A hybrid Ant Colony Algorithm for the exam timetabling problem

R. Abounacer, J. Boukachour, B. Dkhissi, A. El Hilali Alaoui

A hybrid Ant Colony Algorithm for the exam timetabling problem

Algorithme de Colonie de Fourmis hybride pour le problème de planification horaire des examens

R. Abounacer[(1,2)], J. Boukachour[(2)], B. Dkhissi[(1)], A. El Hilali Alaoui[(1)]

1. Laboratory of Modeling and Scientific Computing
Faculté des Sciences et techniques de Fès Saiss, Maroc
rachida1630@yahoo.com, dkhissi_btissam@yahoo.fr, elhilali_fstf2002@yahoo.fr
2. CERENE, ISEL Quai Frissard B.P. 1137 - 76063 Le Havre Cedex France
boukachour@univ-lehavre.fr

RÉSUMÉ. Vue l'augmentation du nombre d'étudiants dans les établissements scolaires et universitaires, le nombre d'examens à passer par chaque étudiant et les réformes pédagogiques actuelles, les planifications classiques des cours et des examens ne sont plus suffisantes, ce qui a amené les chercheurs opérationnels et les informaticiens à chercher des nouvelles méthodes pour résoudre le problème d'emploi du temps des examens. Notre travail consiste à planifier les examens de telle sorte à maximiser le temps de séparation entre deux examens consécutifs pour chaque étudiant et ceci en utilisant l'algorithme de colonies de fourmis hybridé avec une technique de recherche locale.

ABSTRACT. Due to increased student numbers and regulation changes educational institutions that allow for greater flexibility, operations researchers and computer scientists have renewed their interest in developing effective methods to resolve the examination timetabling problem. Thus, in the intervening decades, important progress was made in the examination timetabling problem with appearance of adaptation of meta-heuristics. This paper presents a hybridization of the Ant Colony Algorithm and a Complete Local search with Memory heuristic, in order to maximize as much as possible; the free time between consecutive exams for each student, while respecting the conflict constraints, a student cannot sit more than one exam in the same timeslot.

MOTS-CLÉS : problème de planification horaire des examens, algorithme de colonie de fourmis, recherche locale complète avec mémoire.

KEYWORDS: exam timetabling problem, ant colony algorithm, complete local search with memory.

Reçu le 17 juin 2009,
révisé le 11 mai 2010,
accepté le 12 juin 2010.

Revue ARIMA, vol. 12 (2010), pp. 15-42

1. Introduction

The examination timetabling problem is concerned with assigning a set of exams to a limited number of timeslots, subject to a set of constraints. In the literature [17], these constraints are usually categorized into two types: hard constraints and soft constraints. Both of them vary widely from one institution to another according to resource limitations such as the total capacity of the establishment and according to the institutions desired objectives, for example the reduction of the planning horizon. The hard constraints (or feasibility constraints) must be satisfied under all circumstances; they determine whether or not a solution is feasible. The soft constraints are desirable but not absolutely decisive; they can be used to measure the quality of the feasible solutions.

Due to the large variety of examination timetabling problems presented and investigated, a lot of constraints exist. The most important of these constraints are discussed in a review by Carter *et al.* [17]. They are as follows:

– Hard constraints:

- A student cannot sit more than one exam in the same timeslot
- Certain exams must be assigned to the same timeslot
- The maximum seats of the university (the total capacity of the university) may not be exceeded at any timeslot
- Some exams must precede others (the precedence constraint)
- Some exams must be assigned only to a subset of the available timeslots
- No student must sit more than *x* examinations in any *y* consecutive timeslots

– Soft constraints:

- Spread conflicting exams as even as possible for each student [51, 41, 9, 13, 11, 39, 43]. (Two exams are in conflict if they have common students).
- Minimize the number of timeslots needed [19, 18]
- Minimize the number of students sitting two exams in a room on the same day [15, 39]
- Minimize the number of students for two or more exams on consecutive days [12, 39]
- Minimize the number of students sitting two adjacent exams the same day [47, 45]

The high level of research interest in examination timetabling problem has led to the establishment of different variants of the problem called *Toronto a*, *Toronto b*,

Toronto c, *Toronto d* and *Toronto e* [44] and defined according to a set of objectives listed in Table1, which have provided a way for meaningful scientific comparisons and the exchange of research achievements.

Variants	**Objectives**
Toronto *a*	To minimize the number of timeslots needed
Toronto *b*	To space out conflicting exams within limited timeslots
Toronto *c*	To minimize students sitting two exams in a row on the same day
Toronto *d*	Same as above, and to minimize students sitting two exams overnight
Toronto *e*	To minimize students sitting two adjacent exams the same day

Table1. *Variants of the Toronto's examination timetabling problem*

These different variants are studied in the literature, resolved by several approaches and tested using a set of benchmark datasets.

This work deals with the resolution of the *Toronto b* variant. This variant is modelled as a combinatorial optimization problem, which aims to maximize the free time between two consecutive exams for each student, subject to the conflict constraint which requires that a student may not sit for more than one exam in the same timeslot.

Mathematically the variant *Toronto b* can be formulated as indicated below by the formulas (1) and (2). The objective function (1) represents proximity between exams. If a student has two consecutive exams then a penalty value equal to 16 is assigned. Two exams with one empty timeslot between them will be assigned a penalty value of 8. Two empty timeslots correspond to a penalty of 4 and so on. In order to have a relative measure, this sum is divided by the total number of students.

The hard constraint (2) ensures that no student can sit for more than one exam in the same timeslot.

$$Min \frac{\sum_{h=1}^{T}\sum_{k=1}^{T}\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} E_{ij}\, x_{ih}\, x_{jk}\, D(h,k)}{M} \quad (1)$$

Subject to

$$\sum_{h=1}^{T} E_{ij} x_{ih}\, x_{jh} = 0 \qquad \forall\, i, j = 1,\ldots, N \;\; i \neq j \tag{2}$$

Where:

– N is the number of exams

– M is the number of students

– T is the number of timeslots

– The matrix $E = (E_{ij})_{N\times N}$ where each element E_{ij} is the number of students that have to take both exams i and j.

– The decision variable is:

$$x_{ih} = \begin{cases} 1 & \text{if the exam i is affected to the timeslot h} \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

– The penalty term D(h,k) is defined as follow:

$$D(h,k) = \begin{cases} 2^{5-|h-k|} & \text{if } 1 \leq |h-k| \leq 5 \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

h and *k* are two different timeslots of the planning horizon.

In the following section, various methods are presented of resolving the examination timetabling problem, as found in the literature. The third section is devoted to our proposed approach, which is a hybridization of Ant Colony algorithm and Complete Local search with Memory method. The last section is dedicated to the experimental results.

2. Related work

Garey and Johnson [34] have proved that the examination timetabling problem is NP-hard. In the last decade, several approaches have been proposed and appeared efficient to resolve the examination timetabling problem. We give an overview of different approaches of resolution. Many successful approaches mix a number of techniques.

2.1. Graph based sequential techniques

Graph based sequential techniques are widely studied methods which were developed during the early days of research on timetabling problems. They are used in sequential (or constructive) solution methods to order the events that are not yet scheduled according to the difficulties of scheduling them into a feasible timeslot (without violating any hard constraints).

