

# A survey of search methodologies and automated system development for examination timetabling

R. Qu · E.K. Burke · B. McCollum · L.T.G. Merlot ·  
S.Y. Lee

Received: 18 July 2006 / Accepted: 28 January 2008 / Published online: 29 October 2008  
© Springer Science+Business Media, LLC 2008

**Abstract** Examination timetabling is one of the most important administrative activities that takes place in all academic institutions. In this paper, we present a critical discussion of the research on exam timetabling which has taken place in the last decade or so. This last ten years has seen a significantly increased level of research attention for this important area. There has been a range of insightful contributions to the scientific literature both in terms of theoretical issues and practical aspects. The main aim of this survey is to highlight the new trends and key research achievements that have been carried out in the last decade. We also aim to outline a range of relevant important research issues and challenges that have been generated by this body of work.

We first define the problem and discuss previous survey papers. Within our presentation of the state-of-the-art

methodologies, we highlight recent research trends including hybridisations of search methodologies and the development of techniques which are motivated by raising the level of generality at which search methodologies can operate. Summarising tables are presented to provide an overall view of these techniques. We also present and discuss some important issues which have come to light concerning the public benchmark exam timetabling data. Different versions of problem datasets with the same name have been circulating in the scientific community for the last ten years and this has generated a significant amount of confusion. We clarify the situation and present a re-naming of the widely studied datasets to avoid future confusion. We also highlight which research papers have dealt with which dataset. Finally, we draw upon our discussion of the literature to present a (non-exhaustive) range of potential future research directions and open issues in exam timetabling research.

---

R. Qu (✉) · E.K. Burke  
Automated Scheduling, Optimisation and Planning (ASAP)  
Group, School of Computer Science, University of Nottingham,  
Nottingham NG8 1BB, UK  
e-mail: [rxq@cs.nott.ac.uk](mailto:rxq@cs.nott.ac.uk)

E.K. Burke  
e-mail: [ekb@cs.nott.ac.uk](mailto:ekb@cs.nott.ac.uk)

B. McCollum  
School of Computer Science, Queens University, Belfast  
University Road, N. Ireland BT7 1NN, UK  
e-mail: [b.mccollum@qub.ac.uk](mailto:b.mccollum@qub.ac.uk)

L.T.G. Merlot  
Quantm Ltd, 4/333 Flinders Lane, Melbourne 3000, Australia  
e-mail: [liam.merlot@quantm.net](mailto:liam.merlot@quantm.net)

S.Y. Lee  
Mechanical and Industrial Engineering, University of Toronto,  
5 King's College Road, Toronto, ON M5S 3G8, Canada  
e-mail: [lee@mie.utoronto.ca](mailto:lee@mie.utoronto.ca)

**Keywords** Examination timetabling · Benchmark problem datasets · Search methodologies

## 1 Introduction

Timetabling problems arise in various forms including educational timetabling (e.g. Burke et al. 2004d), nurse scheduling (e.g. Burke et al. 2004b), sports timetabling (e.g. Easton et al. 2004) and transportation timetabling (e.g. Kwan 2004). They have represented a challenging and important problem area for researchers across both Operational Research and Artificial Intelligence since the 1960s. Recent years have seen an increased level of research activity in this area. This is evidenced by (among other things) the emergence of a series of international conferences on the Practice and Theory on Automated Timetabling (PATAT)

(Burke and Carter 1998; Burke and De Causmaecker 2003; Burke and Erben 2001; Burke and Ross 1996; Burke and Rudova 2007; Burke and Trick 2005), and the establishment of a EURO (European Association of Operational Research Societies) working group on automated timetabling (see <http://www.asap.cs.nott.ac.uk/watt>).

Burke et al. (2004d) focused upon a definition of general timetabling, which covers many cases:

A timetabling problem is a problem with four parameters:  $T$ , a finite set of times;  $R$ , a finite set of resources;  $M$ , a finite set of meetings; and  $C$ , a finite set of constraints. The problem is to assign times and resources to the meetings so as to satisfy the constraints as far as possible.

Among the wide variety of timetabling problems, educational timetabling is one of the most widely studied. It is one of the most important and time-consuming tasks which occur periodically (i.e. annually, quarterly, etc.) in all academic institutions. The quality of the timetabling has a great impact on a broad range of different stakeholders including lecturers, students and administrators (see Romero 1982, Sabin and Winter 1986). Variants of educational timetabling include school timetabling (class-teacher scheduling), university course timetabling, exam timetabling, faculty timetabling and classroom assignment. It has been observed that course and exam timetabling are relatively close problems (Schaerf 1999) but very significant differences do exist (McCollum 2007). This survey will concentrate on examination timetabling.

