Introducing a Novel parameter in Generation of Course Timetable with Genetic Algorithm

Ravitashaw Bathla, Shubham Jain, Rajeev Singh

Department of Computer Engineering

G.B. Pant University of Agriculture and Technology

Pantnagar, Uttarakhand, India

27th December, 2013

Outline

- Abstract
- Introduction
- Related Work
- Problem Description
- Genetic Algorithm
- Proposed Methodology
- Results and Discussions
- Conclusion
- Future Work

Motivation

Genetic Algorithm produces result which is randomly selected from the solution space. This solution is not appropriate most of the times, even though it is fit.

The Happiness parameter is introduced for the faculty and the students, to provide comfort with the timetable.

Introduction
The Timetable Problem

Definition: The task of assigning events, such as lectures, exams to a limited set of timeslots and rooms, within the limits of specified constraints.

It is an NP-Hard Problem.

Introduction
Type of University Timetable Problem

- Course Timetable Problem
  - One event per room per timeslot
  - Fixed number of timeslots

- Exam Timetable Problem:
  - Multiple events per room at one time
  - Variable timeslots

Introduction
Constraints

- Hard Constraints
  - All should be satisfied for the solution to be feasible
  - Higher Priority

- Soft Constraints
  - Maximum of the soft constraints should be obeyed
  - Lower priority than Hard Constraints

Related Work

- Bambrick (1997), presents Genetic Algorithm for timetable problem along with repair strategy. All types of feasible and non-feasible solution exists together in the solution space with fitness count of 1.0.
- Abdullah and Turabieh (2008), local search along with genetic Algorithm is used for optimization. An additional computational cost is incurred using local search and it is performed on final solution which limits the final solution
- Yang and Hat (2011), used an additional data structure for storing information about events with zero penalty value

Problem Description
The University Course Timetable

- The course timetable of Department of Computer Engineering, G.B. Pant University of Agriculture and Technology, Pantnagar was used for generating the timetable.

- A course is represented as Course Number, followed by Lectures, Tutorials and Laboratories such as Operating System with course number TCT324 having total 3 credits is represented as TCT324(2-1-1). Tutorials are not included in the total number of credit hours.

Problem Description
Constraints

The Hard Constraints are:

1. A classroom must not be booked twice at one time.
2. Each event must be allotted exactly once.
3. Students must not have multiple classes simultaneously.
4. The classroom must have enough capacity to hold the allotted group of students.
5. Instructors must not be booked twice for one particular timeslot.
6. An instructor must not be allotted a lecture, laboratory or tutorial when he/she is unavailable. E.g. an instructor might have prior commitments.

The soft constraints have been exempted from consideration since they vary greatly with respect to departments and universities.

Problem Description
Sample Data

S. No.	Abbrv.	Course Name	Credits	Year
1	DS	Data Structure	3(3-1-0)	II
2	DSS	Discrete Structure	3(1-0-0)	II
3	FLAT	Finite Lang. and Automata Theory	3(3-1-0)	II
4	FCCS	Fund. of Computer Comm. Systems	4(3-1-1)	II
5	PL	Programming Language	3(3-1-0)	II
6	SYP	System Programming	4(3-0-1)	III
7	DBMS	Database Management System	4(3-0-1)	III
8	ORS	Operational Research	2(2-1-0)	III
9	MIS	Management Information System	3(2-0-1)	III
10	CA	Computer Architecture	3(2-1-1)	IV
11	DMW	Data Mining and Warehousing	3(2-1-1)	IV
12	CGA	Computer Graphics and Animations	4(3-0-1)	IV

Genetic Algorithm

- Derived from process of evolution in nature as stated by Charles Darwin
- Each individual is represented as Chromosome
- A group of Chromosomes form a population.
- These chromosomes undergo evolution process to produce new offsprings (individuals).

Genetic Algorithm
Evolutionary Process

- Initially, chromosomes are randomly generated equal to the population size
- The chromosomes with maximum fitness are used for breeding
- The chromosomes produced after breeding are named as Generation
- The produced offspring are evaluated for fitness value
- The individual with maximum fitness value is added into the population and the one with least fitness value is discarded from the population, thus keeping constant population size.
- This process is repeated depending upon the requirement.

