International Journal of Applied Engineering Research ISSN 0973-4562 Volume 10, Number 14 (2015) pp 35002-35005

Cryptanalysis of Data Encryption Standard Cipher System using Heuristic Search Algorithms

G.Selvi
Research Scholar
Faculty of Information&Communication Engineering,
Anna University, Chennai, India
Email: sellviig@gmail.com

Dr. T. Purushothaman
Associate Professor,
Department of Computer Science and Engineering,
Government College of Technology, Coimbatore, India
Email : purushgct@yahoo.com

Abstract - Cryptanalysis using heuristics is significant in attacks of block ciphers. A lot of work has already been done on this. Almost all of them deal with cryptanalysis either by using classical heuristic search algorithms or a modified version of those algorithms. Hence, in this paper, cryptanalysis is applied to block ciphers such as Data Encryption Standard (DES) and Simplified DES (S-DES) using heuristic search algorithms such as Bees Algorithm (BA), Bacterial Foraging Optimization (BFO) algorithm and Cuckoo search (CS) algorithm with a principal focus on the strength of the heuristic attacks over these block ciphers. The determination and comparison of the time consumed by the aforesaid algorithms to break DES and S-DES encrypted ciphers with select frequently used heuristic search algorithms i.e. Genetic Algorithm (GA), Particle Swarm Optimization (PSO) algorithm, Simulated Annealing (SA) and Tabu Search (TS) algorithm.

Keywords: Data Encryption Standard (DES), Simplified DES (S-DES), Bees Algorithm (BA), Bacterial Foraging Optimization (BFO) algorithm, Cuckoo search (CS)

Introduction

Information processing revolution has made significant progress: yet, it also suffers from concurrent and persistent danger- security threat. To outwit this cyber menace, cryptographic algorithm comes into play through a process of encryption and decryption through encryption algorithm, either symmetric or asymmetric. In a symmetric encryption, a single key is used for both encryption and decryption whereas in asymmetric encryption, different keys are used [12]. Block cipher systems belong to symmetric cryptographic systems [10] [1] and are designed to encrypt data in specific size [9]. Cryptanalysis can, thus be defined as the study of such ciphers, cipher text or cryptosystems with a view of finding weakness in them [8].

A Few Historical Inputs

A knowledge of the earlier studies is useful for the present study. In 2005, Laskari *et al.* [2] have established the particle swarm optimization method to address a problem introduced by the cryptanalysis of block-cipher cryptosystems. In 2007, Nalini *et al.* [3] studied the attacks on DES and S-DES using some prominent heuristics search algorithms such as GA, adaptive GA, PSO, TS and SA. In 2008 [3], they established the applicability of a couple of optimization heuristics to cryptanalysis studies in DES. In 2009, Hasan Mohammed Hasan Husei *et al.* [4] have discussed the Genetic Algorithm for the cryptanalysis of DES-like systems to find out the underlying key. In 2009, Poonam Garg [5] made the cryptanalysis of S-DES using memetic algorithm, genetic algorithm and simulated annealing and investigated the performance for the cryptanalysis on SDES. In 2010, Siva Sathya *et al.* [6] described a Multi-population Genetic Algorithm called Nomadic Genetic Algorithm (NGA) for breaking the message encrypted by DES. In 2011, Vimalathithan *et al.* [7] applied Genetic Swarm Optimization algorithm in the field of cryptanalysis for attacking DES by combining the effectiveness of Genetic algorithm and Particle Swarm optimization.

This paper attempts a structured cryptanalysis of DES and S-DES algorithms using these newly developed algorithms (BA, BFO and CS) along with proven algorithms such as PSO, GA, MA, TS and SA and also the conventional cryptanalysis based on unigram, bigrams and even trigrams. Section 3 contains description and pseudo code on BFO, BA and CS. Section 4 has a brief overview of DES and S-DES Algorithm. Section 5 is an analysis of results of cryptanalysis. Section 6 forms the conclusion. The working platform is MATLAB.