An excellent survey of examination timetabling was published in Carter (1986) and an insightful *follow up* paper appeared in Carter and Laporte (1996). However, a significant number of research papers in the area have been published since 1996. This paper will concentrate upon the research that has appeared since the publication of (Carter and Laporte 1996). The last decade has seen the establishment of a collection of benchmark exam timetabling problems (Carter et al. 1996) which have been used by many of the examination timetabling research papers that have appeared since 1996. However, there has been some confusion in the literature caused by the existence of different benchmark problem datasets with the same names. This paper aims to eradicate such confusion by presenting a definitive re-naming of the sets and by clarifying the situation over which papers dealt with which problems.

### 1.1 Examination timetabling problems

Exam timetabling problems can be defined as assigning a set of exams into a limited number of timeslots (time periods) and rooms (of certain capacity), subject to a set of constraints. The complexities and the challenges presented by timetabling problems arise from the fact that a

large variety of constraints, some of which contradict each other, need to be satisfied in different institutions (Burke et al. 1996a, Carter et al. 1994). In the timetabling literature, constraints are usually categorised into two types: hard constraints and soft constraints, which are explained below:

- *Hard Constraints* cannot be violated under any circumstances (mainly due to physical restrictions). For example, *conflicting* exams (i.e. those which involve common resources such as students) cannot be scheduled simultaneously. A timetable which satisfies all of the hard constraints is usually said to be *feasible*.
- *Soft Constraints* are desirable but are not absolutely critical. In practice, it is usually impossible to find feasible solutions that satisfy all of the soft constraints. Soft constraints vary (and sometimes conflict with each other) from one institution to another in terms of both the type and their importance (Burke et al. 1996a). The most common soft constraint in the exam timetabling literature is to spread conflicting exams as much as possible throughout the examination session so that students can have enough revision time between exams. An example of another soft constraint which may conflict with this is to schedule all the large exams as early as possible to allow enough time for marking. The quality of timetables is usually measured by checking to what extent the soft constraints are violated in the solutions generated.

For a more formal definition of timetabling problems see Burke et al. (2004d), de Werra (1985) and Carter et al. (1996).

Due to the large variety of problems presented and investigated, it would be neither practical nor beneficial to present a comprehensive list of all the hard and soft constraints that occur in timetabling research. We list some of the key hard and soft constraints for exam timetabling in Tables 1 and 2, respectively. We believe that these cover most of the constraints that have appeared in the literature. It can be observed that they can be roughly grouped as time related (No. 1 in Table 1 and Nos. 1–7 in Table 2) or resource related (No. 2 in Table 1 and Nos. 8–11 in Table 2). Most of the survey papers reviewed in Sect. 1.2 present lists of constraints

**Table 1** Primary hard constraints in examination timetabling problems

Primary hard constraints	
1.	No exams with common resources (e.g. students) assigned simultaneously.
2.	Resources need to be sufficient (i.e. the number of students assigned to a room needs to be below the room capacity, enough rooms are needed for all of the exams).

**Table 2** Primary soft constraints in examination timetabling problems

Primary soft constraints	
1.	Spread conflicting exams as evenly as possible, or not in $x$ consecutive timeslots or days.
2.	Groups of exams required to take place at the same time, on the same day or at one location.
3.	Exams to be consecutive.
4.	Schedule all exams, or largest exams, as early as possible.
5.	An ordering (precedence) of exams needs to be satisfied.
6.	Limited number of students and/or exams in any timeslot.
7.	Time requirements (e.g. exams (not) to be in certain timeslots).
8.	Conflicting exams on the same day to be located nearby.
9.	Exams may be split over similar locations.
10.	Only exams of the same length can be allocated into the same room.
11.	Resource requirements (e.g. room facility).

in exam and general timetabling. The hard constraints listed in Table 1 and the first soft constraint in Table 2 are those that have been mostly covered by the research in the literature.

We will begin this critical review of the research area by overviewing a number of surveys that have appeared in the literature since the 1960s. Many of these papers cover educational timetabling in general and thus include discussions of examination timetabling in addition to other discussions.

## 1.2 Previous surveys on educational timetabling

An early survey by Miles (1975) provided a useful bibliography of early developments in computer aided timetabling. Another well-known early survey by Schmidt and Strohlein (1979), including more than 200 references, covered almost all the work on timetabling before 1979. de Werra (1985) introduced various mathematical (graph theoretical) models and briefly overviewed methods for class–teacher and course timetabling based on graph colouring and network flow methods. The author noted that exam timetabling and course scheduling were similar to each other although there were differences between them. In de Werra (1997) the same author introduced some requirements in timetabling into restricted graph colouring models and reviewed some mathematical programming formulations.

Carter (1986) presented a review of the early research on practical applications of examination timetabling in several universities. He reviewed a variety of graph heuristics and pointed out that none of the algorithms/packages had been implemented in more than one institution. There was no standard data on which comparisons could be carried out. Also, measures of a problem’s difficulty did not exist. Carter and Laporte (1996) updated the above survey to summarise

the algorithmic approaches from 1986 to 1996. The criteria for the discussion was that the method should be either tested on real data or implemented in real world applications. They categorised the methods into four types: cluster methods, sequential methods, generalised search (meta-heuristics) and constraint based techniques. They observed that the approaches implemented in practice were relatively simple variants of different methods and only addressed a subset of the constraints in the problems. The authors concluded by suggesting that timetabling researchers should report test results on benchmark problems to gain a better understanding of the various approaches taken in exam timetabling. As we will see later in this paper, this is what has happened since 1996.

