, .851, .825, .870
50	-.222, .584, .711, -.154, .349, .987, -.976, .974, .785, .890, .870
51	-.267, .660, .796, .177, .463, .501, -.411, -.635, .997, .580, 9.05
52	-.440, .581, .248, -.906, .989, .987, -.778, .711, .851, .630, .928
53	-.425, .766, .741, -.535, .721, .894, -.377, .707, .997, .401, .849
54	-.539, .721, .878 -.559, .491, .921, -.991, .867, .940, .911, .888
55	-.363, .657, .951, .157, .614, .204, -.374, -.944, .835, .683, .947
56	-.363, .663, .951, .157, .665, .179, -.330, -.944, .835, .620, .913
57	-.720, .656, .696, .227, .539, .498, -.553, -.351, .901, .564, .927
58	-.207, .753, .593, .615, .373, .492, -.566, -.364, .750, .536, .814
59	-.705, .745, .551, .642, .399, .918, -.997, -.876, .958, .567, .901
60	.259, .550, .925, -.564, .981, .965, -.986, .981, .497, .960, .793

61	-.365, .670, .696, .227, .427, .464 -.394, -.351, .972, .564, .855
62	-.314, .648, .911, .167, .861, .188, -.405, -.624, .906, .579, .872
63	-.717, .658, .927, .120, .669, .934, -.995, -.633, .973, .567, .925
64	-.374, .736, .844, -.534, .994, .836, -.271, .110, .905, .470, .834
65	-.225, .754, .836, -.701, .996, .857, -.309, -.100, .999, .612, .813
66	-.265, .736, .844, -.534, .966, .507, -.271, .487, .906, .623, .834
67	-.720 .670, .696, .227, .427. .500, -.560, -.351, .972, .564, .927
68	-.451, .726, .930, -.261, .995, .228, -.399, -.239, .537, .622, .894
69	-.119, .639, .864, -.564, .952, .927, -.680, .973, .575, .645, .820
70	-.717, .658, .840, .620, .539, .927, -.998, -.897, .875, .485, .925
71	-.118, .695, .864, -.493, .983, .507, -.916, .873, .809, .974, .840
72	-.586, .698, .958, .293, .976, .855, -.848, -.851, .502, .496, .919
73	-.720, .670, .713, .157, .704, .500, -.562, -.201, .795, .536, .972
74	-.717, .658, .689, .642, .539, .934, -.995, -.831, .883, .567, .925
75	-.119, .639, .864, -.564, .952, .927, -.680, .973, .575, .645, .887
76	.411, .572, .951, -.556, .990, .467, -.606, .976, .497, .960, .792
77	-.147, .748, .888, .217, .572, .282, -.994, .180, .100, .977, .801
78	-.252, .696, .844, -.588, .841, .769, -.971, .996, .810, .991, .840
79	-.217, .730, .880, .217, .572, .274, -.994, .114, .100, .977, .801
80	.260 .584, .921, -.564, .998, .504, -.606, .973, .497, .960, .793
81	-.462, .600, .845, .649, .411, .226, -.450, -.863, .951, .500, .917
82	-.512, .711, .551, .746, .402, .924, -.100, -.748, .893, .600, .917
83	-.374, .736, .844, -.544, .994, .836, -.271, -.559, .905, .470, .834
84	-.720, .670, .797, .166, .704, .500, -.562, -.281, .795, .518, .972
85	-.703, .744, .696, .166, .699, .500, -.568, -.172, .484, .688, .914
86	-.103, .695, .864, -.493, .983, .507, -.927, .744, .998, .974, .840
87	-.538, .721, .878, -.559, .491, .921, -.991, .936, .940, .996, .888
88	-.579, .744, .989, .180, .839, .965, -.997, -.5.15, .462, .536, .902
89	-.538, .668, .878, -.601, .491, .922, -.991, .922, .940, .996, .925

90	-.187, .753, .925, -.564, .919, .380, -.591, .998, .570, .963, .793
91	.191, .654, .888, -.564, .956, .499, -.606, .973, .497, .960, .803
92	-.425, .766, .740, -.540, .666, .912, -.383, .738, .997, .412, .849
93	-.412, .718, .888, -.550, .992, .458, -.991, .973, .940, .996, .884
94	-.222, .584, .702, -.154, .349, .997, -.992, .965, .785, .890, .870
95	-.291, .579, .541, -.101, .348, .986, -.994, .959, .851, .828, .870
96	-.462, .600, .845, .649, .411, .226, -.450, -.863, .978, .500, .917
97	-.149, .713, .863, -.620, .967, .790, -.361, .194, .985, .600, .841
98	-.125, .552, .772, .180, .597, .443, -.786, .134, .905, .803, .844
99	-.107, .696, .864, -.372, .983, .530, -.916, .873, .722, .974, .840
100	-.425, .766, .740, -.535, .721, .912, -.383, .707, .997, .412, .849

Table 6.4: Solutions of Example 3.

Chapter 7

Conclusions and Future Scope

In this chapter we have given the conclusions of the thesis and the future scope of the research. This chapter is organized as follows: Section 7.1 gives the conclusion and Section 7.2 gives the future scope.

