# Three-phase simulated annealing algorithms for exam scheduling

**Conference Paper** · August 2003
DOI: 10.1109/AICCSA.2003.1227522 · Source: IEEE Xplore



**3 authors:**

Nashat Mansour
Lebanese American University
**90** PUBLICATIONS **1,104** CITATIONS
SEE PROFILE

Abbas Tarhini
Lebanese American University
**23** PUBLICATIONS **86** CITATIONS
SEE PROFILE

Vatche Ishakian
Bentley University
**43** PUBLICATIONS **296** CITATIONS
SEE PROFILE

# Three-Phase Simulated Annealing Algorithms for Exam Scheduling

**Nashat Mansour, Abbas Tarhini, Vaje Ishakian**
Computer Science Program, Lebanese American University, Mme Curie st., Beirut,
Lebanon, 1102-2801
E-mail: nmansour@lau.edu.lb

**Abstract**
*Scheduling of final exam usually results in conflicts and inconvenience. Conflicts occur when simultaneous exams are scheduled for the same student, and inconvenience to a student refers to consecutive exams or more than two exams on the same day. A good exam schedule should aim to minimize conflicts and the two inconvenience factors based on weight that are user-assigned to these three factors and subject to some constraints such as the number and capacities of classrooms. Scheduling final exams for large numbers of courses and students in universities is an intractable problem. In this work, we decompose the problem into three phases and propose simulated annealing algorithms for these phases. Hence, we refer to our solution methods as 3-phase simulated annealing (3PSA). We empirically compare 3PSA with a 4-phase clustering-based heuristic algorithm using realistic data. Our experimental results show that 3PSA produces good exam schedules, which are better than those of the clustering heuristic procedure.*

**Keywords**
Applications, heuristics, exam scheduling, simulated annealing.

## INTRODUCTION

Manual scheduling of final exams for large numbers of courses and students by a university's Registrar's Office invariably results in conflicts or inconvenience in the exam schedule of many students. Conflicts occur where simultaneous exams are scheduled for the same student, and inconvenience to a student refers to consecutive exams or more than two exams on the same day. A good exam schedule would aim to minimize conflicts and the two inconvenience factors based on weights that are user-assigned to these three factors. Such a schedule may also be subject to constraints, such as predefined number of days and limited-capacity classrooms.

Several algorithms for solving the exam scheduling problem have been developed. These algorithms have been surveyed in [3,4,7]. Burke and Petrovic [3] classify existing algorithms into four types. One type is sequential methods that order exams and then assign them sequentially into valid time periods. Ordering is done by heuristics based on an estimation of how difficult it is to schedule the exams. Examples of sequential methods can be found in [1,6]. The second type of algorithms is cluster methods in which the set of exams is divided into groups/clusters based on hard constraints and then the clusters are assigned to time periods to satisfy soft constraints. Examples of cluster methods can be found in [5,12,14]. The third type of algorithms is contraint-based. In these algorithms, the problem is modelled as a set of variables (exams) to which resources (periods and rooms) have to be assigned to satisfy a number of constraints. Examples of these algorithms can be found in [2,8]. The fourth type of algorithms is meta-heuristic methods, which are based on genetic algorithms, simulated annealing, tabu search, and hybrid approaches. Examples of these algorithms can be found in [9,10,11,15,16].

Previous work on exam scheduling has been based on a variety of problem models and solution approaches. We are interested in the solution approaches that are based on decomposing the problem into sub-problems or phases. In this work, we decompose the exam scheduling problem into three phases. In each phase, we address one or more constaints. Then, we solve each by a simulated annealing algorithm. Hence, our proposed method is referred to as 3-phase simulated annealing (3PSA) method. We empirically compare 3PSA with an existing four-phase method that uses heuristics in each phase. Our results, which are based on realistic data, show that 3PSA produce better results in terms of the number of exam conflicts and inconvenient exams.

The rest of the paper is organized as follows. Section 2 presents the problem formulation used and a brief background on simulated annealing. Section 3 describes the proposed 3PSA algorithms. Section 4 presents the empirical results. Section 5 contains our conclusions.

