Scatter Search and Graph Heuristics for the Examination Timetabling Problem

Drifa Hadjidj[1] and Habiba Drias[2]
[1]Département d'Informatique, Université M'hamed Bouguerra, Algérie
[2]Institut National d'Informatique-INI-, Algérie

Abstract: *Examination timetabling problem is an optimization problem which consists in assigning a set of exams to a set of contiguous time slot, satisfying a set of constraints. The problem falls in the category of the NP-Complete problems and is usually tackled using heuristic methods. In this paper we describe a solution algorithm and its implementation based on the graph heuristics and the evolutionary meta-heuristic called scatter search which operates on a set of solutions by combining two or more elements. New solutions are improved before replacing others according to their quality and diversity. The implementation of the algorithm has been experimented on the popular carter's benchmarks and compared with the best recent results.*

Keywords: *Examination timetabling, scatter search, evolutionary, meta-heuristic, graph heuristics.*

Received December 4, 2006; accepted May 3, 2007

1. Introduction

Timetabling problems are generating considerable interest from researchers across the fields of operations research and artificial intelligence. A general timetabling problem includes assigning a set of events (courses, exams, meetings, etc.,) into a limited number of timeslots (periods). All timetabling problems share the same basic characteristics of the general timetable problem but, the amount of variance between different timetable problems is so greet as to make general solutions impracticable to implement. In our study we are mainly concerned with one type of university timetabling problem: the examination timetabling problem.

This problem regards the scheduling for a number of exams in a given set of periods avoiding overlap of exams with common students and spreading the exams for student as much as possible. However, as polices differ from university to university in defining the quality of exam timetabling; it becomes difficult to give a universal definition for this problem.

At the most basic form of this problem any feasible timetable must satisfy the constraint:

- No student may have two exams in the same period.

And in some cases:

- The seating capacity for any period should not be exceeded.

A university may also add other constraints, for example:

- If a student has to take two exams in any one day, there must be at least one complete period between the two exams.
- One exam must be held before another one
- Some exams may require the same period
- An exam may require a particular room

Thus, we have tow kinds of constraints: hard and soft.

- *Hard constraints* are constraints that can not be violated for physical reasons or administration rules For example; a person cannot be assigned to two different exams at the same time. Solutions which do not violate any of the hard constraints are called feasible solutions.
- *Soft constraints* are mainly preferences constraints, desirable but not essential. In most real world situations no solutions can be found to satisfy all the soft constraints. Therefore, a useful measure of the quality of a timetable can be taken to be the number of violations of these soft constraints. Minimising these violations is one of the overriding objectives. Generally, the violated soft constraints are aggregated into an objective function, which serves as an index of the solution quality. Thus, the goal of the examination timetabling process is to produce the feasible timetable of the highest possible quality (minimum value of the particular objective function).

Note that the examination timetabling problem is an NP-hard problem [27] which means that it is not possible in a reasonable time to carry out an exhaustive search for the timetable.

For instance, the number of ways to schedule 68 exams into 28 periods is 28^{68} and the vast majority of these timetables are infeasible, because hard constraints are not always satisfied. Thus the problem is to find good timetables in this vast search space which contains some feasible solutions. The manuel solution of timetabling usually requires several days and may be unsatisfactory to some respects. Many of the exam timetabling algorithms which have been developed are very problem-specific and cannot be used to find high quality solutions to a wide range of problem instances. In general, no one algorithm will give best results for all exam timetabling problems to which it is applied. Some algorithms will perform well on one subset of problems but fail when different constraints are introduced

In this paper we describe a solution algorithm and its implementation based on the evolutionary meta-heuristic called *scatter search* which operates on a set of solutions by combining tow or more elements. New solutions are improved before replacing others according to their quality and diversity. The implementation of the algorithm has been experimented on the popular carter's benchmarks and compared with the best recent results.

The paper starts with the description of the exam timetabling problem. In section 3 we present previous approaches followed in section 4 by the general template of scatter search. In section 5 we present our algorithm followed by the computational experimentation with the performance obtained, before concluding our contribution on the field.