7.1 Conclusions

In this thesis, we have developed evolutionary multiobjective approaches to construct desired Boolean functions having the best trade-offs among their properties. We have also included a new concept of biasedness in our method(chapter 5) and got success to get those functions that could not be possible by the method without biasedness(Chapter 2). Apart from construction of Boolean functions, we have also developed a new method based on NSGA-II to solve a set of linear inequalities. The primary aim of my research work has been to develop new evolutionary methods by using heuristic technique to construct Boolean functions by optimizing their properties and to solve a system of linear inequalities. In Chapter 1(Introduction), we have given necessary back ground related to our work to understand the thesis. Also,in this chapter, we have briefly described the Boolean functions and their properties, and a comprehensive literature review related to our work is given. In Chapter 2, evolutionary multiobjective approaches is developed based on two objective nonlinearity and resiliency and in

Chapter 3, evolutionary multiobjective approaches is developed based on 2 objective nonlinearity and autocorrelation to construct balanced Boolean functions having the best trade-offs between the properties respectively.
In Chapter 4, Evolutionary multiobjective approach is developed based on 3 objectives nonlinearity, resiliency and autocorrelation to construct balanced Boolean functions.
In Chapter 5, the concept of biasedness is introduced in the proposed method to get the desired Boolean functions of 7 and 8 variables. Observe that such functions could not be constructed by the method developed in Chapter 2.
In Chapter 6, we have developed a method based on NSGA-II to solve a general system of linear inequalities. The main advantage of using NSGA-II is that multiple solutions, if exist, of a problem are obtained in one single simulation run.

7.2 Future Scope

The following areas of future work have been identified.

1. extansion of research work contained in this thesis.

2. Development of new heuristic methods for construction of Boolean functions and S-boxes to improve their cryptographic properties.

3. Development of future methods should further aim to simultaneously optimize multiple cryptographic properties which will be dependent on the limitations of property combination co-existence.

4. New methods for constructing strong Boolean functions and s-boxes which are a combination of heuristic techniques.

5. extension of the method discussed for the system of linear inequalities to the system of nonlinear inequalities

Bibliography

[1] Aguirre H., Okazaki H. and Fuwa Y., *"An evolutionary multiobjective approach to design highly non-linear Boolean functions"*, Proc. GECCO'07, 749-756, 2007.

[2] Barnhart C., Johnson E. L. , Nemhauser G. L. , M. W. P. and Savelsbergh P. H. V., *"Branch-and-Price: column generation for solving huge integer programs"*, Operations Research, 46(3), 316-329, 1998.

[3] Bector C.R., Chandra S., and Dutta J., *"Principles of Optimization Theory"*, Narosa publishing House, New Delhi, 2005.

[4] Beelen P. and Leander G., *"A new construction of highly nonlinear S-boxes P"*, Cryptogr, Commun. DOI 10.1007/s12095-011-0052-4, Springer Science Business Media, LLC, 4, 65-77, 2012.

[5] Benders J. F., *"Partitioning procedures for solving mixed-variables programming problems"*, Numerische Mathematik, 4, 238-252, 1962.

[6] Ben-Isreal A. and Charnes A., *"On some problems of diophantine programming"*, Cahiers du Centre dEtudes de Recherche Operationelle, 4, 215-280, 1962.

[7] Biham E. and Shamir A., *"Differential cryptanalysis of DES like cryptosystem"* ,Journal of Cryptology, 4, 3-72, 1991.

[8] Biham E. and Shamir A., *"Differential cryptanalysis of the full 16-round DES"*, Advanced in Cryptology-CRYPTO' 92, LNCS, 740, 487-496, 1992.

[9] Bharti S.K. and Singh S.R., *"Solving multi objective linear programming problems using intuitionistic fuzzy optimization method: a comparative approach"*, Proc. Int. conf. on Reliability, Infocom Technologies and Optimization, Jan. 29-31, 2013.

[10] Bharti S.K., Nishad A. N. and Singh S.R., *"Solution of multi objective linear programming problems in intuitionistic fuzzy environment"*, Proc. second Int. Conf. on Soft Computing Jaipur, 2012.

[11] Burnett L, Millan W., Dawson E. and Clark A., *"Simpler methods for generating better Boolean functions with good cryptographic properties"*, Australasian Journal of Combinatorics, 29, 241 - 247, 2004.

[12] Burnett L., *"Heuristic optimization of Boolean functions and substitutions-Boxes for cryptography"*, Ph.D. Thesis, Information Security Institute, Feculty of Information Technology, Queensland University of Technology, 2005.

[13] Carlet C., *"On the propagation criteria of degree l and order k"*, Advances in Cryptology-EUROCRYPT'98, LNCS, 1403, 462-474, 1998.

[14] Chee S., Lee S. and Kim K., *"Semi-bent functions"*, Advancess in Cryptology-ASIACRYPT'94, LNCS, 917, 107-118, 1995.

[15] Chen D., Batson R. G. and Dand Y., *"Applied Integer Programming: Modelling and Solution"*, John Wiley and Sons, 2010.

[16] Cohen G., Honkala I., Litsyn S. and Lobstein A., *"Covering Codes"*, Elsevier, Amsterdam, 1997.

[17] Clark D. C. and Gonzalez R. C. *"Optimal solution of linear inequalities with applications to pattern recognition"*,IEEE Trans. on Pattern Analysis and Machine Intelligence, 3(6), 643-655, 1981.

[18] Clark J.A. and Jacob J. L. , *"Two-stage optimization in the design of Boolean functions"*, Proc. 5th Aus. Conf. on Info., Security and Privacy- ACISP 2000, LNCS, 1841, 242-254, 2000.

[19] Clark J. A., Jacob J. L., Stepney S., Maitra S. and Millan W., *"Evolving Boolean function satisfying multiple criteria"*, INDOCRYPT,2551 , 246-259, 2002.

[20] Deb, K., *"Multi-objective evolutionary algorithms: introducing bias among pareto-optimal solutions"*, Advances in Evolutionary Computing, 263-293, 2003.

[21] Deb K., Pratap A., Agarwal S, and Meyarivan T., *"A fast and elitist multi-objective genetic algorithm"*, IEEE Trans. on Evolutionary Computation, 6(2), 182-197, 2002.

[22] Dawson E. and Clark A., *"Divide and conquer attack on certain classes of stream cipher"*, CRYPTOLOGIA, 18(1), 25-40,1994.

[23] Dawson E., Millan W. and Simpson L., *"Designing Boolean functions for cryptography application"*, Proc. General Algebra Conference(AAA58), Verlag Johannes Heyn, 1-22, 1999.