## BACKGROUND

### Exam Scheduling Problem

Given that A exams are to be taken by students over B days, where E exam periods can be done per day, the exam scheduling problem consists of assigning A exams to Π (=B*E) exam periods, within specified classrooms. The objective is to minimize the conflict and the unfairness factors, which are: (i) The number of students having simultaneous exams, (ii) The number of students having

consecutive exams, and (iii) The number of students having two or more exams on the same day.

In this work, we assume the following conditions and constraints:

(i) The user should be provided with the flexibility of assigning weights to the three conflict and unfairness factors.
(ii) A limited number of exam periods, Π, is predefined.
(iii) A limited predefined number of classrooms, R, are available for exams.
(iv) Room capacities ($\Psi_1$, $\Psi_2$, …, $\Psi_R$) are taken into consideration in assigning exams to rooms (second feasibility condition). Also, more than one exam can be assigned to the same room at the same time if they fit.
(v) The last period of one day is considered to be consecutive to the first period of the next day.

Scheduling problems can be represented by graphs. Let G ($V_G$, $E_G$) be a graph in which: vertex $v_i \in V_G$ represents an exam to be scheduled and $|V_G| = A$; vertex weight $w_i$ represents the number of students taking exam $v_i$; edge $e \in E_G$ joining two vertices $v_i$ and $v_j$ represents the existence of students taking both exams $v_i$ and $v_j$; weight of edge e, $w_{ij}$, represents the number of students taking both exams $v_i$ and $v_j$.

The exam scheduling problem can be expressed as a modified weighted-graph coloring problem, where we color the vertices of a graph using a specified maximum number of colors (exam periods), Π, such that the numbers of conflicting and unfair exams are minimized and the constraints (listed above) are satisfied. A solution to the exam-scheduling problem is henceforth denoted as the configuration C. Note that each color corresponds to an exam period and all vertices having the same color represent the exams that are assigned to the same period.

Let c(v) be the color of vertex v, and $\xi = \{ c_1, c_2, \ldots, c_\pi \}$ be the set of ordered, available colors; that is, $|\xi| = \Pi$ = maximum number of available colors, and $(c_i - c_{i-1}) = 1$ for i=2,3,…, Π. The conflicting and unfair exams are given by the following factors:

(i) $S_{SE}$, the total number of students having conflicting simultaneous exams. That is, $S_{SE} = \sum w_{ij}$ for all i and j such that c(i) =c(j)
(ii) $S_{CE}$, the total number of students having consecutive exams. That is, $S_{CE} = \sum w_{ij}$ for all i and j such that $|c(i)- c(j)| = 1$.
(iii) $S_{ME}$, the total number of students having two or more exams per day. That is, $S_{ME} = \sum w_{ij}$ for all i and j such that c(i) and c(j) refer to exam periods on the same day.
(iv) $\rho_{ik} = 1$ if the capacity of room i is exceeded in period k, i.e. if room i was assigned a larger number of students than $\Psi_i$ (capacity of room i); otherwise, it is equal to zero. Hence, the inner summation in $\sum_{1 \le k \le \Pi} \sum_{1 \le i \le R} \rho_{ik}$ gives the total number of rooms violated in a period, whereas the outer summation gives the total number of rooms violated in all periods.

### Simulated Annealing

Simulated annealing is based on ideas from physics and is analogous to the physical annealing of a solid [13]. To coerce some material into a low-energy state, it is heated and then cooled very slowly, allowing it to come to thermal equilibrium at each temperature. At each temperature in the cooling schedule, the Metropolis algorithm simulates the behavior of the system.

