Omega 39 (2011) 598–607

Contents lists available at ScienceDirect

Omega

journal homepage: www.elsevier.com/locate/omega

An integrated hybrid approach to the examination timetabling problem

Hamza Turabieh *, Salwani Abdullah

Data Mining and Optimisation Research Group, Center for Artificial Intelligence Technology, Universiti Kebangsaan Malaysia, 43600 Bangi, Selangor, Malaysia

A R T I C L E I N F O

Article history:
Received 3 March 2010
Accepted 31 December 2010
Processed by B. Lev
Available online 21 January 2011

Keywords:
Great deluge approach
Examination timetabling
Heuristics
Metaheuristics

A B S T R A C T

This paper is derived from an interest in the development of automated approaches to tackle examination timetabling problems effectively. We propose a hybrid approach that incorporates effective heuristic operators within the great deluge algorithm. The latter was chosen because of consistently good performances being reported within the examination timetabling research domain. The additional heuristic procedures further enhance the overall effectiveness of this integrated hybrid approach. These procedures are drawn from methodologies that have appeared in the literature under term the "electromagnetic-like mechanism". The aim is to move sample points towards a high quality solution while avoiding local optima by utilising a calculated force value. This value, which is calculated dynamically, is treated as a decay rate in determining the level within the great deluge algorithm. To evaluate the proposed algorithm, we carry out experimental work on two types of examination timetabling datasets. All the related results and analysis obtained illustrate that this hybrid approach is effective when compared with existing approaches in the literature.

1. Introduction

The domain of automated timetabling has seen significant research activity from across the multi-disciplinary spectrum of Operational Research and Artificial Intelligence. The general domain covers educational timetabling [1–6], nurse rostering [7], sports timetabling [8,9], transportation timetabling [10–12], machine scheduling [13,14], and flowshop scheduling [15–17]. This paper is concerned with university examination timetabling, which is a particularly important administrative activity in modern academic institutions. This research area has attracted particular attention over recent years. Examination timetabling has been overviewed in [18–21].

These surveys and overviews demonstrate that there have been many diverse techniques that have been applied to the problem. Examples of early heuristics techniques based on graph colouring include [22–24]. Constraint-based techniques have been applied successfully to examination timetabling problems, particularly, when hybridising with other methods [25,26].

In recent years, interest in metaheuristics for exam timetabling has increased significantly. Such methods include Tabu Search [27–29], Simulated Annealing [30,67,68] and Evolutionary Algorithms [31]. Other methodologies applied to examination timetabling problems include Ant Algorithms [32], Great deluge [33–36], GRASP [37,38], case-based reasoning [39,40], Bio-inspired methods [41,69], very large neighbourhood search [42–44], and hybrid Variable Neighbourhood Search [45]. Hybridisation of different methodologies plays a key role in many of the methodologies in the literature. A particularly effective example of a hybrid method was presented by Caramia et al. [46].

Hyper-heuristics have also been applied in recent years, to examination timetabling [47–49]. Here, one of the motivations is to raise the level of generality of the search methodologies. Examination timetabling methods have been reviewed and discussed in considerable details in [18–21].

The remainder of the paper is organised as follows: Section 2 briefly explains the examination timetabling problem. Section 3 gives a specification of the examination timetabling problem with the formulation, which includes related soft and hard constraints. Section 4 depicts the benchmark datasets for examination timetabling problems. Section 5 presents the proposed algorithm. Experimental results are discussed in Section 6, followed by some brief concluding comments in Section 7.

2. Examination timetabling problems

The examination timetabling problem can be thought of as an assignment problem where a set of examinations must all be scheduled within a certain time period subject to a set of constraints [18,19]. The variety of constraints that are addressed in the scientific literature has increased in recent years and there is considerable scope for this to continue. The reason for this is that more and more complex issues from real-world problems are being incorporated into the models in the scientific literature as the issue of closing the gap between theory and practice in automated timetabling becomes a major aspect of the

* Corresponding author.
E-mail addresses: hamza@ftsm.ukm.my, turabieh@gmail.com (H. Turabieh), salwani@ftsm.ukm.my (S. Abdullah).

0305-0483/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.omega.2010.12.005

international scientific agenda [50]. Indeed, the 2nd International Timetabling Competition (ITC2007) [51] attempted to pursue this agenda by introducing three tracks (one on exam and two on course timetabling) where the problems incorporated more real-world constraints. The common elements in examination timetabling problems normally include:

- a set of exams to schedule;
- a given fixed or variable length of timeslots; and
- a set of student exam enrolments that define the clash between exams.

There are two categories of constraints, which are often known as hard and soft constraints. Hard constraints are sometimes referred to as "conflict" constraints and are defined to be those which must be satisfied and which cannot be violated under any circumstances. If these hard constraints are satisfied, then the solution (timetable) can be called feasible. On the other hand, soft constraints (sometimes referred to as "side constraints") are considered to be less essential and the violation of these constraints (although acceptable) is undesirable. The violation of these soft constraints has an associated penalty by which the quality of a feasible solution is often measured.

The survey conducted by Burke et al. [52] found that the constraints vary widely from institution to institution in British Universities. Examples of hard constraints encountered in practical examination timetabling are:

- No student should be required to sit two examinations at once.
- There must be sufficient resources e.g. rooms and seating capacity at any given timeslot, respectively.
- Certain examinations are subject to precedence constraints, for example exam A must take place before/after exam B.
- Certain examinations must take place in a specific room.

The common accepted hard constraints for examination timetabling problems are (i) no student is asked to sit two examinations at the same time and (ii) there must be enough seating capacity.

Some soft constraints as reported by Burke et al. [52], are:

- Students should not be scheduled to sit for two consecutive examinations.
- Student should not be scheduled to sit for more than one examination in a day.
- Each student's examination should be spread as evenly as possible over the exam period.

One of the most prominent soft constraint in the exam timetabling literature is to spread conflicting exams as much as possible throughout the examination period because this features in the widely studied Toronto benchmarks [24].

3. Problem description

The problem descriptions employed in this paper are divided into two parts:

- *Problem I*: This problem is an uncapacitated examination timetabling problem where a room capacity requirement is not considered while constructing a timetable. This problem has been introduced by Carter et al. [24] along with a set of 13 real-world instances from a variety of educational institutions, that have been accepted benchmark datasets for over a decade (for more details see [18,19]).
- *Problem II*: This represents an exam timetabling model that incorporates a significant number of real-world constraints. This formulation was introduced as part of the 2nd International Timetabling Competition (ITC2007). Problem II is considered to be more challenging than Problem I because of the high number of hard/soft constraints compared to Problem I.

3.1. Problem I

The input for this problem can be stated as follows. It is adapted from the description presented in Burke et al. [7]:

- E_i is a collection of N examinations ($i=1,\ldots,N$).
- T is the number of timeslots.
- $C=(c_{ij})_{N\times N}$ is the conflict matrix where each record, denoted by c_{ij} ($i, j\in\{1,\ldots,N\}$), represent the number of students taking the exams i and j.
- M is the number of students.
- t_k ($1\le t_k\le T$) specifies the assigned timeslots for exam k ($k\in\{1,\ldots,N\}$).

The hard and soft constraints considered are "no students should sit two examinations simultaneously" and "spread exams as evenly as possible throughout the timetable", respectively.