2. Description Problem

The examination timetabling problem can be described as a given set of P periods and a set of N exams to place each exam in some period avoiding overlaps of exams with common students and spreading the exams for student as much as possible. Here we assume that all exams have the same duration period and we give a version based on [15, 19, 30] to allow comparison of quality of solutions produced by the best recent approaches. It is common in timetabling community to employ the penalty function for solution evaluation proposed by in [19] on his wide range of real world timetabling benchmark problems, available via internet: ftp://ftp.mie.utoronto.ca/pub/carter/tesprob/.

This penalty function enables comparison of quality of solutions produced by different approaches. Thus in our experiment we use the method of solution evaluation shown in equation 2 for measuring student spread based on proximity to minimize. The hard constraints that must be observed are presented in equations 3 and 4. Equation 3, stats that every event should be scheduled once, and only once in the timetable. Equation 4 specifies that no conflicting exams should be scheduled within the same period. As in [19] we assume that we have not a limitation on available seating for each period.

The penalty function sums all the penalties and divides the obtained sum by the number of students. The final result is the average penalty for each student. The smaller its value the better the solution is. Note that whenever a student has to sit two examinations s periods apart, such penalties are: $w_1 = 16$, $w_2 = 8$, $w_3 = 4$, $w_4 = 2$, $w_5 = 1$. Thus the notation used is as follows:

The minimization problem is defined as minimize

$$\left(\sum_{i=\square}^{N-\square}\sum_{j=i+\square}^{N}\sum_{p=\square}^{P}\sum_{q=\square}^{P}\square^{\square-|q-p|}\, t_{ip}\, t_{jq}\, c_{ij} + \sum_{i=\square}^{N}\square\square\square\square\, t_{i(P+1)}\right)$$

(1)

$$\text{If}\quad \begin{cases} \square \le |q-p| \le \square \\ \text{and 0 otherwise} \end{cases} \qquad (2)$$

Subject to

$$\sum_{p=\square}^{P+\square} t_{ip} = \square \qquad \forall i \in \{1, \ldots\ldots N\}$$

(3)

$$\sum_{i=\square}^{N-\square}\sum_{j=i+\square}^{N}\sum_{p=\square}^{P} t_{ip}\, t_{jp}\, c_{ij} = \square$$

(4)

where

$t_{ip} = 1$ if exam i is scheduled in period p, 0 otherwise.

c_{ij} is the number of students attending exam i and j.

M is the total number of students.

In some cases it may not be possible to construct a feasible timetable, each exam that cannot be scheduled in a valid period can be placed in an extra period, the period *(P+1)* which should be heavily penalised [8, 14].

3. Previous Approaches

A large number of variants of the examination timetabling problem have been proposed in the literature depending on the type of the institution and distinct constraints. In other respects, several techniques have been presented to solve each type of timetabling problem.

The early approaches include integer programming, network flow, and graph heuristics [22]. Some of these techniques are either impractical or too simple to solve complex or large timetabling problems.

- *Constraint programming logic* [7] has been employed over the years for timetabling problems.

- *Heuristic sequencing approach* [14, 19, 22], employed in educational timetabling, involves using heuristic, to estimate how difficult each exam will be to place in the timetable and to order them before placing each exam. These heuristics produce substantially better timetables than using a random ordering.

Recently, a large amount of successful research has investigated meta-heuristic approaches for a variety of timetabling problems. These include taboo search simulated annealing and, genetic algorithms [23, 28, 32]. These methods begin with one or more initial timetables and employ search strategies which try to avoid local optima. Extensive work has also been carried out to study other new approaches and methodologies for timetabling problems as more problem solving experience is collected. These include hyper-heuristics [17], hybrid methods [29], cluster methods [4], case-based reasoning [13], hybrid multi-objective evolutionary approach [21], ant algorithms [24], sequential methods with meta-heuristics [15], fuzzy methodologies [5], multicriteria approach, [9], and large neighbourhood search approach [1, 2, 3].

Note that in this paper we are not attempting to directly solve the examination timetabling problem. Instead, we are concerned with underpinning the development of general timetabling systems by investigating scatter search approach for this problem.

