A Novel Similarity Measure for Heuristic Selection in Examination Timetabling

Yong Yang and Sanja Petrovic

School of Computer Science and Information Technology,
The University of Nottingham, NG8 1BB, UK
{yxy, sxp}@cs.nott.ac.uk

Abstract. Metaheuristic approaches to examination timetabling problems are usually split into two phases: an initialisation phase in which a sequential graph colouring heuristic is employed to construct an initial solution and an improvement phase in which the initial solution is gradually improved. Different hybridisations of metaheuristics with sequential heuristics are known to lead to solutions of different quality. A Case Based Reasoning (CBR) methodology has been developed for selecting an appropriate sequential construction heuristic for hybridisation with the Great Deluge metaheuristic. In this paper we propose a new similarity measure between two timetabling problems that is based on fuzzy sets. The experiments were performed on a number of real-world benchmark problems and the results were also compared with other state-of-the-art methods. The results obtained show the effectiveness of the developed CBR system.

1 Introduction

Examination timetabling is an important and difficult task for educational institutions since it requires expensive human and computer resources and has to be solved several times every year. Timetabling can be defined to be the problem of allocating a set of examinations over a limited number of time periods subject to constraints in such a way as to generate no conflicts between any two examinations. For example, no student should be required to attend two examinations at the same time and no student should have two examinations on the same day.

The timetabling problem can be represented as an undirected weighted graph where vertices represent examinations, while edges represent conflicts between examinations (i.e. an edge connects examinations with common students) [24]. To both vertices and edges weights are assigned that correspond to the number of students enrolled in the examinations and the number of students enrolled in two examinations that are in conflict, respectively. For illustration purposes, a simple timetabling problem (with four examinations) is shown in Figure 1. For example, the weight of Math is 30 because 30 students are enrolled in this course. The edge connecting AI and PA1 is assigned weight 9 because there are nine students who are enrolled in both examinations. The timetabling problem is closely linked to the graph colouring problem [24], which is concerned with

E. Burke and M. Trick (Eds.): PATAT 2004, LNCS 3616, pp. 247–269, 2005.

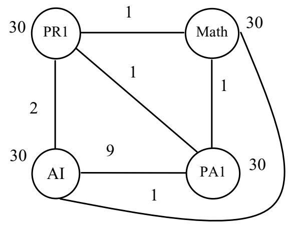

Fig. 1. A simple example of examination timetabling problem

the colouring of the vertices in such a way that no two adjacent vertices are coloured by the same colour. In the context of examination timetabling, colours correspond to time periods. In Figure 1, it can be seen that at least four different time periods are required to solve the problem since no two examinations which are in conflict with each other should be scheduled in the same time period.

Both the examination timetabling problem and the graph colouring problem are known to be NP-complete [29]. However, the examination timetabling problem has an additional wide variety of hard and soft constraints [24]. Hard constraints are those that must be completely satisfied. Solutions which do not violate hard constraints are called feasible solutions. Soft constraints are not essential to the feasibility of a timetable, but their satisfaction is highly desirable. In practice, the quality of an examination timetable is evaluated by some measure of satisfaction of soft constraints since it is usually impossible to fully satisfy all of them. A thorough review of the variety of constraints imposed on examination timetabling can be seen in [4].

1.1 Heuristics for Examination Timetabling

The complexity and the large size of the real-life university examination timetabling problems required development of different heuristics which were employed with reasonable success for their solving over the last 40 years [12], [18], [38]. Early research was focused on sequential heuristics for solving graph colouring problems [7], [20], [21], [46]. The main idea of these heuristics is to schedule examinations one by one, starting from the examination which is evaluated as the most "difficult" for scheduling. Different heuristics measure the "difficulty" of each examination in different ways. The drawback of these heuristics is that they have different performance on varied problem instances [23].

In recent years, there has been an increased interest in application of metaheuristics to examination timetabling problem solving because these techniques can take into consideration soft constraints and are usually able to generate more satisfactory solutions than sequential heuristics alone can do. In practice, timetabling problems are usually solved by a two-phase approach that consists of initialisation and improvement phases. In the first phase, an initial solution is iteratively constructed by using an appropriate sequential heuristic. The improvement phase gradually improves the initial solution by using a metaheuristic such as Simulated Annealing [31], [35], [45], Memetic Algorithm [8], [13], [15], GRASP [25] and Tabu Search [27], [47]. However, the performance of some metaheuristics is known to be highly dependent on the parameter values [6]. For example, it is well known that the settings of the cooling parameters have a great importance to the successful application of Simulated Annealing [45]. Furthermore, the performance of many approaches may vary from one problem instance to another, because they were developed specifically for solving one particular class of real-world problems [2].

In practice, a timetable administrator needs to make a great effort to select an appropriate (successful) sequential heuristic for hybridisation with a metaheuristic, and then needs to "tailor" the chosen heuristics by utilising the domain-specific knowledge to obtain a preferred solution for a given problem. Recently, the development of more general timetabling approaches that are capable of solving a variety of problems with different characteristics equally well, has attracted the attention of the timetabling community. In particular, the research into hyper-heuristics for examination timetabling gave promising results. Hyper-heuristics is defined as "the process of using (meta-)heuristics to choose (meta-)heuristics to solve the problem in hand" [5]. Terashima-Marín et al. [44] introduced an evolutionary approach as a hyper-heuristic for solving examination timetabling problems. In their approach, a list of different sequential heuristics, parameter value settings, and the conditions for swapping sequential heuristics are encoded as chromosomes. The timetable is built by using the best chromosome found by a genetic algorithm. Burke et al. [6] proposed a hyper-heuristic for timetabling problems in which the selection of heuristics is controlled by a tabu search algorithm. Tabu search approaches were employed within the hyper-heuristic framework that searched for different permutations of graph heuristics for solving both exam and course timetabling problems [3], [10].

1.2 Case-Based Heuristic Selection

Case-based reasoning (CBR) [32] is an artificial intelligence methodology which is an effective alternative to traditional rule-based systems. It is in particular useful for generating intelligent decisions in weak-theory application domains [26]. CBR stems from the observation that similar problems will have similar solutions [34]. Rather than defining a set of "IF THEN" rules or general guidelines, a CBR system solves a new problem by reusing previous problem solving experience, stored as cases in the case base. In CBR, a new input problem is usually solved by four steps: retrieve a case that is the most similar to the new problem, reuse and revise the solution of the retrieved case to generate a solution for the new problem, and retain the new input problem and its solution as a new case in the case base.

In the domain of scheduling, there have been some attempts to resort to CBR for achieving the intelligent heuristic selection so that the flexibility and robustness of scheduling is enhanced. Current CBR systems for heuristic selection fall into two categories: algorithm reuse and operator reuse. The basic underlying assumption of the CBR systems in the first category is that it is likely that an approach proved to be effective for solving a specific problem will be also effective for solving a similar problem. In these CBR systems, a case contains a problem representation and an algorithm known to be effective for its solving. Schmidt [43] proposed a CBR framework to choose an appropriate algorithm for a given production scheduling problem. Schirmer [42] designed a similar CBR system for solving project scheduling problems and showed that the CBR system outperformed a number of metaheuristics. A case-based reasoning system was developed by Burke et al. for solving university course timetabling problems [9], [11].