Burke et al. constructed a questionnaire in 1996 on exam timetabling (Burke et al. 1996a) and sent it to 95 British universities, of which 56 replied. The issues concerned included:

- The structure of the problems (i.e. size, complexity and constraints, etc.),
- How the problems were solved, and
- The objective of the timetabling problem (i.e. what constitutes good solutions).

The resultant data was analysed to provide information on the constraints involved between exams, students, departments, timeslots and rooms. In addition to the 13 constraints originally listed in the questionnaire, another 19 constraints were provided by the universities, demonstrating that in reality there is a wide variety of requirements among different institutions. It was found that just 21% of the universities used some form of computational help. Where timetables were constructed manually, half of the institutions did not base their solution on the previous year’s timetable, requiring a workload of many months. The paper suggested some appropriate properties of automated timetabling systems that could be utilised in practice. This paper also provided some insight into the pertinent issues that impacted upon real world exam timetabling problems at the time. Burke et al. (1997) presented a brief introduction to automated university exam timetabling research. The paper concentrated on techniques for university timetabling which were popular at the time.

Bardadym (1996) considered different issues in computer-aided management systems for timetabling. He discussed problems, requirements, data representations and mathematical models. Solution methods from the 1960s to the 1990s were also overviewed mainly covering heuristics, meta-heuristics and algorithmic tools for integration in decision support systems. Meta-heuristics and interactive timetabling were seen as the new wave of computer-aided timetabling systems. Open issues for future timetabling research were also discussed.

Wren (1996) illustrated a useful and interesting link between scheduling, timetabling and rostering by studying an example of the Travelling Salesman Problem. He concluded that the similarity between timetabling and staff rostering may lead to successful cross-fertilisation on different types of problems. Indeed, recent research (as we shall see later) has provided some evidence to support this.

Schaerf (1999) in his survey, looked at the formulations of school, course and exam timetabling and declared that it is difficult to make a distinction between the latter two. Based on the definitions of variants of these problems, solution techniques particularly from artificial intelligence were classified and reviewed. Possible future directions were presented including specific techniques, standardisation, approximability, the design of a powerful constraint language, and the combination of, and comparisons against, different techniques.

Burke and Petrovic (2002), and in a follow up paper (Petrovic and Burke 2004), presented overviews of recent research conducted on university (course and exam) timetabling that had been carried out in their group including hybrid evolutionary algorithms, meta-heuristics, multi-criteria approaches, case-based reasoning techniques and adaptive approaches. An outline of research on sequential, clustering, constraint based techniques and meta-heuristic methods was also provided. Future directions highlighted knowledge based systems and approaches which aim to raise the generality of timetabling systems.

An article by Burke et al. (2004d) discussed the application of graph colouring methods to timetabling. The authors considered class–teacher, course, exam and sports timetabling. This paper highlights the role that graph colouring methods have played in the timetabling literature over the last 40 years or so. Indeed, it points out that graph colouring ideas are incorporated in several modern hybrid meta-heuristic techniques.

Reviews concerning specific techniques for timetabling have also appeared and this, to a certain extent, reflects the speed at which timetabling research developments have been made. Burke and Landa Silva (2004) reviewed memetic algorithm methods that have been proposed to solve scheduling and timetabling problems. A number of issues concerning the design of memetic algorithms were discussed. These include the domain knowledge that has been incorporated into the methodology to deal with infeasibility, local search design concerning intensification mechanisms and the balance between genetic search and local search. The paper also drew attention to the design of self-adaptive memetic algorithms as an area for future timetabling and scheduling research. Multi-objective meta-heuristic techniques have been reviewed by Landa Silva et al. (2004) for problems including educational timetabling. The paper covered recent multi-objective techniques including multi-phased approaches and multi-criteria evolutionary

techniques. Issues that were discussed include problem formulations, problem domain knowledge and strategies used in local search.

Although not specifically a survey paper, McCollum (2007) has discussed a number of papers in both course and exam timetabling with the aim of setting a research agenda to bridge the gap between timetabling research and practice. McCollum identified the development of timetables in institutions as a multiphase procedure and presented different real world modelling processes for exam and course timetabling. Of particular interest are the highlighted research challenges for both exam and course timetabling.

Schaerf and Di Gasparo (2007) raised some particularly interesting and important issues in timetabling research. They discussed measurability and reproducibility in university (course and exam) timetabling. In addition to highlighting the importance of these issues when conducting research, the authors discussed practices that can contribute to the improvement of these two aspects. Indeed, the international competition in 2002 has set a standard on course timetabling. Our observations on the benchmark exam timetabling below and the construction of the benchmark datasets at <http://www.asap.cs.nott.ac.uk/resources/data.shtml> highlight the importance of setting up standard scientific comparisons in timetabling research.