4. Scatter Search Approach

Scatter search is an evolutionary meta-heuristic population based for solving combinatorial and linear optimization problems. It has been proposed in 1960's by [25, 26] for combining decision rules and problem constraints. It uses strategies to construct solutions by combining others and updating the set of reference solutions used for combination. In fact, scatter search can be implemented in multiple ways and offers many alternatives for exploiting its fundamental ideas [25, 26]. Specifically this approach may be sketched as follows:

- *Diversification generation method* used to generate a starting set of solutions to guarantee a critical level of diversity and to rebuild a subset of the population during the search.
- *Improvement method* used to improve in quality new solutions obtained by the diversification generation method or by the solution combination method.
- *Reference set update method* builds and maintains a reference set consisting of the b_1 best solutions according to their quality, and the b_2 most diverse solutions (some scatter search procedures only uses the quality criterion to update).
- *Subset generation method* produces subsets of solutions with two or more elements from the reference set to be combined as a basis for creating combined solutions.
- *Solution combination method* transforms a given subset of solutions produced by the subset generation method into one or more combined solution using a problem dependent combination operator.

5. Scatter Search for the Examination Timetabling Problem

In this section, we describe the solution approach that we have developed for the examination timetabling problem using scatter search. The components are organized in some way as described in section 4.

5.1. Diversification Generation Method

In order to create a population of distinct solutions, we combine heuristics with the process of tournament selection. First, we generate a subset of exams randomly, we order them with respect to some heuristic, and then we assign them one by one into feasible timeslots without violating any hard constraint and with the lowest penalty. The size of the subset is given as percentage of the full examinations set [15, 19].

A variety of sequential heuristics can be used to construct initial solutions. They sort examinations on the basis of the estimated difficulty of their scheduling. The most interesting are: Largest Degree (LD) heuristic and Saturation Degree (SD) heuristic which are two widely studied graph coloring heuristics for applications to timetabling problems [6, 15, 19], see algorithm 1.

In LD exams are ordered decreasingly by the number of conflicts they have with other exams. This heuristic aims to schedule the most conflicting exams first. For SD exams that are not yet scheduled are ordered increasingly by the number of feasible timeslots available at that time. The priorities of the exams thus change dynamically according to the situations encountered at each step of the solution construction

Algorithm 1 diversification generation method, for each member of the population composed by *P* size timetables

A. Randomly built a tournament of exams with size at a set proportion of total size (for example 20% of total size).
B. Select in turn an exam from tournament using one of the precedent ordering strategies (to break ties we use the LD strategy).
C. Schedule the selected exam in the valid period causing least penalty. If there is a tie between periods schedule the exam at the earliest of the

periods. If no such period exists then add to the list of unscheduled exams.

Another way of maintaining the diversity in the population is to consider, from time to time, some assignments (we use 5% of the total number of exams) of exams to periods which have not been frequently used before.

Thus, a frequency matrix $N \times P$ is updated at each iteration of the scatter search process. When we wish to increase the diversity, we create a partial solution with $c \times N$ (we use $c= 0.05$) assignments of exams to periods which were not frequently used. The remaining $N\text{-}c \times N$ assignments of exams to periods are then computed by the diversification generation method.

5.2. Improvement Method

This algorithm iteratively inspects the neighbourhood (the set of candidate solutions, produced from the current one by moving one exam) and replaces the current solution by the candidate with lowest penalty. The hill climbing used to improve the solutions does not accept zero improvement and will not spend time exploring regions of no immediate improvement such as plateaus.

Algorithm 2 improvement method, for each period *p* in the timetable taken in some random order:

A. Take each exam e in period p in some random order.
B. Calculate the penalty of scheduling e in every other valid period.
C. If the best of these has low penalty than p, then schedule e in this period. Using the precedent ordering strategy SD, try scheduling each unscheduled exam in some valid period
D. If any improvement is found, repeat.

Note that at this stage, the SD strategy produces better results than the other graph coloring heuristics: the resulting timetabling solutions have no unscheduled exams.

5.3. Reference Set Update Method

The construction of the initial reference set starts with the selection of the best b_1 solutions from the population *P*. These solutions are added to the reference set *Refset* and deleted from *P*. For each solution in *P-refset* we compute the minimum of the distances to the solutions in *Refset*, then, we select the solution with the maximum of these minimum distances. This solution is added to *Refset* and deleted from *P* and the minimum distances are updated. Thus the reference set *Refset* composed by *b* solutions is a subset of the population that consists of both $Refset_1$ composed by b_1 high quality solutions and $Refset_2$ composed by b_2 diverse solutions. It is used to generate new solutions by applying the solution combination method.