The CBR scheduling systems in the second category iteratively reuse the operators for solving a new input problem. A case in these systems describes a context in which a previously used scheduling operator proved to be successful. Miyashita and Sycara [36] built a CBR system called CABINS for solving job scheduling problems in which sub-optimal solutions were improved by iteratively employing a number of move operators, selected by CBR. Petrovic, Beddoe, and Berghe [37] developed a CBR system for nurse rostering problems in which the constraint satisfaction procedure was driven by iterative application of the scheduling repair operators employed in previously encountered similar situations. Burke, Petrovic, and Qu [19] proposed a novel case based hyper-heuristic for solving timetabling problems. A timetable was iteratively constructed by using a number of heuristics, which were selected by a CBR controller.

In general, the CBR systems' effectiveness depends on the proper definition of the similarity measure, because it determines which case will be used for solving a new input problem. In the current CBR scheduling systems for heuristic selection, cases are usually represented by the sets of attribute-value pairs, while the similarity between two cases is calculated as the distance between their attribute sets. The attributes and their weights can be set either empirically [36], [37], [42] or by employing knowledge discovery methods [19].

The objective of our research is to raise the level of generality of metaheuristic approaches to examination timetabling problems. A CBR system [40], [41] based on algorithm reuse was developed which produced high quality solutions for a range of different examination timetabling problems. The CBR system selected an appropriate sequential heuristic for the initialisation of the Great Deluge algorithm (GDA) [28]. GDA has been chosen due to its simplicity of use in terms of required parameters and high-quality results that it produced for examination timetabling problems. It has been shown that sequential heuristic selected for the initialisation phase had a great impact on the quality of the final solution [41]. In addition, a sequential heuristic which provided a "good" starting point for the GDA search in solving a particular timetabling problem, was proved to be good for the GDA initialisation in solving a similar timetabling problem.

Our research is focused on the application of sequential heuristics in the initialisation phase of the GDA. In the CBR system developed, a case consists of a description of an examination timetabling problem and the sequential heuristic that was used to construct a good initial solution for the GDA applied to the problem. The selection of the sequential heuristic for a new input problem comprises the following steps. The similarity between the new input problem and each problem stored in the case base is calculated. A case which is the most similar to the new input problem is retrieved, and the associated sequential heuristic of the retrieved case is used for the GDA initialisation for a new input problem.

In this paper, we discuss different representations of timetabling problems and corresponding similarity measures. The first representation takes into consideration the number of students involved in examinations and uses weighted graph representation of the timetabling problem [40]. The second representation does not consider number of students and uses unweighted graph representation [41]. We propose a new similarity measure based on weighted graph representation, which instead of using crisp number of students involved in the conflicts uses linguistic terms (*Low*, *Medium*, *High*) to evaluate the importance of conflicts between two examinations. Fuzzy sets are used to model these linguistic terms.

The paper is organised as follows. Section 2 provides a brief introduction to GDA and different sequential heuristics that are used for the initialisation phase. Section 3 describes briefly two different similarity measures based on the weighted and unweighted graph representation of timetabling problems, and introduces a new fuzzy similarity measure. Section 4 briefly introduces the retrieval process in our CBR system. Section 5 presents a series of experiments on benchmark problems that were carried out to evaluate the performance of the new CBR system. The final conclusions are presented in Section 6.

2 Great Deluge Algorithm and Sequential Heuristics

Great Deluge Algorithm (GDA) is a local search method proposed by Dueck [28]. Compared to the well known Simulated Annealing approach, GDA uses a simpler acceptance rule for dealing with the move that leads to a decrease in the solution quality. Such a worse intermediate solution can be accepted if the value of the objective function of the solution is smaller than a given upper boundary value, referred to as “water-level”. Water-level is initially set to be the penalty of the initial solution multiplied by a predefined factor. After each move, the value of the water-level is decreased by a fixed rate, which is computed based on the time that is allocated for the search (expressed as the total number of moves). One important characteristic of the GDA is that better solutions could be obtained with the prolongation of the search time of the algorithm [1]. This may not be valid in other local search algorithms in which the search time cannot be controlled. Burke et al. developed a GDA algorithm for examination timetabling [1]. The authors proposed to use the total number of moves, which expresses the computational time that the user is willing to spend, in the calculation of water level.

A variety of sequential heuristics can be used to construct initial solutions for the GDA. Five different heuristics are used in this research:

1. Largest Degree, which schedules examinations with the largest number of conflicts first,
2. Largest Enrolment, which priorities for scheduling examinations with the largest student enrolment,
3. Largest Colour Degree, which prioritises examinations with the largest number of conflicts that they have with already scheduled examinations,
4. Largest Weighted Degree, which estimates the difficulty of scheduling of each examination by the weighted conflicts, where each conflict is weighted by the number of students who are enrolled in both examinations, and
5. Least Saturation Degree, which schedules examinations with the least number of available periods for placement first.

They can be further hybridised with Maximum Clique Detection [30], Backtracking [33], and Adding Random Elements [17]. In total, 40 different sequential heuristics are investigated. The details of these heuristics are given in [41].

3 Similarity Measures for Examination Timetabling Problems

A properly defined similarity measure has a great impact on the CBR system. On the other hand, similarity measure is tightly connected with the representation of the cases. In this section we will briefly introduce two different similarity measures between examination timetabling problems based on different graph representations, which we investigated in our previous research work. A new similarity measure will be introduced next, which addresses some deficiencies of the previous ones.

3.1 Similarity Measure Based on Weighted Graph Representation

A timetabling problem is represented by a undirected weighted graph $G = (V, E, \alpha, \beta)$, where V is the set of vertices that represent examinations, $E \subseteq V \times V$ is the finite set of edges that represent conflicts between examinations, $\alpha : V \mapsto \mathsf{N}^+$ assigns a positive integer weight to each vertex that corresponds to the number of students enrolled in the examination, $\beta : E \mapsto \mathsf{N}^+$ is an assignment of weight to each edge which corresponds to the number of students enrolled in two examinations that are in conflict. The similarity measure between a new input problem $G_q = (V_q, E_q, \alpha_q, \beta_q)$ and a problem stored in the case base $G_s = (V_s, E_s, \alpha_s, \beta_s)$ is based on the graph isomorphism, which is known to be a NP-complete problem. An isomorphism is presented by a vertex-to-vertex correspondence $f : V_q \to V_s$ which associates vertices in V_q with those in V_s. In our notation, vertices and edges of graph G_q are denoted by Latin letters, while those of graph G_s are denoted by Greek letters.