The simulated annealing algorithm (SA) simulates the natural phenomenon by a search (perturbations) process in the solution space (energy landscape) optimizing some objective function, OF (energy). It starts with some initial solution at a high (artificial) temperature and then reduces the temperature gradually to a freezing point. Figure 1 gives an outline of an SA algorithm

```
Initial random configuration
Determine initial temperature T(0);
Determine freezing temperature T_f ;
while (T(i) > T_f and not converged) do
        repeat N times
                perturb();
                if (ΔOF ≤ 0 ) then
                        update()        /* accept */
                else
                        if (random() < e^(- ΔOF / T(i))) then
                                update()        /*accept*/
                        else
                                reject_purturbation();

        save_best_sofar();
        T(i) = θ * T(i);
endwhile
```

**Figure 1. Outline of an SA algorithm.**

### 3-PHASE SIMULATED ANNEALING ALGORITHM

Our phased-solution is based on decomposing the exam scheduling problem into three sub-problems or phases. In the first phase, the number of simultaneous exams, $S_{SE}$, is minimized. In he second phase, the total sum of consecutive exams, $S_{CE}$, and multiple exams per day, $S_{ME}$, are minimized. In the third phase, the objective is to assign the exams to the available rooms without exceeding room capacities.

In the following subsections, we propose using simulated annealing (SA) algorithms for solving the three sub-problems in the three phases. Hence, we refer to our

approach as three-phase simulated annealing (3PSA) solution approach.

### Phase-1: Minimizing $S_{SE}$

In Phase-1, we assign the A exams to Π periods with the only objective of minimizing the number of simultaneous exams, $S_{SE}$. The input to this phase is the graph G with A vertices; each vertex represents an exam and each edge has a weight equal to the number of students enrolled in both adjacent exams. We propose a simulated annealing algorithm SA-1 that aims to solve the Phase-1 sub-problem.

The system to be coerced into a low-energy state in this phase is represented by the configuration C, which is implemented as an array of A elements. Each element is given by a value c(i), which represents the period (color) to which an exam i is assigned. The color-value of c(i) is some $c_j \in \xi$. The initial configuration is randomly generated. That is, the color-value assigned to the A array elements are randomly selected from the set ξ. The system energy in this phase is given by the objective function OF1 = $S_{SE}$. In the annealing process, the configuration-system, C, goes through changes until it reaches a minimal or near-minimal value for $S_{SE}$ at freezing temperature. These changes take place over many iterations, in an SA algorithm, which are shown in Figure 1 in the outer while-loop and the inner repeat-loop. The outer loop determines the cooling schedule of the annealing process and the inner loop implements the major step in an SA algorithm, which is the Metropolis step.

An iteration of the Metropolis step consists of a perturbation operation, an accept/reject criterion, and a thermal equilibrium condition. Perturbation to the configuration C is done by randomly selecting an array element (with color $c_i$) and assigning it another color $c_j$ (randomly-selected from ξ). The acceptance criterion checks the change in OF1 due to the perturbation. If the change decreases OF1 (down-hill move), the perturbation is accepted and C is updated. However, if the perturbation causes OF1 to increase (up-hill move), it is accepted only with probability $e^{-\Delta OF1 / T(i)}$, where T(i) is the respective temperature value. The advantage of this procedure is that the down-hill moves are always accepted and hence the objective function tends to be minimized. Further, the controlled up-hill moves also prevent the evolving solution (configuration) from being prematurely trapped in a bad local minimum. However, at very low (near-freezing) temperature values, T(i), up-hill moves are no longer accepted. This Metropolis step is repeated N = Π*A times at every temperature after which thermal equilibrium is considered to be reached.

As for the cooling schedule, the initial temperature, T(0), is set to a value that yields a high initial acceptance probability of 0.93 for up-hill moves applied to the initial configuration. The freezing temperature, $T_f$, is set to a value that makes this probability extremely small ($2^{-30}$), so that only down-hill moves are allowed at this point. The cooling schedule used in this work is simply given by T(i+1) = θ*T(i), with θ = 0.95. The algorithm terminates when the freezing point is reached or when the best-so-far candidate solution does not improve for a number of temperature values, say 20, in the cold part of the annealing schedule.