[24] Dhal G., *"Combinatorial properties of Fourier Motzkin elimination"*, Electronic J. of Linear Algebra, 16, 334-346, 2007.

[25] Ding C., Xiao G. and Shan W., *"The stability theory of strean ciphers"*, LNCS, 561, 1991.

[26] Dobbirtin H., *"Construction of bent function and balanced Boolean function with high nonlinearity"*, Fast Software Encryption, FSE, 94, LNCS, 1008, 61-74, 1994.

[27] Dress A. W. M. and Wenzel W. , *"A new look at the greedy algorithm"*, Appl. Math. Lett., 3, 33-35, 1990.

[28] Dress A. W. M. and Wenzel W., *"Valuated matroids"*. Adv. Math., 93, 214-250, 1992.

[29] Eckart Z., *"Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications"*, Doctoral Dissertation ETH 13398, Swiss Federal Institute of Technology, Switzerland, 1999.

[30] Eckart Z. and Lothar T., *"Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approch"*, IEEE Trans. on Evolutionary Computation, 3(4), 257-271, 1999.

[31] Fedorova M. and Tarannikov Y., *"On the constructing of highly nonlinear resilient Boolean functioms by means of special matrices"*, Process in Cryptology-INDOCRYPT 2001, LNCS, 2247, 254-266, 2001.

[32] Filiol E., *"Decimal attack of stream cipher"*, Progress in Cryptology, LNCS- INDOCRYPT 2000, 1977, 31-42, 2000.

[33] Fonseca C. M. and Fleming P. J., *"Genetic algorithms for multiobjective optimization: formulation, discussion and generalization"*, In Genetic Algorithms: Proc. Fifth Int. Conf. S. Forrest, ed., San Mateo, CA: Morgan Kaufmann, 416-423, 1993.

[34] Forsgren A.and Murray W., *"Newton methods for large-scale linear inequality-constrained minimization"*, SIAM J. Optimization, 7(1), 162 - 176, 1997.

[35] Genova K. and Guliashki V., *"Linear integer programming method and approaches-survey"*, Cybernetics and Information Technolies, Bulgarian Academy of Science, 11, 3-25, 2011.

[36] Gong G. and Khoo K., *"Additive autocorrelation of resilient Boolean functions"*, Selected areas in Cryptography, SAC 2003, LNCS, 3006, 275-290, 2004.

[37] Goyal R., Yadav S. P. and Kishor A., *"Design of Boolean functions satisfying multiple criteria by NSGA-II"*, Proc. Int. Conf. on Soft Computing for Problem Solving (SocProS 2011), AISC 130, 461-468, 2011.

[38] Glover F., *"A new foundation for a simplified primal integer programming algorithms"*, Operations Research, 13, 879-919, 1965.

[39] Glover F., *"Tabu search: A tutorial"*, J. Interfaces, 20(4) 74-94, 1990.

[40] Guignard M. and Kim S., *"Lagrangian decomposition: A model yielding stronger Lagrangian bounds"*, Mathematical Programming, 39, 215-228, 1987.

[41] Guillot P., *"Cryptographical Boolean functions construction from linear codes"*, BFCA'05, 141-167, 2005

[42] Gupta K. C. and Sarkar P., *"Improved construction of nonlinear resilient s-boxes"*, Advances in Cryptology-ASIACRYPT 2002, LNCS, 2501, 466-483, 2002.

[43] Gupta P., Mehlawat, Mukesh K. and Mittal G., *"Asset portfolio optimization using support vector machines and real coded genetic algorithm"*, J. Global Optimization, 53(2), 297-315, 2012.

[44] Hoffman K. L. and Padberg M., *"Solving airline crew scheduling problems by branch-and-cut"*, Management Science, 39, 657-682, 1993.

[45] Holland J. H., *"Genetic algorithms and the optimal allocation of trials"*, SIAM J. Computing, 2(2), 88-105, 1973.

[46] Izbenko Y., Kovtun V. and Kuznetsov A.,*"The design of Boolean functions by modified hill climbing method"*, Information Technology, Sixth Int. Conf., 356 - 361, 2009.

[47] Johansson T. and Jonsson F., *"Fast correlation attacks based on Turbo code techniques"*, Advances in Cryptology- CRYPTO'99, LNCS,1666, 181-197, 1999.

[48] Johansson T. and Jonsson F., *"Improved fast correlation attack on Stream cipher via convolutional codes"*, Advances in Cryptology-EUROCRYPT'99, LNCS, 1592, 347-362, 1999.

[49] Johansson T. and Pasalic E., *"A construction of resilient functions with high nonlinearity"*, IEEE Trans. on Information Theory, 49(2), 494-501, 2003.

[50] Kavut S. and Yucel M. D., *"Improved cost function in design of Boolean functions satisfying multiple critaria"*, INDOCRYPT, 121-134, 2003.

[51] Khoo K. and Gong G., *"New constructions for resilient and highly nonlinear Boolean functions"*, Information Security and Privacy, ACISP 2003, LNCS, 2727, 498-509, 2003.

[52] Kurosawa K. and Satoh T., *"Design of SAC/PC(l) of order k Boolean functions and three other cryptographic criteria"*, Advances in Cryptology-EUROCRYPT'97, LNCS,1233, 434-449, 1997.

[53] Kirkpatrick S. J., Gelatt C. D. and Vecchi M. P., *"Optimization by simulated annealing"*, J. Science, 220(4958), 671-680, 1983.

[54] Knowres J. and Corne D., *"The Pareto archived evolution strategy:A new base line for multiobjective optimization"*,Proc. 999 Congress on Evolutionary Computation, IEEE Press, 98-105, 1999.

[55] Korovin K., Tsiskaridze N. and Voronkov A., *"Implementing conflict resolution"*, Ershov Memorial Conference 2011,LNCS, 7162, 362-376, 2012.