The similarity degree between two vertices, $a \in V_q$ and $\chi \in V_s$, determined by correspondence f is denoted by $DS_f(a, \chi)$ and calculated in the following way:

$$DS_f(a,\chi) = \begin{cases} \min(\alpha_q(a), \alpha_s(\chi)) & \text{if } f(a) = \chi \\ 0 & \text{otherwise}. \end{cases} \tag{1}$$

Similarly, $DS_f(x,\gamma)$ represents the similarity degree between two edges determined by correspondence f, where $x = (a,b) \in E_q$ and $\gamma = (\chi,\delta) \in E_s$ and is calculated as follows:

$$DS_f(x,\gamma) = \begin{cases} \min(\beta_q(x), \beta_s(\gamma)) & \text{if } f(a) = \chi \text{ and } f(b) = \delta \\ 0 & \text{otherwise}. \end{cases} \tag{2}$$

The label ϕ is used to denote an extraneous vertex or edge in a graph, which is not mapped by correspondence f. $DS_f(a,\phi)$, $DS_f(\phi,\chi)$, $DS_f((a,b),\phi)$ and $DS_f(\phi,(\chi,\delta))$ are set to be equal to 0.

Finally, the similarity degree $\mathrm{SIM1}_f(G_q,G_s)$ between the graphs G_q and G_s determined by correspondence f is calculated in the following way:

$$\mathrm{SIM1}_f(G_q,G_s) = \frac{F_v + F_e}{M_v + M_e} \tag{3}$$

where

$$F_v = \sum_{a\in V_q} \sum_{\chi \in V_s} DS_f(a,\chi) \tag{4}$$

$$F_e = \sum_{x\in E_q} \sum_{\gamma \in E_s} DS_f(x,\gamma) \tag{5}$$

$$M_v = \min\left(\sum_{a\in V_q} \alpha_q(a), \sum_{\chi\in V_s} \alpha_s(\chi)\right) \tag{6}$$

$$M_e = \min\left(\sum_{x\in E_q} \beta q(x), \sum_{\gamma\in E_s} \beta_s(\gamma)\right). \tag{7}$$

Note that the value of $DS_f(G_q,G_s) \in [0,1]$ is subject to correspondence f. The task is to find a correspondence f that yields as high value of $DS_f(G_q,G_s)$ as possible.

The results obtained using the weighted graph representation and described similarity measure are given in [40] (the normalisation of $\mathrm{SIM1}_f(G_q,G_s)$, performed by dividing with $(M_v + M_e)$ is calculated here differently than in [40] due to the changes in the retrieval process which will be described in Section 4).

3.2 Similarity Measure Based on Unweighted Graph Representation

A timetabling problem is represented by a graph $G = (V,E)$. The numbers of students who are sitting examinations and are involved in examination conflicts are not taken into consideration.

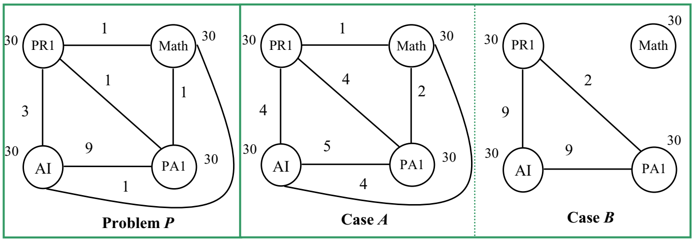

Fig. 2. New problem P and the case base with cases A and B

The similarity degree $DS_f(a,\chi)$ between two vertices in G_q and G_s determined by correspondence f is calculated in the following way:

$$DS_f(a,\chi) = \begin{cases} 1 & \text{if } f(a) = \chi \\ 0 & \text{otherwise}\,. \end{cases} \tag{8}$$

Similarly, the calculation of the similarity degree $DS_f(x,\gamma)$ between two edges determined by correspondence f, where $x = (a,b) \in E_q$ and $\gamma = (\chi,\delta) \in E_s$, is given by

$$DS_f(x,\gamma) = \begin{cases} 1 & \text{if } f(a) = \chi \text{ and } f(b) = \delta \\ 0 & \text{otherwise}\,. \end{cases} \tag{9}$$

In such a definition of similarity between two timetabling problems a mapped pair of vertices/edges in two graphs contributes to the similarity by a constant value 1 (independently from a number of students involved in the mapped vertices/edges). Finally, the similarity degree $\mathrm{SIM2}_f(G_q,G_s)$ between G_q and G_s determined by correspondence f is calculated in the following way:

$$\mathrm{S}IM2_f(G_q,G_s) = \frac{F_v + F_e}{M_v + M_e} \tag{10}$$

where

$$F_v = \sum_{a\in V_q}\sum_{\chi\in V_s} DS_f(a,\chi) \tag{11}$$

$$F_e = \sum_{x\in E_q}\sum_{\gamma\in E_s} DS_f(x,\gamma) \tag{12}$$

$$M_v = \min(|V_q|,|V_s|) \tag{13}$$

$$M_e = \min(|E_q|,|E_s|) \tag{14}$$

where $|V|$ and $|E|$ denote the cardinality of the sets V and E, respectively. Experimental results show that the similarity measure SIM2 on average outperforms SIM1 on benchmark problems established within university timetabling community [41].

Table 1. Similarity between timetabling problems P and A, B, by similarity measure SIM1

	P and A	P and B
F_v	$30 + 30 + 30 + 30 = 120$	$30 + 30 + 30 + 30 = 120$
F_e	$3 + 5 + 1 + 1 + 1 + 1 = 12$	$3 + 1 + 9 = 13$
M_v	$\min(120, 120) = 120$	$\min(120, 120) = 120$
M_e	$\min(16, 20) = 16$	$\min(16, 20) = 16$
$\mathrm{SIM1}(P,^*)$	$(120 + 12)/(120 + 16) = 0.97$	$(120 + 13)/(120 + 16) = 0.978$

3.3 Fuzzy Similarity Measure Based on Weighted Graph Representation

The similarity measure SIM1 is investigated further. In order to find a case in the case base that is similar to the new timetabling problem, i.e. to establish a "good" isomorphism between two graphs, two issues are considered. Firstly, it is necessary to find a "good" correspondence between vertices/edges of the new timetabling problem and the one stored in the case base. Secondly, weights of the vertices/edges should have equal or similar values. However, it was noticed that in some situations the similarity measure SIM1 will give priority to a graph with less similar structure to the new input problem but with the same (high) weights of the corresponding vertices/edges over a graph with more similar structure but different weights of the corresponding vertices/edges.

To illustrate this observation let us consider three timetabling problems whose structures are given in Figure 2: a new input problem P and problems A and B which are stored in the case base. Let us suppose that the established graph isomorphism(s) associates vertices in P and those in $A(B)$ that have the same examination names. The similarities between P and A and B are given in Table 1.