Several graph based sequential techniques have been successfully applied to examination timetabling problem. In 1996 Carter, Laporte and Lee [19] studied five ordering strategies on real and randomly generated exam timetabling problems: saturation degree (SD: gives priority to the examinations having the least number of available timeslots for placements), largest degree (LD: gives priority to the exams having the largest number of conflicts), largest weighted degree (LWD: prioritises the examination having the largest weighted conflicts, where each conflict is weighted by the number of students enrolled in two examinations), largest enrolment (LE: prioritises the examinations with the largest student enrolments) and Largest color degree (LC: gives priority to the examinations with the largest number of conflicts with already scheduled examinations). In 1964, Broder [7] used the largest degree strategy to resolve the final examination scheduling problem. In 1968, Wood [52] used the largest enrolment strategy in examination timetabling problem. Saturation degree was defined by Brelaz in 1979 [6]. In 2004, Burke and Newall [14] investigated a dynamic ordering strategy which ordered the exams adaptively during the problem solving in an iterative process. Asmuni et al. [4] in 2004 have employed a Fuzzy methodology which has applied successfully in a wide range of real world applications. In the examination timetabling context, a fuzzy expert system is used to rank exams based on an assessment of how difficult it is to schedule the exams taking into account multiple criteria. By considering more than one criteria to rank the exams, it is hoped that rankings are produced that better reflect the actual difficulty of placing the exam, as several factors are simultaneously taken into account.

2.2. Constraint based techniques

The constraint based techniques are aimed at simplifying the resolution of a problem or demonstrating the non-existence of solution. Thus, they do not constitute a technique for resolving the problem, but provide instead a set of techniques for rewriting the constraints, thereby facilitating the exploration of the solution set, without modify it. Schematically, at each step, the following cycle is repeated:

– Choice of a subset of constraints and generation of an induced constraint

– Put in evidence of a redundant constraint relative to the induced constraint

– Modification of the set of the constraints: addition of the induced constraint and exclusion of the redundant constraint

Practically, that can consist to remove a set of decision variables whose values do not belong to any solution: this filtering of the fields avoid many attempts at solutions that lead to eventual failure. These techniques also allow simplifying the expression of the constraints, for example by elimination of the redundant constraints.

The investigation of constraint based techniques to exam timetabling has attracted the attention of the timetabling community for many years. In [22], the authors have modelled an exam timetabling problem faced by a French school as a constraint satisfaction problem and implemented a two-phase enumeration algorithm: "*preassignment*" and "*final assignment*" phases. In 2003 Merlot et al. [39] employed a constraint based technique to produce initial solutions, in a similar way to [5] using the Optimization Programming Language (OPL) [37], which is a new modelling language for combinatorial optimization that simplifies the formulation and solution of optimization problems. Subsequently, a simulated annealing and a hill climbing method (which are stochastic optimization procedures that are widely applicable and effective in several problems), were used to improve the solutions. In 2004, Duong and Lam [31] employed a constraint based technique to generate initial solutions for a simulated annealing methodology for the exam timetabling problems at Ho Chi Minh City University of Technology.

2.3. Local search based techniques

In local search techniques, the first step is to obtain any feasible solution to the problem. Given some feasible solution, all the "nearby" solutions are considered, and the better among them is chosen. This process is repeated till a locally optimal solution (a solution which is better than all its nearby solutions) is obtained.

For the examination timetabling problem, efficient algorithms exist for refining arbitrary points in the search space into better solutions. Such algorithms, called local search based techniques, define a neighbourhood's structure, typically based on initial crude solutions. This structure is used to iteratively improve an initial solution. In 1998, Thompson and Dowsland [49] carried out valuable work to develop a two-stage approach where feasible solutions from the first stage were improved in the second stage by simulated annealing concerning soft constraints. In 2001, Di Gaspero and Schearf [24] carried out a valuable investigation on a family of tabu search-based techniques whose neighbourhoods were in regions around hard or soft constraints. In the same year, White and Xie [50] developed a four-stage tabu search called OTTABU for the exam timetabling problem at the University of Ottawa. The solutions were gradually improved by considering more constraints at each stage. Also, in 2001, Caramia *et al.* [16] developed a fine-tuned local search method where a greedy scheduler assigned exams

into the least possible number of timeslots and a “penalty decreaser” improved the timetable without increasing the number of timeslots. In 2002, Di Gaspero [23] presented a family of Tabu Search algorithms for the examination timetabling problem. The algorithm relies on multiple neighbourhoods and is based on a token-ring search which circularly employs recolor (change single exam) and shake (swapping groups of exams), followed by kickers (change sequence of single exams) to further improve the obtained solutions obtained in 2001[24]. The technique extended the idea of diversifying the search from local optima. In 2003, Merlot *et al.* [39] employed a simulated annealing approach initialised by constraint programming techniques and followed by hill climbing to further improve the solution. The hill climbing approach, proposed for the timetabling problem by Appleby *et al.* [3] in 1961, inspects iteratively the neighbourhood and replaces the current solution by a candidate with better fitness. In 2004 Yang and Petrovic [43] employed case-based reasoning to choose graph heuristics which are then used to construct initial solutions for the “great deluge” algorithm. They obtained the best results reported in the literature for several Toronto instances. In 2004, Burke *et al.* [9] studied a variant of simulated annealing, called the “great deluge” algorithm which is proposed by G. Dueck in 1993 [30]. This algorithm, like simulated annealing, may accept worse candidate solutions. The worse solution is accepted if its fitness is less than or equal to some given upper limit, called a *level*. The value of *level* does not depend upon the current solution: at the beginning, the initial value (the only input parameter) of *level* is equal to the initial cost and at every iteration, it is lowered by a fixed decay rate. In 2006, Burke *et al.* [8] investigated variants of variable neighbourhood search and obtained the best results in the literature across some benchmark Exam Timetabling Toronto datasets. The results were further improved by using standard genetic algorithms to intelligently select subsets of neighbourhoods. Also in 2006, Abdullah *et al.* [1] developed a large neighbourhood search based on an improved graph construction methodology that had originally been developed by Ahuja and Orlin [2] for different optimization problems.

2.4. Population based techniques

Population-based methods deal with a set (i.e., a population) of solutions rather than with a single solution. At each iteration, a one of a number of techniques is applied to the current population to generate the population of the next generation. Therefore, population-based algorithms provide a natural, intrinsic way for the exploration of the search space.

Various population based techniques have been proposed to resolve the examination timetabling problem. Burke *et al.* [15] in 1996 developed a mimetic algorithm which employs light and heavy mutation operators to reassign single exams and sets of exams, respectively, with the aim of escaping from local optima. Burke and Newall [12] in 1999

studied the decomposition of the exam timetabling problems by iteratively assigning a subset of the *n* most difficult exams as measured by graph heuristics. The exams assigned in previous stage are fixed and the sub-problem at the current stage is solved using the Mimetic Algorithm investigated. Terashima-Marin *et al.* [48] in 1999 designed a clique-based crossover operator on timetabling problems that was transferred into graph colouring problems. Erben [32] in 2001 developed a grouping genetic algorithm where appropriate encoding and fitness functions were studied. Sheibani [46] in 2002 built a special mathematical model whose purpose is to minimize interference of exam times for each student, and developed a standard genetic algorithm for solving exam timetabling problems in training centres with the objective of maximising the intervals between the exams. The fitness value for each chromosome is calculated according to the product of a weight and a distance between two exam subjects.

