A New Biological Operator in Genetic Algorithm for Class Scheduling Problem

R.Lakshmi
Assistant Professor
Department of Computer Science
Pondicherry University
Puducherry - India

K.Vivekanandhan,
Professor
Department of Computer Science &Engineering
Pondicherry Engg. College
Puducherry - India

R.Brintha
Department of Computer Science
Pondicherry University
Puducherry - India

ABSTRACT
This paper describes an innovative approach to solve Class Scheduling problem which is a constraint combinatorial NP hard problem. From the wonders of natural evolution, an important phenomenon of RNA interference induced silencing complex (RISC) can be used as Interference Induced Silencing operator and it is incorporated into the Genetic Algorithm to solve any practical problems like Class Scheduling problem. The aim of this research is to create an automated system for class scheduling problem using Genetic Algorithm to the extent by a new biologically inspired operator, Interference Induced Silencing (IIS) operator that it can be used to set the instant specific preferences to generate the effective time table with the probabilistic operators like crossover and mutation. The framework of the fitness function has considered the hard constraints and the soft constraints. The results were proved to be efficient than the simple Genetic algorithm.

Keywords
Class Scheduling Problem, Genetic Algorithm, Interference Induced Silencing Operator, Swap Mutation, Preference Settings, Hard Constraint and Soft Constraint.

1. INTRODUCTION
Genetic algorithm is an adaptive heuristic search algorithm based on the principle of natural genetics and survival of the fittest**.** Genetic algorithms search for solutions represented as points on the search space, by emulating biological selection and reproduction. In a Genetic Algorithm (GA), the parameters of the problem to be optimized are encoded into a finite length string, usually a string of bits. Each parameter is represented by a portion of the string. The string is called a chromosome, and each bit is called a gene**.** Each chromosome is given a measure of "fitness" by the fitness function, sometimes called the objective or evaluation function. The fitness of a chromosome determines its ability to survive and reproduce offspring. The "least fit" or weakest chromosomes of the population are displaced by more fit chromosomes. The transition rules used by the GA for evolving a population from one generation to the next are called genetic recombination and reproduction operators. Holland was also the first to explicitly propose crossover and other recombination operators. However, the seminal work in the field of genetic algorithms came in 1975, with the publication of the book Adaptation in Natural and Artificial Systems [1]. A Genetic algorithm [13] requires a process of initialization of population, evaluation of fitness, and reproduction which involves selection, crossover and mutation operators is represented in the following flowchart.

.

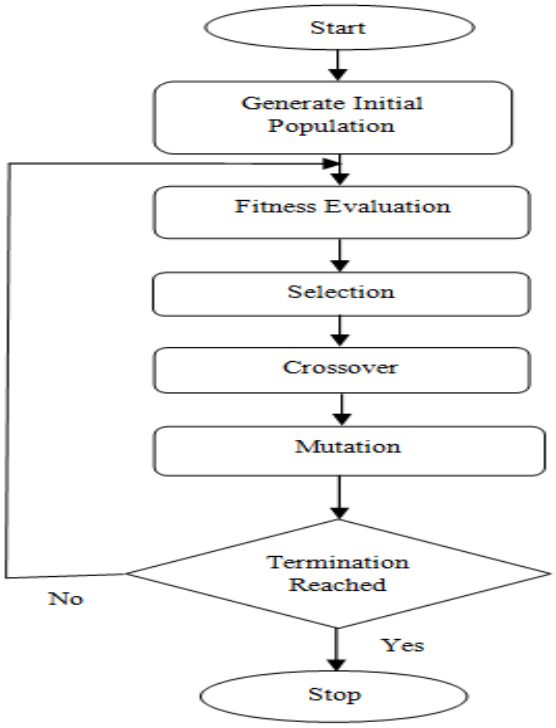


Figure 1: The Evolutionary Mechanism

The flowchart describes that the Genetic Algorithm begins with a set of candidate solutions (individuals/chromosomes) called population. A next generation population is obtained from solutions of old population called as fittest individuals. Candidate solutions which are chosen to form new solutions referred as offspring are selected according to their fitness. This process is repeated until a pre specified condition is met.

The rest of this paper is organized as follows. Section 2 discusses in greater detail the Class Scheduling Problem (CSP). Section 3 describes a brief summary of genetic algorithm approach to the CSP. Section 4 explains the working concept of interference induced silencing concept in GA for the CSP problem. Section 5 gives the results and statistics and Section 6 gives the conclusion.