Similarity measure SIM1 evaluates case B to be more similar (although slightly) to new problem P than case A. Obviously, following the definition of similarity SIM1 the weights of the corresponding edges of P and B that are equal contribute more to the similarity than the corresponding edges of P and A which do not have the same weights. However, graph P has the same structure as graph A, but is structurally very different to graph B. These observations motivated the definition of the new similarity measure SIM3 to improve the effectiveness of the previously developed CBR system [40], [41]. This similarity measure does not consider vertex weights but only edge weights because they indicate the size of the conflict between the examinations. The corresponding edges will still contribute to the similarity between two graphs, but their contribution needs to be smaller than their weights. The procedure for calculation of the contribution of the edge weights to the similarity measure consists of two steps:

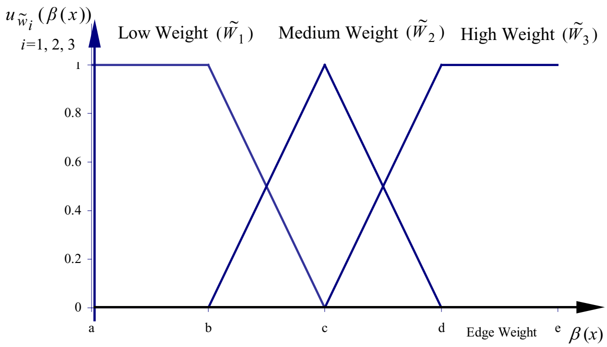

Fig. 3. Membership functions defined for fuzzy sets *Low Weight*, *Medium Weight*, *High Weight*

Step I The corresponding edges of the two graphs are classified to sets: *Low Weight*, *Medium Weight* and *High Weight*. In order to avoid a rigid definition of strict boundaries of these sets, fuzzy sets [48, 49] are used for their modelling. Unlike classical sets in which each object is either a member or not a member of a given set, a fuzzy set $\tilde{A}$ defined on a universe of discourse U is characterised by a membership function $u_{\tilde{A}}(x) \in [0, 1]$ that assigns to each object $x \in U$ a degree of membership of x in $\tilde{A}$. The membership functions for three fuzzy sets *Low Weight* ($\tilde{W}_1$), *Medium Weight* ($\tilde{W}_2$) and *High Weight* ($\tilde{W}_3$) are given in Figure 3.

Parameters a, b, c, d, e are defined in the following way. Parameter a defines the lower bound of the set *Low Weight* and is set to be 1 (weight of edges are positive integers). Parameter b is calculated as the mean value of all edge weights in the graph:

$$b = \frac{\sum\limits_{x \in E} \beta(x)}{|E|} \,. \tag{15}$$

The assumption is that the edges whose weight is smaller than the mean weight have high degree of membership to *Low Weight*. Parameter e is set to be the maximum edge weight in the graph:

$$e = \max_{x \in E} \beta(x) \,. \tag{16}$$

Parameters c and d are set to divide the $[b, e]$ interval into equal sizes:

$$c = b + \frac{e - b}{3} = \frac{2b + e}{3} \tag{17}$$

$$d = b + 2\frac{e - b}{3} = \frac{b + 2e}{3} \,. \tag{18}$$

The result of step I is the classification of the corresponding edge weights in the established graph isomorphism given by a triplet

$$\left(u_{\widetilde{low\_wt}}(\beta(x)), u_{\widetilde{med\_wt}}(\beta(x)), u_{\widetilde{high\_wt}}(\beta(x))\right)$$

which denotes a membership degree of edge x to three fuzzy sets: *Low Weight*, *Medium Weight* and *High Weight*.

Step II Based on the classification obtained in step I, the weight of the edge is assigned a real number W_x which determines its contribution to the similarity measure between two graphs. Experiments indicated that real number should not be greater than the average edge weight in the graph. It is calculated using the formula

$$W_x = \frac{\sum_{i=1}^{3} h_i u_{\tilde{w}_i}(\beta(x))}{\sum_{i=1}^{3} u_{\tilde{w}_i}(\beta(x))} \tag{19}$$

where h_1 is set to be 1; h_2 is set as mean of h_1 and h_3; h_3 is set as mean weight of all edges' weights of the graph of the new input timetabling problem.

The similarity degree between two vertices a and χ on correspondence f is defined as follows:

$$DS_f(a, \chi) = \begin{cases} 1 & \text{if } f(a) = \chi \\ 0 & \text{otherwise}. \end{cases} \tag{20}$$

The similarity degree between two edges x and γ, where $x = (a, b) \in E_q$ and $\gamma = (\chi, \delta) \in E_s$, on correspondence f is denoted by $DS_f(x, \gamma)$:

$$DS_f(x, \gamma) = \begin{cases} \min(W_x, W_\gamma) & \text{if } f(a) = \chi \text{ and } f(b) = \delta \\ 0 & \text{otherwise} \end{cases} \tag{21}$$

where W_x and W_γ are the new edge weights for edges x and γ, respectively.

Similarity degree $\mathrm{SIM3}_f(G_q, G_s)$ between two undirected weighted graphs G_q and G_s on correspondence f is calculated as

$$\mathrm{SIM3}_f(G_q, G_s) = \frac{F_v + F_e}{M_v + M_e} \tag{22}$$

where

$$F_v = \sum_{a \in V_q} \sum_{\chi \in V_s} DS_f(a, \chi) \tag{23}$$

$$F_e = \sum_{x \in E_q} \sum_{\gamma \in E_s} DS_f(x, \gamma) \tag{24}$$

$$M_v = (|V_q|, |V_s|) \tag{25}$$

$$M_e = \min\left(\sum_{x \in E_q} W(x), \sum_{\gamma \in E_s} W_\gamma\right) \tag{26}$$

where M_v and M_e are the maximum values that F_v and F_e can take, respectively.

The procedure for calculation of the similarity between case P and cases A and B from the case base is illustrated by an example given in Figure 2. The calculation of "new weights" of edges with which they will contribute to the similarity measure are given in Table 2, while Table 3 presents the calculation of new similarities between cases P and A and B. According to this new similarity measure, case A is more similar to case P than case B.

4 Retrieval Process

A case base may contain a large number of cases. The retrieval process of the CBR system has to establish a graph isomorphism between a new problem and all cases in the case base. In order to enable the faster retrieval a filtering phase is introduced which retrieves the subset of cases from the case base using a set of features, that we refer to as *shallow properties*. They reflect the size and the complexity of the problem: f_1, the number of examinations; f_2, the number of enrolments; f_3, the number of time periods available; f_4, the density of the conflict matrix (calculated as the ratio of the number of examinations in conflict to the square of the total number of examinations).

The nearest neighbour is used to calculate the similarity degree of two cases based on the shallow properties, represented by feature sets F_q and F_s:

$$\mathrm{SIM}_{\mathrm{shallow}}(F_q, F_s) = 1 - \sqrt{\frac{1}{n}\sum_{i=1}^{n} \mathrm{distance}\left(f_{q_i}, f_{s_i}\right)^2} \tag{27}$$

where n is the number of features, f_{q_i} and f_{s_i} are the values of the ith feature in F_q and F_s, respectively, and the *distance* between two feature values f_{q_i} and f_{s_i} is computed as

$$\mathrm{distance}(f_{q_i}, f_{s_i}) = \left| \frac{f_{q_i} - f_{s_i}}{f_{\max_i} - f_{\min_i}} \right| \tag{28}$$

where $f_{\max_i}$ and $f_{\min_i}$ are the maximum and minimum values of the ith feature recorded in the case base.