Naji Azimi [40] in 2004 implemented an ant colony system and compared it with simulated annealing, tabu search and a genetic algorithm under a unified framework for solving a systematically designed exam timetabling problems. Our ACO approach is different from that of Azimi's one. This difference lies in the choice of the various parameters. Azimi generated randomly some instances of various sizes. Our approach has been tested using datasets widely employed in exam timetabling literature. Dowsland and Thompson [29] in 2005 developed ant algorithms based on the graph colouring model studied in [21] for solving exam timetabling problems without soft constraints.

2.5. Multi-criteria techniques

In the majority of approaches on timetabling, weighted costs of violations of different constraints are summed and used to indicate the quality of the solutions. However, in real world circumstances, the simple sum of costs on different constraints cannot always take care of the situation in such cases. Multi-criteria techniques have been studied in timetabling with the aim of handling different constraints easily by considering a vector of constraints instead of a single weighted sum.

Burke *et al.* [10] in 2001 developed a two-stage multi-criteria approach dealing with nine criteria in exam timetabling problems. A multi-criteria method called compromise programming [53] was used where the quality of the solutions was evaluated by the distance between them to an ideal point representing optimal solutions concerning all criteria. This technique was further studied in [42] by Petrovic and Bykov. Their multi-criteria approach is based on the "great deluge" algorithm [30].

3. Proposed approach

In the first of this section, we define the constructive meta-heuristics which build solutions iteratively from problem-specific solution components. All constructive heuristics take an empty solution and successively add solution components to build a complete typically feasible solution to a problem. Generally, the solutions built by constructive meta-heuristics are improved by the application of a local search procedure [33, 27] which represents the gradual improvement of a current solution(s) starting from initial one(s) until some stopping condition is satisfied.

ACO is a constructive meta-heuristic in which successive populations of artificial ants build solutions, guided by a model of the solutions that may be produced, called a pheromone representation. Thus, the ant colony algorithm provides good starting points for local search [28]. So, in the following, we present the two techniques used in our proposed approach to resolve the examination timetabling problem: the ACO and the CLM.

3.1. Ant colony algorithm for the examination timetabling problem

Ant colony optimization (ACO) is a technique inspired by the work of biologists. It has been taken up by several scientists and mathematicians and has been widely exploited and developed in the early 90s [20, 25, 26]. ACO is a simulation of a set of agents that cooperate to find a solution of an optimization problem by relying on uncomplicated communication. The term stigmergy is used to describe this kind of behavioural feedback.

Stigmergic effects are present generally in the collective behaviour of many social insects and in ant's colonies in particular. Many of these interactions are mediated by pheromones, volatile chemicals secreted by individual insects that cause changes in other individuals' behaviours. So ants exchange information by laying down pheromones on their way back to the nest when they have found food. In that way, they collectively develop a complex network of trails, connecting the nest in the most efficient way to the different food sources.

The ants use a stochastic construction technique that employs probabilistic decisions on the basis of artificial pheromone trails and heuristic information. The stochastic component allows the ants to explore a greater number of solutions. At the same time, the use of heuristic information helps guide the ants toward promising search regions using the collective interaction of a population of ants. Pheromone is accumulated during iterations through a learning mechanism implied in the pheromone update rule. In particular, ants move from node to node on a constructed graph using a transition rule that favours shorter edges.

The pheromone trail is updated of their generated tours based on a pheromone update rule. This rule consists of depositing a quantity of pheromone proportional to the quality of the corresponding tour; and thereafter evaporating the pheromone trail to avoid an unlimited accumulation of pheromone that could lead to premature convergence to a suboptimal solution region. This mechanism favours the exploration of the search space.

In the following of this section, an adaptation of the ACO to the examination timetabling problem is presented. The aim is to obtain the best assignments of the available timeslots to the examinations by maximizing the separation between two consecutive exams for each student according to the formula (1) and respecting the conflict constraint presented by the equation (2).

3.1.1. Solution representation

Each ant aims to build a feasible solution minimizing the cost expressed by the formula (1). The solution provided by each ant is a matrix with two rows and a number of columns equal to the number of exams. The first row represents exams; the second represents the assigned timeslot to the corresponding exam as shown in figure1.

$$\begin{pmatrix} 1 & 2 & 3 & 4 & . & . & . & . & N-1 & N \\ 12 & 1 & 6 & T & . & . & . & . & 1 & 10 \end{pmatrix}$$

Figure1. *Representation of the solution matrix*

The proposed approach consists of initially scheduling exams sequentially, beginning with the exams which are considered to be most “difficult” for scheduling. Thus, the set of exams presented in the first row of the matrix (figure1) are sorted using two strategies; first, the sort is commenced with the exams with largest number of conflicts (LD). We note that the number of conflicts for exam *x* is defined as the number of other exams which one or more students in common with exam *x*. Second, if two exams have the same number of conflicts, then an exam with largest student enrolment has the priority to be sorted (LE).

We note that the choice of scheduling the most “difficult” exams (in term of number of conflicts and number of enrollments) aims to avoid the situation in which some exams cannot be assigned to any time period without violating hard constraints.

3.1.2. Graph representation

Each ant is an agent who, in order to build a feasible solution minimizing the cost, operates on a bipartite graph G = (U, V, L), where U and V are the two sets of vertices and L is the set of edges.

The first set of the vertices U is the set of the examinations made in a predefined order according to (LE) and (LD) strategies. The second set V of the vertices is the set of the timeslots. The set L is the set of edges connecting each vertex in U to one in V, such that L = {(i, t) / i ∈ U, t ∈ V}. Figure2 shows the adopted graph representation.

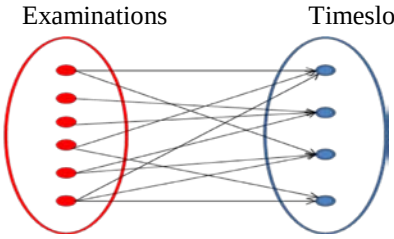

Figure2. *Structure of the graph associated to the proposed approach*

All ants are initially placed on the first exam. Each ant has to assign to the current exam an adequate timeslot, according to the movement strategy presented in the following section, such that the edges are not bi-directional. Thus, after choosing a suitable timeslot for the current exam, the ant takes one's place on the next exam. We note that each element of the set of examinations must be visited once only, unlike to the elements of the timeslots set which can be visited more than once because each timeslot can be assigned to several exams not having common students. The tour of each ant is finished once all the exams are planned, and the process is repeated.

3.1.3. The strategy of ant's movement

The basic ingredient of ACO is the use of a probabilistic solution construction mechanism based on stigmergy. Initially, all the ants are placed on the first examination according to the defined schedule in section 3.1.2. The ant υ must choose the best timeslot *h* to move to, taking into account the conflict constraint in the sense that if the current exam *i* has common students with another exam to which a timeslot *t* is already assigned, then it is necessary to avoid the assignment of this timeslot to the current exam.