Lewis (2008) has recently surveyed metaheuristic algorithms applied to university (both course and exam) timetabling problems. He mainly concentrated on evolutionary algorithms, ant colony optimisation, tabu search, simulated annealing, and iterated local search. The literature was reviewed in three categories: one-stage optimisation algorithms, two-stage optimisation algorithms, and algorithms that allow relaxations. An analysis of the research was carried out, with an emphasis on the hard and soft constraints that were addressed. The author also highlighted the importance of benchmark data in timetabling research.

In addition to the surveys that have appeared over the years, an online bibliography was prepared by Kingston (1995) in 1995 and includes more than 1000 references on automated timetabling.

From the above brief discussion, we can see that there are a number of surveys in the literature concerning different issues that have impacted upon exam timetabling research. We also provide, in Appendix A, a list of PhD theses that have appeared during the years, where extensive reviews have been carried out upon specific aspects of course and exam timetabling. However, there is no comprehensive review which has dealt with the large body of exam timetabling research that has appeared in the last decade. This body of work includes a number of state-of-the-art approaches and has introduced a wide variety of diverse and successful methodologies. This paper aims to build on the survey paper of Carter and Laporte (1996)



to provide a modern discussion of the methods and techniques that have been developed for this important problem. With this in mind, we will not discuss in detail the work that appeared before 1996. We aim to keep our bibliography of examination timetabling papers and our classification tables (see later on) up to date on the following web page <http://www.asap.cs.nott.ac.uk/resources/data.shtml>. We would be very grateful if authors could contact us as new papers appear in order to regularly update this public resource. Although we have covered all the relevant papers of which we are aware, we may have inadvertently omitted relevant papers that have already appeared. If so, we apologise and would welcome the opportunity to add them to the information available at our web site.

The next sub-sections review the timetabling systems, languages and tools that were developed in the last decade. Furthermore, issues on timetabling models and complexity are also discussed. Section 2 will then review the timetabling approaches and methodologies that have appeared. We have organised it according to the different techniques that have been investigated. For each technique reviewed, a corresponding summarising table is presented in Appendix B. In Sect. 3, we summarise all the work on three sets of benchmark exam timetabling problems, and clarify some issues concerning the consistency of the benchmark dataset introduced by (Carter et al. 1996). Finally, we conclude the paper by summarising the research trends of the last ten years or so and presenting some future directions.

### 1.3 Timetabling systems

Over the years, a number of timetabling systems for both course and exam timetabling have appeared in the literature. However, many of them (especially before 1996) were specially developed for, and implemented at, particular institutions (Carter and Laporte 1996).

Hansen and Vidal (1995) presented a nationwide exam timetabling system which was reported to have been in use since 1992 to solve the problems of centralised planning of both oral and written examinations for 248 high schools in Denmark. The complex problem with a variety of objectives was described and solved by a four-phase process dealing with different objectives using different techniques and heuristics. Some issues including the preparation of data and the scheduling of exams were discussed. Experiences during the development of the system and other practical issues including the maintenance of centralised information and communications were also discussed (Hansen et al. 1995).

Lim et al. (2000) developed a timetabling system, which was a 3-tier client/server application, for both the course and exam timetabling problems at the National University

of Singapore. The problems and the overall manual process were described. In the exam portion of the system, exams were weighted by three measures and assigned into the timeslots one by one using constraint propagation by an arc consistency algorithm. The timetables were generated by the system in a much shorter time and compared favourably with the manually generated ones from the previously used procedure. Ho et al. (2001) further developed the system by using a Tabu Search algorithm, which employed four moving operators to improve the solutions obtained. The Push Forward Insertion heuristic, which has been employed in vehicle routing problems, was used to help spread the exams across timetables. Real problems were used to test the approach.

Dimopoulou and Miliotis (2001) developed a timetabling system to deal with both the course and exam timetabling problems at Athens University of Economics and Business. Firstly, an Integer Programming method was developed based on the MPCODE and XPRESS-MP packages. This approach was employed to assign groups of courses to groups of timeslots for the course timetabling problem. Based on the course timetables, the initial exam timetables were built and were adjusted repeatedly by a heuristic approach which dealt with a number of constraints. This provided good and feasible solutions with minimum effort.

### 1.4 Timetabling languages and tools

Over the years, timetabling researchers have employed some general packages, such as ECLiPse for constraint logic programming (Ajili and Wallace 2003), to build timetabling systems. However, some packages and languages which are specialised for timetabling have also appeared to support representations and comparisons in timetabling research. This is a goal which (if it could be achieved) would undoubtedly benefit timetabling research. However, it is true to say that the various attempts to suggest such standards have not yet been built upon and widely adopted by the community.

Burke et al. (1998a) presented general requirements (generality, completeness and practicability) for building a standard data format for general timetabling problems based on set theory and logic. Examples were given to show how common constraints were modelled by using this data format. The objective is to provide an open way of making comparisons on results and exchanging data in timetabling research.