An improved new solution *t* replaces another in $Refset_1$ if its evaluation is better than the worst evaluation solution in $Refset_1$. If *t* is not kept due to its quality it replaces another in $Refset_2$ if its diversity is better then the worst one in $Refset_2$. The distance functions between two solutions t and t' used to measure the diversity in reference set *Refset is:*

$$\partial(t,t') = \sum_{i=1}^{P}\sum_{j=1}^{E}\left|t_{ij} - t'_{ij}\right| \quad (5)$$

5.4. Subset Generation Method Construction

This subset generation procedure consists of creating different subsets of *Refset* as basis for the subsequent combination method. The approach for doing it is organized to generate different collections of subsets from *Refset* [25, 26]. Each subset is generated only once. The smallest subsets (denoted by 2-subsets) consist of all 2-elements subsets from *Refset.* The subsets a-subsets, a $\in$ {3, 4} are built by augmenting each (a-1)-subset with the best evaluation solution not included in the subset. Finally, c-subsets c$\in$ {5...|*Refset*|} contains the best c-elements.

In our implementation, we found that most of the searching power can be attributed to the combination of the 2-subsets.

5.5. Solution Combination Method

The combination method as well as the improvement method is context dependent. The method we implement consists of generating a new solution from the combination of each 2-subset solutions, see Algorithm 3. The solutions generated are subjected to the improvement method, before they are considered for membership in *Refset.* The combination method continues until there is no change in *Refset* and all the subsets have been used. At this point, the *Refset* is partially rebuilt with a diversification update.

Algorithm 3 solution combination method, we combine the two timetables of a subset in order to produce a new timetable:

A. Initially, all exams that are not scheduled in either of the two parents are inserted in the not yet scheduled list set.
B. We then consider the first period in both timetables
C. If an exam is scheduled in this period in both parents then it is scheduled in this period in the child timetable.
D. Exams that are scheduled in this period in only one of the parents are added to the not scheduled list.

E. We then attempt to schedule the exams from the not scheduled list into the current period in the child timetable in order with one of the measures described below (to break ties we use the LD strategy).
F. Exams that cannot be scheduled are left in the not scheduled list and passed on to the next period.

If both parents have unscheduled events we consider that the main problem is scheduling all exams within the limited number of periods. Therefore we use the largest weighted degree heuristic where the highest priority is given to the examination with the largest sum of the weighted conflicts: each conflict is weighted by the number of students involved.

If either or both of the parents do not have unscheduled exams, we use the least conflicting with previous period heuristic which orders the exams by their number of conflicts with the exams that have already been scheduled in the child's previous period. This is aimed at reducing the number of near clashes [12, 15].

6. Experimental Experiences

The purpose of this experiment is to evaluate the performance of our approach for the examination timetabling problem. We assume that there are no limits on the amount of seating available during each period, in order to enable comparison with a wider range of existing methods based on this assumption. We implement and run the algorithm using Visual C++ 6, on a Pentium III 851 MHz machine with 256 Mb RAM.

The algorithm was tested on 5 real world benchmark exam timetabling problems using the carter's data collection [19]. These datasets presented in Table 1, cover a range of characteristics (on number of exams and conflict matrix density). In Table 2, we show the performance of our approach in comparison with selected recently published results on the datasets. The best results amongst the compared techniques for each dataset are highlighted in bold font.

As the selection of the search parameters has an important effect on the performance of the metaheuristic, computational analyses are conducted and these parameters are fixed as follows:

- The population size (*P*)= 200.
- The reference set size (*b*)= 20.
- The number of quality solutions (b_1)= 10.
- The number of diverse solutions (b_2)= 10.
- The size of the subset in the process of tournament selection= 20.
- The number of iterations= 2.

We can observe that the performance of our algorithm with these parameters is comparable to the others and is better for the UTE-S-92 problem. Another advantage of this approach is that reference set contains a number of high-quality solutions from which the decision-maker could choose the one to use.