The probability P_{ih}^{υ} of selecting a timeslot *h* for the current examination *i*, depends on both the amount of pheromone and the heuristic information present on the edge (i, h).

$$P_{ih}^{\upsilon}(t) = \begin{cases} \dfrac{(\tau_{ih}(t))^{\alpha}(\eta_{ih}^{\upsilon}(t))^{\beta}}{\sum_{l \in J_i^{\upsilon}} (\tau_{il}(t))^{\alpha}(\eta_{il}^{\upsilon}(t))^{\beta}} & \text{if } h \in J_i^{\upsilon} \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

Where J_i^{υ} is the set of timeslot candidates for ant υ being on the vertex i.

The parameters α and β define the relative importance of the pheromone information $\tau_{ih}(t)$ on edge (i, h) and the heuristic information $\eta_{ih}^{\upsilon}(t)$ on edge (i, h):

According to the objective function (1), this expression $\eta_{ih}^{\upsilon}(t)$ is an estimation of the partial solution constructed until the current step and which will guide the ants' search with problem specific information.

$$\eta_{ih}^{\upsilon}(t) = \frac{1}{1 + \sum_{(j,k)\in B^{\upsilon}(t)} E_{ij}\, D(h,k)} \tag{6}$$

Where: E_{ij} is the number of students that have to take both exams *i* and *j*.

And $B^{\upsilon}(t)$ is the set of all couples (examination, timeslot) visited by the ant υ before choosing a timeslot for the current examination *i* in iteration *t*.

D(h, k) is the penalty term defined as follows:

$$D(h,k) = \begin{cases} 2^{5-|h-k|} & \text{if } 1 \le |h-k| \le 5 \\ 0 & \text{otherwise} \end{cases}$$

h and *k* are two different timeslots of the planning horizon.

Thus, the probability P_{ih}^{υ} is a compromise between the heuristic information and the pheromone information in each iteration *t*.

3.1.4. The pheromone update

The pheromone update rule defines the way in which good solutions are reinforced in the pheromone trail by adding a quantity $\Delta\tau$ which is high on the best solution arcs as is shown by the formula (7). Thus, after a complete tour of each ant (when the ant has visited all exams assigning to each one a suitable timeslot), the pheromone on each edge (i, h) connecting the exam i to the timeslot h, is updated by a certain amount of pheromone:

$$\Delta\tau_{ih}(t) = \sum_{\upsilon=1}^{K} \frac{m_{ih}^{\upsilon}(t)}{K} \tag{7}$$

Where K is the total number of the ants.

Where

$$m_{ih}^{\upsilon}(t) = \begin{cases} 1 & \text{if } \upsilon \text{ choose the timeslot h for the exam i} \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

The algorithm would not be complete without the pheromone evaporation process, which is applied to avoid the unlimited accumulation of pheromone and premature convergence toward a suboptimal solution region. The rule for updating trails is:

$$\tau_{ih}(t+1) = \rho\, \tau_{ih}(t) + \Delta\tau_{ih}(t) \quad (9)$$

Where ρ is the pheromone evaporation rate.

3.2. Complete local search with memory

Local search techniques can be classified into those that store attributes of the solutions they visit during their execution, and those that do not. The former type includes heuristics like tabu search while the latter type includes heuristics like simulated annealing. In a wide variety of applications, tabu search and its hybrids have proved superior to memoryless heuristics. In this paper, we use Complete Local search with Memory (CLM) [35], a new neighbourhood search approach that makes effective use of memory structures in a way that is similar to the idea of "Tabu search with structured moves" [36]. This recent heuristic uses memory in an explicit manner, i.e. all explored solutions are recorded in order to avoid the exploration solutions that have been already visited. In addition, the heuristic described here has a backtracking mechanism that allows it to search solution spaces more thoroughly.

The term memory in CLM refers to storage space set aside specifically for storing solutions generated by the approach. The CLM approach keeps track of the solutions visited and prevents their neighbourhoods from being searched again at later stages during its execution. CLM is used to maintain three lists of solutions. The first one, called LIVE, stores solutions that are available to the approach for future consideration called *exploration*. A second list, called DEAD, contains solutions that were in LIVE at some stage, but have already been explored. The third list, called NEWGEN is a temporary store for new solutions being generated by the approach during the current iteration.

In this work, the best solutions generated at each iteration by the ACO are used as starting points for CLM, putting them in LIVE. The two sets DEAD and NEWGEN are initially empty. A threshold is generated, initially using the solutions obtained by the ACO. Thus at each iteration of the ACO, the best value of the objective function (1) is taken from among all those obtained. After the generation of the neighbours, this value is updated if a neighbour has a better evaluation. Then CLM performs a number of iterations until certain stopping conditions are met. During each of these iterations, it picks at most K solutions from LIVE. Each one of these is explored, i.e. it is transferred from LIVE to DEAD and all their neighbours, according to a specific neighbourhood definition, are generated. Several neighbourhood structures have been defined for the examination timetabling problem. The two most used structures are:

– Exchange neighbourhood exams: This type of neighbourhood is done by moving an examination into another available timeslot. Usually, even for problems whose size is limited, this type of neighbourhood does not allow a fast convergence since the moving

in the solution space is low. In addition, the previous assignments and timetable conflicts can also make the movement of a single examination impossible.

– Exchange neighbourhood timeslots: all exams of two randomly selected timeslots are exchanged. This method allows exchange timeslots while respecting the timetable constraints.

In this work, the second structure is used because it allows a permutation of the timeslots while respecting the conflict constraint unlike to the first structure which requires a checking of the conflict constraint and then a correction of the introduced neighbour.

From all generated neighbours, those whose objective values are lower than the threshold value are checked for membership in LIVE, DEAD, and NEWGEN. The neighbour is put into NEWGEN if it does not exist in any set. Clearly, if a neighbour is member of one of these three sets, it can be discarded. If it is in LIVE, then it will be explored in a future iteration. If it is in DEAD, then it has already been explored. Finally, if it is in NEWGEN, then it has already been generated in the current iteration. When all neighbours of all K solutions picked have been considered, then the solutions in NEWGEN are transferred to LIVE and the iteration has been completed.

There are a number of parameters and decisions that influence CLM:

– The initial solution(s).

– K, the number of solutions to be picked from LIVE at each iteration.

– The neighbourhood definition: The type of neighbourhood is defined by the way moves of made in an examination timetabling vicinity to introduce another called neighbour.

– The threshold value: This parameter controls the way in which CLM accepts non-improving moves in the sense that a solution with evaluation higher (resp. lower) than the threshold value is not accepted in the case of the minimization problem (resp. the maximization problem).

– The stopping criterion: Clearly, the algorithm must stop when LIVE is empty. When this condition is satisfied, the heuristic has explored all solutions that can be reached from the initial solution. Consequently this stopping rule determines the longest execution times, and returns the best solution possible. However, additional conditions can be imposed to lead to early termination of the search, such as terminating after a predefined number of iterations have been reached, or terminating if no improvement is recorded after a number of iterations.

– The order in which the solutions are stored in and picked from LIVE: In [35], three possibilities are suggested: first, choose the solution in LIVE with the best objective value. Second, choose the solution that allows the best improvement on exploration, this condition takes the search along a path which appears to bring about the best results at

that stage. Third, choose randomly any solution. In this work, the last condition, i.e. *random choice*, seems to be the fastest rule to apply in most circumstances, since it does not require sorting LIVE on any iteration. However the results obtained by applying such a rule are unpredictable.