The cases whose similarity with the new problem is greater than the predefined threshold (empirically set to be 0.6) are passed to the Tabu Search algorithm [39] which searches for the best graph isomorphism SIM1 in terms of defined similarity measures (SIM1, SIM2 or SIM3) between the new problem and the retrieved subset of cases. Finally, the general similarity measure is calculated between the new problem C_q and a case C_s from the subset of cases, using the formula

$$\mathrm{SIM}(C_q, C_s) = \mathrm{SIM}_{\mathrm{shallow}}(F_q, F_s) \cdot \mathrm{SIM}_f(G_q, G_s)\,. \tag{29}$$

Table 2. Calculation of "new weights" in graphs P, A and B

	Problem P			Case A			Case B			$DS_f(x,\gamma)$	$DS_f(x,\gamma)$
Edge	Weight	Step I	Step II	Weight	Step I	Step II	Weight	Step I	Step II	$x \in E_P,\ \gamma \in E_A$	$x \in E_P,\ \gamma \in E_B$
(PR1, AI)	3	(0.72, 0.28, 0)	1.23	4	(0, 0.8, 0.2)	2	9	(1, 0, 0)	2.67	1.23	1.23
(PR1, Math)	1	(1, 0, 0)	1	1	(1, 0, 0)	1	–	–	–	1	0
(PR1, PA1)	1	(1, 0, 0)	1	4	(1, 0, 0)	2	2	(0, 0, 1)	1	1	1
(PA1, Math)	1	(1, 0, 0)	1	2	(1, 0, 0)	1	–	–	–	1	0
(PA1, AI)	9	(0, 0, 1)	2.67	5	(0, 0, 1)	2.67	9	(0, 0, 1)	2.67	2.67	2.67
(AI, Math)	1	(1, 0, 0)	1	4	(1, 0, 0)	2	–	–	–	1	0
Sum	16		7.9	20		10.67	20		6.34	$F_e = 7.9$	$F_e = 4.9$

Table 3. Similarity between timetabling problem P and A, B, by the new similarity measure SIM3

Similarity	F_v	M_v	F_e	M_e	SIM3 (P, *)
P and A	4	4	7.9	$\min(7.9, 10.67) = 7.9$	$(4 + 7.9)/(4 + 7.9) = 1.0$
P and B	4	4	4.9	$\min(7.9, 6.34) = 6.34$	$(4 + 4.9)/(4 + 6.34) = 0.86$

5 Experimental Results

The experiments were performed on a number of real-world examination problems from different universities that has been collected and used as benchmark problems. The objectives of the experiments are

- to compare different similarity measures;
- to investigate whether the new similarity measure can enable retrieval of the most effective sequential heuristics for the benchmark problems;
- to evaluate the new CBR system performance by comparing it with the other state-of-the-art approaches to examination timetabling.

ftp://ftp.mie.utoronto.ca/pub/carter/testprob/ gives the benchmark problems. Their characteristics are shown in Table 4.

The cost function of these problems takes into consideration the spread of student's examinations. The cost function was adopted in the research on university examination timetabling and enables comparison between different timetabling approaches. It can be described by the following formula [23]:

$$w_s = \frac{32}{2^s}, \quad s \in \{1, 2, 3, 4, 5\} \tag{30}$$

where w_s is the cost given to the solution whenever a student has to take in two examinations scheduled s periods apart from each other. Experiments were run on a PC with a 1400 MHz Athlon processor and 256 MB RAM.

5.1 Case Base Initialisation

In our experiments, the initial case base was seeded with a number of examination timetabling problems that were randomly generated (more details are given in [41]). Seeding problems differ in three parameters: the number of examinations (n), the number of students (s), and the density of the conflict matrix (d). Three seeding problems were created for each combination of these parameters, which are random variables with a normal distribution where mean of $n \in \{100, 200, 300, 400\}$, mean of $s \in \{10 \times n,\ 20 \times n\}$, and mean of $d \in \{0.07, 0.15, 0.23\}$. For each n, s and d, the proportion of the standard deviation and the mean was set as 0.05. Thus, 72 ($3 \times 4 \times 2 \times 3$) different seeding problems were generated for the case base.

Table 4. Examination timetabling benchmark problems

Data	Institution	Periods	No. of Exams	No. of Students	No. of Enrolments	Density of Conflict Matrix
car-f-92	Carleton University, Ottawa	32	543	18,419	55,522	0.14
car-s-91	Carleton University, Ottawa	35	682	16,925	56,877	0.13
ear-f-83	Earl Haig Collegiate Institute, Toronto	24	190	1,125	8,109	0.29
hec-s-92	Ecole des Hautes Etudes Commercials, Montreal	18	81	2,823	10,632	0.20
kfu-s-93	King Fahd University Dharan	20	461	5,349	25,113	0.06
lse-f-91	London School of Economics	18	381	2,726	10,918	0.06
rye-s-93	Ryerson University, Toronto	23	486	11,483	45,051	0.07
sta-f-83	St Andrew's Junior High School, Toronto	13	139	611	5,751	0.14
tre-s-92	Trent University, Peterborough, Ontario	23	261	4,360	14,901	0.18
uta-s-92	Science, University of Toronto	35	622	21,276	11,793	0.08
ute-s-92	Faculty of Engineering, University of Toronto	10	184	2,750	11,793	0.08
yor-f-83	York Mills Collegiate Institute, Toronto	21	181	941	6,034	0.27

In order to find the best initialisation heuristic for each seeding problem, the GDA initialised by each sequential heuristic was run for 5 times using 20×10^6 iterations (this value was set empirically), while the water-level was set to 1.3 (this value is taken from [14]). These values for the number of iterations and for the water-level will be employed in most of the experiments presented in this paper. Finally, three case bases were established: a small, a middle and a large case base with 24, 48 and 72 cases, respectively.

5.2 Evaluation of Similarity Measures

The purpose of this set of experiments is to evaluate the effectiveness of the proposed similarity measure SIM3. This new similarity measure is also compared with the similarity measures SIM1 and SIM2.

Having established three case bases and defined three different similarity measures, each combination of a case base and a similarity measure was employed to choose a sequential heuristic for each of the 12 benchmark problems. We adopted the method described in [41] to evaluate whether the retrieved sequential heuristic is effective for the benchmark problem. For each benchmark problem, the GDA was run five times initialised by each sequential heuristic. After that sequential heuristics were sorted in ascending order by the average final solution cost obtained. The rank of the sequential heuristic H for the problem P is denoted by $R(H, P)$.

The System Effectiveness Degree $SED(P)$ indicates the distance between the sequential heuristic used in the case retrieved from the case base denoted by H^{CB} and the heuristic H^{best} which is the best for the GDA initialisation for the benchmark problem P ($R(H^{\mathrm{best}}, P) = 1$). It is calculated as

$$SED(P) = 1 - \frac{R(H^{\mathrm{CB}}, P) - 1}{N - 1} \tag{31}$$

where N is the total number of heuristics used for the GDA initialisation. A high value of SED indicates the high effectiveness of the retrieved sequential heuristic. For each combination of the case base and the similarity measure, the average $SED(P)$ values were computed for all benchmark problems and are shown in Figure 4.