7. Conclusion

In this paper we have developed and implemented an approach based on the scatter search evolutionary meta-heuristic to the examination timetabling. The results by this method, enhanced by the use of the most performing graph heuristics, are encouraging. We plant to improve our work by considering other soft constraints and implementing a parallel scatter search to reduce the CPU computation time.

Table 1. Characteristics of the problems used for testing.

Datasets	No. of Exams	No. of Students	No. of Periods	Density
HEC-S-92	81	2823	18	0.42
STA-F-83	139	611	13	0.14
UTA-S-92	622	21266	35	0.13
UTE-S-92	184	2750	10	0.08
EAR-F-83	190	1125	24	0.27

Table 2. Results comparison.

Datasets	Ute92	Uta92	Sta83	Hec92	Ear83
Our Scatter Search approach	**22.94**	3.52	138.12	10.63	30.90
Abdullah et al. (2006) [1]	24.21	3.63	150.28	10.28	34.87
Abdullah et al. (2006) [3]	26.00	3.60	159.00	10.80	36.00
Asmuni et al. (2004) [5]	27.78	3.75	160.42	11.78	37.02
Burke et al. (2004) [8]	25.70	3.50	159.10	10.60	35.00
Burke et al. (2005) [10]	35.40	4.52	158.20	-	45.60
Burke et al. (2007) [11]	28.01	**2.88**	158.19	12.25	37.92
Burke et al. (2002) [12]	25.83	3.20	168.73	11.54	37.05
Burke et al. (2006) [16]	24.82	3.06	157.03	10.15	32.76
Caramia et al. (2001) [18]	24.40	3.50	158.20	**9.20**	**29.30**
Carter et al. (1996) [19]	25.80	3.50	161.50	10.80	36.40
Casey et al. (2002) [20]	25.40	-	**134.90**	10.80	34.80
Cote et al. (2005) [21]	25.30	3.50	157.00	10.40	34.20
DiGaspero et al. (2001) [23]	29.00	4.20	160.80	12.40	45.70
Eley et al. (2006) [24]	27.70	3.80	157.20	11.30	36.80
Kendall et al. (2005) [28]	27.12	3.77	157.38	10.74	38.14
Merlot et al. (2003) [29]	25.10	3.50	157.30	10.60	35.10
Petrovic et al. (2003) [30]	25.33	3.14	165.27	10.83	33.75
White et al. (2004) [33]	29.00	4.44	158.00	12.90	45.80
Yang et al. (2004) [34]	25.53	3.14	151.15	10.83	33.70

References

[1] Abdullah S., Ahmadi S., Burke E., and Dror M., “Investigating Ahuja-Orlin's Large Neighbourhood Search Approach for Examination Timetabling,” *OR Spectrum*, vol. 29, no. 2, pp. 351-372, 2007.

[2] Abdullah S., Ahmadi S., Burke E., Dror M., and McCollum B., “A Tabu Based Large Neighbourhood Search Methodology for the Capacitated Examination Timetabling Problem,” *Journal of Operational Research Society,* vol. 58, no. 11, pp. 1494-1502, 2007.

[3] Abdullah S. and Burke E., “A Multi-start Large Neighbourhood Search Approach with Local Search Methods for Examination Timetabling,” *in Long D., Smith S., Borrajo D., and McCluskey L. (eds.)*, *The International Conference on Automated Planning and Scheduling (ICAPS‘2006)*, Cumbria, UK, pp. 334-337, 2006.

[4] Arani T. and Lotfi V., “A Three Phased Approach to Final Exam Scheduling,” *IIE Transactions*, vol. 21, no. 1, pp. 86-96, 1989.

[5] Asmuni H., Burke E., Garibaldi J., and McCollum B., “Fuzzy Multiple Ordering Criteria for Examination Timetabling,” *in Burke E. and Trick M. (eds.)*, *Practice and Theory of Automated Timetabling (PATATV’2004)*, Pittsburgh, Lecture Notes in Computer Science Springer-Verlag, vol. 3616, pp. 334-353, 2005.

[6] Brelaz D., “New Methods to Colour the Vertices of a Graph,” *Communications of the Association for Computing Machinery*, vol. 22, no. 4, pp. 251-256, 1979.

[7] Boizumault P., Delon Y., and Peridy L., “Constraint Logic Programming for Examination Timetabling,” *Journal of Logic Programming,* vol. 26, no. 2, pp. 217-233, 1996.