The pseudocode of the CLM heuristic is provided in Figure 3. Note that it is a template heuristic. Apart from the choice of neighborhood structures, one has to decide on the memory size, the stopping rule, the criteria for picking solutions from LIVE, and the values of k and τ.

```
Input: Problem instance I, initial solution s0, k, τ .
Output: A solution to I.
    LIVE ← s0 ;
    DEAD ← Ø ;
     While stopping rule is not satisfied
            NEWGEN ← Ø;
            Chosen ← 0;
            While chosen < k and LIVE ≠Ø;
                    Choose a solution s from LIVE;
                    Chosen ← chosen + 1;
                    Transfer s from LIVE to DEAD;
                    Generate neighbors of s with objectives better than a threshold τ ;
                    For each neighbor n_s of s
                            If n_s is not in LIVE, DEAD or NEWGEN
                                If sufficient memory is available
                                    Add n_s to NEWGEN;
                                Else
                                    Transfer all nodes from NEWGEN to LIVE;
                                    Goto postproc;
          For each s in NEWGEN
              Transfer s from NEWGEN to LIVE;
Postproc: for each solution s in LIVE
            Remove n from LIVE;
            Obtain a locally optimal solution n_l from n using local search;
            Add n_l to DEAD;

Return the best solution in DEAD;
```

Figure3. *Pseudocode of a complete local search with memory heuristic*

Figure 4 describes the proposed approach of ant colony algorithm coupled with a local search with memory approach. Noting that *cost(s)* is calculated according to the objective function (1) and *best** is the best solution obtained by the ACO.

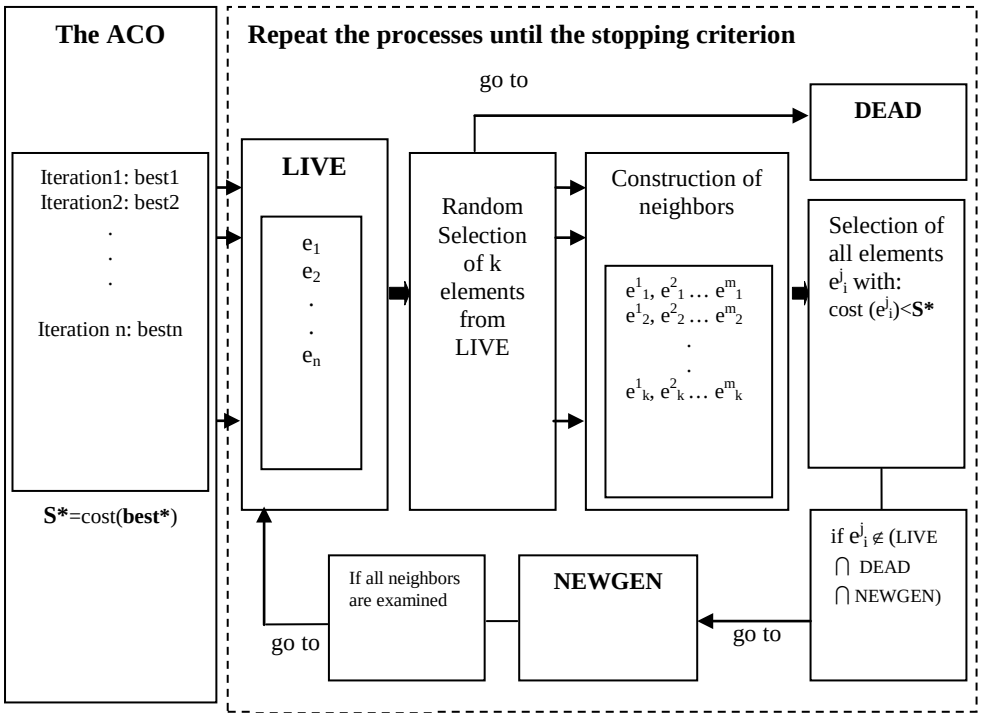

Figure4. *Illustration of the hybrid ACO*

4. Experimental results

This section aims firstly to test the proposed approach of resolution on a set of Carter's real-world data, secondly to apply the approach to a real-world exam timetabling problem face by the Faculty of Sciences and Techniques of Fez (FSTF).

4.1. Tests on Carter's Benchmarks

In 1996, Carter, Laporte and Lee in [19] introduced a set of 13 real-world exam timetabling problems from three Canadian highs schools, five Canadian universities, one American university, one British university and one university in Saudi Arabia. Over the

years they were widely employed as datasets in exam timetabling research. The benchmarks can be downloaded from: ftp://ftp.mie.utoronto.ca/pub/carter/testprob/.

Each instance is characterised by five values:

– The number of examinations

– The number of students: is the number of students in each instance.

– The number of enrolment: To calculate the number of enrollment in each instance, a vector E was defined where each element E_i is the number of students enrolled in exam i. The number of enrollment is the total sum of students enrolled in each exam:

$$\sum_{i=1}^{\text{Number of exams}} E_i$$

– The number of timeslots

– The density of conflict: To indicate the density of the conflicting exams in each of the instances, a conflict matrix C was defined where each element c_{ij}=1 if exam i conflicts with exam j (have common students), or c_{ij}=0 otherwise. The density of conflict represents the ratio between the number of elements of value '1' and the total number of elements in the conflict matrix. We note that the maximum density of conflict is 1.0.

Two variants of objectives were defined in the original dataset:

– Minimize the number of timeslots needed for the problem (corresponding to *Toronto a* in Table 1)

– Minimize the average cost per student (corresponding to *Toronto b* in Table 1).

For the first objective, the aim is to find feasible timetables of the shortest length. For the second objective, an evaluation function was defined to calculate the cost of the timetables generated, the aim is to space out the conflicting exams within a limited number of timeslots.

In the literature, two versions of the data were circulated and were tested by different approaches. The characteristics of the first version that has appeared more in the literature and which is used in this paper are listed in Table 2.

Data	Number of examinations	Number of students	Number of enrollments	Number of timeslots	Density of conflict
car-s-91	682	16925	56877	35	0.13
car-f-92	543	18419	55522	32	0.14
ear-f-83	190	1125	8109	24	0.27
hec-s-92	81	2823	10632	18	0.42
kfu-s-93	461	5349	25113	20	0.06
lse-f-91	381	2726	10918	18	0.06
rye-s-93	486	11483	45051	23	0.07
sta-f-83	139	611	5751	13	0.14
tre-s-92	261	4360	14901	23	0.06
uta-s-92	622	21266	58973	35	0.13
ute-s-92	184	2749	11793	10	0.08
yor-f-83	181	941	6034	21	0.29

Table2. *Characteristics of the Torontob Benchmark Datasets*

4.2. Adjustment of the parameters

Several test runs were carried out in order to determine the required parameters appropriately.

Different parameters of the ACO approach are fixed as follows:

– Number of iterations : 1000

– Number of ants : 20

– Initial pheromone amount $\tau_0 = 0.5$

– Evaporation factor $\rho = 0.4$

– Relative importance of pheromone factor $\alpha = 2$

– Relative importance of heuristic information factor $\beta = 3.5$

Different parameters of the CLM heuristic are fixed as follows:

– The number of iteration : 100

– The size of NEWGEN : 200

– The size of LIVE : 250

– The size of DEAD : 300

4.3. Computational results

According to the objective function presented in section (1.2) by the equation (1), Table3 presents the best results obtained over 40 runs, using ACO alone and then using ACO with the CLM approach.