An objective function is formulated to represent the soft constraint as

$$\min \frac{\sum_{i=1}^{N-1} F(i)}{M} \tag{1}$$

where

$$F(i)=\sum_{j=i+1}^{N} c_{ij}\,proximity(t_i,t_j) \tag{2}$$

$$proximity(t_i,t_j)=\begin{cases} 2^5/2^{|t_i-t_j|} & \text{if} \quad 1\le |t_i-t_j|\le 5 \\ 0 & \text{otherwise}\end{cases} \tag{3}$$

subject to

$$\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} c_{ij}\lambda(t_i,t_j)=0$$

where

$$\lambda(t_i,t_j)=\begin{cases} 1 & \text{if} \quad t_i=t_j \\ 0 & \text{otherwise}\end{cases} \tag{4}$$

Eq. (2) represents the cost for an exam i which is given by the proximity value multiplied by the number of students in conflict. A proximity value between two exams is presented in Eq. (3). Eq. (4) presents the hard constraint, i.e. it represents a clash-free requirement where no student can sit for two or more exams at the same time.

3.2. Problem II

The benchmark instances considered for problem are taken from the third track of the 2nd International Timetabling Competition (ITC 2007) [51] (http://www.cs.qub.ac.uk/itc2007/index.htm). Eight cases have been introduced. A set of hard and soft constraints are drawn from real-world problems and are listed in Tables 1 and 2.

A feasible timetable is one in which all examinations have been assigned to a period and room and there is no violation of the hard constraints. The objective function is to minimise the violation of the soft constraints as given by [51]

$$\min \sum_{s \in S} (w^{2R} C_S^{2R} + w^{2D} C_S^{2D} + w^{PS} C_S^{PS}) + w^{NMD} C_S^{2NMD} + w^{FL} C^{FL} + w^{p} C^{p} + w^{R} C^{R} \quad (5)$$

Each dataset has its own weight as shown in Table 3 [51].

4. An integrated approach

4.1. The encoding scheme

We employed a direct encoding scheme to represent a candidate solution. Fig. 1(a) shows an example of the encoding scheme for Problem I. We can see that 4 exams are assigned in timeslot t_1, i.e. e_1, e_5, e_7, e_{10}, whilst e_{21}, e_{18} and e_{25} are allocated in t_2. We used a similar encoding scheme for Problem II. The only difference is that a room parameter is taken into account (as shown in Fig. 1(b)). It can be observed that exam e_1 is scheduled in timeslot t_1 at room r_1, whilst e_{13} and e_4 are scheduled in timeslot t_T at room r_3.

4.2. Neighbourhood search

In the methodology presented in this paper, two types of neighbourhood search operation have been employed. They can be outlined as follows:

Nbs_1: select two exams at random and swap timeslots
Nbs_2: choose a single exam at random and move to a new random feasible timeslot

The neighbourhood is selected randomly and applied within the great deluge algorithm.

Table 1
Hard constraints.

Hard constraints	Explanation
H1	There cannot be any students sitting for more than one exam at the same time
H2	The total number of students assigned to each room cannot exceed the room capacity
H3	The length of exams assigned to each timeslot should not violate the timeslot length
H4	Some sequences of exams have to be respected, e.g. *Exam_A* must be schedule after *Exam_B*
H5	Room related hard constraints must be satisfied, e.g. *Exam_A* must be scheduled in Room 80

4.3. Great deluge algorithm (GD)

The great deluge algorithm is a local search procedure that was introduced by Dueck [53]. It was applied to examination timetabling by Burke et al. [33] and to university course timetabling by Burke et al. [54]. It requires two parameters represented by the requirements of the user: the computational time that is available and an estimate of the quality of the required solution. The method accepts a worse solution if the quality value is less than or equal to some given upper boundary value, called the "*level*". In this approach, the "*level*" is initially set to be the objective function value of the initial solution. During its run, the "*level*" is iteratively lowered by a dynamic force decay rate calculated from the EM algorithm. The pseudo code for the great deluge algorithm is shown in Fig. 2.

4.4. Electromagnetic-like mechanism and particle swarm optimisation

We now employ the Great geluge algorithm described above within an approach that we denote as EM, because it draws upon methods in the literature that have been called the electromagnetic-like mechanism [55]. This EM method shares the same

Table 3
The associate weight of ITC2007 collection of examination datasets.

Data sets	w^{2D}	w^{2R}	w^{PS}	w^{NMD}	w^{FL}	w^{P}	w^{R}
Exam_1	5	7	5	10	100	30	5
Exam_2	5	15	1	25	250	30	5
Exam_3	10	15	4	20	200	20	10
Exam_4	5	9	2	10	50	10	5
Exam_5	15	40	5	0	250	30	10
Exam_6	5	20	20	25	25	30	15
Exam_7	5	25	10	15	250	30	10
Exam_8	0	150	15	25	250	30	5

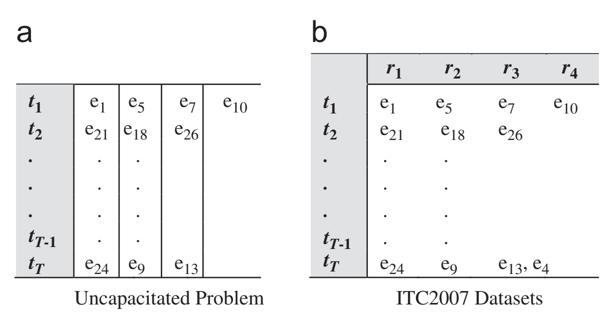

Fig. 1. Encoding scheme: (a) uncapacitated problem and (b) ITC2007's datasets.

Table 2
Soft constraints.

Soft constraints	Mathematical symbol	Explanation
S1	C_S^{2R}	*Two exams in a row*: Minimise the number of consecutive exams in a row for a student
S2	C_S^{2D}	*Two exams in a day*: Student should not be assigned to sit more than two exams in a day. Of course, this constraint only becomes important when there are more than two examination periods in the same day
S3	C_S^{PS}	*Periods spread*: All students should have a fair distribution of exams over their timetable
S4	C_S^{2NMD}	*Mixed durations*: The numbers of exams with different durations that are scheduled into the same room has to be minimised as much as possible
S5	C^{FL}	*Larger examinations appearing later in the timetable*: Minimise the number of examinations of large class size that appear later in the examination timetable (to facilitate the assessment process)
S6	C^{P}	*Period penalty*: Some periods have an associated penalty, minimise the number of exams scheduled in penalised periods.
S7	C^{R}	*Room penalty*: Some rooms have an associated penalty, minimise the number of exams scheduled in penalised rooms

```
Great Deluge Algorithm
Set estimated quality of every solution, EstimatedQuality = f(Sol_i) – F_i, where
(i=1,…,population size) and F is a total force taken from Figure 3;
Calculate force decay rate, β = EstimatedQuality/NumOfIte_GD;
Set iteration_GD ← 0;
for (iteration_GD< NumOfIte_GD)
    Define a randomly selected neighbourhood structure (Nbs_1 or Nbs_2) on Sol
    to generate a new solution called Sol*;
    Calculate f(Sol*);
    if (f(Sol*) < f(Sol_best))
        Sol ← Sol*;
        Sol_best ← Sol*;
    else
        if (f(Sol*)≤ level)
            Sol ← Sol*;
        endif
    endif
    level = level - β;
    Increase iteration_GD by 1;
end for
```

Fig. 2. Great deluge algorithm.

concepts with the Particle Swarm Optimisation (PSO) method [56] where the goal is that particles in the population move towards the most promising area of the search space. Within EM, the position changed based on the total force that effects the particle in the search space. Kennedy and Eberhart [57] reported that there are five general concepts that affect the performance of swarm intelligence, i.e.