[8] Burke E., Bykov Y., Newall J., and Petrovic S., “A Time-Predefined Local Search Approach to Exam Timetabling Problems,” *IIE Transactions on Operations Engineering*, vol. 36, no. 6, pp. 509-528, 2004.

[9] Burke E., Bykov Y., and Petrovic S., “A Multicriteria Approach to Examination Timetabling,” *in Burke E. and Erben W. (eds.)*, *Practice and Theory of Automated Timetabling (PATAT’2000)*, Lecture Notes in Computer Science Springer-Verlag, Constance, Germany, vol. 2079, pp. 118-131, 2001.

[10] Burke E., Dror M., McCollum B., Petrovic S., and Qu R., “Hybrid Graph Heuristics within a Hyper-heuristic Approach to Exam Timetabling Problems,” *in Golden B., Raghavan S., and Wasil E. (eds.)*, *The Next Wave in Computing, Optimization and Decision Technologies,* Maryland, vol. 29, no. 1, pp. 79-91, 2005.

[11] Burke E., McCollum B., Meisels A., Petrovic S., and Qu R., “A Graph-Based Hyper Heuristic for Educational Timetabling Problems,” *European Journal of Operational Research,* vol. 176, no. 1, pp. 177-192, 2007.

[12] Burke E. and Newall J., “Enhancing Timetable Solutions with Local Search Methods,” *in Burke E. and Causmaecker P. (eds.)*, *Practice and Theory of Automated Timetabling (PATAT’2002)*, Lecture Notes in Computer Science Springer-Verlag, Gent, Belgium, vol. 2740, pp. 195-206, 2003.

[13] Burke E., Petrovic S., and Qu R., “Case Based Heuristic Selection for Timetabling Problems,” *Journal of Scheduling,* vol. 9, no. 2, pp. 115-132, 2006.

[14] Burke E. and Newall J., “Solving Examination Timetabling Problems Through Adaption of Heuristic Ordering,” *Annals of Operations Research*, vol. 129, no. 2, pp. 107-134, 2004.

[15] Burke E., Newall J., and Weare R., “A Simple Heuristically Guided Search for the Timetable Problem,” *in Alpaydin E. and Fyte C. (eds.)*, *The International ICSC Symposium on Engineering of Intelligent Systems (EIS'98),* University of La Laguna, Spain, ICSC Academic Press, pp. 574-579, 1998.

[16] Burke E. and Bykov Y., “Solving Exam Timetabling Problems with the Flex-Deluge Algorithm,” *in Proceedings of the Practice and Theory of Automated Timetabling Conference (PATAT’2006)*, Brno, Czech, pp. 167-180, 2006.

[17] Burke E., Kendall G., and Soubeiga E., “A Tabu Search Hyperheuristic for Timetabling and Rostering,” *Journal of Heuristics*, vol. 9, no. 6, pp. 451-470, 2003.

[18] Caramia, M., DellOlmo P., and Italiano G., “New Algorithms for Examinations Timetabling,” *in Naher S. and Wagner D., (eds.)*, *Algorithm Engineering 4th International Workshop WAE 2000,* Saarbrucken, Germany, Lecture Notes in Computer Science, Springer-Verlag, vol. 1982, pp. 230-241, 2001.

[19] Carter M., Laporte G., and Lee S., “Examination Timetabling: Algorithmic Strategies and Applications,” *Journal of the Operational Research Society*, vol. 47, no. 3, pp. 373-383, 1996.

[20] Casey S. and Thompson J., “A Hybrid Algorithm for the Examination Timetabling Problem,” *in Burke E. and Causmaecker P. (eds.)*, *Practice and Theory of Automated Timetabling (PATAT’2002),* Gent Belgium, Lecture Notes in Computer Science Springer-Verlag, vol. 2740 pp. 205-230, 2002.

[21] Côté P., Wong T., and Sabourin R., "Application of a Hybrid Multi-Objective Evolutionary Algorithm to the Uncapacitated Exam Proximity Problem," *in Burke E. and Trick M. (eds.), Practice and Theory of Automated Timetabling (PATAT'2004)*, Pittsburgh, Lecture Notes in Computer Science Springer-Verlag, vol. 3616, pp. 294-312, 2005.