Topical Heuristics

Bees Algorithm (BA)

The BA [15] is a population based intelligent optimization algorithm, inspired by the natural foraging mechanism of honeybees. The pseudo code of the bees' algorithm is given below

Initialize the population with random N_{sc} solutions

Evaluate the population fitness

While (termination criteria not met)

- **Select** sites (M_{vs}) for neighborhood search
- **Recruit** bees for selected sites and evaluate fitnesses
- **Select** the fittest bee from each site
- **Assign** remaining $(N_{sc} - M_{vs})$ bees to search randomly and evaluate their fitnesses

End while

Bacterial Foraging Optimization (BFO) Algorithm

The BFO [16] is stimulated by the pattern revealed by bacterial foraging behaviors. The algorithm follows four important mechanisms such as chemotaxis, swarming, reproduction and elimination-dispersal.

i) Chemotaxis: This process represents the movement of an E.coli cell which is represented in computational chemotaxis as

$$\theta^i(j+1,k,l) = \theta^i(j,k,l) + C(i)\frac{\Delta(i)}{\sqrt{\Delta^T\Delta(i)}}$$

where, $\theta^i(j,k,l)$ indicates the i^{th} bacterium at j^{th} chemotactic, k^{th} reproductive and l^{th} elimination dispersal step, $C(i)$ is the step size taken in the random direction specified by the tumble (run length unit) and Δ represents a unit length vector in the random direction.

ii) Swarming: A group of E.coli cells arrange themselves in an itinerant ring by moving up the nutrient gradient when placed among a semisolid matrix with a single nutrient chemo effecter.

iii) Reproduction: This maintains the swarm size constant.

iv) Elimination and Dispersal: Gradual or abrupt changes in the local environment, where a bacterium population resides, may happen due to various causes e.g. a considerable rise in temperature.

A simple pseudo code of the bacterial foraging algorithm [16] [18] [19] is given below

Initialization of BFO parameters
For every elimination-dispersal loop
  For every reproduction step
    For every chemotaxis step
      For each bacteria
        Compute fitness
        Keep the details of the bacteria
        Tumble move by bacteria
        Swim move by bacteria
      End for
    End for
  End for
End for

Cuckoo Search (CS) Algorithm

The CS algorithm [17] was stimulated by the obligate brood parasitism of some cuckoo species by laying their eggs in the nests of other birds.

Each egg in a host nest represents a solution, whereas a cuckoo egg represents a new solution. The goal is to employ the new and possibly better solutions (cuckoos) to replace poor solutions in the nests. In the simplest form, each nest has one egg. The algorithm can be extended to more complex cases, where each nest has several eggs indicating a set of solutions. A simple pseudo code of the cuckoo search [14] algorithm is given below

Generate initial population of host nests
while (termination criteria or maximum number of iterations met)
  Get a cuckoo (i) randomly by Levy flights
  Evaluate its quality/fitness F
  Select a nest among N (say j) randomly
  if (Fi>Fj)
    Replace j by the new solution
  end
  Discard worse nests

Replace new ones via Levy flights but at new locations Keep the best solutions (or nests with quality solutions) Rank the solutions and find the current best
end while

Objective function

In most of the literary works, the unigram, bigram and trigram statistics are used for cryptanalysis. Increased complexity and meaningless results of their study lead to dictionary search resulting in performance of key breaking. The objective function (or fitness function) is given here as

$$F_i = \underset{i \in solutionset}{\arg\min} \left(Decipher(i) - Dictionary(i) \right)$$

where, $Decipher(i) - Dictionary(i)$ shows the difference between the deciphered text using i^{th} key (solution) and the closest dictionary text. Exact cracking of the key makes this term to zero. Here, the objective is designed as a minimization function, which means once the solution which gives zero F, the key is considered to be hacked.

DES and S-DES Algorithm

The DES Algorithm

The DES operates on 64-bit blocks of plaintext by utilizing a 64-bit key. Every eight bit of the 64-bit key is used for parity checking and so by discarding them, 56-bit key is obtained. After performing an initial permutation, the 64-bit input is divided into two parts i.e., right half (R0) and left half (L0), and both half should be in 32 bits in length. It encrypts the plaintext by 16 iterations or rounds. In each round, the following four operations are performed.

a) Using an expansion permutation function, the 32-bit right half of the plaintext is expanded to 48-bit
b) Resultant is XORed with a 48-bit sub-key K1
c) Then it is transformed into a 32-bit output by feeding the input into eight substitution boxes (S-boxes)
d) A straight permutation (P-permutation) is done over the resultant and the output is XORed with the initial left half L_0, in order to obtain the new right half R0. Subsequently, the initial right half R0 now becomes the new left half L_0

The four operations are repeatedly performed using the lastly obtained plain text. After 16 iterations, the right (R16) and left (L16) halves are concatenated and a final permutation, which is the inverse of the initial permutation, is performed to produce the cipher text. It is to be noted that for each of the 16 rounds of DES, a different 48-bit sub keys are generated. These sub-keys are determined by initially dividing the 56-bits into two 28-bits and then both the halves are shifted one or two bits left based on the round number [13].