It is evident that the SED values of SIM3 are higher than those of SIM1 and SIM2 for all three case bases. This result justifies the new fuzzy similarity measure. The experimental results also show that the growth of the size of the case base leads to the retrieval of more effective sequential heuristics.

5.3 System Performance on Benchmark Problems

The following set of experiments aims to investigate the effectiveness of our CBR system by comparing the obtained results with those of other approaches. The CBR system with the similarity measure SIM3 and the large case base were used to solve benchmark problems. In each experiment, our CBR system selected

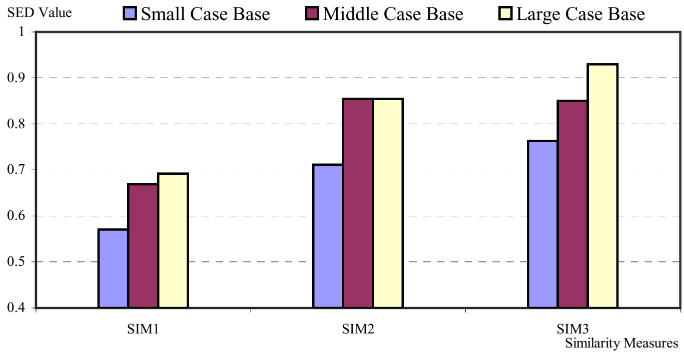

Fig. 4. Performance of different similarity measures

a sequential heuristic for a benchmark problem. The problem was solved by running the retrieved sequential heuristic and the GDA successively for 200×10^6 iterations, five times with varying random number seeds. System Effectiveness Degree *SED* is calculated for each retrieved sequential heuristic. Table 5 shows our results and the best results achieved by the exhaustive search across all heuristics.

It can be seen that CBR succeeded in suggesting the appropriate heuristics for the GDA initialisation and thus resulted in high-quality solutions. The new CBR initialisation was successful in finding the best heuristics for the benchmark problem lse-f-91, sta-f-83 and uta-s-92. For seven problem instances car-f-92, car-s-91, ear-f-83, hec-s-92, kfu-s-93, rye-f-92 and yor-f-83, the retrieved heuristics are among the four best ($0.923 \leq SED \leq 0.974$). It is important to note that the developed CBR initialisation took in average less than 10 minutes for each timetabling problem, while an exhaustive test needed more than six hours.

Table 6 shows the comparison of the average results generated by three other state-of-the-art approaches: the GDA where the initial solution was constructed by saturation degree (SD) [1], the GDA initialised by the adaptive heuristic [14], [16], the GDA where the saturation degree heuristic was applied with the maximum clique detection (MCD) and backtracking (BT) in the initialisation phase (this heuristic was suggested by Carter et al. [22] to be the best constructive heuristic). Each problem instance was solved five times. The time (in seconds) shown is the average time spent on the search. The GDA was also allocated the same number of iterations $200*10^6$ for each approach. In this experiment, we employed the higher number of iterations than in the previous ones in order to compare our results with the published ones. The times shown are different due to the use of computers of different characteristics.

For nine benchmark problems, our CBR system obtained best average results (highlighted by bold characters). For two problems, second best average results were obtained. Even more, for eleven benchmark problems the best value of the cost function was obtained as a result of appropriate GDA initialisation. The obtained results prove the significance of the appropriate initialisation of GDA.

Table 5. Comparison of results for benchmark problems obtained by the exhaustive search and CBR initialisation of the GDA

	Exhaustive test				CBR ($\lvert CB \rvert = 72$, SIM3)			
	Retrieval		Run GDA		Retrieval		Run GDA	
Data	*SED*	Time (s)	Cost	Time (s)	*SED*	Time (s)	Cost	Time (s)
car-f-92	1.00	35700	**3.97**	1080	0.923	491	3.99	1027
car-s-91	1.00	42739	**4.52**	1310	0.948	1733	4.53	1040
ear-f-83	1.00	15245	**34.78**	690	0.949	445	34.87	690
hec-s-92	1.00	20874	**11.32**	1490	0.923	73	11.36	1021
kfu-s-93	1.00	19643	**14.11**	689	0.974	1402	14.35	751
lse-f-91	1.00	15095	**10.78**	595	**1.00**	1170	**10.78**	559
rye-f-92	1.00	20123	**8.74**	862	0.974	683	8.79	699
sta-f-83	1.00	12368	**158.02**	676	**1.00**	91	**158.02**	649
tre-s-92	1.00	16495	**8.03**	730	0.744	972	8.10	844
uta-s-92	1.00	32094	**3.20**	1051	**1.00**	839	**3.20**	1051
ute-s-92	1.00	10755	**25.70**	557	0.769	172	26.10	574
yor-f-83	1.00	26723	**36.85**	1200	0.949	348	36.88	1243

Finally, we also compare our results with those produced by the state-of-the-art timetabling metaheuristics: Simulated Annealing (SA) [35], Tabu search [47], and GRASP [25]. The average of the scores for the twelve problem instances is shown in Table 7.

We can see that our CBR system outperformed other metaheuristics. Our CBR system obtained the best average results for seven benchmark problems and the second best average results for two benchmark problems. In addition, it is clear that the additional time on the case retrieval is required by our CBR system. However, the time spent on the selection of an appropriate sequential heuristic is justified by the quality of the results.

6 Conclusions

Different graph representation of examination timetabling problems and the corresponding similarity measures between two problems have been discussed. They are used in the CBR system for heuristic initialisation of GDA. The experimental results on a range of real world examination timetabling problems prove that the new fuzzy similarity measure based on weighted graph representation leads to the good selection of sequential heuristic for the GDA initialisation. By assigning linguistic terms to the edge weights of the timetabling graphs, the new similarity measure enables the retrieval of the timetabling problem from the case base which is structurally similar to the new problem.

We have also demonstrated that the CBR system with the new similarity measure can efficiently select a good heuristic for the GDA initialisation for

Table 6. Comparison of results for benchmark problems obtained by different initialisation of the GDA

Data	SD			Adaptive			SD & MCD & BT			CBR ($\|CB\| = 72$)			
	GDA Time	Best Cost	Avg. Cost	GDA Time	Best Cost	Avg. Cost	GDA Time	Best Cost	Avg. Cost	Retrieval Time	GDA Time	Best Cost	Avg. Cost
car-f-92	1120	4.03	4.07	416	–	4.10	1220	3.97	4.04	491	1027	**3.93**	**3.99**
car-s-91	1400	4.57	4.62	681	–	4.65	1441	4.62	4.66	1733	1040	**4.50**	**4.53**
ear-f-83	806	34.85	36.04	377	–	37.05	767	33.82	36.26	445	690	**33.71**	**34.87**
hec-s-92	1471	11.27	12.43	516	–	11.54	1411	11.08	11.48	73	1021	**10.83**	**11.36**
kfu-s-93	843	14.33	14.64	449	–	**13.90**	996	14.35	14.62	1402	751	**13.82**	14.35
lse-f-91	646	11.61	11.65	341	–	10.82	675	11.57	11.94	1170	559	**10.35**	**10.78**
rye-f-92	845	9.19	9.66	–	–	–	881	9.32	9.50	683	699	**8.53**	**8.79**
sta-f-83	675	165.12	169.7	418	–	168.73	674	166.07	166.31	91	649	**151.52**	**158.02**
tre-s-92	907	8.13	8.29	304	–	8.35	751	8.19	8.27	972	844	**7.92**	**8.10**
uta-s-92	1070	3.25	3.30	517	–	**3.20**	1101	3.24	3.31	839	1051	**3.14**	**3.20**
ute-s-92	716	25.88	26.05	324	–	**25.83**	653	25.53	26.02	172	574	**25.39**	26.10
yor-f-83	1381	36.17	**36.59**	695	–	37.28	1261	**36.31**	37.27	348	1243	36.53	36.88