### Phase-2: Minimizing $S_{CE}$ and $S_{ME}$

The input of Phase-2 is the output of Phase-1, which is a graph of Π vertices: each vertex represents a block of exams that must be scheduled simultaneously (in the same period); each edge between a pair of vertices has a weight equal to the number of students participating in the exams of the adjacent blocks (corresponding to the pair of vertices). In Phase-1, we color the A exams with Π colors. That is, we end up scheduling the A exams to Π logical periods. In Phase-2, we aim to map the Π logical periods into Π actual periods, such that both $S_{CE}$ and $S_{ME}$ are minimized. That is, we aim to renumber / rearrange the Π blocks of exams so that they end up in suitable, actual periods. We propose SA-2 algorithm for the Phase-2 sub-problem.

The configuration considered by SA-2 is an array of Π elements, where each element refers to the period to which a whole block of exams is assigned. SA-2 includes similar steps to those of SA-1. But the objective function to be minimized is the weighted sum OF2 = $\phi_1 S_{CE} + \phi_2 S_{ME}$. In this work, we assume that minimizing consecutive exams and minimizing multiple exams per day are equally important ($\phi_1=\phi_2=1$). The perturbation operation is done by swapping randomly selected blocks of exams between periods. In SA-2, the Metropolis step is repeated N = $\Pi^2$ times at every temperature. Note that SA-2 does not change the exam blocks produced by SA-1. The output of SA-2 is a schedule of exams to actual periods (and days).

### Phase-3: Assigning Exams to Rooms

The input to Phase-3 is an exam schedule with A elements, each is assigned to a specific exam period and is required to be assigned to a classroom. In this phase, we use SA-3 for assigning the exams in each period to the available rooms, such that no room exceeds its capacity.

SA-3 is applied Π times to the exams in each of the Π periods. Every time, the configuration considered by SA-3 is an array of records. Each record includes: an exam, the period to which the exam is assigned, and the classroom. In this array, the periods are the same for all exams and the rooms are to be determined in this phase. The size of the array is equal to the number of exams assigned in Phase-1 to the particular period. The initial configuration is generated by randomly assigning rooms to exams and the

objective function is OF3 = $\sum_{k=1}^{\Pi} \sum_{i=1}^{R} \rho_{ik}$. A perturbation is done by randomly reassigning a room to a randomly selected exam/element in the array of records. At each temperature, perturbations are repeated (A*R/Π) times. Note that SA-3 does not alter the exam-period that has been assigned by SA-1 and SA-2. The outputs of the Π runs of SA-3 are then merged into an array of A records, each record includes: an exam, the assigned period, and the assigned room. This array represents the solution found by 3PSA.

## EMPIRICAL RESULTS

In this section, we present the results of 3PSA and compare them with those of manual scheduling and an existing four-phase heuristic algorithm, referred to as FESP [14]. These algorithms are applied to real data of university exams obtained from five semesters. These data provide five subject problem instances, SP1-SP5, illustrated in Table 1. This table shows the number of exams (A), number of available classrooms (R), the number of students, and the total number of student enrolment for all exams-courses.

The exam schedules produced by the algorithms are evaluated in terms of the five metrics: number of students having simultaneous exams ($S_{SE}$), number of students having consecutive exams ($S_{CE}$), number of students having two or more exams per day ($S_{ME}$), total number of rooms assigned with more students than their capacities over all exam periods, and execution time. We first apply the algorithms to the five subjects problems with a typical number of exam periods (at the university where data is obtained), namely Π = 32 periods for B = 8 days and E = 4 periods per day. Then, we consider the results of the algorithms for tighter and more relaxed number of exam periods, specifically for Π = 20, 24, 28, 36, 40. All experiments were done on a PC with a Pentium III, 850 MHz, and 512 MB RAM.

Tables 2–6 give the results of the 3PSA and FESP and of manual scheduling for the five subject problems SP1–SP5. These results apply for a typical number of 32 exam periods (over 8 days). Tables 7–11 give the results of the four algorithms for different exam periods $20 \leq \Pi \leq 40$ (and different exam days). Table 12 gives the range of execution times of the implementation of the algorithms over the five problems. From these results, we infer the following findings:

(a) 3PSA produces 'good' exam schedules. This finding is based on the low values obtained for $S_{SE}$, $S_{CE}$, and $S_{ME}$ relative to the total number of students. 3PSA also manages to fit students in the available rooms except for SP2, SP4, and SP5 where Π = 20 and 24. Further, 3PSA allows a reduction in the number of exam days while keeping the values of the first three metrics reasonable using the same number of rooms. In our examples, 3PSA allows a reduction of at least one day (i.e., from 32 to 28 periods).