Tsang et al. (1999) developed a language to specify constraint satisfaction problems so that constraint programming systems can be easily implemented for exam timetabling problems. The aim was to build a high level system which abstracted the details as much as possible so that end users,

without knowledge of both the constraint programming and host languages, can focus on the problem specific information. A real exam timetabling problem was used as an example to build the constraint satisfaction system.

Reis and Oliveira (2001) proposed a language, called UniLang, which used a list of synonyms to naturally represent data, various constraints, quality measures and solutions for general university timetabling problems. Eight classes of sub-problems in timetabling were defined and several examples were presented to interpret the language proposed. The language was converted into constraint logic programming in ECLiPse (Ajili and Wallace 2003) for different problems.

Di Gaspero and Schaerf (2003) introduced a software tool called EASYLOCAL++ for the implementation of a family of local search algorithms (Hill Climbing, Simulated Annealing and Tabu Search) on general timetabling problems. This represented an object-oriented framework which consisted of a hierarchy of abstract classes to take care of different aspects of local search. The main characteristics of the tool were reusability and generality, which were interpreted using examples from school timetabling, course timetabling and exam timetabling. It was employed to develop a family of Tabu Search methods in (Di Gaspero and Schaerf 2001).

De Causmaecker et al. (2002) discussed how the Semantic Web can be used in timetabling. They studied, layer by layer, how this technology can be applied to interpret problem specific knowledge in timetabling using XML. An upper level timetabling ontology was presented to demonstrate the ability to support the fast development of applications on different timetabling problems, whose constraints and resources can be easily identified.

Chand (2005) proposed a constraint based general model where constraints are grouped as domain, spread and CountResource, based upon which timetabling data and constraints can be transferred into a relational database. Examples of exam timetabling data by Burke et al. (1994) and course timetabling data by Goltz and Matzke (1999) were presented using the model proposed. The author declared that the format can be extended to include other constraints and can be applied to different languages. The author also provided a brief review of the relevant work on modelling timetabling data.

Ranson and Ahmadi (2007) briefly reviewed and analysed the limitations of the existing timetabling languages/standards. Based on the STTL language, which was developed for school timetabling in (Kingston 2001), they designed a flexible language-independent timetabling model incorporating the features of object-oriented programming and the UML language.

## 1.5 Models and complexity issues

Over the last ten years, important issues concerning the models and the complexity of timetabling problems have been discussed in the literature. However, so far, there are still no universally accepted complete models. Not much deployment work has been carried out on this topic. The main area of research on complexity issues has been on school timetabling (Schaerf 1999).

Cooper and Kingston (1996) represented timetabling problems using a timetabling specification language called TTL. They proved that timetabling problems are NP-complete in five independent ways which actually occur in practice. The prospects of overcoming the special cases in real problems were discussed.

de Werra et al. (2002) studied the complexity of some variants of class-teacher timetabling problems. A simple model of the problem was first given, followed by an extension where the classes were partitioned into groups. The authors showed that when there was a teacher giving lectures to three groups of classes, besides giving lectures to individual groups, the problem is NP-complete. A polynomial procedure to find a timetable (of a certain number of timeslots) based on network flows was given for the problems where there were at most two groups of classes.

## 2 Examination timetabling approaches/techniques

There has been a significant amount of exam timetabling research in the last ten years. We note that many of the successful methodologies that have appeared in the literature represent hybridisations of a number of techniques. Thus, the classification presented here, where possible, is by the main technique. However, several of the methodologies could have appeared in two or more of the classifications.

### 2.1 Graph based sequential techniques

The paper by Welsh and Powell (1967) represented a very important contribution to the timetabling literature. It built the bridge between graph colouring and timetabling, which led to a significant amount of later research on graph heuristics in timetabling (e.g. Mehta 1981, 1982). In exam timetabling problems, the exams can be represented by vertices in a graph, and the hard constraint between exams is represented by the edge between the vertices. The graph colouring problem of assigning colours to vertices, so that no adjacent vertices have the same colour, then corresponds to the problem of assigning timeslots to exams. Different soft constraints (such as those listed in Table 2) need to be considered separately and evaluated to represent a measure of solution quality.

**Table 3** Widely studied ordering strategies for examination timetabling problems

Heuristics	Ordering strategy
Saturation degree (Brelaz 1979)	increasingly by the number of timeslots available for the exam in the timetable at the time
Largest degree (Broder 1964)	decreasingly by the number of conflicts the exams have with the other exams
Largest weighted degree (Carter et al. 1996)	the same as Largest Degree but weighted by the number of students involved
Largest enrolment (Wood 1968)	decreasingly by the number of enrolments for the exam
Random ordering	randomly order the exams
Colour degree	decreasingly by the number of conflicts the exam has with those scheduled at the time

The basic graph colouring based timetabling heuristics are constructive methods that order the exams by how difficult they are to be scheduled. The exams are then assigned, one by one, into the timeslots. A broad range of ordering strategies and their modified variants appear in the timetabling literature (Carter 1986). We list, in Table 3, some of the widely employed ordering strategies, most of which are based upon graph colouring heuristics. A random ordering method has also been employed in the literature to introduce randomness in hybrid approaches and to provide comparisons.