Table 7. Comparison of results for benchmark problems obtained by different metaheuristics

	SA [35]			Tabu [47]			GRASP [25]			CBR ($\|CB\| = 72$)			
Data	Time	Best Cost	Avg. Cost	Time	Best Cost	Avg. Cost	Time	Best Cost	Avg. Cost	Retrieval Time	GDA Time	Best Cost	Avg. Cost
car-f-92	233	4.3	4.4	–	4.63	4.69	–	4.4	4.7	491	1027	**3.93**	**3.99**
car-s-91	296	5.1	5.2	–	5.73	5.82	–	5.4	5.6	1733	1040	**4.50**	**4.53**
ear-f-83	26	35.1	35.4	–	45.8	45.6	–	34.8	35.0	445	690	**33.71**	**34.87**
hec-s-92	5.4	**10.6**	**10.7**	–	12.9	13.4	–	10.8	10.9	73	1021	10.83	11.36
kfu-s-93	40	**13.5**	**14.0**	–	17.1	17.8	–	14.1	14.3	1402	751	13.82	14.35
lse-f-91	35	10.5	11.0	–	14.7	14.8	–	14.7	15.0	1170	559	**10.35**	**10.78**
rye-f-92	70	**8.4**	**8.7**	–	11.6	11.7	–	–	–	683	699	8.53	8.79
sta-f-83	5	157.3	157.4	–	158	158	–	**134.9**	**135.1**	91	649	151.52	158.02
tre-s-92	39	8.4	8.6	–	8.94	9.16	–	8.7	8.8	972	844	**7.92**	**8.10**
uta-s-92	233	3.5	3.6	–	4.44	4.49	–	–	–	839	1051	**3.14**	**3.20**
ute-s-92	9	**25.1**	**25.2**	–	29.0	29.1	–	25.4	25.5	172	574	25.39	26.10
yor-f-83	30	37.4	37.9	–	42.3	42.5	–	37.5	38.1	348	1243	**36.53**	**36.88**

most of the benchmark problems, and even more it outperforms the other state-of-the-art solution approaches based on GDA. This research makes a further contribution to the attempt of development of a general metaheuristic framework for timetabling, which is not tailored for a particular timetabling problem, i.e. works well on a range of different timetabling problems. We believe that this new similarity measure along with the proposed CBR methodology are also applicable to other domains such as personnel scheduling, job shop scheduling, and project scheduling.

References

1. Burke, E. K., Bykov, Y., Newall, J. P., Petrovic, S.: A Time-Predefined Local Search Approach to Exam Timetabling Problems. IIE Trans. Oper. Eng. **36** (2004) 509–528
2. Burke, E. K., Eckersley, A. J., McCollum, B., Petrovic, S., Qu, R.: Similarity Measures For Exam Timetabling Problems. In: Proc. 1st Multidisciplinary International Conference on Scheduling: Theory and Applications (2003) 120–136
3. Burke, E. K., Dror, M., Petrovic, S., Qu, R.: Hybrid Graph Heuristics in a Hyper-Heuristic Approach to Exam Timetabling. In: Golden, B.L., Raghavan, S., Wasil, E. A. (eds.): The Next Wave in Computing, Optimization and Decision Technologies. Springer, Berlin (2005) 79–92
4. Burke, E. K., Elliman, D. G., Ford, P. H., Weare, R. F.: Examination Timetabling in British Universities—A Survey. In: Burke, E., Ross, P.: The Practice and Theory of Automated Timetabling I (PATAT'95, Selected Papers). Lecture Notes in Computer Science, Vol. 1153, Springer, Berlin (1996) 76–92
5. Burke, E. K., Hart, E., Kendall, G., Newall, J., Ross, P., Schulenburg, S.: Hyper-Heuristics: An Emerging Direction in Modern Search Technology. Chapter 16 in: Glover, F., Kochenberger, G. (eds.): Handbook of Meta-heuristics, Kluwer, Dordrecht (2003) 457–474
6. Burke, E. K., Kendall, G., Soubeiga, E.: A Tabu Search Hyper-heuristic for Timetabling and Rostering. J. Heuristics **9** (2003) 451–470
7. Burke, E., Kingston, J., De Werra, D.: Applications to Timetabling. Section 5.6 in: Gross, J., Yellen, J. (eds.): Handbook of Graph Theory, Chapman and Hall/CRC Press, London (2004) 445–474
8. Burke, E. K., Landa, J. D.: Design of Memetic Algorithms for Scheduling and Timetabling Problems. In: Krasnogor, N., Hart, W., Smith, J. (eds.): Recent Advances in Memetic Algorithms and Related Search Technologies. Springer, Berlin (2004) 289–312
9. Burke, E. K., MacCarthy, B., Petrovic, S., Qu, R.: Structured Cases in CBR—Reusing and Adapting Cases for Time-Tabling Problems. Knowledge-Based Syst. **13** (2000) 159–165
10. Burke, E., MacCarthy, B., Petrovic, S., Qu, R.: Knowledge Discovery in a Hyper-heuristic Using Case-Based Reasoning for Course Timetabling. In: Burke, E., De Causmaecker, P.: Practice and Theory of Automated Timetabling IV (PATAT'02, Selected Papers). Lecture Notes in Computer Science, Vol. 2740. Springer, Berlin (2003) 276–287
11. Burke, E. K., MacCarthy, B., Petrovic, S., Qu, R.: Multiple-Retrieval Case Based Reasoning for Course Timetabling Problems. J. Oper. Res. Soc. (2005) accepted for publication