The S-DES Algorithm

The encryption operation, an 8-bit block of plaintext and a 10-bit key are used as input and an 8-bit block of cipher text is produced as output, whereas, for decryption operation, an 8-bit block of cipher text and the same 10-bit key are used as input and the original 8-bit block of plaintext is produced as output. The encryption algorithm is performed in five steps: (1) an initial permutation (IP), (2) a complex function called f_K, where both permutation and substitution operations are performed based on the key input, (3) a simple permutation function (SW), which switches the two halves of the data, (4) again performing the function f_K, and (5) inverse permutation (IP^{-1}). The same steps are followed for the decryption operation, where the output of encryption i.e., 8-bit cipher text is decrypted to produce the original plaintext [11].

Experiments and Results

The cryptanalysis of DES and SDES using classical heuristics such as GA,PSO,SA,TS and newly emerged heuristics such as BA,BFO and CS were done in the working platform of MATLAB.

Experiments

Ten experiments are conducted to crack 10-bit key of S-DES and 16-bit key of DES. In every experiment, different keys with same size were used. Every experiment has five rounds of cracking in which different plain texts but same key were used. The plain text and keys that are used in different rounds of cracks and experiments are given in Table 1 as an illustration.

Table 1 The plain text and 10-bit keys that are used in various rounds and experiments of cryptanalysis of S-DES

Experiment no	Round no	1	2	3	4	5
	Plain text	Hi	How	Hello	Encrypt	Positive
1		0001000001	0001000001	0001000001	0001000001	0001000001
2		0001000100	0001000100	0001000100	0001000100	0001000100
3		0001000101	0001000101	0001000101	0001000101	0001000101
4		0001000111	0001000111	0001000111	0001000111	0001000111
5		0001001011	0001001011	0001001011	0001001011	0001001011
6		0001001100	0001001100	0001001100	0001001100	0001001100
7		0001001111	0001001111	0001001111	0001001111	0001001111
8		0001010001	0001010001	0001010001	0001010001	0001010001
9		0001010011	0001010011	0001010011	0001010011	0001010011
10		0001010100	0001010100	0001010100	0001010100	0001010100

Table 2 The plain text and 16-bit keys are used in various rounds and experiments of cryptanalysis of DES

Experiment no	Round no	1	2	3	4	5
	Plain text	Hi	How	Hello	Encrypt	Positive
1		0100011001000100	0100011001000100	0100011001000100	0100011001000100	0100011001000100
2		0100011101000101	0100011101000101	0100011101000101	0100011101000101	0100011101000101
3		0100100001000001	0100100001000001	0100100001000001	0100100001000001	0100100001000001
4		0100110001010011	0100110001010011	0100110001010011	0100110001010011	0100110001010011
5		0100110101001011	0100110101001011	0100110101001011	0100110101001011	0100110101001011
6		0100111101001001	0100111101001001	0100111101001001	0100111101001001	0100111101001001
7		0101000001010011	0101000001010011	0101000001010011	0101000001010011	0101000001010011
8		0101000101000010	0101000101000010	0101000101000010	0101000101000010	0101000101000010
9		0101001101000001	0101001101000001	0101001101000001	0101001101000001	0101001101000001
10		0101010001001000	0101010001001000	0101010001001000	0101010001001000	0101010001001000

In the first round of first experiment (of cracking 16-bit DES key), *0100011001000100* and *Hi* are used as key and plain text respectively. In the subsequent rounds of experiments, the size of the plain text was increased and the cracking of same key was done. But in the subsequent experiments, the set of plain text was unchanged whereas the cracking was done for different keys (like *0100011101000101, 0100100001000001 and 0100110001010011* in experiments 2, 3 and 4 respectively and so on). Table 1 show the experimental data that are used

in cracking 10-bit S-DES key and Table 2 also presents the experimental data that are used in cracking 16-bit DES key.