2. CLASS SCHEDULING PROBLEM

Class Scheduling Problem (CSP) is characterized as constraint satisfaction problem that belongs to NP hard problems. For simple cases, heuristic search algorithm is used to find the optimal solution. When the requirements and inputs become complex, GA is used to find the good solution. Class scheduling task has always been solved by manually almost in all educational institutions. While framing up the time table enormous factors have to be taken into consideration. Considering all factors and also the time table is made by an experienced person, but still it is not satisfactory for everyone, because it does not satisfy all the requirements. For example if faculties want to impose their preconditions, then it will be a very big hectic work when we do manual arrangements in the time table. These precondition requirements and many constraints have been resolved effectively by the proposed system. Several approaches have been applied for solving class scheduling problem namely Hidden Markov Model (HMM), Simulated Annealing (SA), Knowledge based Genetic algorithms and other heuristic methods [7] [9] [11]. The class scheduling problem involves allocation of classes to professor, students, fixing rooms for the time-slot, assigning subjects, etc. An effective class scheduling problem is to satisfy the requirements and utilizing the space and human resources effectively. Making a Class schedule involves the constraint satisfaction which is categorized into hard constraints and soft constraints are given below.

2.1 Hard Constraints (HC)

Hard constraints are those that are adhered to the CSP [14], which may leads, to infeasible solution, when it is broken. The hard constraints used here are:

- HC1: Professor must have one class at a time.
- HC2: Students must have one class at a time.
- HC3: Rooms should accommodate all the Students in a class
- HC4:Laboratory schedule should not be overlapped by more than one course
- HC5: To allocate the lab hour to a particular course group, the laboratory should have sufficient equipments.

2.2 Soft Constraints (SC)

Soft constraints [15] are those that can be violated according to the situation, and still it leads to the feasible solution. The breaches must be minimized. It is not convenient for either students or professors, and it also makes more difficult to understand the academic schedules in departments. Some of the soft constraints identified are as follows:

- SC1: Professors may prefer the time slot
- SC2: Professors may prefer the classroom
- SC3:Allotment of classes distributed evenly over the week
- SC4:Classrooms that are closer to the parent department are allotted
- SC5:No repetition of the subject in the same day
- SC6:Professor should not handle more than one subject for the same class
  SC7: Classrooms should not be allocated that are much larger than the size of the class

These constraints are referred to as preferences in CSP.

The objects of the Class Scheduling Problem and their description are given in the following table:

Table 1: Object description for CSP

Professor	The Professor class has an ID and the name of the professor. It also contains a list of classes that a professor teaches.
Students Group	The Students Group class has an ID and the name of the student group, as well as the number of students (size of group). It also contains a list of classes that the group attends.
Classroom	The Room class has an ID and the name of the classroom, as well as the number of seats and information about equipments like computers, LCD projectors, etc. If the classroom has computers, it is expected that there is a computer for each seat. IDs are generated automatically.
Course	The Course class has an ID and the name of the course.
Course Class	It holds a reference to the course to which the class belongs, a reference to the professor who teaches, and a list of student groups that attend the class. It also stores how many seats (sum of student groups' sizes) are needed in the classroom, if the class requires computers in the classroom, and the duration of the class (in hours).

Once the valid chromosome is built by the objects of the Class Scheduling Problem that are shown in the Table1, the rooms (classroom, laboratory) and timeslots are assigned, in order to resolve the conflicts raised in the academic activities of each department due to those hard constraints and soft constraints.

Figure2 given below shows the time table slots for theory and laboratory of the Department of Computer Science. There are 35 time slots numbered from 1 to 35 allotted for a Class Group in a week is scheduled in the figure. The laboratory time-slots for the courses M.C.A and Engineering are shown in the time table and the symbol '*' is referred as free time-slots that can be utilized by other Departments.

Theory

Timeslot / Days	Mon	Tue	Wed	Thu	Fri
9-10	1	2	3	4	5
10-11	6	7	8	9	10
11-12	11	12	13	14	15
12-13	16	17	18	19	20
LUNCH BREAK					
14-15	21	22	23	24	25
15-16	26	27	28	29	30
16-17	31	32	33	34	35

Lab

Timeslot / Days	Mon	Tue	Wed	Thurs	Fri
9-10	M.C.A III	M.C.A I	*	*	*
10-11					
11-12	*	*		M.C.A II	
12-13					
LUNCH BREAK					
14-15	E-I	*	E-IV	E-III	E-II
15-16					
16-17					

Figure2: Institute Time Table

3. GENETIC ALGORITHM APPROACH

Many versions of GAs have been successfully applied to real-life situations. The general approach in applying GAs to timetabling problems is to use a GA to evolve an appropriate permutation and then use a heuristic method to construct feasible solutions according to the permutation, so as to satisfy the hard constraints and soft constraints. GA involves encoding, fitness evaluation, recombination techniques and these can be applied to our class scheduling problem is as follows in the subsequent sections.