12. Burke, E. K., Meisels, A., Petrovic, S., Qu, R.: A Graph-Based Hyper Heuristic for Timetabling Problems. Eur. J. Oper. Res. (2005) accepted for publication
13. Burke, E. K., Newall, J. P., Weare, R. F.: A Memetic Algorithm for University Exam Timetabling. In: Burke, E., Ross, P.: The Practice and Theory of Automated Timetabling I (PATAT'95, Selected Papers). Lecture Notes in Computer Science, Vol. 1153, Springer, Berlin (1996) 241–250
14. Burke, E. K., Newall, J. P.: Enhancing Timetable Solutions with Local Search Methods. In: Burke, E., De Causmaecker, P.: Practice and Theory of Automated Timetabling IV (PATAT'02, Selected Papers). Lecture Notes in Computer Science, Vol. 2740. Springer, Berlin (2003) 195–206
15. Burke, E. K., Newall, J. P., Weare, R. F.: Initialisation Strategies and Diversity in Evolutionary Timetabling. Evol. Comput. **6** (1998) 81–103
16. Burke, E. K., Newell, J. P.: Solving Examination Timetabling Problems Through Adaptation of Heuristic Orderings. Ann. Oper. Res. **129** (2004) 107–134
17. Burke, E. K., Newell, J. P., Weare, R. F.: A Simple Heuristically Guided Search for the Timetable Problem. In: Proceedings of the International ICSC Symposium on Engineering of Intelligent Systems (University of La Laguna). Academic, New York (1998) 574–579
18. Burke, E. K., Petrovic, S.: Recent Research Directions in Automated Timetabling. Eur. J. Oper. Res. **140** (2002) 266–280
19. Burke, E. K., Petrovic, S., Qu, R.: Case Based Heuristic Selection for Timetabling Problems. J. Scheduling (2006) accepted for publication
20. Carter, M. W.: A Survey of Practical Applications on Examination Timetabling. Oper. Res. **34** (1986) 193–202
21. Carter, M. W., Laporte, G.: Recent Developments in Practical Course Timetabling. In: Burke, E., Ross, P.: The Practice and Theory of Automated Timetabling I (PATAT'95, Selected Papers). Lecture Notes in Computer Science, Vol. 1153, Springer, Berlin (1996) 3–21
22. Carter, M. W., Laporte, G., Chinneck, J. W.: A General Examination Scheduling System. Interfaces **24** (1994) 109–120
23. Carter, M. W., Laporte, G., Lee, S. Y.: Examination Timetabling: Algorithmic Strategies and Applications. J. Oper. Res. Soc. **47** (1996) 373–383
24. Carter, M. W., Johnson, D. G.: Extended Clique Initialisation in Examination Timetabling. J. Oper. Res. Soc. **52** (2001) 538–544
25. Casey, S., Thompson, J.: GRASPing the Examination Scheduling Problem. In: Burke, E., De Causmaecker, P.: Practice and Theory of Automated Timetabling IV (PATAT'02, Selected Papers). Lecture Notes in Computer Science, Vol. 2740. Springer, Berlin (2003) 232–246
26. Deng, P. S.: Using Case-Based Reasoning Approach to the Support of Ill-structured Decisions. Eur. J. Oper. Res. **93** (1996) 511–521
27. Di Gaspero, L., Schaerf, A.: Tabu Search Techniques for Examination Timetabling. In: Burke, E., Erben, W.: The Practice and Theory of Automated Timetabling III (PATAT'00, Selected Papers). Lecture Notes in Computer Science, Vol. 2079. Springer, Berlin (2001) 104–117
28. Dueck, G.: New Optimization Heuristics. J. Comput. Phys. **104** (1993) 86–92
29. Garey, M. R., Johnson, D. S.: Computers and Intractability a Guide to the Theory of NP-completeness. Freeman, San Francisco (1977)
30. Gendreau, M., Soriano, P., Salvail, L.: Solving the Maximum Clique Problem Using a Tabu Search Approach. Ann. Oper. Res. **41** (1993) 385–403
31. Johnson, D.: Timetabling University Examinations. J. Oper. Res. Soc. **41** (1990) 39–47

32. Kolodner, J.: Case-Based Reasoning. Morgan Kaufmann, San Mateo, CA (1993)
33. Laporte, G., Desroches S.: Examination Timetabling by Computer. Comput. Oper. Res. **11** (1984) 351–360
34. Leake, D. B.: CBR in Context: the Present and Future, Case-Based Reasoning: Experiences, Lessons, and Future Directions. AAAI Press/MIT Press, Menlo Park, CA (1996)
35. Merlot, L. T. G., Boland, N., Hughes, B. D., Stuckey, P. J.: A Hybrid Algorithm for the Examination Timetabling Problem. In: Burke, E., De Causmaecker, P.: Practice and Theory of Automated Timetabling IV (PATAT'02, Selected Papers). Lecture Notes in Computer Science, Vol. 2740. Springer, Berlin (2003) 205–232
36. Miyashita, K., Sycara, K.: CABINS: A Framework of Knowledge Acquisition and Iterative Revision for Schedule Improvement and Reactive Repair. Artif. Intell. **76** (1995) 377–426
37. Petrovic, S., Beddoe, G. R., Berghe, G. V.: Storing and Adapting Repair Experiences in Employee Rostering. In: Burke, E., De Causmaecker, P.: Practice and Theory of Automated Timetabling IV (PATAT'02, Selected Papers). Lecture Notes in Computer Science, Vol. 2740. Springer, Berlin (2003) 149–166
38. Petrovic, S., Burke, E.: Educational Timetabling. Chapter 45 in: Leung, J. (ed.): Handbook of Scheduling: Algorithms, Models, and Performance Analysis. Chapman and Hall/CRC Press, London (2004), 45.1–45.23
39. Petrovic, S., Kendall, G., Yang, Y.: A Tabu Search Approach for Graph-Structured Case Retrieval. In: Proc. STarting Artificial Intelligence Researchers Symposium (France). IOS Press (2002) 55–64
40. Petrovic, S., Yang Y., Dror, M.: Case-based Initialisation of Metaheuristics for Examination Timetabling. In: Kendall, G., Burke, E., Petrovic, S., Gendreau, M. (eds.): Multidisciplinary Scheduling Theory and Applications. Springer, Berlin (2005) 289–308
41. Petrovic, S, Yang, Y., Dror, M.: Use of Case Based Reasoning In Solving Examination Timetabling Problems. Technical Report NOTTCS-TR-2004-6, University of Nottingham, UK (2004)
42. Schirmer, A.: Case-Based Reasoning and Improved Adaptive Search for Project Scheduling. Naval Res. Log. **47** (2000) 201–222
43. Schmidt, G.: Case-Based Reasoning for Production Scheduling. Int. J. Product. Econ. **56/7** (1998) 537–546
44. Terashima-Marín, H., Ross, P., Valenzuela-Rendón, M.: Evolution of Constraint Satisfaction Strategies in Examination Timetabling. In: Proceedings of the Genetic and Evolutionary Conference (1999) 635–642
45. Thompson, J. M., Dowsland, K. A.: Variants of Simulated Annealing for the Examination Timetabling Problem. Ann. Oper. Res. **63** (1996) 105–128
46. Welsh, D. J. A., Powell, M. B.: An Upper Bound on the Chromatic Number of a Graph and its Application to Timetabling Problems. Comput. J. **10** (1967) 85–86
47. White, G. M., Xie, B. S., Zonjic, X.: Using Tabu Search With Longer-Term Memory and Relaxation to Create Examination Timetables. Eur. J. Oper. Res. **153** (2004) 80–91
48. Zadeh, L. A.: Fuzzy Sets. Inform. Control **8** (1965) 338–353
49. Zadeh, L. A.: The Concept of a Linguistic Variable and its Application to Approximate Reasoning. Inform. Sci. **8, 9** (1975) 199–249, 43–80, respectively