- *Proximity*: The population should represent a simple search space in order to minimise the computation time.
- *Quality*: The population should represent some quality factors in terms of the solutions.
- *Diversity*: The population should not be driven along excessively narrow channels.
- *Stability*: The population should not change its mode of behaviour every time the environment changes.
- *Adaptability*: The population must change its behaviour when it is appropriate to do so.

In this paper, we hybridise EM with the great deluge algorithm. We believe that this satisfies all five of the concepts of swarm intelligence outlined above. We now present a detailed description of the hybridisation of EM with the great deluge algorithm.

The first mention of the electromagnetic-like mechanism in the literature was in [55]. Since then, the approach has been applied to a wide variety of domains such as flowshop scheduling [58], vehicle routing [59], array pattern optimisation [60], and numerical optimisation [61,62].

The idea behind EM is to consider each sample particle to be an electrical charge. It is a metaheuristic algorithm with an attraction–repulsion mechanism that aims to move solutions toward high quality solutions. This mechanism can avoid the sample particle (a timetable in our case) from being trapped in a local optimum. Each candidate solution has a charge (related to the objective function value) which determines the magnitude of attraction or repulsion of the solution over the sample population. The better the solution quality of the particle, the higher the charge that the particle has. The electrostatic force between two particle charges is directly proportional to the magnitude of each charge and inversely proportional to the square of the distance between the charges [55,63]. The charge (q_i) of timetable i is as follows:

$$q_i = \exp\left(-T\frac{f(x_i)-f(x_{best})}{\sum_{k=1}^{m}(f(x_k)-f(x_{best}))}\right) \tag{6}$$

where q_i is the charge for timetable i; $f(x_i)$ is the penalty of timetable i; $f(x_k)$ is the penalty of timetable k; $f(x_{best_i})$ is the penalty of best timetable; m is the population size; and T is the number of timeslots.

A better solution encourages other particles to converge to attractive regions, while a bad solution discourages particles away from region. These particles move along with the total force. The following formulation is the total force of particle i:

$$F_i = \sum_{j \neq i}^{m}\left\{\begin{array}{ll} (f(x_j)-f(x_i))\frac{q_i q_j}{\|f(x_j)-f(x_i)\|^2} & \text{if} \quad f(x_j) < f(x_j) \\ (f(x_i)-f(x_j))\frac{q_i q_j}{\|f(x_j)-f(x_i)\|^2} & \text{if} \quad f(x_j) \geq f(x_j) \end{array}\right\}, \quad \forall i \quad \text{(particle)} \tag{7}$$

4.5. A hybrid metaheuristic approach for exam timetabling

We utilise the graph colouring heuristics (i.e. the hybridisation of least saturation degree, largest degree first and largest enrolment first heuristics) in constructing the initial solutions. The incorporation between the EM algorithm and the Great deluge methodology is presented in Fig. 3 as an improvement approach. Step 1 is an initialisation stage where the quality of the initial solution, best solution, level and the maximum number of iterations (which represent the duration of the search process) are initialised. In Step 2, the total force, F, is calculated for each solution. Then the great deluge algorithm is invoked. In the great deluge algorithm, the total force is used to calculate the estimated quality of the new solution and the decreasing rate (in this paper, we refer to it as the dynamic force decay rate).

Fig. 4 illustrates an example of three solutions (represented as Sol_1, Sol_2 and Sol_3). The charge (q) and force (F) for each solution must be calculated first. Then the great deluge algorithm is employed with the aim of reducing the penalty cost based on the F value. For example, with solution Sol_1, the F value is 0.235104, thus the estimated quality of the new solution (denoted as Estimated Quality in Fig. 3) is 3.3059 (i.e. Penalty$-F$=3.541–0.235104). Note that the “Penalty” value is referring to the objective function for each problem considered in this paper. The great deluge algorithm will try to reduce the penalty cost

```
Electromagnetic-like Mechanism and Great Deluge Algorithm
  Step 1: Initialization
        Set initial solution as Sol;
        Calculate the initial penalty cost, f(Sol);
        Set best solution, Sol_best ← Sol;
        Set initial level: level ← f(Sol);
        Set total number of iterations, NumOfIte;
        Set number of iterations for Great Deluge, NumOfIte_GD;
        Set iteration ← 0;
  Step 2: Evaluation
     do while (iteration < NumOfIte)
        Calculate total force, F, for each timetable based on EM;
        Apply a dynamic force decay rate great deluge algorithm (as in Figure 2);
        Increase iteration by 1;
     end do
```

Fig. 3. A hybridisation of the electromagnetic-like mechanism and the Great deluge algorithm.

	Penalty	Charge	Force
Sol_1	3.541	0.0941702	0.235104
Sol_2	3.531	0.4346310	0.194763
Sol_3	3.553	0.2457278	0.032197

Great Deluge →

	Penalty	Charge	Force
Sol_1	3.3763	0.1060713	0.107349
Sol_2	3.3362	0.0025137	0.186730
Sol_3	3.5208	0.2903313	0.081937

Fig. 4. Illustrative example for uta92 I with three solutions.

by at least the (Penalty-F) value (if possible) while searching for a new solution. However, in this experiment, the great deluge algorithm is able to reduce the penalty value more than the (Penalty-F) value, while in the case of Sol_3, the great deluge algorithm cannot reduce the solution to the estimated quality; however, this will lead to a change in the exploration status of solution movements. For example, after employing the great deluge algorithm, the quality of solution Sol_1 is 3.3003 (which is less than the Estimated Quality, 3.3059). Note that this example is taken from our experiment on the uta92 I uncapacitated dataset (see Section 5).

5. Benchmark dataset specification

5.1. Problem I: uncapacitated datasets

The specification of the uncapacitated examination timetabling problems is shown in Table 4, which can be found at http://www.cs.nott.ac.uk/~rxq/data.htm. These benchmarks were first introduced in [24]. The notation we employ is that used on [18,19] to distinguish between different versions of the datasets that have circulated over the years.

5.2. Problem II: competition datasets

The competition (ITC 2007) had 12 datasets (of which only 8 are generally available). These datasets are collected from different universities. The data is available at: http://www.cs.qub.ac.uk/itc2007/Login/SecretPage.php. The competition is described in detail in [51]. Table 5 shows the specification of competition datasets considered in this paper.

6. Experimental results

The proposed algorithm was implemented using Matlab and simulations were performed on an Intel Pentium 4 2.33 GHz computer. Table 6 shows the parameters for the proposed algorithm which were determined after some preliminary experiments. They are similar to the parameter used by Birbil [55] and Birbil and Fang [63]. We ran the experiments for 100,000 iterations for the main hybrid algorithm, and 1000 iterations for great deluge algorithm

Table 4
Uncapacitated datasets examination datasets features.