3.1 Chromosome Representation

Encoding the class scheduling problem into gene form is represented as a collection of units of information (rooms and time-slots) referred as a Chromosome. For designing a weekly schedule, the chromosome is represented as two dimensional array of size 7*5*no. of rooms. In this encoding, the first row represents the room number, the second row represents the time slots. There is an additional hash map which is used in the chromosome representation to obtain the first time-space slot at which a class commences .Each hour of a class has a separate entry in the hash table, but there is only one entry per class in the hash map. For example, if a class starts at 11.00 a.m. and continues for two hours, in the hash table it maps the entries of 11a.m and 12.00p.m.

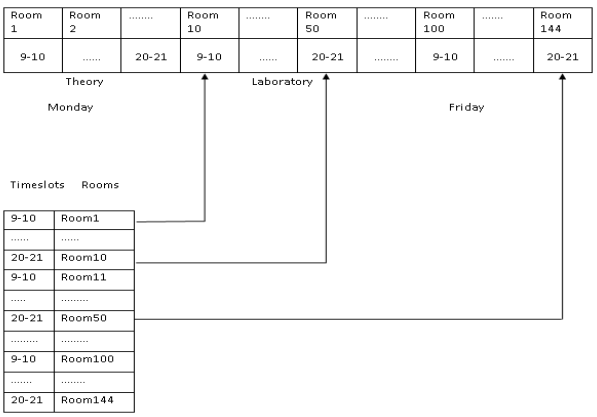


Figure 3: Chromosome Representation

3.2 Selection Operator

After generating the initial population, selection operator is used to make more copies of better strings in a new population. The Selection process is based on fitness values. Chromosomes that are evaluated with higher values have more chances to be selected to reproduce, whereas, those with low values will be either ignored or used in recombinant mechanism which may produce good strings for next generation. The process of natural selection causes those individuals that encode successful structures to produce copies more frequently. To strengthen the generation of a new population, the reproduction of the individuals in the current population is necessary. In this paper, Roulette wheel selection [10] is used as selection operator. The working principle of roulette wheel selection method is as follows,

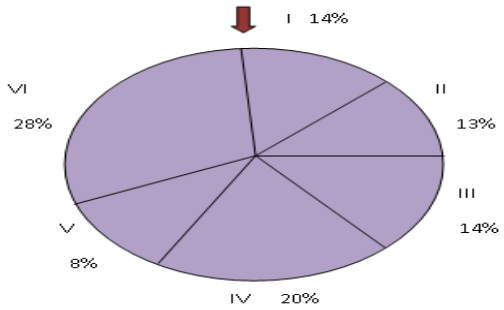


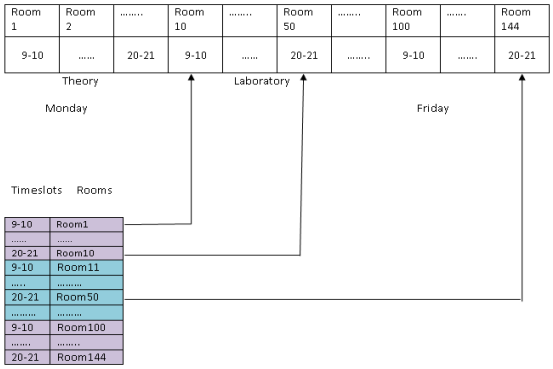
Figure 4: Roulette Wheel Selection Mechanism

In the Roulette wheel selection method the first step is to calculate the cumulative fitness of the whole population through the sum of the fitness of all individuals. Then the probability of selection `is calculated for each individual. An array is built containing cumulative probabilities of the individuals. So, 'n' random numbers are generated in the range and for each random number as an array element which can have higher value is searched for the chromosome (room no., time slot). Therefore, individuals are selected according to their probabilities of selection in the roulette wheel. Here the sixth individual (room no., timeslot) have the highest fitness than others. So the probability of selecting the sixth individual is higher than selecting the other individuals from the wheel. The number of offspring produced for the next generation from this genetic operator is based on the selection pressure.