[56] Kurosawa K., Satoh T. and Yamamoto K., *"Highly nonlinear t-resilient functions"*, J. Universal Computer Science, 3(6), 721-729, 1997.

[57] Land A. H. and Doig A. G., *"An automatic method for solving discrete programming problems"*, Econometrica, 28, 97-520, 1960.

[58] Little J. D. C., Murthy K. G., Sweeney D. W. and Karel C., *"An algorithm for the traveling salesman problem"*, Operations Research, 11, 972-989, 1963.

[59] Maitra S., *"Correlation immune Boolean functions with very high nonlinearity"*, Cryptography, eprint.iacr.org/2000/054.ps.

[60] Maitra S., *"Highly nonlinear balanced Boolean functions with very good autocorrelation property"*, Workshop on Coding and Cryptography, WCC 2001, Electric Notes in Discrete Mathematics, 6, 1-557, 2001.

[61] Maitra S., *"On nonlinearity and autocorrelation properties of correlation immune Boolean functions"*, J. Information Science and Engerineeing, 20, 305-323, 2004.

[62] Maitra S. and Pasalic E., *"Further constructions of resilient Boolean functions with very high nonlinearity"*, IEEE Trans. on Infor. Theory, 48(7), 1825-1834, 2002.

[63] Maity S. and Johansson T., *"Construction of cryptographically important Boolean functions"*, Progress in Cryptology- INDOCRYPT-2002, LNCS, 2551, 234-245, 2002.

[64] Maity S. and Maitra S., *"Minimum distance between bent and 1-resilient Boolean functions"*, Fast Software Encryption, FSE 2004, LNCS, 3017, 143-160, 2004.

[65] Matsui M., *"Linear cryptanalysis method for DES cipher"*, Advances in Cryptology- EUROCRYPT 93, LNCS, 765, 386 - 397, 1994.

[66] McFarland R.L., *"A family of difference sets in non-cyclic groups"*, J. Combinatorial Theory, 15, 1-10, 1973.

[67] McWilliams F.J. and Sloane N.J.A., *"The theory of error-correcting codes"*, North-Holland, 2, 168-181, 1978.

[68] Meier W. and Staffelbach O., *"Fast correlation attacks on certain stream ciphers"*, J. Cryptology, 1(3), 159-176, 1989.

[69] Millan W., Clark A. and Dawson E., *"An effective genetic algorithm for finding highly non-linear Boolean functions"*, Proc. First Int. Conf. on Info. and Comm. Security, LNCS, 1334, 149-158, 1997.

[70] Millan W., Clark A. and Dawson E., *"Heuristic desing of cryptographically strong balanced Boolean functions"*, Proc. Advances in Cryptology - EUROCRYPT 98, LNCS, 1403, 489-499, 1998.

[71] Minsky M., *"Steps towards artificial intelligence"*, Proc. Institute of Radio Engineers, 49(1), 8-30, 1961.

[72] Mishra S. K. and Jaiswal M., *"Optimality conditions and duality for non-differentiable multiobjective semi-infinite programming"*, Vietnam J. Mathematics, 40, 331-343, 2012.

[73] Mohanty B. K. and Vijayaraghavan T. A. S., *"A multiobjective programming problem and its equivalent goal programming problem with approperiate priorities and aspiration levels: a fuzzy approach"*, Computers and Research, 22, 771-778, 1995.

[74] Nagaraja G. and Krishna G., *"An algorithm for the solution of linear inequalities"* IEEE Trans. on Computers, 23(4), 421-427, 1974.

[75] Nemhauser G. L. and Wolsey L. A., *"Integer and Combinatorial Optimization"*, John Wiley and Sons, 1988.

[76] Orsi R., Helmke U. and Moore J. B., *"A Newton-like method for solving rank constrained linear matrix inequalities"*, Proc. 43rd IEEE Conf. on Decision and Control , 3138-3144, 2004.

[77] Padberg M. W. , *"Combinatorial optimization"*, Mathematical Programming Study, 12, 1-7, 1980.

[78] Pasalic E., *"A design of Boolean functions resistant to (fast) algebraic cryptanalysis with efficient implementation"*, Cryptogr. Commun., LLC, 4, 25-45, 2012.

[79] Puchinger J., *"Combining Metaheuristics and Integer Programming for Solving Cutting and Packing Problems"*, PhD Thesis, Vienna University of Technology, Institute of Computer Graphics and Algorithms, January 2006.

[80] Rothaus O.S., *"On bent functions"*, J. Combinatorial Theory, 20(3), 300-305, 1976.

[81] Roy B., *"A brief outline of research on correlation immune functions"*, Information Security and Privacy, ACISP 2002, LNCS, 2384, 379-394,2002.

[82] Sarkar P. and Maitra S., *"Construction of nonlinear Boolean functions with important cryptographic properties"*, Advances in Cryptology-EUROCRYPT 2000, LNCS,1807, 485-506, 2000.

[83] Sarkar P. and Maitra S., *"Nonlinearity bounds and construction of resilient Boolean functions"*, Advances in Cryptology - Crypto 2000, LNCS, 1880, 515-532, 2000.

[84] Seberry J., Zhang X. M. and Zheng Y., *"The relationship between propagation characteristics and nonlinearity of cryptographic functions"*, J. Universal Computer Science, 1(2), 136-150, 1995.

[85] Seberry J., Zhang X. M. and Zheng Y., *"Nonlinearity and propagation characteristics of balanced Boolean functions"*, Information and Computation, 119(1) , 1-13, 1995.

[86] Siegenthaler T., *Correlation immunity of nonlinear combining functions for cryptographic applications"*, IEEE Trans. on Information Theory, 30(5), 776-780, 1984.

[87] Siegenthaler T., *Decrypting a class of stream cipher using ciphertext only"*, IEEE Trans. on Computers, 34(1), 2010-2017, 1985.