Approach	car-s-91	car-f-92	ear-f-83	hec-s-92	kfu-s-93	lse-f-91	rye-s-93	sta-f-83	tre-s-92	uta-s-92	ute-s-92	Yor-f-83
ACO	8.1	6.6	50.1	13.1	22.7	17.7	17.4	164.2	11.5	5.1	35.3	53.2
hybrid ACO	6.9	5.9	42.4	11.0	17.3	14.9	14.0	155.7	9.9	4.5	32.0	44.1

Table3. *Results obtained using the ACO, then using the hybrid ACO*

We can observe from Table3, that results given by the hybridization of the ACO with the CLM approach are better than those given by ACO alone, for all instances. In fact, the ants guide the search process into promising regions of the solution space where the complete local search with memory approach can find good solutions.

The following section aims to compare our results with those produced by the state-of-the-art of examination timetabling problem in term of solution quality. Table4 represents different results obtained by several authors. The last row of the table contains the best results obtained by our approach. The best results are presented in bold, and "-" means that the corresponding problem is not tested or a feasible solution cannot be obtained in the literature. We have not listed computational times for the following reasons: Firstly, comparisons across very different platforms over the years are impossible. Secondly, examination timetabling is a problem which is almost always tackled weeks or months before the timetable will be used.

Data	Carter 1996	Carami a 2001	Gasper o 2001	Gasper o 2002	Merlot 2003	Burke 2004	Burke 2004	Yang 2005	Abdul ah 2006	Burke 2006	Our approa ch
car-s-91	7.1	6.6	6.2	5.7	5.1	5.0	4.8	4.5	5.2	4.6	6.9
car-f-92	6.2	6.0	5.2	-	4.3	4.3	4.2	3.9	4.4	4.0	5.9
ear-f-83	36.4	29.3	45.7	39.4	35.1	36.2	35.4	33.7	34.9	32.8	42.4
hec-s-92	10.8	9.2	12.4	10.9	10.6	11.6	10.8	10.8	10.3	10.0	11.0
kfu-s-93	14.0	13.8	18.0	-	13.5	15.0	13.4	13.8	13.5	13.0	17.3
lse-f-91	10.5	9.6	15.5	12.6	10.5	11.0	10.4	10.3	10.2	10.0	14.9
rye-s-93	7.3	6.8	-	-	-	-	8.9	8.5	8.7	-	14.0
sta-f-83	161.5	158.2	160.8	157.4	157.3	161.9	159.1	158.3	159.2	159.9	155.7
tre-s-92	9.6	9.4	10.0	-	8.1	8.4	8.3	7.9	8.4	7.9	9.9
uta-s-92	3.5	3.5	4.2	41.1	3.5	3.4	3.4	3.1	3.6	3.2	4.5
ute-s-92	25.8	24.4	27.8	-	25.1	27.4	25.7	25.3	26.0	24.8	32.0
yor-f-83	41.7	36.2	41.0	39.7	37.4	40.8	36.7	36.3	36.2	37.2	44.1

Table4. *Results of the Toronto b Dataset*

Comparing our results with those presented in the literature, it seems that our approach is superior to some of other approaches. So, for the sta-f-83 benchmark, we find the best solution. For seven benchmarks, our approach obtained greater or equal values compared to some of those presented in the literature. For the rye-s-93 benchmark instance, our algorithm finds a feasible solution, unlike 50% of approaches of literature which are not tested or feasible solutions cannot be obtained.

4.4. Application of the resolution approach

These encouraging results led us to address a real-world exam timetabling problem at the Faculty of Sciences and Techniques of Fez (FSTF). This instance is relatively less complicated compared to Carter's instances in terms of the conflict density, as shown in Table5.

In order to give a feasible planning for the FSTF, it was essential to include an additional constraint which takes into account the total capacity of the establishment. This constraint requires that the total capacity, at each timeslot, is not exceeded. Adapting the same notations used above, this constraint can be expressed as follows:

$$\sum_{i=1}^{N} E_{ii} x_{ih} \leq C \qquad \forall h = 1 \ldots T \tag{10}$$

Where

C is the total capacity of the establishment.

E_{ii} is the number of students that are enrolled in exam i.

$$x_{ih} = \begin{cases} 1 & \text{if the exam i is affected to the timeslot h} \\ 0 & \text{otherwise} \end{cases}$$

Data	Number of examinations	Number of students	Number of enrollments	Number of timeslots	Density of conflict	Total Capacity
FSTF-09	198	1876	7691	12	0.026	660

Table5. *The FSTF dataset*

Our problem is particularly characterized by only four timeslots per day, in contrast with the Toronto instances which are characterized by 6 timeslots per day. According to the penalty term presented by formula (4) for the Toronto instances with 6 timeslots per day, the penalty term for the FSTF instance with 4 timeslots per day, is presented by the following formula (11), which aims to separate two consecutive exams for each student, at the very least, with two timeslots. So, if a student has two consecutive exams then a penalty value equal to 4 is assigned. Two exams with one empty timeslot between them will be assigned a penalty value of 2. Two empty timeslots correspond to a penalty of 1.

$$D(h,k) = \begin{cases} 2^{3-|h-k|} & \text{if } 1 \leq |h-k| \leq 3 \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

h and k are two different timeslots of the planning horizon.

Table6 shows the best planning obtained by our approach with a cost equal to 2.76. The first column of table 6 represents the set of the timeslots (T_1, …, T_{12}). Each row of the table presents the set of examination indices associated with the relevant timeslot.

T1	3	188	39	28	11	156	17	192	184	177	173	141	118	93	85	54	49	47	132	129	122			
T2	2	197	41	34	14	181	152	126	123	107	97	92	86	72	71									
T3	1	162	42	29	23	193	179	174	171	169	160	139	135	95	88	83	78	68	53	48	44	131	130	
T4	116	113																						
T5	35	15	189	157	43	142	24	180	168	149	137	125	109	104	98	90	87	80	73	70	66			
T6	7	194	164	32	12	36	145	20	22	182	176	170	159	151	147	110	100	75	69	60	58	50		
T7	117	114																						
T8	4	196	40	31	8	155	25	21	120	190	185	166	158	134	124	111	106	101	96	94	89	79	65	63
T9	0	187	163	13	16	119	27	175	167	153	146	143	138	136	84	64	61	56	51	45	178			
T10	112	115																						
T11	5	195	38	33	9	165	144	26	18	183	172	127	108	102	82	76	74	62	59	55	133	121	128	
T12	6	186	154	37	30	10	19	191	161	150	148	140	105	103	99	91	81	77	67	57	52	46		

Table6. *Planning of the FSTF examinations using the hybrid ACO*

This planning allows to the most of the students to have a maximum separation between two consecutive exams. For example, the students enrolled at the exams 154, 155, 156 and 157, have a large time between two consecutive exams as mentioned in Table7.

Exams	154	155	156	157
Timeslots	T12	T8	T1	T5

Table7. *Example of a subset of exams*

5. Conclusion

In this paper, we have shown a hybrid Ant Colony Algorithm with the Complete Local Search with Memory approach. The proposed algorithm is able to solve large real world exam timetabling problems and to generate results which are compared with high performance algorithms from the literature. We conclude the paper by presenting a real-world examination timetabling problem faced by The Faculty of Sciences and Techniques of Fez.