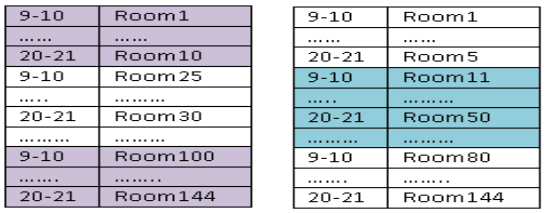
3.3 Crossover Operator

Crossover is a primary genetic operator [14] that used to mate two chromosomes (parents) to produce a new chromosome referred as offspring in the next generation. This process is used to build better sequence of genes by taking the best characteristics from each parent. Crossover occurs during evolution according to a user-definable crossover probability. In this paper two point crossover operator is utilized and illustrated below.

Chromosome



Offspring



Parent1 Parent 2

Figure 5: Crossover Mechanism

The figure 5 shows the two crossover points and its site is indicated in a white color in parent1 and a blue color in parent2 are chosen randomly and gene set is copied from the beginning to the first point from the first parent then the gene set between the crossover points is selected from the second parent, remaining gene set from the second crossover point is selected from the first parent produces the new offspring chromosome. This technique joins data in the hash map of two parents, and then it creates a vector of slots according to the content of the new hash map.

3.4 Mutation Operator

Mutation is a genetic operator [16] that changes the gene sequence by altering one or more gene values in a chromosome from its initial state to reproduction state. This may have the probability of resulting in entirely new gene values being added to the gene mating pool. With these new gene values, the genetic algorithm may be able to arrive at better solution than was previously possible.

Each chromosome comprises a set of timeslots and a set of rooms. In this paper swap mutation is used, which replaces the time slot or a room slot of a chromosome with a randomly generated timeslot or a room slot to obtain better solutions.

.

4. INTERFERENCE INDUCED SILENCING

The Interference Induced Silencing mechanism [19] [20] is inspired from the biological process RNAi [21] which moderates the activity of the genes within living cells. RNAs are the direct products of genes, and these small RNAs can bind to other specific messenger RNA (mRNA) molecules and either increase or decrease their activity, for example to prevent an mRNA from producing a protein, by silencing their activity with the help of RNA interference. It has an important role in defending cells against parasitic genes – viruses and transposons and also used to silence gene expression in a range of organisms. Most of the genes are turned off or silenced appropriately preventing from doing their work of protein synthesis. For example, thousand of genes are active only during embryo development, but remain silent in healthy adults. Similarly, certain genes have to be activated in skin and to be silenced in the heart, liver and other organs. The genes associated with embryogenesis are inappropriately reactivated causing the explosion of uncoordinated cell growth which is the cause of several types of cancers. RNAiinterferes this process and prevents the further explosion to suppress reactivation. Hence, RNAi could be defined as a mechanism by which genes associated with embryogenesis are kept silent.

4.1 Interference Induced Silencing in Class Scheduling Problem

The profound concept, Interference Induced silencing operator is used to silence the gene set of a chromosome that undergoes recombination [3]. The recombination operator such as crossover and mutation may disrupt the gene set that leads to good solutions. To avoid disruption during recombination, the proposed Interference Induced Silencing operator works at gene level silences those gene sets which are considered as instant specific preferences in a class scheduling problem. During crossover the parent chromosomes that are combined together within the crossover points produces the offspring that may or may not satisfy the constraints of the Class scheduling problem. When the constraints are not adhered in the offspring, then the preferences or some of the constraints had been disrupted. Therefore to satisfy the constraints in a chromosome, the gene set should be silenced or preserved by the IIS operator.

In figure 6 given below shows that the two chromosomes had undergone the crossover mechanism which results in swapping the gene set with respect to the crossover points. Sometimes we may have to give importance to the preferences in a chromosome. For example, senior faculty members as the Head of the Department asks for the first hour (9-10) in the forenoon session for all week days, he/she may be allotted. In such situations, simple Genetic algorithm could set different fitness values based on their constraints which affect the feasible solutions. In a simple GA, during crossover operation, that particular time slot (9.00 a.m to 10.00 a.m) gets disrupted and does not set the preference in the time table.