[88] Silver E. A., *"An overview of heuristic solution methods"*, J. Operational Research Society, 55, 936-956, 2004.

[89] Srinivas, N., and Deb, K., *"Muiltiobjective optimization using nondominated sorting in genetic algorithms"*, Evolutionary Computation, 2, 221-248, 1994.

[90] Taha H. A., *"Operations Research - An Introduction "*, Prentice Hall of India, New Delhi, 7th edition, 2007.

[91] Tang D., Zhang W. and Tang X., *"Construction of balanced Boolean functions with high nonlinearity and good autocorrelation properties"*, Des. Codes Cryptogr, 67(1), 77-91, 2013.

[92] Tang Y., Reed P. and Wagener T., *"How effective and efficient are multiobjective evolutionary algorithms at hydrologic model calibration"*, Hydrology and Earth System Sci, 10, 289-307, 2006.

[93] Tarannikov Y., *"New constructions of resilient Boolean funtions with maximal nonlinearity"*, Fast Software Encryption, FSE 2001, LNCS, 2355, 66-77, 2002.

[94] Tarannikov Y., *"On resilient Boolean functions with maximal nonlinearity"*, Progress in Cryptology- INDOCRYPT 2000, LNCS, 1977, 19-30, 2000.

[95] Thavaneswaran A., Appadoo S. S. and Samanta M., *" Random coefficient GARCH models"*, Mathematical and Computer Modelling, 41(6-7), 723-733, 2005.

[96] Verma A.K. and Ramesh P.G., *"Multi-objective initial preventive maintenance scheduling for large engineering plants"*, Int. J.Reliability, Quality and Safety Engineering, 14(3), 241-250, 2007.

[97] Verma A.K., Srividya A. and Ramesh P.G., *"A systems approach to integrated e-maintenance of large engineering plants"*, Int. J. Systems Assurance Engineering and Management, 1(3), 239-245, 2010.

[98] Wang Y. and Pham H., *A multi-objective optimization of imperfect preventive maintenance policy for dependent competing risk systems with hidden failure"*, IEEE Trans. on Reliability, 60(4), 770-781, 2011.

[99] Warmack R.E. and Gonzalez R.C., *"An algorithm for the optimal solution of linear inequalities and its application to pattern recognition"*, IEEE Trans. on Computers, 22(12), 1065-1075, 1973.

[100] Wei Y., Ouyang N. and Hu N., *"New construction of Boolean functions which satisfy multiple cryptographic criteria"*, Communications and Networking in China, CHINACOM 2009, 1-6, 2009.

[101] Wu C. K. and Dawson E., *"Construction of correlation immune Boolean function"*, Australasian J. of Combinatorics, 21, 141- 166, 2000.

[102] Young R. D., *"A primal (all integer) integer programming algorithm"*, J. Research of the National Bureau of Standards, 69B, 1965, 213-250.

[103] Youssef A. M. and Gong G., *"Boolean functions with large distance to all bijective monomials: an odd case"*, Selected Areas in Cryptography, SAC, 2001, LNCS, 2259, 49-59, 2001.

[104] Zhang X. M. and Zheng Y., *"On nonlinear resilient Boolean functions"*, Advances in Cryptology- EUROCRYPT'95, LNCS, 921, 274-288, 1995.

[105] Zhang X. M., Zheng Y. and Imai H., *"Relating differential distribution table to other properties of substitution boxes"*, Design, Codes and Cryptograpphy, 19, 45-63, 2000.

[106] Zheng Y and Zhang X. M., *"New results on correlation immunity"*, Information Security and Cryptology-ICISC 2000, LNCS, 2015, 49-63, 2001.

Bibliography

[1] Aguirre H., Okazaki H. and Fuwa Y., *"An evolutionary multiobjective approach to design highly non-linear Boolean functions"*, Proc. GECCO'07, 749-756, 2007.

[2] Barnhart C., Johnson E. L. , Nemhauser G. L. , M. W. P. and Savelsbergh P. H. V., *"Branch-and-Price: column generation for solving huge integer programs"*, Operations Research, 46(3), 316-329, 1998.

[3] Bector C.R., Chandra S., and Dutta J., *"Principles of Optimization Theory"*, Narosa publishing House, New Delhi, 2005.

[4] Beelen P. and Leander G., *"A new construction of highly nonlinear S-boxes P"*, Cryptogr, Commun. DOI 10.1007/s12095-011-0052-4, Springer Science Business Media, LLC, 4, 65-77, 2012.

[5] Benders J. F., *"Partitioning procedures for solving mixed-variables programming problems"*, Numerische Mathematik, 4, 238-252, 1962.

[6] Ben-Isreal A. and Charnes A., *"On some problems of diophantine programming"*, Cahiers du Centre dEtudes de Recherche Operationelle, 4, 215-280, 1962.

[7] Biham E. and Shamir A., *"Differential cryptanalysis of DES like cryptosystem"* ,Journal of Cryptology, 4, 3-72, 1991.

[8] Biham E. and Shamir A., *"Differential cryptanalysis of the full 16-round DES"*, Advanced in Cryptology-CRYPTO' 92, LNCS, 740, 487-496, 1992.

[9] Bharti S.K. and Singh S.R., *"Solving multi objective linear programming problems usingh intuitionistic fuzzy optimization method: a comparative approach"*, Proc. Int. conf. on Reliability, Infocom Technologies and Optimization, Jan. 29-31, 2013.

[10] Bharti S.K., Nishad A. N. and Singh S.R., *"Solution of multi objective linear programming problems in intuitionistic fuzzy environment"*, Proc. second Int. Conf. on Soft Computing Jaipur,2012.

[11] Burnett L, Millan W., Dawson E. and Clark A., *"Simpler methods for generating better Boolean functions with good cryptographic properties"*, Australasian Journal of Combinatorics, 29, 231247, 2004.