Genetic Algorithm Operators

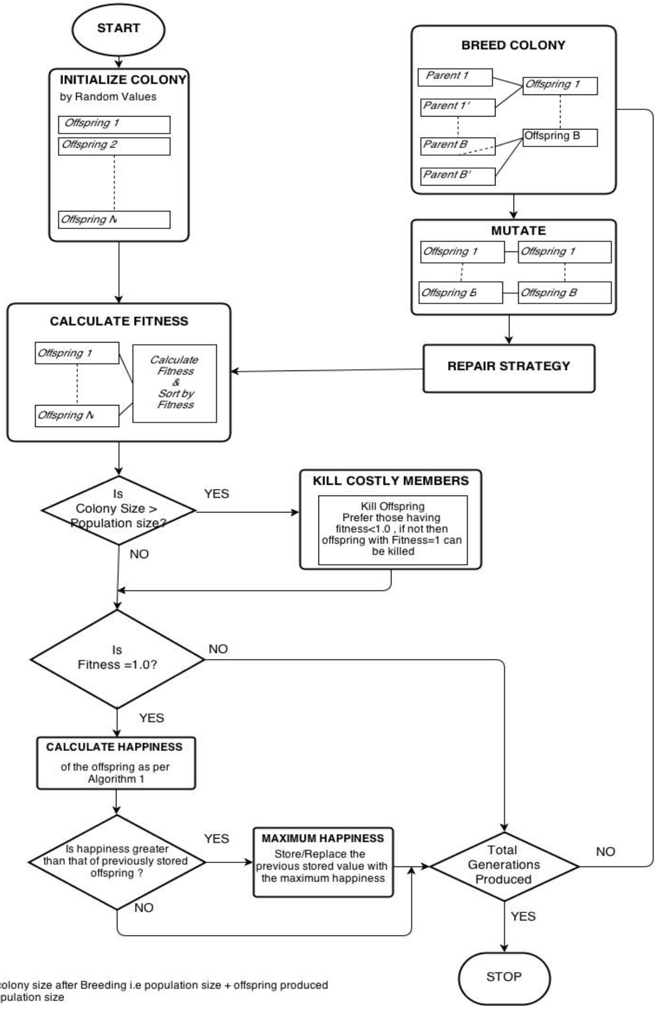
Crossover

Parent 1:	1010100001111 0001100110000110
Parent 2:	1010111011101 1000011101011010
Offspring:	1010100001111 1000011101011010

Mutation

Before Mutation:	101011101101110010101
After Mutation:	101011100101110010101

Proposed Methodology



START

INITIALIZE COLONY by Random Values

Offspring 1
Offspring 2
Offspring N

CALCULATE FITNESS

Offspring 1
Offspring N
Calculate Fitness & Sort by Fitness

Is Colony Size > Population size?
YES
NO

KILL COSTLY MEMBERS

Kill Offspring Prefer those having fitness<1.0 , if not then offspring with Fitness=1 can be killed

Is Fitness =1.0?
NO
YES

CALCULATE HAPPINESS

of the offspring as per Algorithm 1

Is happiness greater than that of previously stored offspring ?
YES
NO

MAXIMUM HAPPINESS

Store/Replace the previous stored value with the maximum happiness

Total Generations Produced
NO
YES

STOP

BREED COLONY

Parent 1
Parent 1'
Parent B
Parent B'
Offspring 1
Offspring B

MUTATE

Offspring 1
Offspring B
Offspring 1
Offspring B

REPAIR STRATEGY

where, B =colony size after Breeding i.e population size + offspring produced
N=population size

Proposed Methodology
The Happiness Parameter

- The Happiness parameter calculated happiness of the entire table
- The Happiness value is calculated after implementation of Genetic Algorithm

The Happiness Parameter:

It is the sum of free timeslots an instructor has after which no lectures are scheduled for that particular day, for all days of the week. A penalty is added for lectures at odd times like 08:00-09:00 (early morning) and/or 14:00-15:00 (immediately after lunch)