A self-evident extension would be to adapt the resolution approach proposed to resolve the other variants of the examination timetabling problem.

6. References

[1] Abdullah S., Ahmadi S., Burke E.K., Dror M. *« Investigating Ahuja-Orlins large neighbourhood search for examination timetabling* ». Accepted by *OR Spectrum*, 2006.

[2] Ahuja R.K., Orlin J.B., Sharma D. « *Multi-exchange neighborhood search algorithm for capacitated minimum spanning tree problem* ». *Mathematical Programming*, 91. 71-97, 2001.

[3] Appleby J. S., Blake D. V., Newman E. A. « *Techniques for Producing School Timetables on a Computer and their Application to other Scheduling Problems* ». *The Computer Journal* 3. 237-245, 1961.

[4] Asmuni H., Burke E.K., Garibaldi J., McCollum B. « *Fuzzy multiple ordering criteria for examination timetabling* ». In E.K. Burke and M. Trick (eds). (2005). Selected papers from the *5th International Conference on the Practice and Theory of Automated Timetabling*. *Springer Lecture Notes in Computer Science*, 3616. 334-353, 2004.

[5] Boizumault P., Delon Y., Peridy L. « *Constraint logic programming for examination timetabling* ». *Journal of Logic Programming*, 26(2). 217- 233, 1996.

[6] Brelaz D. « *New methods to color the vertices of a graph* ». Communication of the *ACM*, 22(4). 251-256, 1979.

[7] Broder S. « *Final examination scheduling* ». Communications of the *ACM*, 7. 494-498, 1964.

[8] Burke E.K., Eckersley A.J., McCollum B., Petrovic S., Qu R. « *Hybrid variable neighbourhood approaches to university exam timetabling* ». Technical Report *NOTTCS-TR-2006-2*, School of CSiT, University of Nottingham, 2006.

[9] Burke E.K., Bykov Y., Newall J.P., Petrovic S. « *A time-predefined local search approach to exam timetabling problems* ». *IIE Transactions*, 36(6). 509-528, 2004.

[10] Burke E.K., Bykov Y., Petrovic S. « *A multicriteria approach to examination timetabling* ». In E.K. Burke and W. Erben (eds). (2001). *Practice and Theory of Automated Timetabling*, selected papers from *the 3rd International Conference*. *Springer Lecture Notes in Computer Science*, 2079. 118-131, 2001.

[11] Burke E.K., Dror M., Petrovic S., Qu R. « *Hybrid graph heuristics in hyper-heuristics applied to exam timetabling problems* ». In B.L. Golden, S. Raghavan and E.A. Wasil (eds). The Next Wave in *Computing, Optimization, and Decision Technologies*. Springer, Maryland, 79-91, 2005.

[12] Burke E.K., Newall J.P. « *A multi-stage evolutionary algorithm for the timetable problem* ». *IEEE Transactions on Evolutionary Computation*, 3(1). 63-74, 1999.

[13] Burke E.K., Newall J.P. « *Enhancing timetable solutions with local search methods* ». In E.K. Burke and P. De Causmaecker (eds). (2003). *Practice and Theory of Automated Timetabling*: Selected Papers from the 4th International Conference. *Springer Lecture Notes in Computer Science*, 2740. 195-206, 2003.

[14] Burke E.K., Newall J.P. « *Solving examination timetabling problems through adaptation of heuristic orderings* ». *Annals of Operational Research*, 129. 107-134, 2004.

[15] Burke E.K., Newall J.P., Weare R.F. « *A memetic algorithm for university exam timetabling* ». In E.K. Burke and P. Ross (eds). (1996). *Practice and Theory of Automated Timetabling*: Selected Papers from the 1st International Conference. *Springer Lecture Notes in Computer Science*, 1153. 241-250, 1996.

[16] Caramia M., DellOlmo P., Italiano G.F. « *New algorithms for examination timetabling* ». In S. Naher and D. Wagner (eds). *Algorithm Engineering 4th International Workshop, Proceedings WAE* 2000. *Springer Lecture Notes in Computer Science*, 1982, 230-241, 2001.

[17] Carter M.W., Chinneck J.W., Laporte G. «*A general examination scheduling system*». *Interfaces*, 24. 109-120, 1994.

[18] Carter M.W., Johnson D.G. « *Extended clique initialisation in examination timetabling* ». *Journal of Operational Research Society*, 52. 538-544, 2001.

[19] Carter M.W., Laporte G., Lee S.Y. « *Examination timetabling: Algorithmic strategies and applications* ». *Journal of Operational Research Society*, 47(3). 373-383, 1996.

[20] Colorni A., Dorigo M., Maniezzo V. « *Distributed optimization by ant colonies* ». F. Varela, P. Bourgine . *Proc. Europ. Conf. Artificial life, Elsevier, Amsterdam*, 1991.

[21] Costa D., Hertz A. « *Ant can colour graphs* ». *Journal of Operational Research Society*, 48. 295-305, 1997.

[22] David P. « *A constraint-based approach for examination timetabling using local repair techniques* ». In E.K. Burke and M.W. Carter (eds). (1998). *Practice and Theory of Automated Timetabling.* Selected Papers from the 2nd International Conference. *Springer Lecture Notes in Computer Science*, 1408. 169-186.

[23] Di Gaspero L. « *Recolour, shake and kick: A recipe for the examination timetabling problem* ». In E.K. Burke and P. De Causmaecker (eds). (2002). Proceedings of the *4th International Conference on Practice and Theory of Automated Timetabling.* 21st-23rd August 2002. KaHo St.-Lieven, Gent, Belgium. 404-407, 2002.

[24] Di Gaspero L., Schaerf A. « *Tabu search techniques for examination timetabling* ». In E.K. Burke and W. Erben (eds). (2001). *Practice and Theory of Automated Timetabling*: Selected Papers from the 3rd International Conference. *Springer Lecture Notes in Computer Science*, 2079. 104-117, 2001.

[25] Dorigo M. « *Optimization, learning and natural algorithms* ». Doctoral dissertation, politecnico di Milano, Italy, 1992.

[26] Dorigo M., Maniezzo V., Colorni A. « *Ant system : optimization by a colony of cooperating agents* ». *IEEE transaction on systems, Man and Cybernetics*. 29-41, 1996.

[27] Dorigo M., Stützle T. « *The ant colony optimisation metaheuristic: Algorithms, applications and advances* ». In F. Glover and G. Kochenberger (eds), *Handbook of Metaheuristics*, Vol. 57 of *International Series in Operations Research and Management Science*, Kluwer Academic Publishers, Boston, MA, 251–285, 2002.

[28] Dorigo M., Stützle T. « *Ant Colony Optimization* », MIT Press, 2004.

[29] Dowsland K.A., Thompson J. « *Ant colony optimization for the examination scheduling problem* ». *Journal of Operational Research Society*, 56. 426-438, 2005.

[30] Dueck G. « *New optimization heuristics: the great deluge and the recordtorecord travel* ». *Journal of Computational Physics*, 104. 86-92, 1993.