[22] De Werra D., "An Introduction to Timetabling Problem," *European Journal of Operational Research,* vol. 19, no. 2, pp. 151-162, 1985.

[23] Di Gaspero L. and Schaerf A., "Tabu Search Techniques for Examination Timetabling," *in Burke E. and Erben W. (eds.), Practice and Theory of Automated Timetabling (PATAT'2000),* Constance, Germany, Lecture Notes in Computer Science Springer-Verlag, vol. 2079, pp. 104-117, 2001.

[24] Eley M., "Ant Algorithms for the Exam Timetabling Problem," *in Proceedings of the Practice and Theory of Automated Timetabling Conference (PATAT'2006)*, Brno, Czech, pp. 167-180, 2006.

[25] Glover F., Laguna M., and Marti R., "Fundamentals of Scatter Search and Path Relinking," *Control and Cybernetics,* vol. 29, no. 3, pp. 653-684, 2000.

[26] Glover F., Laguna M., and Marti R., "Scatter Search and Path Relinking: Advances and Applications," *in Glover F. and Kochenberger G. (eds.), Handbook of Meta-Heuristics*, Kluwer, pp. 1-36., 2003.

[27] Karp R., "Reducibility Among Combinatorial Problems," *in Miller R. and Thatcher J. (eds.), Complexity of Computer Computations,* Plenum Press, New York, pp. 85-103, 1972.

[28] Kendall G. and Mohd N., "An Investigation of a Tabu Search Based Hyperheuristic for Examination Timetabling," *in Kendall G., Burke E., and Petrovic S. (eds.), Multidisciplinary Scheduling: Theory and Applications*, 1st International Conference (MISTA'03), Nottingham, UK, pp. 309-328, 2005.

[29] Merlot L., Boland N., Hughes B., and Stuckey P., "A Hybrid Algorithm for Examination Timetabling Problem," *in Burke E. and Causmaecker P. (eds.), Practice and Theory of Automated Timetabling (PATAT'2002),* Lecture Notes in Computer Science Springer-Verlag, Gent, Belgium,, vol. 2740, pp. 207-231, 2003.

[30] Petrovic S., Yang Y., and Dror M., "Case-Based Initialisation for Examination Timetabling," *in Proceedings of the 1st Multidisciplinary International Conference on Scheduling: Theory and Applications (MISTA'2003)*, Nottingham, UK, pp. 137-154, 2003.

[31] Schaerf A., "A Survey of Automated Timetabling," *Artificial Intelligent Review*, vol. 13, no. 2, pp. 87-127, 1999.

[32] Thompson J. and Dowsland K., "A Robust Simulated Annealing Based Examination Timetabling System," *Computers and Operations Research*, vol. 25, no. 7, pp. 637-648, 1998.

[33] White G., Xie B., and Zonjic S., "Using Tabu Search with Longer-Term Memory and Relaxation to create Examination Timetables," *European Journal of Operational Research,* vol. 153, no. 16, pp. 80-91, 2004.

[34] Yang Y. and Petrovic S., "A Novel Similarity Measure For Heuristic Selection in Examination Timetabling," *in Burke E. and Trick M. (eds.), Practice and Theory of Automated Timetabling (PATAT'2004)*, Lecture Notes in Computer Science Springer-Verlag, Pittsburgh, vol. 3616, pp. 377-396, 2005.

Drifa Hadjidj is a lecturer in the Department of Computer Science at Boumerdes University, Algeria. She holds the engineering degree in computer science in 1985, and the Master degree in operational research in 1999, from Algiers USTHB University. Her current research interests are in the area of timetabling problems, meta-heuristics, computational complexity, drawing graphs, and applications of operational research and artificial intelligence.

Habiba Drias holds the Master degree in computer science from Case Western Reserve University, Cleveland OHIO USA in 1984, and the Doctorate degree prepared at Paris6 University from Algiers USTHB University in 1993. She has directed the computer science institute of USTHB and then the laboratory of research in artificial intelligence for many years. She has over a hundred published papers in the domain of artificial intelligence, e-commerce, and computational complexity and the satisifiability problem. She is currently the president of the National Institute of Informatics-INI-.