Happiness Parameter Formula

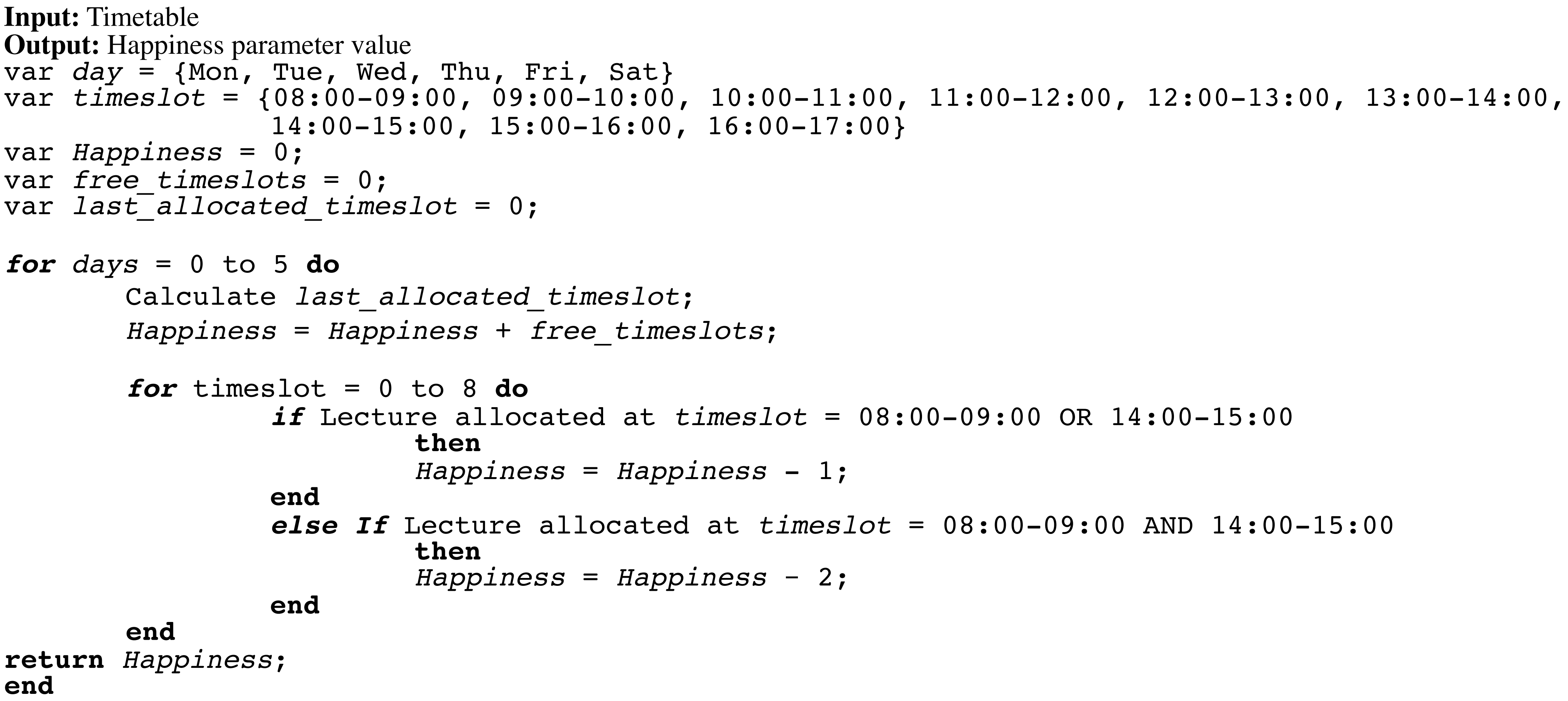
$$Happiness = \sum_{i=0}^{n} \{(Total\ Time\ slots(i.e.8) - Position\ of\ last\ lecture\ for\ i\ day) - p\}$$

Where,

i = Number of Working days {Mon, Tue, Wed, Thu, Fri, Sat}, and

$$\mathbf{Penalty\ ,p} = \begin{cases} -1, & \textbf{\textit{If a lecture is scheduled at}}\ 08{:}00 - 09{:}00\ \textbf{\textit{OR}}\ 14{:}00 - 15{:}00 \\ -2, & \textbf{\textit{If a lecture is scheduled at}}\ 08{:}00 - 09{:}00\ \textbf{\textit{AND}}\ 14{:}00 - 15{:}00 \end{cases}$$

Happiness Parameter Algorithm



Input: Timetable
Output: Happiness parameter value

```
var day = {Mon, Tue, Wed, Thu, Fri, Sat}
var timeslot = {08:00-09:00, 09:00-10:00, 10:00-11:00, 11:00-12:00, 12:00-13:00, 13:00-14:00,
                14:00-15:00, 15:00-16:00, 16:00-17:00}
var Happiness = 0;
var free_timeslots = 0;
var last_allocated_timeslot = 0;

for days = 0 to 5 do
       Calculate last_allocated_timeslot;
       Happiness = Happiness + free_timeslots;

       for timeslot = 0 to 8 do
                if Lecture allocated at timeslot = 08:00-09:00 OR 14:00-15:00
                         then
                         Happiness = Happiness - 1;
                end
                else If Lecture allocated at timeslot = 08:00-09:00 AND 14:00-15:00
                         then
                         Happiness = Happiness - 2;
                end
       end
return Happiness;
end
```

Result and Discussion
Specification for GA

S. No.	Parameters	Quantity
1	Number of Generations	20,000
2	Population Size	50
3	Number of Working Days	6
4	Number of Timeslots each day	8

The simulation was performed using Intel Core i3 processor 2.4Ghz on Windows 7 Operating System.