[31] Duong T.A., Lam K.H. « *Combining constraint programming and simulated annealing on university exam timetabling* ». In Proceedings of the *2nd International Conference in Computer Sciences, Research, Innovation & Vision for the Future (RIVF2004)*, Hanoi, Vietnam, February 2-5, 2004, 205-210, 2004.

[32] Erben W. « *A grouping genetic algorithm for graph colouring and exam timetabling* ». In E.K. Burke and W. Erben (eds). (2001). *Practice and Theory of Automated Timetabling*: Selected Papers from the 3rd International Conference. *Springer Lecture Notes in Computer Science*, 2079. 132-156, 2001.

[33] Feo T. A. and Resende M. G. C. « *Greedy randomized adaptive search procedures* ». *Journal of Global Optimization* 6. 109–133, 1995.

[34] Garey M., Johnson D. « *Computers and intractability a guide to the theory of NP-completeness* ». W.H. Freeman, 1977.

[35] Ghosh D, Sierksma G. « *Complete local search with memory* ». *Journal of Heuristics*, 8. 571–84, 2002.

[36] Glover F. « *Tabu search—part II* ». *ORSA Journal on Computing*, 2:4-32, 1990.

[37] Hentenryck P.V. « *The OPL Optimization Programming Language* » , 1999.

[38] Hussin N.M.,Kendall G. « *A tabu search hyper-heuristic approach to the examination timetabling problem at the MARA university of technology* ». In E.K. Burke and M. Trick (eds). (2005). Selected Papers from the *5th International*

Conference on the Practice and Theory of Automated Timetabling. Springer Lecture Notes in Computer Science, 3616. 199-218, 2004.

[39] Merlot L.T.G., Boland N., Hughes B.D., Stuckey P.J. « *A hybrid algorithm for the examination timetabling problem* ». In E.K. Burke and P. De Causmaecker (eds). (2003). *Practice and Theory of Automated Timetabling*. Selected Papers from the 4th International Conference. *Springer Lecture Notes in Computer Science*, 2740. 207-231, 2003.

[40] Naji Azimi Z. « *Comparison of metaheuristic algorithms for examination timetabling problem* ». *Applied Mathematics and Computation*, 16(1-2). 337-354, 2004.

[41] Paquete L., Stutzle T. « *Empirical analysis of tabu search for the lexicographic optimization of the examination timetabling problem* ». In E.K. Burke and P. De Causmaecker (eds). (2002). Proceedings of the *4th International Conference on Practice and Theory of Automated Timetabling*. 21st-23rd August 2002. KaHo St.-Lieven, Gent, Belgium. 413-420, 2002.

[42] Petrovic S., Bykov Y. « *A multiobjective optimisation technique for exam timetabling based on trajectories* ». In E.K. Burke and P. De Causmaecker (eds). (2003). *Practice and Theory of Automated Timetabling*, selected papers from the 4th International Conference. *Springer Lecture Notes in Computer Science*, 2740. 179-192, 2003.

[43] Petrovic S.,Yang Y. « *A Novel similarity measure for heuristic selection in examination timetabling* ». In E.K. Burke and M. Trick (eds). (2005). Selected Papers from *the 5th International Conference on the Practice and Theory of Automated Timetabling.* Springer Lecture Notes in *Computer Science*, 3616. 377-396, 2004.

[44] Qu R., Burke E. K., McCollum B., Merlot L.T.G., Lee S. Y. « *A Survey of Search Methodologies and Automated Approaches for Examination Timetabling* ». *Computer Science Technical Report*, NOTTCS-TR-2006-4, 2006.

[45] Ross P., Marin-Blazquez J.G., Hart E. « *Hyper-heuristics applied to class and exam timetabling problems* ». In Proceedings of the 2004 *Congress on Evolutionary Computation*, 1691-1698, 2004.

[46] Sheibani K. « *An evolutionary approach for the examination timetabling problems* ». In E.K. Burke and P. De Causmaecker (eds). (2002). *Proceedings of the 4th International Conference on Practice and Theory of Automated Timetabling*. 21st-23rd August 2002. KaHo St.-Lieven, Gent, Belgium. 387-396, 2002.

[47] Terashima-Marin H., Ross P., Valenzuela-Rendon M. « *Evolution of constraint satisfaction strategies in examination timetabling* ». In Proceedings of the *Genetic and Evolutionary Conference*, Orlando, Florida, 635-642, 1999.

[48] Terashima-Marin H., Ross P., Valenzuela-Rendon M. « *Clique-based crossover for solving the timetabling problem with Gas* ». In: Schoenauer M. et al (eds).

Proceedings of the 1999 *Congress on Evolutionary Computation*. Washington: *IEEE Press*. 1200-1206, 1999.

[49] Thompson J., Dowsland K. « *A robust simulated annealing based examination timetabling system* ». *Computer & Operations Research*, 25. 637-648, 1998.

[50] White G.M., Xie B.S. « *Examination timetables and tabu search with longer-term memory* ». In: E.K. Burke and W. Erben (eds). (2001). *Practice and Theory of Automated Timetabling*: Selected Papers from the 3rd International Conference. *Springer Lecture Notes in Computer Science*, 2079. 85-103, 2001.

[51] White G.M., Xie B.S., and Zonjic S. « *Using tabu search with longer term memory and relaxation to create examination timetables* ». *European Journal of Operational Research*, 153(16). 80-91, 2004.

[52] Wood D.C. « *A system for computing university examination timetables* ». *The Computer Journal*, 11(1). 41-47, 1968.

[53] Zeleny M. « *A concept of compromise solutions and method of displaced ideal* ». *Computers & Operations Researh*, 1(4). 479-496, 1974.

7. Biography

– Rachida Abounacer is a PhD student of the Laboratory of Modeling and Scientific Calcul and CERENE laboratory, she is a member of Operational Research and Computer group at the Faculty of Sciences and Techniques of Fez, Morocco. She works on scheduling problems and metaheuristics.

– Jaouad Boukachour is an Associate Professor of Computer Sciences at the University of Le Havre, France. His research interests include: Scheduling Problems, Operational Research, and Supply Chain Management. He has supervised a number of PhD researchers in areas such as logistics and scheduling aircraft landings. Currently, he is supervising six PhD students working on traceability, modelling road traffic, job shop scheduling, scheduling aircraft landings and vehicle routing. He has published more than 30 referred research papers. Within the French CPER 2006 (State-Region Project Contract), he was responsible for Modelling Optimisation and Simulation of physical and information flows in an industrial logistics project. Currently, he heads two projects about tracking container shipments, funded by French National Research Agency (ANR) and CPER 2008.

– Btissam Dkhissi is a PhD student of the Laboratory of Modeling and Scientific Calcul at the Faculty of Sciences and Techniques of Fez, Morocco, she is a member of Operational Research and Computer group. She works on staff scheduling, timetabling problems, vehicle routing and metaheuristics methods.

– Ahmed El Hilali Alaoui is a PhD of Operational Research at the Faculty of Sciences and Techniques of Fez, Morocco. His research interests include: Scheduling Problems and Operational Research. He is responsible for the operational research and computer group, and he is supervising 10 PhD students working on job shop scheduling, scheduling aircraft landings, vehicle routing and optimization algorithms. He is member of the La Société Marocaine de Recherche Opérationnelle (SOMARO).