Datasets	Number of timeslots	Number of examinations	Number of students	Conflict density
car92	32	543	18,419	0.14
car91	35	682	16,925	0.13
ear83 I	24	190	1125	0.27
hec92 I	18	81	2823	0.42
kfu93	20	461	5349	0.06
lse91	18	381	2726	0.06
pur93 I	43	2419	30,032	0.03
rye92	23	486	11,483	0.07
sta83 I	13	139	611	0.14
tre92	23	261	4360	0.18
uta92 I	35	622	21,267	0.13
ute92	10	184	2750	0.08
yor83 I	21	181	941	0.29

Table 5
ITC 2007 examination datasets features.

Datasets	D1	D2	D3	D4	D5	D6	D7	Conflict density
Exam_1	7891	7833	607	54	7	12	0	5.05
Exam_2	12,743	12,484	870	40	49	12	2	1.17
Exam_3	16,439	16,365	934	36	48	170	15	2.62
Exam_4	5045	4421	273	21	1	40	0	15.0
Exam_5	9253	8719	1018	42	3	27	0	0.87
Exam_6	7909	7909	242	16	8	23	0	6.16
Exam_7	14,676	13,795	1096	80	15	28	0	1.93
Exam_8	7718	7718	598	80	8	20	1	4.55

Note: **D1**: Number of students. **D2**: Number of actual students in the datasets. **D3**: Number of exams. **D4**: Number of timeslots. **D5**: Number of rooms. **D6**: Period hard constraints. D7: Room hard constraints.

Table 6
Parameter setting for the proposed algorithm.

Parameter	**Value**
Generation number	100,000
NumOfIte_GD	1000
Population size	50

(NumOfIte_GD) with population size 50. These parameter values have been selected based on preliminary experiments.

6.1. *Problem I: the uncapacitated problem and the Toronto benchmarks*

The first series of experiments carried out in this section attempts to space out students' examinations throughout the examination period. Termination is based on the number of generations, and is initially set at 100,000 iterations. We compare our results with the best known results in the literature [18,19] on the thirteen timetabling instances shown in Table 4.

The best results and the average out of 5 runs are shown in Table 7 with different random seeds. From Table 7, we can see that our approach is able to produce the best known result (denoted in bold) in the literature on two of the thirteen problems (tieing with [45] on kfu93). These results have been generated by an iterated algorithm with novel improvement factors [46]; heuristic selection by employing a novel similarity measure [40]; and a variable neighbourhood approach which employed a genetic algorithm to select a subset of the neighbourhood [45].

We are particularly interested to compare our results with the other results in the literature that employed a hybrid combination of approaches. The algorithms compared in the table are described as follows:

- M1: Merlot et al. [25]—constraint programming was used as an initialisation procedure for simulated annealing and hill climbing
- M2: Burke and Newall [34,35]—a great deluge algorithm with adaptive ordering as the initialisation
- M3: Côté et al. [31]—a hybrid multi-objective evolutionary algorithm
- M4: Abdullah et al. [42]—a large neighbourhood search approach hybridised with local search methods
- M5: Yang and Petrovic [40]—fuzzy sets were employed for hybridisation of great deluge and graph heuristics
- M6: Qu and Burke [49]—a hybridisation within a graph-based hyper-heuristic
- M7: Qu et al. [18,19]—a dynamic hybridisation of different graph colouring heuristics
- M8: Burke et al. [45]—a hybrid variable neighbourhood search with a genetic algorithm
- M9: Caramia et al. [46]—an iterated algorithm with novel improvement factors

Table 8 shows the comparison of results between these hybrid algorithms. Again, the best results out of 5 runs are shown in bold.

From Table 8, we can see that our hybrid algorithm produce two best results (denoted in bold). We are interested to compare our results with M2, that also employed a great deluge algorithm. Our hybrid algorithm outperforms M2 in all instances except for rye92 dataset (tieing on car92 dataset). We believe this is due to the capability of our hybrid approach to enhance the exploration process.

Fig. 5 shows box plots that illustrate the distribution of solution quality for the uncapacitated datasets. We can see that there is a close gap between the best, average and worse solution qualities which demonstrates that the approach is robust.

Figs. 6 and 7 show the trends of the quality of the best solutions found throughout the search process when the technique is applied to two datasets, i.e. sta83 I and uta92 I, respectively. They are very similar. The *x*-axis represents the number of iterations, while the *y*-axis represents the penalty cost. Each point represents the best solution found in each iteration. An analysis of the diagrams illustrates the trend of cost improvement, i.e. at the beginning of the search process, the relatively steep slope of the curves indicates a large decrease in the penalty cost (high quality improvement with respect to the solution). However, as the number of iterations increases, the penalty cost is gradually decreased until there is little improvement towards the end of the search process.

Fig. 8 shows the behaviour of our hybrid metaheuristic on the sta83 I dataset. The plotted line shows the correlation between the number of iterations and the overall solution quality. The algorithm offers more flexibility in accepting a worse solution at the beginning and middle of the search process. Towards the end of the search process, the worse solution is unlikely to be accepted. The relative improvement of the solution becomes small (but still there is an improvement).

Table 7
Comparison between best known results.

Dataset	Our method		Best known results	
	Best	Average		
car91	4.80	4.894	**4.50**	Yang and Petrovic [40]
car92	4.10	4.28	**3.9**	Burke et al. [45]
ear83 I	34.92	35.93	**29.3**	Caramia et al. [46]
hec92 I	10.73	10.827	**9.2**	Caramia et al. [46]
kfu93	**13.00**	13.53	**13.0**	Burke et al. [45]
lse91	10.01	10.21	**9.6**	Caramia et al. [46]
pur93 I	4.73	4.84	**3.7**	Caramia et al. [46]
rye92	9.65	9.74	**6.8**	Caramia et al. [46]
sta83 I	158.26	159.69	**156.9**	Burke et al. [45]
tre92	**7.88**	7.975	7.9	Burke et al. [45]
uta92 I	3.20	3.274	**3.14**	Yang and Petrovic [40]
ute92	26.11	27.00	**24.4**	Caramia et al. [46]
yor83 I	36.22	36.27	**34.9**	Burke et al. [45]

Table 8
Comparison results on other combination approaches.

Instance	Our approach	M1	M2	M3	M4	M5	M6	M7	M8	M9
car91	4.80	5.1	4.65	5.2	5.2	**4.50**	5.16	5.11	4.6	6.6
car92	4.10	4.3	4.10	4.2	4.4	3.93	4.16	4.32	**3.9**	6.0
ear83 I	34.92	35.1	37.05	34.2	34.9	33.7	35.86	35.56	32.8	**29.3**
hec92 I	10.73	10.6	11.54	10.2	10.3	10.83	11.94	11.62	10.0	**9.2**
kfu93	**13.00**	13.5	13.90	14.2	13.5	13.82	14.79	15.18	**13.0**	13.8
lse91	10.01	10.5	10.82	11.2	10.2	10.35	11.15	11.32	10.0	**9.6**
pur93 I	4.73	–	–	–	–	–	–	–	–	**3.7**
rye92	9.65	–	8.7	8.8	8.7	8.53			–	**6.8**
sta83 I	158.26	157.3	168.73	157.2	159.2	158.3	159	158.88	**156.9**	158.2
tre92	**7.88**	8.4	8.35	8.2	8.4	7.92	8.6	8.52	7.9	9.4
uta92 I	3.20	3.5	3.20	3.2	3.6	**3.14**	3.59	3.21	3.2	3.5
ute92	26.11	25.1	25.83	25.2	26.0	25.39	28.3	28	24.8	**24.4**
yor83 I	36.22	37.4	37.28	36.2	36.2	36.35	41.81	40.71	**34.9**	36.2