Results

The results that are obtained for the given heuristic search algorithms after carrying out all the above mentioned experiments are illustrated here. Firstly, the convergence graph of all the algorithms is affixed to visualize the performance of them in every round of cracking a 16-bit and 32-bit DES keys and S-DES key. Secondly, the CPU time for cracking the keys is determined in every round of experiment. Figures 1.1 to 1.3 represent the Cipher breaking performance by the algorithms BFO, BA, and CS over a 16-bit DES key *0100011001000100*.

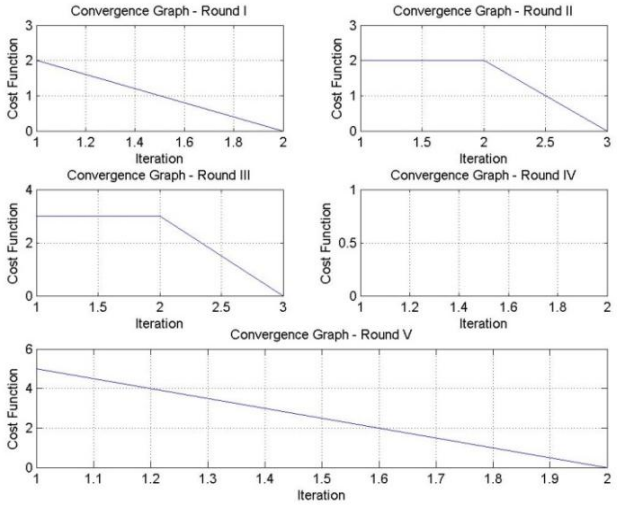

Figure 1.1 Cipher breaking performance by BFO over a 16-bit DES key *0100011001000100*

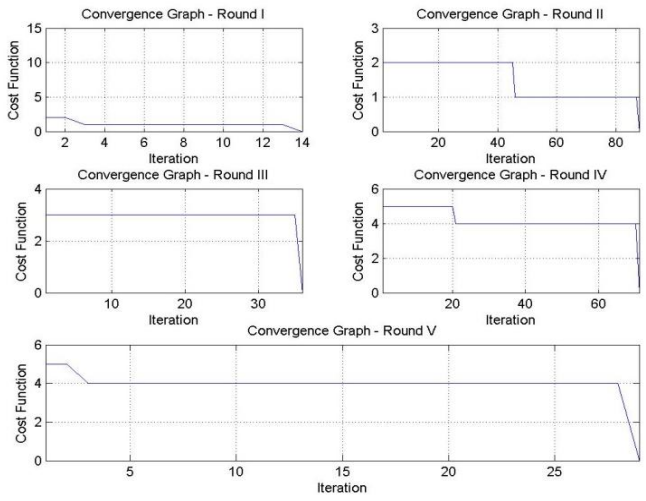

Figure 1.2 Cipher breaking performance by BA over a 16-bit DES key *0100011001000100*

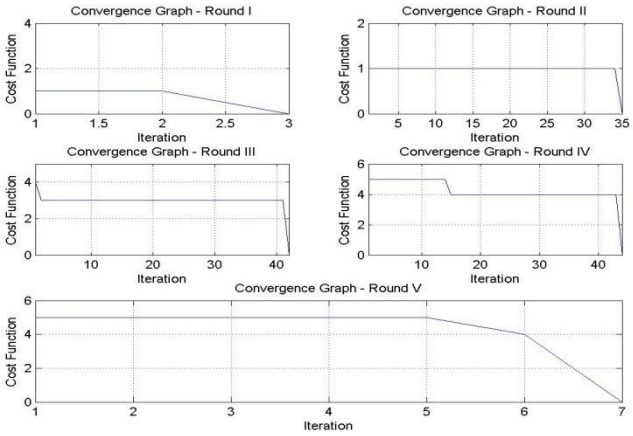

Figure 1.3 Cipher breaking performance by CS over a 16-bit DES key *0100011001000100*

Analysis

In the first round of first experiment, BA wins by taking only 0.62 seconds to break the key whereas SA, CS, PSO and TS take 1.02, 3.49, 1.20 and 2.86 seconds respectively. In this way, BA made 13 wins out of 50 rounds of experiments, while SA, CS, PSO and TS made 13, 7, 9 and 8 wins respectively. This analytical value takes BA and SA at the highest position because they achieve a success rate of 0.26 (=13/50) or 26%. CS, PSO and TS could make only 0.14, 0.18 and 0.16 success rates respectively, which are only 54%, 69%, 62% of the success rate of SA and BA.