[12] Burnett L., *"Heuristic optimization of Boolean functions and substitutions-Boxes for cryptography"*, Ph.D. Thesis, Information Security Institute, Feculty of Information Technology, Queensland University of Technology, 2005.

[13] Carlet C., *"On the propagation criteria of degree l and order k"*, Advances in Cryptology-EUROCRYPT'98, LNCS, 1403, 462-474, 1998.

[14] Chee S., Lee S. and Kim K., *"Semi-bent functions"*, Advancess in Cryptology-ASIACRYPT'94, LNCS, 917, 107-118, 1995.

[15] Chen D., Batson R. G. and Dand Y., *"Applied Integer Programming: Modelling and Solution"*, John Wiley and Sons, 2010.

[16] Cohen G., Honkala I., Litsyn S. and Lobstein A., *"Covering Codes"*, Elsevier, Amsterdam, 1997.

[17] Clark D. C. and Gonzalez R. C. *"Optimal solution of linear inequalities with applications to pattern recognition"*,IEEE Trans. on Pattern Analysis and Machine Intelligence, 3(6), 643-655, 1981.

[18] Clark J.A. and Jacob J. L. , *"Two-stage optimization in the design of Boolean functions"*, Proc. 5th Aus. Conf. on Info., Security and Privacy- ACISP 2000, LNCS, 1841, 242-254, 2000.

[19] Clark J. A., Jacob J. L., Stepney S., Maitra S. and Millan W., *"Evolving Boolean function satisfying multiple criteria"*, INDOCRYPT,2551 , 246-259, 2002.

[20] Deb, K., *"Multi-objective evolutionary algorithms: introducing bias among pareto-optimal solutions"*, Advances in Evolutionary Computing, 263-293, 2003.

[21] Deb K., Pratap A., Agarwal S, and Meyarivan T., *"A fast and elitist multi-objective genetic algorithm"*, IEEE Trans. on Evolutionary Computation, 6(2), 182-197, 2002.

[22] Devson E. and Clark A., *"Divide and conquer attack on certain classes of stream cipher"*, CRYPTOLOGIA, 18(1), 25-40,1994.

[23] Devson E., Millan W. and Simpson L., *"Designing Boolean functions for cryptography application"*, Proc. General Algebra Conference(AAA58), Verlag Johannes Heyn, 1-22, 1999.

[24] Dhal G., *"Combinatorial properties of Fourier Motzkin elimination"*, Electronic J. of Linear Algebra, 16, 334-346, 2007.

[25] Ding C., Xiao G. and Shan W., *"The stability theory of strean ciphers"*, LNCS, 561, 1991.

[26] Dobbirtin H., *"Construction of bent function and balanced Boolean function with high nonlinearity"*, Fast Software Encryption, FSE, 94, LNCS, 1008, 61-74, 1994.

[27] Dress A. W. M. and Wenzel W. , *"A new look at the greedy algorithm"*, Appl. Math. Lett., 3, 33-35, 1990.

[28] Dress A. W. M. and Wenzel W., *"Valuated matroids"*. Adv. Math., 93, 214-250, 1992.

[29] Eckart Z., *"Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications"*, Doctoral Dissertation ETH 13398, Swiss Federal Institute of Technology, Switzerland, 1999.

[30] Eckart Z. and Lothar T., *"Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approch"*, IEEE Trans. on Evolutionary Computation, 3(4), 257-271, 1999.

[31] Fedorova M. and Tarannikov Y., *"On the constructing of highly nonlinear resilient Boolean functioms by means of special matrices"*, Process in Cryptology-INDOCRYPT 2001, LNCS, 2247, 254-266, 2001.

[32] Filiol E., *"Decimal attack of stream cipher"*, Progress in Cryptology, LNCS- INDOCRYPT 2000, 1977, 31-42, 2000.

[33] Fonseca C. M. and Fleming P. J., *"Genetic algorithms for multiobjective optimization: formulation, discussion and generalization"*, In Genetic Algorithms: Proc. Fifth Int. Conf. S. Forrest, ed., San Mateo, CA: Morgan Kaufmann, 416-423, 1993.

[34] Forsgren A.and Murray W., *"Newton methods for large-scale linear inequality-constrained minimization"*, SIAM J. Optimization, 7(1), 162 - 176, 1997.

[35] Genova K. and Guliashki V., *"Linear integer programming method and approaches-survey"*, Cybernetics and Information Technolies, Bulgarian Academy of Science, 11, 3-25, 2011.

[36] Gong G. and Khoo K., *"Additive autocorrelation of resilient Boolean functions"*, Selected areas in Cryptography, SAC 2003, LNCS, 3006, 275-290, 2004.

[37] Goyal R., Yadav S. P. and Kishor A., *"Design of Boolean functions satisfying multiple criteria by NSGA-II"*, Proc. Int. Conf. on Soft Computing for Problem Solving (SocProS 2011), AISC 130, 461-468, 2011.

[38] Glover F., *"A new foundation for a simplified primal integer programming algorithms"*, Operations Research, 13, 879-919, 1965.

[39] Glover F., *"Tabu search: A tutorial"*, J. Interfaces, 20(4) 74-94, 1990.

[40] Guignard M. and Kim S., *"Lagrangian decomposition: A model yielding stronger Lagrangian bounds"*, Mathematical Programming, 39, 215-228, 1987.

[41] Guillot P., *"Cryptographical Boolean functions construction from linear codes"*, BFCA'05, 141-167, 2005

[42] Gupta K. C. and Sarkar P., *"Improved construction of nonlinear resilient s-boxes"*, Advances in Cryptology-ASIACRYPT 2002, LNCS, 2501, 466-483, 2002.

[43] Gupta P., Mehlawat, Mukesh K. and Mittal G., *"Asset portfolio optimization using support vector machines and real coded genetic algorithm"*, J. Global Optimization, 53(2), 297-315, 2012.