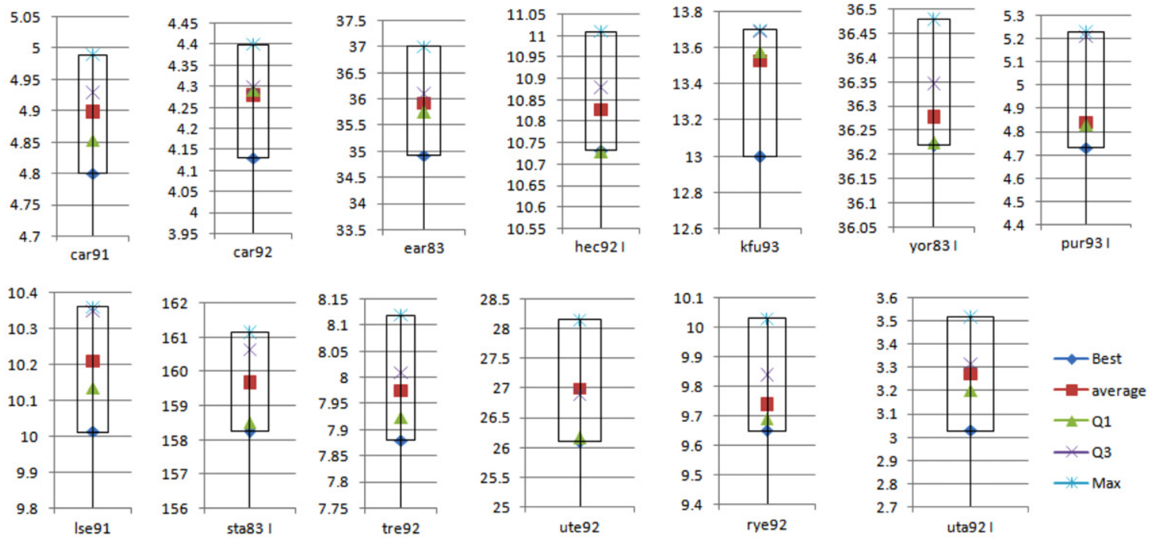

Fig. 5. Box plots of the penalty costs for all instances.

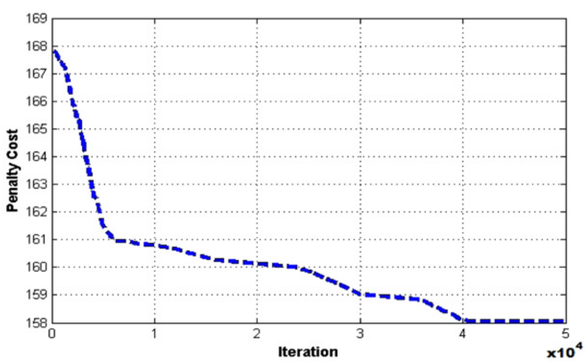

Fig. 6. sta83 I convergence.

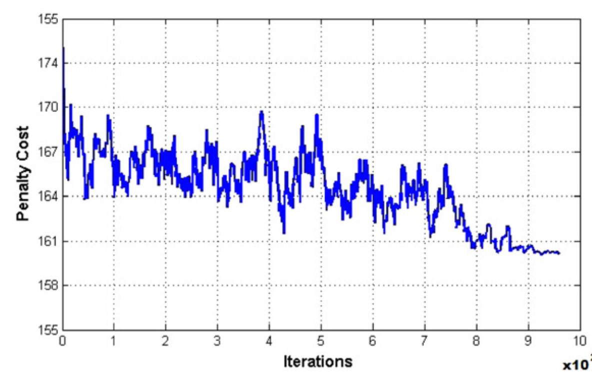

Fig. 8. The behaviour of hybrid metaheuristic algorithm on the sta83 I dataset.

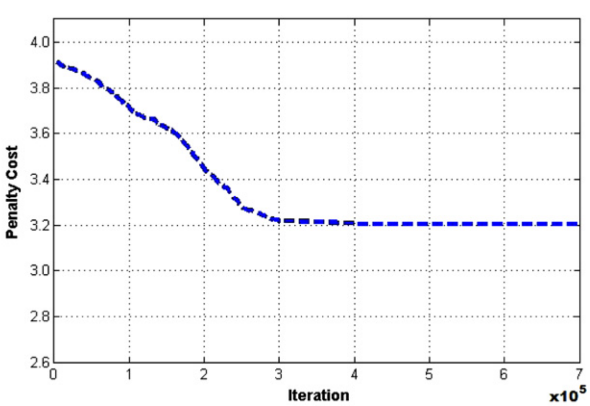

Fig. 7. uta92 I convergence.

6.2. *Problem II: the real-world competition problem (ITC2007)*

The second series of experiments addresses the problem of minimising the weighted sum of the soft constraints represented in Eq. (5). We used the same parameter settings as in Table 6. The only difference is that the termination criterion is set to 600 seconds (as set in the competition). The algorithm was run 11 times (as in the ITC2007 competition) on each dataset with different random seeds. Table 9 shows the best, median and worst solutions obtained together with upper and lower quartiles (denoted as Q1 and Q3).

Table 9
Best, median and worse solutions.

Datasets	Best	Q1	Median	Q3	Worst
Exam_1	4368	4464	4587	4654	5196
Exam_2	390	400	400	405	410
Exam_3	9830	10,152	10,830	10,942	11,557
Exam_4	17,251	17,494	19,125	20,015	20,647
Exam_5	3022	3220	3287	3448	3546
Exam_6	25,995	26,395	26,700	27,715	29,130
Exam_7	4067	4224	4263	4382	4573
Exam_8	7519	7740	7984	8146	11,007

Fig. 9 shows the box plots of the solution quality on the ITC2007 datasets. For most instances, the range between best, median and worse solutions is relatively wide except for the *Exam_1* and *Exam_8* datasets. We believe that this could be due to the limited execution time because at the beginning of the search process, the algorithm tends to carry out exploration by easily accepting worse solutions and then towards the end of the search process, convergence will happen. This is consistent with the results obtained on the uncapacitated examination timetabling datasets (see Section 6.1), when given extra execution time, the algorithm is able to produce a number of best known results. We also believe that the complexity of the problems is different