As regards breaking the 16-bit and 32-bit DES keys in every round of experiment, BFO breaks 32-bit key with a success rate of 4%. In breaking 16-bit key, SA, BA, CS, PSO and TS achieve a success rate of 16%, 22%, 22%, 22% and 18% respectively, while in breaking 32-bit key, they achieve 28%, 6%, 20%, 22% and 20%. In both the cases, PSO makes a consistent success rate of 22%.

SA, PSO and BA break ten different S-DES keys faster than the other algorithms. BA takes only 0.51-0.83 seconds to crack five different cipher texts that are enciphered by the S-DES key '0001000001' whereas SA and PSO takes 0.57-1.21 seconds and 0.63-1.15 seconds, respectively. But among the 10 keys, PSO breaks 6 of the keys efficiently rather than the other two, whereas, BA and SA breaks individually performs over 2 of the keys.

As a best case, SA takes only 1-1.5 seconds to break a 16-bit key and 0.48-5.77 seconds to break a 32-bit key; even the size of the plain text varies. Tables 3.1 to 3.3 show cumulative statistical performance of the heuristic attacks over S-DES, 16-bit DES key and 32-bit DES key.

Table 3.1 Cumulative statistical performance of the heuristic attacks over S-DES

Sl. No	Algorithms	Mean	S.D.
1	SA	0.90	0.45
2	GA	$7.58\ X10^{13}$	$9.83\ X10^{123}$
3	BFO	6.61	2.36
4	BA	0.96	0.75
5	CS	4.05	6.89
6	PSO	0.80	0.40
7	TS	1.94	2.36

Table 3.2 Cumulative statistical performance of the heuristic attacks over 16-bit DES key

Sl. No	Algorithms	Mean	S.D.
1	SA	5.13	6.65
2	GA	$2.57X10^{13}$	$0.01X10^{13}$
3	BFO	72.31	187.60
4	BA	13.36	33.27
5	CS	10.46	19.94
6	PSO	2.98	2.39
7	TS	13.46	29.99

Table 3.3 Cumulative statistical performance of the heuristic attacks over 32-bit DES key

Sl. No	Algorithms	Mean	S.D.
1	SA	31.12	48.07
2	GA	$1.75\ X10^{13}$	$1.01\ X10^{13}$
3	BFO	422.81	654.27
4	BA	286.87	995.75
5	CS	40.64	86.17
6	PSO	21.47	26.17
7	TS	42.96	112.75

At the end of the average of all the performance in hacking DES keys, PSO takes only 3 seconds to hack 16-bit key, which is58%, 4%, 22%, 28% and 22% of the average time taken by SA, BFO, BA, CS and TS respectively. For hacking a 32-bit DES key, PSO takes only 21.47 seconds, which is just only 67%, 5%, 7%, 52% and 49% of the average time taken by SA, BFO, BA, CS and TS respectively.

Conclusion

In this paper, the cryptanalysis over DES and S-DES is made using the most popular heuristic search algorithms such as GA, TS, PSO and SA along with CS, BA and BFO algorithms. The study helped to analyze the weakness of the block ciphers when heuristic based attacks were made against them. After the analysis, the following conclusions are drawn: BFO is weak in attacking the block ciphers, BA and CS are successful only in a few cases, PSO has the highest success rate in attack, TS's defence is rare whereas SA is constant in defence and GA has the worst heuristic attack over the block ciphers. The study sums up through comparison the strengths and weaknesses of all these algorithms. To conclude, if any block cipher is robust against PSO based attack, it can as well be the best against any other heuristic based attacks.

References

[1] Khaled M. Alallayah, Waiel F. Abd El-Wahed, Mohamed Amin and Alaa H. Alhamami, "Attack of Against Simplified Data Encryption Standard Cipher System Using Neural Networks" Journal of Computer Science,Vol.6,No.1, pp. 29-35,2010.

[2] E.C. Laskari, G.C. Meletiou,Y.C. Stamatioud and M.N. Vrahatis, "Evolutionary computation based cryptanalysis:A first study", Journal of nonlinear analysis,Vol.63, No.5-7,pp.e823-e830, 2005.