Result and Discussion

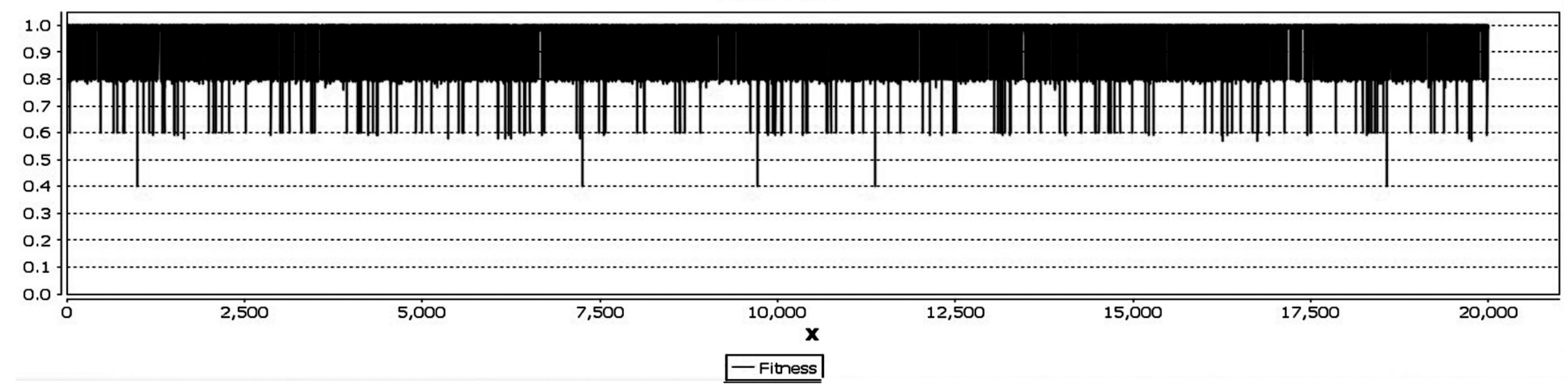
S. No.	G_0	$G(H_{neg})$	H_{max}	$G(H_{max})$	H_{min}	$G(H_{min})$	H_{avg}
1	14288	13566	5	2	-12	19	-4.608133
2	14317	13488	6	4	-13	6	-4.650268
3	14228	13300	6	1	-14	1	-4.556508
4	14190	13476	6	8	-12	5	-4.834179
5	14254	13514	6	11	-13	4	-4.687877
6	14427	13537	6	10	-13	17	-4.635198
7	14300	13192	7	2	-13	2	-4.419161
8	14422	13461	8	1	-12	6	-4.498613
9	14290	13298	5	9	-12	14	-4.348775
10	14158	12842	6	15	-13	3	-4.323138

Where,

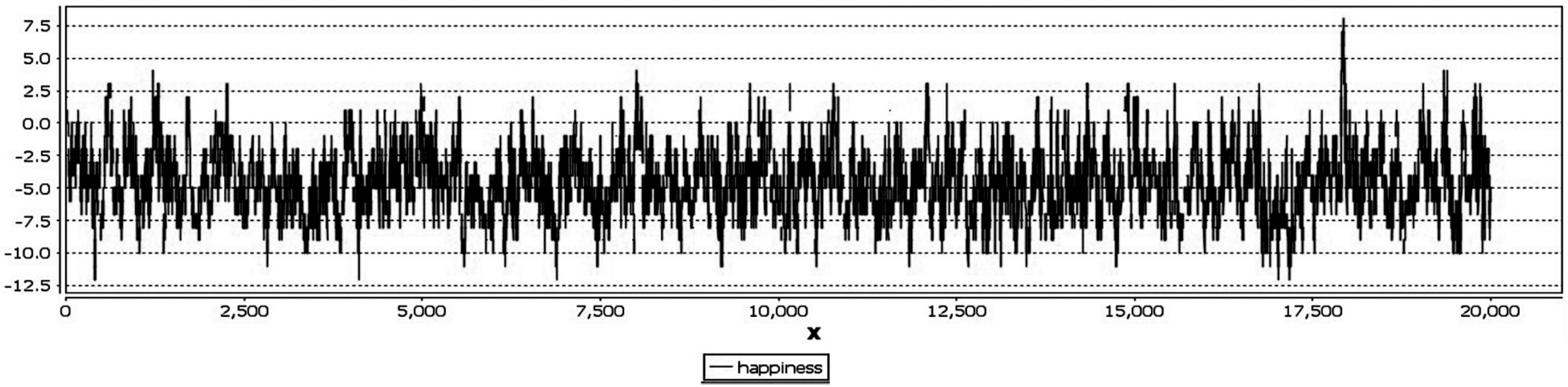
G_0 = Generations with Fitness = 1.0,
$G(H_{max})$ = Generations with Maximum Happiness,
$G(H_{min})$ = Generations with Minimum Happiness
$G(H_{neg})$ = Generations with negative Happiness,