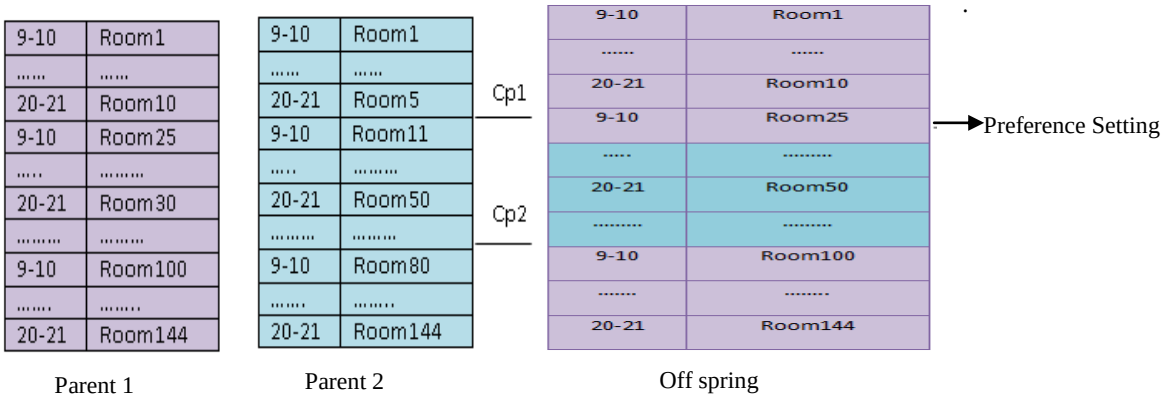


Figure 6: Effects of Interference Induced Silencing Mechanism after Two Point Crossover

When we apply our proposed operator after crossover which sets the stated preference and satisfy the requirement of the Head of the Department. If it is not done, the chromosome structure will be changed for every constraint needed and it does not meet the requirements. So, to make any preference set in a chromosome that is not disrupted by crossover mechanism is done effectively by the proposed system, Interference Induced Silencing operator is shown in the above figure 6.

Several repair mechanism [22] and crossover operators [23] have been applied to course time tabling problem, but that again got some problems.

Mask operator has also been proposed [24] to mask the gene in a chromosome. The problem with the mask operator is, need to set another string (bit associated mask string) to mask/silence the preferences (genes) in a chromosome. Moreover the utility of a bit mask string is not immediately apparent. If you set the mask bits distantly in a mask string, sometimes by mistake the masked bit value is changed, the chromosome structure is also changed and provides infeasible solutions. If the genes/preferences are more in a candidate chromosome, the mask bit mechanism fails to correct the choice of preferences (genes) in a chromosome.

Considering all aspects, interference induced silencing mechanism is preferred to exactly solve the matters of preference settings in a NP hard problem.

5. RESULTS AND STATISTICS

In this paper, we examined the proposed method and the results were compared with the simple GA. The following are one of the GA parameters from many used in Class Scheduling problem during the experimentation:

No. of chromosomes: 100 (Population Size)
Selection Method : Roulette Wheel Selection
Selection Pressure (or)
Selection Rate: 10%
Crossover: Two Point Crossover
Crossover% 88%
Mutation Operator Swap Mutation
Mutation Rate 2%
A number of preferences were set and the results obtained were tabulated in table2 which is given below.

Table 2: Results of Proposed System, Standard Genetic Algorithm

Preference Number	Convergence Speed in Generations	
	SGA	SGA with IIS
1,1	865	377
2,2	845	413
3,3	925	526
4,4	1006	312
5,5	804	298

In the above table the preference is indicated in a two dimensional array. The first column represents the day of the week. The second column represents the hour that to be preferred by the faculty members. The number of generations, the execution taken place with the given preferences with and without interference induced silencing is shown in the table2. From the results, it is clearly proved that the proposed system perfectly done the preference settings for every constraints in a minimum number of runs/generations.

The proposed system has been experimented with various rates of genetic parameters namely the selection, crossover and mutation and the results were compared with the standard

Genetic algorithm. The performance comparison of simple GA with our proposed system is shown in the following graph.

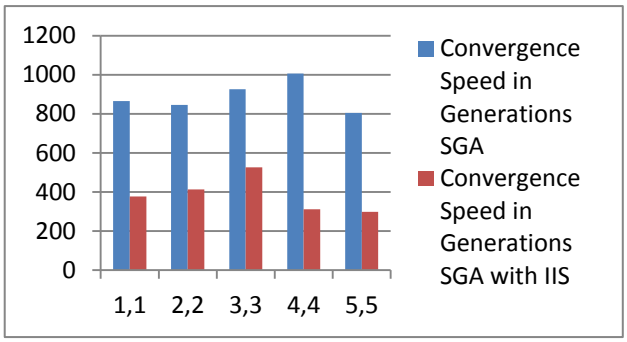


Figure 7: Performance Comparison

From many, few data samples have been taken and generated the performance graph. When comparing the efficiency in terms of convergence time, our proposed system is wise for instance specific preference setting problems

6. CONCLUSION

This paper explores the ways to set preferences in CSP problem using genetic algorithm. After analyzing the pros and cons of few approaches which have been used to solve the course time tabling problem, it is guaranteed that the new biological operator interference induced silencing perfectly does the preference settings in a given problem and proved to be satisfactory in not violating any hard constraints and soft constraints. At present research work is being focused on the application of Interference Induced Silencing in pattern recognition and gene sequence alignment in bioinformatics.