[3] Nalini N and Raghavendra Rao G, "Cryptanalysis of Block Ciphers via Improvised Particle Swarm Optimization and Extended Simulated Annealing Techniques", International Journal of Network Security, Vol.6, No.3, pp.342–353, May 2008.

[4] Hasan Mohammed Hasan Husei, Bayoumi I. Bayoumi,Fathy Saad Holail, Bahaa Eldin M. Hasan and Mohammed Z. Abd El-Mageed, " A Genetic Algorithm for Cryptanalysis with Application to DES-like Systems", International Journal of Network Security, Vol.8, No.2, pp.177–186, Mar. 2009.

[5] Poonam Garg, Cryptanalysis of SDES via Evolutionary Computation Techniques", (IJCSIS) International Journal of Computer Science and Information Security,Vol. 1, No. 1,7 pages, May 2009.

[6] Siva Sathya S,Chithralekha T and AnandaKumar P, "Nomadic Genetic Algorithm for Cryptanalysis of DES 16", International Journal of Computer Theory and Engineering, Vol. 2, No. 3, pp. 411-415, June 2010.

[7] Vimalathithan R and Valarmathi M.L, "Cryptanalysis of DES using Computational Intelligence", WSEAS Transactions On Computers ,Vol. 10,No.7, pp.210-219, 2011

[8]Srinivasa Rao K V,Rama Krishna M and Bujji Babu D,"cryptanalysis Of a Feistal Type Block Cipher by Feed Forward Neural Network Using Right Sigmoidal Signals", International Journal of Soft Computing,Vol.4,No.3,pp.131-135,2009.

[9]Soukaena H. Hashem, Mohammad A. AL-Hamami and Alaa H. AL-Hamami,"Developing a Block-Cipher-Key Generator Using Philosophy of Data Fusion

Technique",Journal of Emerging Trends in Computing and Information Sciences,Vol.2, No.5,pp.222-227, May 2011.

[10]Laskari E C, Meletiou G C, Stamatiou Y C and Vrahatis M N,"Applying evolutionary computation methods for the cryptanalysis of Feistel ciphers",Journal of Applied Mathematics and Computation - AMC , Vol.184, No. 1, pp. 63-72, 2007,doi: 10.1016/j.amc.2005.11.176.

[11]Poonam Garg,"A Comparison of Memetic & Tabu Search for the Cryptanalysis of Simplified Data Encryption Standard Algorithm", Journal of Theoretical and Applied Information Technology, Vol.4,No.4,pp.360-366,Apr 2008.

[12] Omary F, Tragha A, Bellaachia A and Mouloudi A, "Design and Evaluation of Two Symmetrical Evolutionist-Based Ciphering Algorithms",International Journal of Computer Science and Network Security,Vol.7, No.2,pp.181-190, February 2007, doi=10.1.1.103.5113

[13] Vishwanath Patel, R. C. Joshi and A. K. Saxena, "FPGA Implementation Of DES Using Pipelining Concept With Skew Core Key-Scheduling",Journal of Theoretical and Applied Information Technology(JATIT),pp.295-300,2005-2009.

[14] Ehsan Valian, Shahram Mohanna and Saeed Tavakoli,"Improved Cuckoo Search Algorithm For Feedforward Neural Network Training",International Journal of Artificial Intelligence & Applications (IJAIA), Vol.2, No.3,pp.36-43,July 2011, doi : 10.5121/ijaia.2011.2304

[15] Pham DT, Ghanbarzadeh A, Koc E, Otri S, Rahim S and Zaidi M., "The Bees Algorithm" Technical Note, Manufacturing Engineering Centre, Cardiff University, UK, 2005

[16] K. M. Passino, "Bacterial Foraging Optimization", International Journal of Swarm Intelligence Research, Vol.1, No.1, pp.1-16, 2010.

[17] X.-S. Yang; S. Deb ,"Cuckoo search via Lévy flights" In proceedings of World Congress on Nature & Biologically Inspired Computing (NaBIC 2009) pp. 210–214, 2009

[18] Acharya, Panda, Mishra and Lakshmi, "Bacteria Foraging Based Independent Component Analysis", In proceedings of International conference on Computational Intelligence and Multimedia Applications, pp.527-531, December 2007

[19] Subramanian and Padma, "Optimal Design Of Single Phase Transformer Using Bacterial Foraging Algorithm", International Journal of Engineering Science and Technology, Vol.3, No.4, 2011