Graph based heuristics which underpinned simple constructive methods played a very important role in the early days of timetabling research (Carter 1986). Although originally presented as techniques (albeit simple ones) in their own right, they are still being employed and adapted within hybridised methods in the current research literature. Their great strength is that they can provide reasonably good results within a small computational time and are very easy to implement. They are often used to construct initial solutions, or to build good portions of solutions before improvement techniques are applied (see more details in the following sections). A recent article by Burke et al. (2004d) overviewed graph colouring techniques for general timetabling.

In a particularly influential paper, Carter et al. (1996) studied the first five ordering strategies in Table 3 on real and randomly generated exam timetabling problems. Largest cliques, which are the largest sub-graphs where each of the vertices is adjacent to all of the others, were used to build initial solutions, based on which graph colouring heuristics and backtracking techniques were employed to construct the solutions. The idea is that the size of the largest clique determines the least number of timeslots required for the problem. The results indicated that none of the heuristics outperformed any of the others over all of the problems tested. Another important contribution from this work is the introduction of a set of 13 exam timetabling problems, which

became standard benchmarks in the field. They have been widely studied and used by different approaches during the years (see Table 6). We call this the University of Toronto data and discuss it further in Sect. 3.1. Carter and Johnson (2001) investigated sub-graphs which are sufficiently dense (*almost cliques*) on 11 of these instances. They observed that in real exam timetabling problems it was usually the case that there were many largest cliques, and that the *almost cliques* mechanism can potentially extend and improve the above approach.

Burke et al. (1998c) studied the effect of introducing a random element into the employment of graph heuristics (Saturation Degree, Colour Degree and Largest Degree in Table 3) by developing two variants of selection strategies: (1) tournament selection that randomly chooses one from a subset of the first exams in the ordered list; and (2) bias selection that takes the first exam from an ordered list of a subset of all of the exams. These simple techniques, when tested on three of the Toronto datasets, improved the pure graph heuristics with backtracking in terms of both the quality and diversity of the solutions.

Burke and Newall (2004) investigated an adaptive ordering strategy which dynamically ordered the exams during the problem solving in an iterative process. It was observed that a fixed pre-defined heuristic (employed as a measure of difficulty) in a traditional sequential strategy (as shown in Table 3) does not always perform well over the full range of problems. A heuristic modifier was designed to update the ordering of the exams according to the experience obtained with regard to the difficulty of assigning them in the previous iterations. Extensive experiments were carried out on 11 of the Toronto datasets, and another benchmark (which we call the Nottingham data, see Sect. 3.2). This approach was shown to be simple and effective (comparable with, and occasionally better than, state-of-the-art approaches). The method is not dependent on the initial ordering of exams.

Fuzzy logic was employed by Asmuni et al. (2005) to order the exams to be scheduled based on graph colouring heuristics on the Toronto datasets. The idea is that when ordering the exams by how difficult they are, fuzzy functions can be used to give an appropriate evaluation. It was seen that different fuzzy functions need to be used on different problems to obtain the best results. In (Asmuni et al. 2007a), a fuzzy system was developed to build a new evaluation function based on a series of rules to evaluate the quality of timetables where multiple criteria were involved. The approach was further improved by tuning the fuzzy rules and better results were obtained.

Corr et al. (2006) developed a neural network from which a measure of the difficulty of assigning exams during the timetable construction can be obtained by recursively inputting the updated solution construction states. The objective is to adaptively assign the most difficult exams at an early stage of solution construction. The neural network was trained by storing the states of timetable construction (feature vectors) using three graph heuristics. The work has demonstrated the feasibility of employing neural network based methods as an adaptive and generally applicable technique on timetabling problems.

Due to the limitations of constructive methods, where early assignments may lead to situations where no feasible timeslots are available for exams later on in the construction process, *backtracking* was studied (e.g. Carter et al. 1996). This process unassigns the early conflicting exams to allocate the current ones. A *look ahead* technique was also studied in (Burke and Newall 1999) to address this issue (for more details see the discussion on memetic algorithms in Sect. 4.2).

As mentioned earlier, techniques which hybridise graph heuristics with other methods are still appearing in the most modern exam timetabling research literature. The employment of graph heuristics within hyper-heuristics is discussed in Sect. 2.6.

## 2.2 Constraint based techniques

Constraint logic programming (Freuder 2005; Van Hentenryck 1989) and constraint satisfaction techniques (Brailsford et al. 1999) have their origins in Artificial Intelligence research. Such methods have attracted the attention of researchers in timetabling due to the ease and flexibility with which they can be employed for timetabling problems. Exams are modelled as variables with finite domains. Values within the domains (representing the timeslots and rooms) for the variables are assigned sequentially to construct the solutions for the problems. Early research focused on finding feasible solutions (i.e. satisfying all hard constraints). Brailsford et al. (1999) introduced various searching methods on constraint satisfaction problems and demonstrated

that this technique could be applied to optimisation problems.