H_{max} = Maximum Happiness
H_{min} = Minimum Happiness
H_{avg} = Average Happiness

Result and Discussion



Simulation Graph with Fitness = 1.0



Simulation Graph with Maximum Happiness = 8.0

Result and Discussion

$$\textit{Probability of selecting solution with negative Happiness} = \frac{\textit{Total number of Generations with negative Happiness}}{\textit{Total Number of Generations having Fitness}}$$

S. No.	G_0	$G(H_{neg})$	Probability of selecting offspring with H_{neg}
1	14288	13566	0.94946
2	14317	13488	0.93855
3	14228	13300	0.93477
4	14190	13476	0.94968
5	14254	13514	0.94808
6	14427	13537	0.93831
7	14300	13192	0.92251
8	14422	13461	0.93336
9	14290	13298	0.93058
10	14158	12842	0.90704

Result and Discussion

- The Tab. 4 represents the probability of selection of a timetable with negative happiness for all the iterations done in Tab. 3.

- The values in Tab. 4 clearly depicts that there are more than 90% (approx.) chances of selection of non-appropriate timetable without using Happiness parameter.

Conclusion

- Adding Happiness parameter with the Genetic Algorithm narrows the solution space with only appropriately feasible solutions only.

- Thus, using this happiness parameter the non appropriate solutions can be discarded from consideration.

Future Work

- Happiness for Individual Instructors
- Versatility of Happiness Parameter
- Using Happiness parameter on Parallel systems

References

- Tim B. Cooper and Jeffrey H. Kingston. The complexity of timetable construction problems. In Edmund Burke and Peter Ross, editors, *Practice and Theory of Automated Timetabling,* volume 1153 of *Lecture Notes in Computer Science,* pages 281-295. Springer Berlin Heidelberg, 1996.
- Manar Hosny and Shameem Fatima. A survey of genetic algorithms for the university timetabling problem. *International Proceedings of Computer Science and Information Technology*, vol. 13, 2011.
- David Come and Peter Ross. Peckish initialisation strategies for evolutionary timetabling. In *Selected papers from the First International Conference on Practice and Theory of Automated Timetabling,* pages 227-240, London, UK, UK, 1996. Springer-Verlag.
- M.F. Zibran. *A Multi-phase Approach to University Course Timetabling.* Canadian theses. University of Lethbridge (Canada), 2007.
- Rhydian Lewis and Ben Paechter. Finding feasible timetables using group-based operators. *IEEE Trans. Evolutionary Computation,* 11(3):397-413, 2007.
- Leon Bambrick. Lecture timetabling using genetic algorithms. *The University of Queensland,* 1997.
- S. Abdullah and H. Turabieh. Generating university course timetable using genetic algorithms and local search. In *Convergence and Hybrid Information Technology, 2008. ICCIT '08. Third International Conference on,* volume 1, pages 254-260, 2008.

References

- Shengxiang Yang and S.N. Jat. Genetic algorithms with guided and local search strategies for university course timetabling. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on,* 41(1):93-106, 2011.
- Kurt A Hacker, John Eddy, and Kemper E Lewis. Efficient global optimization using hybrid genetic algorithms In *9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization,* pages 4-6, 2002.
- J.H. Holland. *Adaptation in natural and artificial systems:* an *introductory analysis with applications to biology, control, and artificial intelligence.* University of Michigan Press, 1975.
- L. Davis. *Handbook of genetic algorithms.* VNR computer library. Van Nostrand Reinhold, 1991.
- Salwani Abdullah, Hamza Turabieh, Barry McCollum, and Edmund K Burke. An investigation of a genetic algorithm and sequential local search approach for curriculum-based course timetabling problems. In *Proc. Multildisciplinary International Conference on Scheduling: Theory and Applications (MISTA 2009), Dublin, Ireland,* pages 727-731, 2009.
- V. Sapru, K. Reddy, and B. Sivaselvan. Time table scheduling using genetic algorithms employing guided mutation. In *Computational Intelligence and Computing Research (ICCIC), 2010 IEEE International Conference on,* pages 1-4, 2010.
- M. Gen and R. Cheng. *Genetic algorithms and engineering design.* Wiley series in engineering design and automation. Wiley, 1997.

Thanks