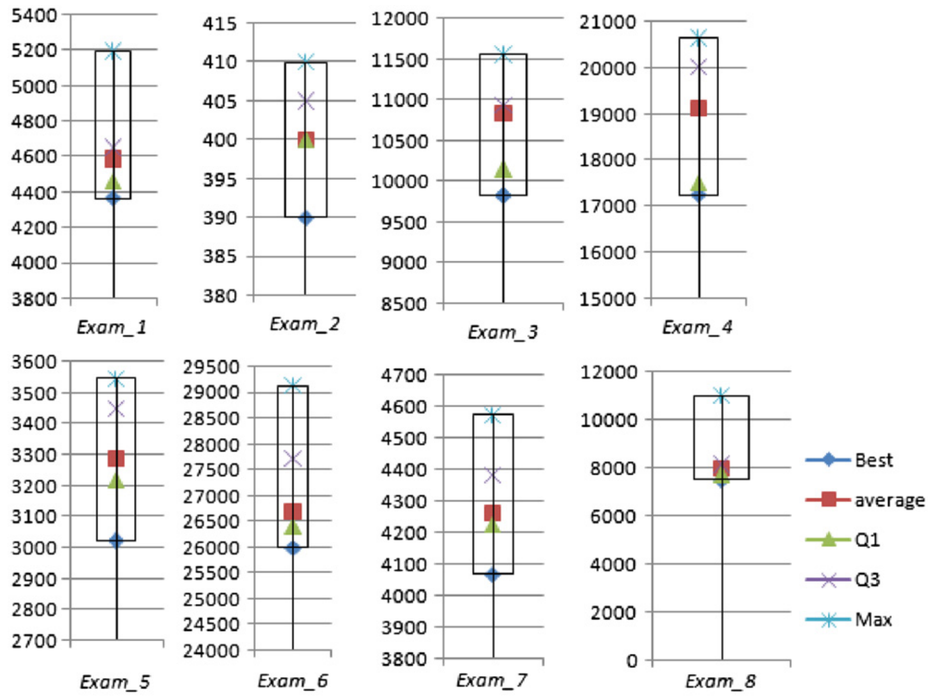

Fig. 9. Box plots of the solution quality for ITC2007 datasets.

Table 10
Results comparison.

Datasets	A1	A2	A3	A4	A5	A6	A7	A8	Our approach	
									Best	Avg.
Exam_1	4370	5905	8006	6670	12,035	4633	4775	4699	**4368**	4607.44
Exam_2	400	1008	3470	623	3074	405	**385**	**385**	390	409.94
Exam_3	10,049	13,862	18,622	–	15,917	9064	8996	**8500**	9830	10,574.57
Exam_4	18,141	18,674	22,559	–	23,582	15,663	16,204	**14,879**	17,251	19,798.56
Exam_5	2988	4139	4714	3847	6860	3042	2929	**2795**	3022	3246.889
Exam_6	26,950	27,640	29,155	27815	32,250	25,880	25,740	**25,410**	25,995	27,405.05
Exam_7	4213	6683	10,473	5420	17,666	4037	4087	**3884**	4067	4328.707
Exam_8	7861	10,521	14,317	–	16,184	7461	7777	**7440**	7519	7969.577

which makes the algorithm behave differently (in relation to the improvement achieved).

Table 10 presents our experimental results compared to other available results in the literature. The algorithms compared in the table are described as follows:

A1: Müller [26]—a three phase constraint-based solver
A2: Gogos et al. [38]—a GRASP-based process
A3: Atsuta et al. [29]—a general purpose solver, combining iterated local search and tabu search
A4: De Smet [64]—local search techniques incorporated within the drools solver, an open-source business rule management system
A5: Pillay [41]—an approach based on cell biology which mimicked cell behaviour
A6: McCollum et al. [36]—an extended great deluge algorithm
A7: Gogos et al. [65]—an improved multi-staged algorithmic process
A8: Gogos et al. [66]—a grid resources approach

The best results are presented in bold. The "–" indicates that the datasets are not tested. The listed results under A1–A5 are obtained from the ITC2007 competition where Müller [26] was the winner at that time. A6, A7 and A8 represent new results which have been published after the ITC2007 competition took place.

Our results are obtained after 11 runs. Note that McCollum et al. [36] applied 51 runs and Gogos et al. [65] ran his approach 100 times, Note that Gogos et al. [66] results are obtained without considering the execution time. It is clearly shown that our results are better than ITC2007s results (A1–A5) under the same number of trials and execution time. Our approach is able to obtain one best result and is competitive with the other current state-of-the-art approaches. In comparison to other participants of the ITC2007, our approach ranked 1st on *Exam_1* dataset and between 2nd and 3rd on the rest of the datasets.

Fig. 10 shows almost the same trend of best solutions found during the search as in Figs. 6 and 7. This shows that the algorithm works similarly on the different datasets despite the differences in the complexity of the datasets and the landscape of the search space.

Fig. 11 shows the performance of the proposed algorithm on the *Exam_1* dataset given extra time (i.e. 6 h). This graph demonstrates how the algorithm explores the search space. Again, the x-axis represents the iteration steps while the y-axis represents the penalty cost. The curve shows that the algorithm begins with an initial solution and improves the results throughout the search

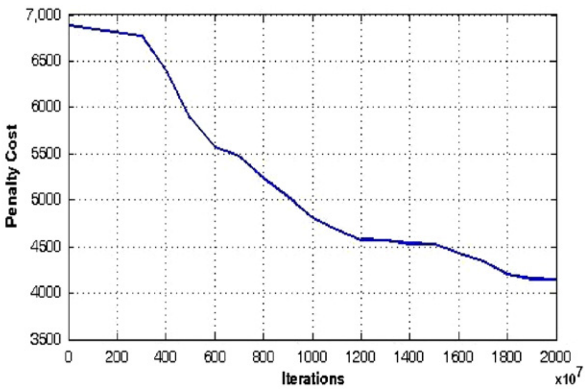

Fig. 10. *Exam_1* convergence.

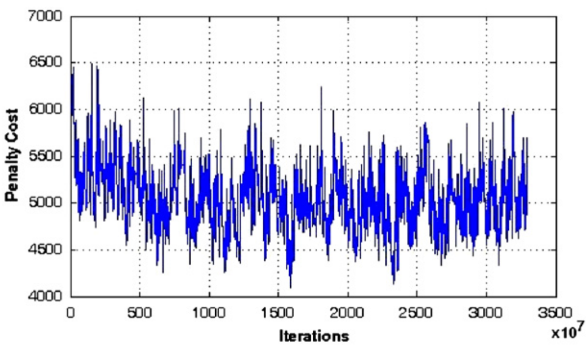

Fig. 11. The behaviour of the proposed algorithm on the *Exam_1* dataset given extra search time.

process. The graphs show fluctuations because the algorithm always accepts the best solution with the worse solution being accepted based on the level. Further analysis on the performance graph shows that prolonging the algorithm does improve the solution (note that the best solution obtained by prolonging the search for this dataset is 4321). We believe that by embedding a good exploration and convergence mechanisms (this is subject to future work); the algorithm will further improve the quality of solutions compared to others.

In general, the algorithm is able to produce some of the best known solutions (on both problems). We believe that one of its key strengths lies in its use of a "dynamic" decreasing rate (we say "dynamic" because the value is recalculated at every iteration) where the level will be decreased based on different values (note that the decreasing rate in a standard great deluge is a predefined constant). Also, another strength is due to the ability to the algorithm to explore different regions of the solution space. Our algorithm works on 50 different solutions at every iteration. Given extra processing time, we are confident that our algorithm is capable of finding better solutions regardless of the nature and complexity of the problems.

7. Conclusions

In this paper, we presented an examination timetabling search methodology that combines the principle of the electromagnetic-like mechanism with a great deluge algorithm. The performance of the approach is tested on two different datasets: the Toronto benchmark and ITC2007 competition datasets. We compare with a set of state-of-the-art approaches from the literature. This approach is simple yet effective, and produces two best results and is consistently good across the all the benchmark problems in comparison with other approaches studied in the literature. With the help of the dynamic decreasing rate that works on a set of different solutions, our approach is capable of finding better solutions for the examination timetabling problem. We are confident that a significant contribution to producing high quality solutions to the uncapacitated timetabling problems and the ITC2007 problems has been made.