Constraint satisfaction techniques are usually computationally expensive due to the fact that the number of possible assignments increases exponentially with the number of variables. They, on their own, cannot usually provide high quality solutions compared with the state-of-the-art approaches (Brailsford et al. 1999) on complex optimisation problems. Different heuristics and techniques are usually integrated to reduce the time complexity for solving practical problems (see Carter and Laporte 1996; White 2000). For example, the *labelling strategy*, where heuristics are usually employed, indicates the order in which the variables are to be initiated.

David (1998) applied constraint satisfaction techniques to model an exam timetabling problem in a French school, the Ecole des Mines de Nantes. Time complexity was crucial. Thus, partial solutions were first obtained. Based on these partial solutions, particular local repairing strategies were employed successively to obtain complete solutions and to make improvements. The approach was run several times with different initial assignments to reduce the chance of missing good solutions. It was employed successfully in the school and can usually generate solutions within one second.

Reis and Oliveira (1999) developed an examination timetabling system based on ECLiPSe (Ajili and Wallace 2003), which is a Prolog based platform for developing various extensions in constraint logic programming. A set of hard and soft constraints in the problem were built into a constraint satisfaction model, where set variables were employed and handled by the libraries in ECLiPSe. Its application on random data and a large real exam timetabling problem in the University of Fernando Pessoa in Porto demonstrated the efficiency of the model.

Merlot et al. (2003) employed constraint programming in a similar way to that of (Boizumault et al. 1996) using OPL (Van Hentenryck 1999), an optimisation programming language, to produce initial solutions. Then a Simulated Annealing and a hill climbing method (see Sect. 2.3 below) were used to improve the solutions. Variables (exams) were ordered by the sizes of their domains (available timeslots) and scheduled into the earliest timeslots one by one. The pure constraint programming obtained the best result for one of the Toronto datasets. The overall hybrid approach was tested on problems at the University of Melbourne, two variants of Toronto instances and the Nottingham data (see Sect. 3). This approach obtained the best results reported in the literature on several instances of the Toronto and Nottingham datasets at the time.

Duong and Lam (2004) also employed constraint programming to generate initial solutions for a Simulated Annealing methodology for the exam timetabling problems at



HoChiMinh City University of Technology. Backtracking and forward checking were employed to reduce the searching effort. The labelling strategy dynamically ordered the variables (exams) by a number of factors such as the size of the domain and the number of students.

Le Huédé et al. (2006) carried out an interesting piece of research on integrating multi-criteria optimisation in constraint programming. Different search strategies were defined for different criteria and were dynamically chosen during the search. Variables were ordered by using the largest degree heuristic, with labelling strategies defined for each criteria. Small instances of exam timetabling problems have been used to evaluate different strategies in constraint programming with a complete search. For large instances, a partial search employing iterative weakening methods was carried out to solve 6 modified Toronto problems (relaxing the number of timeslots).

In summary, recent research on constraint based techniques represents hybridisations with other techniques. The labelling strategy is usually integrated with different problem specific heuristics for variable ordering and is crucial to the success of the method. The development of some powerful constraint programming systems/languages (e.g. ECLIPSE (Ajili and Wallace 2003), CHIP (Simonis 1995), OPL (Van Hentenryck 1999)) significantly supported the construction of complete exam timetabling systems in real world applications. However, only particular problems at different institutions have been tackled with this approach in the literature. Also, the comparisons (with one exception—see below) have been against manually produced solutions. No comparisons have been made between constraint based techniques and other state-of-the-art approaches on the same problems, except for the paper by Merlot et al. (2003) which presents a method that is evaluated on the Toronto and Nottingham data. It is worth noting, though, that this method did produce the best results in the literature on some of these benchmark problems.

### 2.3 Local search based techniques

Local search based techniques (Pirlot 1996) (e.g. Tabu Search, Simulated Annealing and their variants) and Evolutionary Algorithms (see Sect. 2.4) are often classified as meta-heuristics (Burke and Kendall 2005; Glover and Kochenberger 2003). Local search methods are a family of general techniques which solve problems by searching from an incumbent solution to its *neighbourhood*. Different neighbourhood structures and moving operators within the search space distinguish different local search techniques. The search is guided by a defined objective function, which is used to evaluate the quality of the generated timetables.

These techniques represent a large part of the work that has appeared in the last decade. They have been applied on

a variety of timetabling problems, mainly because different constraints can be handled relatively easy. The performance and efficiency of these techniques are highly dependent upon the parameters and search space properties (e.g. connectivity, ruggedness).

Thus significant domain knowledge is usually built-in to deal with specific problems. A large number of variants and combinations have been investigated. We will first deal with Tabu Search.