(b) 3PSA produces better solutions than FESP, and manual scheduling. This is true for all values recorded in Tables 2–6 except for $S_{ME}$ of SP3. This is also true for of most the values recorded in Tables 7-11. In fact, the sum of $S_{CE}$ and $S_{ME}$ values produced by 3PSA is smaller than that of FESP for all cases. Further, a shortcoming of FESP stands out in Table 11, where it fails to produce feasible solutions that fit the available classrooms for all six values of exam periods.

(c) Table 12 shows that 3PSA and FESP are comparable in terms of execution times.

(d) As expected, both algorithms, 3PSA and FESP, produce better results as Π increases and worse results as Π decreases.

## CONCLUSION

We have presented simulated annealing algorithms for finding exam schedules after decomposing the problem into three phases. The resulting method is referred to as 3PSA. We have empirically evaluated 3PSA using realistic university data for five semesters. The experimental results demonstrate that 3PSA produces good exam schedules and allows a reduction in the number of exam days/periods and yet the number of students with unfair exam schedules is still reasonable.


### Acknowledgment

We thank Mazen Timany and Adonis Shehayeb for helping in coding.

**Table 1. Subject Problems**

| Subject Problem | A | R | # of Students | # of Student Enrolments |
|---|---|---|---|---|
| SP1 | 336 | 21 | 2456 | 9550 |
| SP2 | 357 | 21 | 2489 | 9735 |
| SP3 | 359 | 21 | 2512 | 10836 |
| SP4 | 426 | 21 | 3063 | 12275 |
| SP5 | 477 | 21 | 3115 | 12406 |

**Table 2. Results for Subject Problem SP1, $\Pi = 32$**

| | $S_{SE}$ | $S_{CE}$ | $S_{ME}$ | # Rooms exceeding capacity |
|---|---|---|---|---|
| 3PSA | 0 | 127 | 273 | 0 |
| FESP | 0 | 221 | 286 | 0 |
| Manual | 8 | 267 | 329 | 0 |

**Table 3. Results for Subject Problem SP2, $\Pi = 32$**

| | $S_{SE}$ | $S_{CE}$ | $S_{ME}$ | # Rooms exceeding capacity |
|---|---|---|---|---|
| 3PSA | 0 | 93 | 225 | 0 |
| FESP | 0 | 183 | 264 | 0 |
| Manual | 5 | 251 | 310 | 0 |

**Table 4. Results for Subject Problem SP3, $\Pi = 32$**

| | $S_{SE}$ | $S_{CE}$ | $S_{ME}$ | # Rooms exceeding capacity |
|---|---|---|---|---|
| 3PSA | 0 | 41 | 101 | 0 |
| FESP | 0 | 67 | 98 | 0 |
| Manual | 4 | 115 | 146 | 0 |

**Table 5. Results for Subject Problem SP4, $\Pi = 32$**

| | $S_{SE}$ | $S_{CE}$ | $S_{ME}$ | # Rooms exceeding capacity |
|---|---|---|---|---|
| 3PSA | 0 | 96 | 190 | 0 |
| FESP | 0 | 242 | 368 | 0 |
| Manual | 12 | 283 | 396 | 0 |

**Table 6. Results for Subject Problem SP5, Π = 32**

| | $S_{SE}$ | $S_{CE}$ | $S_{ME}$ | # Rooms exceeding capacity |
|---|---|---|---|---|
| 3PSA | 0 | 63 | 137 | 0 |
| FESP | 0 | 100 | 157 | 2 |
| Manual | 9 | 154 | 231 | 0 |