We also believe that with the increasing complexity of examination timetabling problems in many educational institutions, the proposed approach can be easily adapted with new constraints. Additionally, research on multi-objective optimisation approaches may be a very promising direction to be explored with this algorithm. This will be the subject of our future work.

Acknowledgement

The authors are indebted to the anonymous referees for their comments and suggestions. They significantly improved the paper.

References

[1] Schaerf A. A survey of automated timetabling. Artificial Intelligence Review 1999;13(2):87–127.
[2] Lewis R. A survey of metaheuristic-based techniques for university timetabling problems. OR Spectrum 2008;30(1):167–90.
[3] Burke EK, Petrovic S. Recent research directions in automated timetabling. European Journal of Operational Research 2002;140(2):266–80.
[4] Petrovic S, Burke EK. University timetabling. In: Leung J, editor. Handbook of scheduling: algorithms, models, and performance analysis. Boca Raton: CRC Press; 2004. [chapter 45].
[5] Bardadym VA. Computer-aided school and university timetabling: the new wave. In: Burke EK, Ross P, editors. Lecture notes in computer science: vol. 1153. Practice and theory of automated timetabling, I: selected papers from the 1st international conference Berlin: Springer; 1996. p. 22–45.
[6] de Werra D. An introduction to timetabling. European Journal of Operational Research 1985;19:151–62.
[7] Burke EK, De Causmaecker P, Vanden Berghe G, Van Landeghem H. The state of the art of nurse rostering. Journal of Scheduling 2004;7(6):441–99.
[8] Easton K, Nemhauser G, Trick M. Sports scheduling. In: Leung J, editor. Handbook of scheduling: algorithms, models, and performance analysis. Boca Raton: CRC Press; 2004. [chapter 52].
[9] Drexl A, Knust S. Sports league scheduling: graph- and resource-based models. The International Journal of Management Science (Omega) 2007;35(5): 465–71.
[10] Kwan R. Bus and train driver scheduling. In: Leung J, editor. Handbook of scheduling: algorithms, models, and performance. Boca Ratom: CRC Press; 2004. [chapter 51].
[11] Caramia M, Guerriero F. A heuristic approach to long-haul freight transportation with multiple objective functions. The International Journal of Management Science (Omega) 2009;37(3):600–14.
[12] Adlakha V, Kowalski K, Lev B. A branching method for the fixed charge transportation problem. The International Journal of Management Science (Omega) 2010;38(5):393–7.
[13] Yang S-J, Yang D-L. Minimizing the makespan on single-machine scheduling with aging effect and variable maintenance activities. The International Journal of Management Science (Omega) 2010;38(6):528–33.
[14] Lee W-C, Wu C-C, Hsu P-H. A single-machine learning effect scheduling problem with release times. The International Journal of Management Science (Omega) 2010;38(1–2):3–11.
[15] Laha D, Sarin S. A heuristic to minimize total flow time in permutation flow shop. The International Journal of Management Science (Omega) 2009;37(3): 734–9.
[16] Li X, Wang Q, Wu C. Efficient composite heuristics for total flowtime minimization in permutation flow shops. The International Journal of Management Science (Omega) 2009;37(1):155–64.
[17] Ribas I, Companys R, Tort-Martorell X. An iterated greedy algorithm for the flowshop scheduling problem with blocking. The International Journal of Management Science (Omega) 2011;39(3):293–301.
[18] Qu R, Burke EK, McCollum B, Merlot L, Lee S. A survey of search methodologies and automated system development for examination timetabling. Journal of Scheduling 2009;12(1):55–89.
[19] Qu R, Burke EK, McCollum B. Adaptive automated construction of hybrid heuristics for exam timetabling and graph colouring problems. European Journal of Operational Research 2009;198(2):392–404.
[20] Carter MW, Laporte G. Recent developments in practical examination timetabling. In: Lecture notes in computer science: vol. 1153. Selected papers from the 1st international conference on practice and theory of automated timetabling. Berlin: Springer; 1996. p. 373–83.