7. REFERENCES

[1] Holland, A text book of "Adaptation in Natural and Artificial Systems, 1975.

[2] Hitoshi Kanoh and Yuusuke Sakamoto, "Interactive Timetabling System Using Knowledge-Based Genetic Algorithms"

[3] S.SivaSathya and S.Kuppuswami, "Gene Silencing for Course Time-Tabling with Genetic Algorithm ".

[4] Sanjay R. Sutar and Rajan S. Bichkar, "University Timetabling based on Hard Constraints using Genetic Algorithm ".

[5] Leon Bambrick, "Lecture Timetabling Using Genetic Algorithms".

[6] Mila S. de Magalh˜aes, Helio J.C. Barbosa, and Laurent E. Dardenn, "Selection-Insertion Schemes in Genetic algorithms for the Flexible Ligand Docking Problem".

[7] Przemyslaw Dymarski,2010," Hidden Markov Models, Theory and Applications".

[8] www.wikipaedia.com

[9] E. Aycan and T. Ayav,"Solving the Course Scheduling Problem Using Simulated Annealing"

[10] Rakesh Kumar and Jyotishree, "Blending Roulette Wheel Selection & Rank Selection in Genetic Algorithms".

[11] E.K. Burke, J.D. Landa Silva, E. Soubeiga, "Multi objective hyper-heuristic Approaches for space allocation and timetabling".

[12] G.M. White and P.W. Chan, "Towards the Construction of Optimal Examination Timetables," INFOR 17, 1979, p.p. 219-229.

[13] Davis L. (1991) "Handbook of Genetic Algorithms" Van Nostrand Reinhold

[14] N.D. Thanh, "Solving Timetabling Problem Using Genetic and Heuristic Algorithms", Eighth ACIS International Conference on Software Engg, Artificial Intelligence, Networking, and Parallel/Distributed Computing, hal. 472- 477, 2007.

[15] R. Raghavjee and N. Pillay, "Using Genetic Algorithms to Solve the South African School Timetabling Problem," Proceedings of Second World Congress on Nature and Biologically Inspired (NaBIC) (2010), pp. 286-292.

[16] O. T. Arogundade, A. T. Akinwale, and O. M. Aweda, "A Genetic Algorithm Approach for a Real-World University Examination Timetabling Problem, International Journal of Computer Applications, Vol. 12, no. 5 (2010), pp. 1-4.

[17] S. Abdullah and A. R . Hamdan, "A hybrid approach for university course timetabling," International Journal of Computer Science and Network Security,, vol. 8, no. 8, 2008.

[18] P. Ko stuch, "The university course timetabling problem with a three-phase approach." International Conference on the Pra ctice and Theory of Automated Timetabling (PATAT V), pp. 109–125, 2005.

[19] Rachel Nash and Tomas Lindahl, "DNA Ligases" Imperial Cancer Research Fund Clare Hall Laboratories, United Kingdom.

[20] Neema Agrawal, P. V. N. Dasaradhi, Asif Mohmmed, Pawan Malhotra, "RNA Interference: Biology, Mechanism, and Applications" , Microbiology and Molecular Biology reviews, Dec. 2003, p. 657–685.

[21] Chen, A text book of fundamentals of microbiology".

[22] George G. Mitchell Diarmuid O'Donoghue David Barnes Mark McCarville , "GeneRepair – Repair Operator for Genetic Algorithms ".

[23] Qingfu Zhang, Jianyong Sun, and Edward Tsang, "An Evolutionary Algorithm With Guided Mutation for the Maximum Clique Problem", IEEE transactions on evolutionary computation, vol. 9, no. 2, April 2005.

[24] Arthur L. Corcoran, Roger L. Wainwright, "Reducing disruption of superior building blocks in genetic algorithms", Proceedings of the, ACM SIGAPP Symposium on Applied Computing February, 2003.