**Table 7. Results for subject problem SP1, $20 \leq \Pi \leq 40$**

| | | 20 | 24 | 28 | 32 | 36 | 40 |
|---|---|---|---|---|---|---|---|
| $S_{SE}$ | 3PSA | 7 | 2 | 0 | 0 | 0 | 0 |
| | FESP | 41 | 5 | 2 | 0 | 0 | 0 |
| $S_{CE}$ | 3PSA | 282 | 219 | 149 | 127 | 89 | 79 |
| | FESP | 368 | 300 | 241 | 221 | 138 | 133 |
| $S_{ME}$ | 3PSA | 568 | 422 | 337 | 273 | 229 | 184 |
| | FESP | 577 | 466 | 374 | 286 | 221 | 181 |
| #Rooms exceeding capacity | 3PSA | 0 | 0 | 0 | 0 | 0 | 0 |
| | FESP | 3 | 2 | 0 | 0 | 0 | 0 |

**Table 8. Results for subject problem SP2, $20 \leq \Pi \leq 40$**

| | | 20 | 24 | 28 | 32 | 36 | 40 |
|---|---|---|---|---|---|---|---|
| $S_{SE}$ | 3PSA | 2 | 0 | 0 | 0 | 0 | 0 |
| | FESP | 22 | 1 | 1 | 0 | 0 | 0 |
| $S_{CE}$ | 3PSA | 240 | 170 | 118 | 93 | 72 | 55 |
| | FESP | 332 | 264 | 212 | 183 | 165 | 97 |
| $S_{ME}$ | 3PSA | 510 | 372 | 297 | 225 | 196 | 160 |
| | FESP | 526 | 422 | 344 | 264 | 216 | 168 |
| #Rooms exceeding capacity | 3PSA | 3 | 1 | 0 | 0 | 0 | 0 |
| | FESP | 4 | 2 | 1 | 0 | 0 | 0 |

**Table 9. Results for subject problem SP3, $20 \leq \Pi \leq 40$**

| | | 20 | 24 | 28 | 32 | 36 | 40 |
|---|---|---|---|---|---|---|---|
| $S_{SE}$ | 3PSA | 1 | 0 | 0 | 0 | 0 | 0 |
| | FESP | 5 | 1 | 0 | 0 | 0 | 0 |
| $S_{CE}$ | 3PSA | 105 | 74 | 52 | 41 | 32 | 23 |
| | FESP | 136 | 106 | 77 | 67 | 54 | 49 |
| $S_{ME}$ | 3PSA | 219 | 169 | 122 | 101 | 78 | 61 |
| | FESP | 216 | 165 | 117 | 98 | 71 | 69 |
| #Rooms exceeding capacity | 3PSA | 0 | 0 | 0 | 0 | 0 | 0 |
| | FESP | 2 | 0 | 0 | 0 | 0 | 0 |

**Table 10. Results for subject problem SP4, $20 \leq \Pi \leq 40$**

| | | 20 | 24 | 28 | 32 | 36 | 40 |
|---|---|---|---|---|---|---|---|
| $S_{SE}$ | 3PSA | 4 | 1 | 0 | 0 | 0 | 0 |
| | FESP | 32 | 8 | 3 | 0 | 0 | 0 |
| $S_{CE}$ | 3PSA | 191 | 137 | 126 | 96 | 59 | 50 |
| | FESP | 431 | 304 | 250 | 242 | 162 | 152 |
| $S_{ME}$ | 3PSA | 395 | 283 | 270 | 190 | 171 | 125 |
| | FESP | 688 | 541 | 379 | 368 | 277 | 226 |
| #Rooms exceeding capacity | 3PSA | 6 | 3 | 0 | 0 | 0 | 0 |
| | FESP | 7 | 2 | 1 | 0 | 0 | 0 |