[21] Carter MW. A survey of practical applications of examination timetabling algorithms. Operations Research Society of America 1986;34(2):193–202.
[22] Balakrishnan N. Examination scheduling: a computerized application. The International Journal of Management Science (Omega) 1991;19(1):37–41.
[23] Weitz, Lakshminarayanan. An empirical comparison of heuristic and graph theoretic methods for creating maximally diverse groups VLSI, design and exam scheduling. Omega 1997;25(4):473–82.
[24] Carter MW, Laporte G, Lee S. Examination timetabling: algorithmic strategies and applications. Journal of the Operational Research Society 1996;47(3):373–83.
[25] Merlot LTG, Boland N, Hughes BD, Stuckey PJ. A hybrid algorithm for the examination timetabling problem. In: Burke EK, De Causmaecker P, editors. Lecture notes in computer science: vol. 2740. Practice and theory of automated timetabling, IV: selected papers from the 4th international conference Berlin: Springer; 2003. p. 207–31.
[26] Müller T. ITC2007 solver description: a hybrid approach. Annals of Operation Research 2009;172(1):429–46.
[27] White GM, Xie BS, Zonjic S. Using tabu search with longer-term memory and relaxation to create examination timetables. European Journal of Operational Research 2004;153(16):80–91.
[28] Di Gaspero L, Schaerf A. Tabu search techniques for examination timetabling. In: Burke EK, Erben W, editors. Lecture notes in computer science: vol. 2079. Practice and theory of automated timetabling, III: selected papers from the 3rd international conference Berlin: Springer; 2001. p. 104–17.
[29] Atsuta M, Nonobe K, Ibaraki T. ITC2007 Track 1: an approach using general CSP solver. 〈www.cs.qub.ac.uk/itc2007〉, (2007).
[30] Duong TA, Lam KH. 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, 2–5 February 2004. p. 205–10.
[31] Côté P, Wong T, Sabouri R. Application of a hybrid multiobjective evolutionary algorithm to the uncapacitated exam proximity problem. In: Burke EK, Trick M, editors. Lecture notes in computer science: vol. 3616. Practice and theory of automated timetabling, V: selected papers from the 5th international conference Berlin: Springer; 2005. p. 151–68.
[32] Dowsland KA, Thompson J. Ant colony optimization for the examination scheduling problem. Journal of Operational Research Society 2005;56:426–38.
[33] Burke EK, Bykov Y, Newall JP, Petrovic S. A time predefined local search approach to exam timetabling problems. IIE Transactions 2004;36(6):509–28.
[34] Burke EK, Newall JP. Enhancing timetable solutions with local search methods. In: Burke EK, De Causmaecker P, editors. Lecture notes in computer science: vol. 2740. Practice and theory of automated timetabling, IV: selected papers from the 4th international conference Berlin: Springer; 2003. p. 195–206.
[35] Burke EK, Newall JP. Enhancing timetable solutions with local search methods, practice and theory of automated timetabling iv: selected revised papers from the 4th international conference on the practice and theory of automated timetabling (PATAT 2002), Gent, Belgium, 21–23 August 2002. In: Burke EK, Decausmaecker P, editors. Lecture Notes in Computer Science, vol. 2740. Berlin: Springer; 2003. p. 195–206.
[36] McCollum B, McMullan PJ, Parkes AJ, Burke EK, Abdullah S. An extended great deluge approach to the examination timetabling problem. In: MISTA09, The 4th multidisciplinary international conference on scheduling: theory and applications, Dublin, August, 2009. p. 424–34.
[37] Casey S, Thompson J. GRASPing the examination scheduling problem. In: Burke EK, De Causmaecker P, editors. Lecture notes in computer science: vol. 2740. Practice and theory of automated timetabling, IV: selected papers from the 4th international conference Berlin: Springer; 2003. p. 232–44.
[38] Gogos C, Alefragis P, Housos E. A multi-staged algorithmic process for the solution of the examination timetabling problem. In: Practice and theory of automated timetabling (PATAT 2008), Montreal, 2008, p. 19–22.
[39] Burke EK, Petrovic S, Qu R. Case-based heuristic selection for timetabling problems. Journal of Scheduling 2006;9:115–32.
[40] Yang Y, Petrovic S. A novel similarity measure for heuristic selection in examination timetabling. In: Burke EK, Trick M, editors. Lecture notes in computer science: vol. 3616. Practice and theory of automated timetabling, V: selected papers from the 5th international conference Berlin: Springer; 2005. p. 377–96.
[41] Pillay A. Developmental approach to the examination timetabling problem 〈www.cs.qub.ac.uk/itc2007〉, 2007.
[42] Abdullah S, Ahmadi S, Burke EK, Dror M. Investigating Ahuja-Orlin's large neighbourhood search approach for examination timetabling. OR Spectrum 2007;29(2):351–72.
[43] Abdullah S, Ahmadi S, Burke EK, Dror M, McCollum B. A tabu based large neighbourhood search methodology for the capacitated examination timetabling problem. Journal of Operational Research Society 2007;58:1494–502.
[44] Meyers C, Orlin JB. Very large-scale neighborhood search techniques in timetabling problems. In: Burke EK, Rudova H, editors. Practice and theory of automated timetabling, VI: selected papers from the 6th international conference, PATAT 2006, Lecture notes in computer science, vol. 3867. Berlin: Springer; 2007. p. 24–39.
[45] Burke EK, Eckersley AJ, McCollum B, Petrovic S, Qu R. Hybrid variable neighbourhood approaches to university exam timetabling. European Journal of Operation Research 2010;206(1):46–53.
[46] Caramia M, Dellolmo P, Italiano GF. Novel local search-based approaches to university examination timetabling. INFORMS Journal on Computing 2009;20(1):86–99.
[47] Bilgin B, Ozcan E, Korkmaz EE. An experimental study on hyperheuristics and exam timetabling. In: Burke EK, Rudova H, editors. Practice and theory of automated timetabling: selected papers from the 6th international conference. Lecture notes in computer science, vol. 3867. Berlin: Springer; 2007. p. 394–412.
[48] Burke EK, McCollum B, Meisels A, Petrovic S, Qu R. A graph based hyperheuristic for exam timetabling problems. European Journal of Operational Research 2007;176:177–92.
[49] Qu R, Burke EK. Hybridisations within a graph based hyper-heuristic framework for university timetabling problems. Journal of the Operational Research Society 2009;60:1273–85.
[50] McCollum B. A perspective on bridging the gap between theory and practice in university timetabling. In: Burke EK, Rudova H, editors. Lecture notes in computer science: vol. 3867. Practice and theory of automated timetabling, VI: selected papers from the 6th international conference Berlin: Springer; 2007. p. 3–23.
[51] McCollum B, Schaerf A, Paechter B, McMullan P, Lewis R, Parkes AJ, et al. Setting the research agenda in automated timetabling: the second international timetabling competition. INFORMS Journal on Computing 2010;22:120–30.
[52] Burke EK, Elliman DG, Ford PH, Weare RF. Examination timetabling in British universities: a survey. In: Burke EK, Ross P, editors. Lecture notes in computer science: vol. 1153. Practice and theory of automated timetabling, I: selected papers from the 1st international conference Berlin: Springer; 1996. p. 76–90.
[53] Dueck G. New optimization heuristics: the great deluge and the record-to-record travel. Journal of Computational Physics 1993;104:86–92.
[54] Burke EK, Bykov Y, Newall JP, Petrovic S. A Time-predefined approach to course timetabling. Yugoslav Journal of Operational Research (YUJOR) 2003;13(2):139–51.
[55] Birbil SI, Fang S-C. An electromagnetism-like mechanism for global optimization. Journal of Global Optimization 2003;25:263–82.
[56] Burke EK, Kendall G. In: Burke EK, Kendall G, editors. Search methodologies: introductory tutorials in optimization and decision support techniques. Springer; 2005. p. 5–18 [chapter 1].
[57] Kennedy J, Eberhart RC. Particle swarm optimization. In Proceedings of the IEEE international conerence on neural networks, Piscataway, NJ, 1995. p. 1942–48.
[58] Tavakkoli-Moghaddam R, Khalili M, Naderi B. Electromagnetism-like mechanism and simulated annealing algorithms for flowshop scheduling problems minimizing the total weighted tardiness and makespan. Knowledge-Based Systems 2009;23(2):77–85.
[59] Yurtkuran A, Emel E. A new hybrid electromagnetism-like Algorithm for capacitated vehicle routing problems. Expert Systems with Applications 2010;37(4):3427–33.
[60] Jhang Jhen-Yan, Lee Kun-Chou. Array pattern optimization using electromagnetism-like algorithm. AEU—International Journal of Electronics and Communications 2009:491–6.
[61] Tsou C-S, Kao C-H. An electromagnetism-like meta-heuristic for multi-objective optimization. In: Proceedings of the 2006 IEEE congress on evolutionary computation, 2006, p. 1172–8.
[62] Gol Alikhani M, Javadian N, Tavakkoli-Moghaddam R. A novel hybrid approach combining electromagnetism-like method with Solis and Wets local search for continuous optimization problems. Journal of Global Optimization 2009;44(2):227–34.
[63] Birbil SI. Stochastic global optimization techniques. PhD thesis, North Carolina State University, 2002.
[64] De Smet G. ITC 2007—Examination Track: 1, 〈www.cs.qub.ac.uk/itc2007〉, 2007.
[65] Gogos C, Alefragis P, Housos E. An improved multi-staged algorithmic process for the solution of the examination timetabling problem. Annals of Operations Research 2010, doi:10.1007/s10479-010-0712-3.
[66] Gogos C, Goulas G, Alefragis P, Housos E. Pursuit of better results for the examination timetabling problem using grid resources. In: Proceedings of the CI-Sched '09, IEEE symposium on computational intelligence in scheduling, 2009, p. 48–53.
[67] Thompson JM, Dowsland KA. Variants of simulated annealing for the examination timetabling problem. Ann Oper Res 1996;63(1):105–28.
[68] Thompson JM, Dowsland KA. A robust simulated annealing based examination timetabling system. Comput Oper Res 1998;25(7–8):637–48.
[69] Sabar NR, Ayob M, Kendall G. Solving examination timetabling problems using honey-bee mating optimization (ETP-HBMO). In: Proceedings of the 4th Multidisciplinary International Scheduling Conference: Theory andApplications (MISTA 2009), 10–12 Aug 2009, Dublin, Ireland, pp. 399–408.