[44] Hoffman K. L. and Padberg M., *"Solving airline crew scheduling problems by branch-and-cut"*, Management Science, 39, 657-682, 1993.

[45] Holland J. H., *"Genetic algorithms and the optimal allocation of trials"*, SIAM J. Computing, 2(2), 88-105, 1973.

[46] Izbenko Y., Kovtun V. and Kuznetsov A., *"The design of Boolean functions by modified hill climbing method"*, Information Technology, Sixth Int. Conf., 356 - 361, 2009.

[47] Johansson T. and Jonsson F., *"Fast correlation attacks based on Turbo code techniques"*, Advances in Cryptology- CRYPTO'99, LNCS,1666, 181-197, 1999.

[48] Johansson T. and Jonsson F., *"Improved fast correlation attack on Stream cipher via convolutional codes"*, Advances in Cryptology-EUROCRYPT'99, LNCS, 1592, 347-362, 1999.

[49] Johansson T. and Pasalic E., *"A construction of resilient functions with high non-linearity"*, IEEE Trans. on Information Theory, 49(2), 494-501, 2003.

[50] Kavut S. and Yucel M. D., *"Improved cost function in design of Boolean functions satisfying multiple critaria"*, INDOCRYPT, 121-134, 2003.

[51] Khoo K. and Gong G., *"New constructions for resilient and highly nonlinear Boolean functions"*, Information Security and Privacy, ACISP 2003, LNCS, 2727, 498-509, 2003.

[52] Kurosawa K. and Satoh T., *"Design of SAC/PC(l) of order k Boolean functions and three other cryptographic criteria"*, Advances in Cryptology-EUROCRYPT'97, LNCS,1233, 434-449, 1997.

[53] Kirkpatrick S. J., Gelatt C. D. and Vecchi M. P., *"Optimization by simulated annealing"*, J. Science, 220(4958), 671-680, 1983.

[54] Knowres J. and Corne D., *"The Pareto archived evolution strategy:A new base line for multiobjective optimization"*,Proc. 999 Congress on Evolutionary Computation, IEEE Press, 98-105, 1999.

[55] Korovin K., Tsiskaridze N. and Voronkov A., *"Implementing conflict resolution"*, Ershov Memorial Conference 2011,LNCS, 7162, 362-376, 2012.

[56] Kurosawa K., Satoh T. and Yamamoto K., *"Highly nonlinear t-resilient functions"*, J. Universal Computer Science, 3(6), 721-729, 1997.

[57] Land A. H. and Doig A. G., *"An automatic method for solving discrete programming problems"*, Econometrica, 28, 97-520, 1960.

[58] Little J. D. C., Murthy K. G., Sweeney D. W. and Karel C., *"An algorithm for the traveling salesman problem"*, Operations Research, 11, 972-989, 1963.

[59] Maitra S., *"Correlation immune Boolean functions with very high nonlinearity"*, Cryptography, eprint.iacr.org/2000/054.ps.

[60] Maitra S., *"Highly nonlinear balanced Boolean functions with very good autocorrelation property"*, Workshop on Coding and Cryptography, WCC 2001, Electric Notes in Discrete Mathematics, 6, 1-557, 2001.

[61] Maitra S., *"On nonlinearity and autocorrelation properties of correlation immune Boolean functions"*, J. Information Science and Engerineeing, 20, 305-323, 2004.

[62] Maitra S. and Pasalic E., *"Further constructions of resilient Boolean functions with very high nonlinearity"*, IEEE Trans. on Infor. Theory, 48(7), 1825-1834, 2002.

[63] Maity S. and Johansson T., *"Construction of cryptographically important Boolean functions"*, Progress in Cryptology- INDOCRYPT-2002, LNCS, 2551, 234-245, 2002.

[64] Maity S. and Maitra S., *"Minimum distance between bent and 1-resilient Boolean functions"*, Fast Software Encryption, FSE 2004, LNCS, 3017, 143-160, 2004.

[65] Matsui M., *"Linear cryptanalysis method for DES cipher"*, Advances in Cryptology- EUROCRYPT 93, LNCS, 765, 386 - 397, 1994.

[66] McFarland R.L., *"A family of difference sets in non-cyclic groups"*, J. Combinatorial Theory, 15, 1-10, 1973.

[67] McWilliams F.J. and Sloane N.J.A., *"The theory of error-correcting codes"*, North-Holland, 2, 168-181, 1978.

[68] Meier W. and Staffelbach O., *"Fast correlation attacks on certain stream ciphers"*, J. Cryptology, 1(3), 159-176, 1989.

[69] Millan W., Clark A. and Dawson E., *"An effective genetic algorithm for finding highly non-linear Boolean functions"*, Proc. First Int. Conf. on Info. and Comm. Security, LNCS, 1334, 149-158, 1997.

[70] Millan W., Clark A. and Dawson E., *"Heuristic desing of cryptographically strong balanced Boolean functions"*, Proc. Advances in Cryptology - EUROCRYPT 98, LNCS, 1403, 489-499, 1998.

[71] Minsky M., *"Steps towards artificial intelligence"*, Proc. Institute of Radio Engineers, 49(1), 8-30, 1961.

[72] Mishra S. K. and Jaiswal M., *"Optimality conditions and duality for non-differentiable multiobjective semi-infinite programming"*, Vietnam J. Mathematics, 40, 331-343, 2012.

[73] Mohanty B. K. and Vijayaraghavan T. A. S., *"A multiobjective programming problem and its equivalent goal programming problem with approperiate priorities and aspiration levels: a fuzzy approach"*, Computers and Research, 22, 771-778, 1995.

[74] Nagaraja G. and Krishna G., *"An algorithm for the solution of linear inequalities"* IEEE Trans. on Computers, 23(4), 421-427, 1974.

[75] Nemhauser G. L. and Wolsey L. A., *"Integer and Combinatorial Optimization"*, John Wiley and Sons, 1988.