**Table 11. Results for subject problem SP5, $20 \leq \Pi \leq 40$**

| | | 20 | 24 | 28 | 32 | 36 | 40 |
|---|---|---|---|---|---|---|---|
| $S_{SE}$ | 3PSA | 2 | 0 | 0 | 0 | 0 | 0 |
| | FESP | 9 | 1 | 0 | 0 | 0 | 0 |
| $S_{CE}$ | 3PSA | 147 | 99 | 80 | 63 | 48 | 36 |
| | FESP | 195 | 130 | 118 | 100 | 64 | 51 |
| $S_{ME}$ | 3PSA | 287 | 246 | 185 | 137 | 123 | 110 |
| | FESP | 316 | 227 | 175 | 157 | 116 | 89 |
| #Rooms exceeding capacity | 3PSA | 3 | 2 | 0 | 0 | 0 | 0 |
| | FESP | 5 | 2 | 3 | 2 | 2 | 1 |

**Table 12. Range of execution times, in minutes**

| | Execution time |
|---|---|
| 3PSA | 2.6-3.8 |
| FESP | 2.4 – 4.0 |

## References


[1] Balakrishnan, N. (1991) Examination Scheduling: a Computerized Application *Omega*, 19 (1), pp. 37-41.

[2] Brailsford, S.C., Potts, C.N., Smith, B.M., (1999) Constraint Satisfaction Problems: Algorithms and Applications. *European Journal of Operational Research*, 119, pp. 557-581.

[3] Burke, E.K., Petrovic S. (2002) Recent Research directions in Automated Timetabling. *European Journal of Operational Research*, 140, pp. 266-280.

[4] Carter, M.W., (1986) A Survey of Practical Applications of Examination Timetabling Algorithms. *Operations Research*, 34, pp. 193-202.

[5] Carter, M., Johnson, D.G. (2001) Extended Clique Initialization in Examination Timetabling. *Journal of the Operational Research Society*, 52 (5), pp. 538-544.

[6] Carter, M.W., Laporte, G., Chinneck, J. (1994) A General Examination Scheduling System, *Interfaces*, 24 (3), pp. 109-120.

[7] Carter, M.W., Laporte, G., (1996) Recent Developments in Practical Examination Timetabling. In: E. Burk, P. Ross (Eds.), *Proceedings of the First International Conference on the Practice and Theory of Automated Timetabling*. Springer LNCS 1153, pp. 3-21.

[8] Deris, S., Omatu, S., Ohta H. (2000) Timetable Planning Using the Constraint Based Reasoning. *Computer and Operations Research*, 27 (9), pp. 819-839.

[9] Di Gaspero, L., Schaerf, A. (2000) Tabu Search Techniques for Examination Timetabling. *Proceedings of the Third International Conference on the Practice and Theory of Automated Timetabling*, Germany, pp.176-179.

[10] Erben, W. (2000) A Grouping Genetic Algorithm for Graph Coloring and Exam Timetabling. *Proceedings of the Third International Conference on the Practice and Theory of Automated Timetabling*, Germany, pp. 397-421.

[11] Ergul, A. (1996) GA-based Examination Scheduling Experience at Middle East Technical University, in E. Burke and P. Ross (Eds.), *The Practice and Theory of Automated Timetabling*, Springer LNCS 1153, pp. 212-226.

[12] Fisher, J.G., Sheir, D.R. (1983) A Heuristic Procedure for Large-Scale Examination Scheduling Problems. Technical Report 417, *Department of Mathematical Sciences*, Clemson University.

[13] Kirkpatrick, S., Gelatt, C., Vecchi, M. (1983) Optimization by Simulated Annealing, *Science*, 220, pp. 671-680.

[14] Lotfi, V., Cerveny, R. (1991) A Final-Exam-Scheduling Package, *Journal of the Operational Research Society*, 42, pp. 205-216.

[15] Tarhini, A., Mansour N. (1998) Natural Optimization Algorithms for Exam Scheduling, *Proceedings of International Conference on Computer Systems and Applications*, March-April, IASTED, ACTA Press, pp. 124-127.

[16] White, G., Xie, B.S. (2000) Examination Timetables and Tabu Search With Longer Term Memory. *Proceedings of the Third International Conference on the Practice and Theory of Automated Timetabling*. Germany, pp.184-201.