[76] Orsi R., Helmke U. and Moore J. B., *"A Newton-like method for solving rank constrained linear matrix inequalities"*, Proc. 43rd IEEE Conf. on Decision and Control , 3138-3144, 2004.

[77] Padberg M. W. , *"Combinatorial optimization"*, Mathematical Programming Study, 12, 1-7, 1980.

[78] Pasalic E., *"A design of Boolean functions resistant to (fast) algebraic cryptanalysis with efficient implementation"*, Cryptogr. Commun., LLC, 4, 25-45, 2012.

[79] Puchinger J., *"Combining Metaheuristics and Integer Programming for Solving Cutting and Packing Problems"*, PhD Thesis, Vienna University of Technology, Institute of Computer Graphics and Algorithms, January 2006.

[80] Rothaus O.S., *"On bent functions"*, J. Combinatorial Theory, 20(3), 300-305, 1976.

[81] Roy B., *"A brief outline of research on correlation immune functions"*, Information Security and Privacy, ACISP 2002, LNCS, 2384, 379-394,2002.

[82] Sarkar P. and Maitra S., *"Construction of nonlinear Boolean functions with important cryptographic properties"*, Advances in Cryptology-EUROCRYPT 2000, LNCS,1807, 485-506, 2000.

[83] Sarkar P. and Maitra S., *"Nonlinearity bounds and construction of resilient Boolean functions"*, Advances in Cryptology - Crypto 2000, LNCS, 1880, 515-532, 2000.

[84] Seberry J., Zhang X. M. and Zheng Y., *"The relationship between propagation characteristics and nonlinearity of cryptographic functions"*, J. Universal Computer Science, 1(2), 136-150, 1995.

[85] Seberry J., Zhang X. M. and Zheng Y., *"Nonlinearity and propagation characteristics of balanced Boolean functions"*, Information and Computation, 119(1) , 1-13, 1995.

[86] Siegenthaler T., *Correlation immunity of nonlinear combining functions for cryptographic applications"*, IEEE Trans. on Information Theory, 30(5), 776-780, 1984.

[87] Siegenthaler T., *Decrypting a class of stream cipher using ciphertext only"*, IEEE Trans. on Computers, 34(1), 2010-2017, 1985.

[88] Silver E. A., *"An overview of heuristic solution methods"*, J. Operational Research Society, 55, 936-956, 2004.

[89] Srinivas, N., and Deb, K., *"Muiltiobjective optimization using nondominated sorting in genetic algorithms"*, Evolutionary Computation, 2, 221-248, 1994.

[90] Taha H. A., *"Operations Research - An Introduction "*, Prentice Hall of India, New Delhi, 7th edition, 2007.

[91] Tang D., Zhang W. and Tang X., *"Construction of balanced Boolean functions with high nonlinearity and good autocorrelation properties"*, Des. Codes Cryptogr, 67(1), 77-91, 2013.

[92] Tang Y., Reed P. and Wagener T., *"How effective and efficient are multiobjective evolutionary algorithms at hydrologic model calibration"*, Hydrology and Earth System Sci, 10, 289-307, 2006.

[93] Tarannikov Y., *"New constructions of resilient Boolean funtions with maximal nonlinearity"*, Fast Software Encryption, FSE 2001, LNCS, 2355, 66-77, 2002.

[94] Tarannikov Y., *"On resilient Boolean functions with maximal nonlinearity"*, Progress in Cryptology- INDOCRYPT 2000, LNCS, 1977, 19-30, 2000.

[95] Thavaneswaran A., Appadoo S. S. and Samanta M., *" Random coefficient GARCH models"*, Mathematical and Computer Modelling, 41(6-7), 723-733, 2005.

[96] Verma A.K. and Ramesh P.G., *"Multi-objective initial preventive maintenance scheduling for large engineering plants"*, Int. J.Reliability, Quality and Safety Engineering, 14(3), 241-250, 2007.

[97] Verma A.K., Srividya A. and Ramesh P.G., *"A systems approach to integrated e-maintenance of large engineering plants"*, Int. J. Systems Assurance Engineering and Management, 1(3), 239-245, 2010.

[98] Wang Y. and Pham H., *A multi-objective optimization of imperfect preventive maintenance policy for dependent competing risk systems with hidden failure"*, IEEE Trans. on Reliability, 60(4), 770-781, 2011.

[99] Warmack R.E. and Gonzalez R.C., *"An algorithm for the optimal solution of linear inequalities and its application to pattern recognition"*, IEEE Trans. on Computers, 22(12), 1065-1075, 1973.

[100] Wei Y., Ouyang N. and Hu N., *"New construction of Boolean functions which satisfy multiple cryptographic criteria"*, Communications and Networking in China, CHINACOM 2009, 1-6, 2009.

[101] Wu C. K. and Dawson E., *"Construction of correlation immune Boolean function"*, Australasian J. of Combinatorics, 21, 141- 166, 2000.

[102] Young R. D., *"A primal (all integer) integer programming algorithm"*, J. Research of the National Bureau of Standards, 69B, 1965, 213-250.

[103] Youssef A. M. and Gong G., *"Boolean functions with large distance to all bijective monomials: an odd case"*, Selected Areas in Cryptography, SAC, 2001, LNCS, 2259, 49-59, 2001.

[104] Zhang X. M. and Zheng Y., *"On nonlinear resilient Boolean functions"*, Advances in Cryptology- EUROCRYPT'95, LNCS, 921, 274-288, 1995.

[105] Zhang X. M., Zheng Y. and Imai H., *"Relating differential distribution table to other properties of substitution boxes"*, Design, Codes and Cryptograpphy, 19, 45-63, 2000.

[106] Zheng Y and Zhang X. M., *"New results on correlation immunity"*, Information Security and Cryptology-ICISC 2000, LNCS, 2015, 49-63, 2001.