#### 2.3.1 Tabu search

Tabu Search (Gendreau and Potvin 2005; Glover and Laguna 1993) explores the search space by not re-visiting a list of recent moves (kept in a *tabu list*). They may, however, be selected if they generated the best solution obtained so far by using an aspiration strategy. Otherwise, the search moves to other solutions even if they are worse than the incumbent solutions, with the aim of escaping from local optima. Parameters need to be fine-tuned in designing the approach and this is very much dependant on the problem in hand. Such parameters include the tabu list and the stopping criteria among others.

Di Gaspero and Schaerf (2001) carried out a valuable investigation on a family of Tabu Search based techniques whose neighbourhoods concerned those which contributed to the violations of hard or soft constraints. Exhaustive and biased selection strategies were also studied. The length of the tabu list is dynamic and the cost function is adaptively set during the search. The authors experimentally demonstrated that the adaptive cost function and the effective selection of neighbourhoods concerning the violations were key features of the approach. Di Gaspero (2002) improved the approach by employing multiple neighbourhoods based on a *token-ring* search which circularly employs *recolour* (change a single exam) and *shake* (swap groups of exams), followed by *kickers* (change a sequence of single exams) to further improve the solutions obtained. The technique extended the idea of diversifying the search from local optima.

White and Xie (2001) developed a four-stage Tabu Search called OTTABU, where solutions were gradually improved by considering more constraints at each stage, for the exam timetabling problem at the University of Ottawa. In addition to the recency short term memory, a frequency long term memory was also used to record the frequency of the most active moves in the search history. The size of the long term memory was set by analysing the number of less important exams in the problem. In (White et al. 2004) this approach was extended where both of the tabu lists could be dynamically relaxed (emptied) after a certain length of search time with no improvement. This approach compared favourably to those in (Carter et al. 1996) and (Di Gaspero and Schaerf 2001) on the Toronto data. The authors experimentally showed that employing long term memory can significantly improve Tabu Search on real-world problems.

Paquete and Stützle (2002) developed a Tabu Search methodology for exam timetabling where ordered priorities were given for the constraints. The constraints were considered in two ways: (1) one constraint at a time from the highest priority, where ties were broken by considering the lower priority constraints; (2) all the constraints at a time, starting from the highest priority. The second strategy obtained better results, while the first strategy was more consistent. The length of the tabu list was adaptively set by considering the number of violations in the solutions. It was observed that the length of the tabu list needed to be increased with the size of the problems.

### 2.3.2 Simulated annealing

Simulated Annealing (Aarts et al. 2005) is motivated by the natural annealing process (Aarts and Korst 1989). The idea is to search a wider area of the search space at the beginning of the process by accepting worse moves with a higher probability, which is gradually decreased as the search continues. A *temperature* is used within a *cooling schedule* to control the probability of the acceptance of worse moves in the search. Many parameters need to be tuned in Simulated Annealing including the initial and final temperatures, and the cooling factor in the cooling schedule. These parameters affect the performance and success of this approach.

Thompson and Dowsland (1998) carried out valuable work to develop a two-stage approach where feasible solutions from the first stage are fed into a Simulated Annealing process in the second stage to improve soft constraint satisfaction. As different objectives were dealt with in different stages in turn, the solutions from the early stage may be poor, and thus a backtracking technique was proposed. Dowsland (1996) also observed that the way in which the neighbourhood was defined, the importance of objectives and the difficulty of objectives significantly affected the process. Based on their work in (Thompson and Dowsland 1996), the authors further investigated the Kempe chain neighbourhood, where chains of exams rather than individual exams were moved. This gave more flexibility to enable the movement of large difficult exams within the timetable. They concluded that the most important factors in Simulated Annealing were the cooling schedule and the way neighbourhoods were defined and sampled. The authors reported that the developed exam timetabling system had been used successfully in Swansea University since 1993.

Bullnheimer (1998) discussed how a model for Quadratic Assignment Problems was adapted to formulate a small scale practical exam timetabling problem at the University of Magdeburg. The models enabled the university administrators to control how much the conflicting exams needed to be spaced out. Simulated Annealing was employed where two sets of neighbourhood structures (moving the timeslots

of exams and moving single exams) were studied. However, the details of the parameters in the algorithm were not given.

Merlot et al. (2003) employed a Simulated Annealing approach initialised by constraint programming techniques (see Sect. 2.2 on “Constraint Based Techniques”) and followed by hill climbing to further improve the solution. A modified Kempe chain neighbourhood was employed. The best results at the time for several of the benchmark instances were achieved by this hybrid approach. Indeed, the method still has some of the best known results. The authors suggested that methods combining solution construction with local search will dominate the future of exam timetabling research.

Duong and Lam (2004) employed Simulated Annealing on the initial solutions generated by constraint programming for the exam timetabling problem at HMC University of Technology. A Kempe Chain neighbourhood was employed in the Simulated Annealing, whose cooling schedule was experimentally set using mechanisms and algorithms. The authors noted that when limited time is given, it is crucial to tune the components in Simulated Annealing to the specific problems to be solved.

Burke et al. (2004a) studied a variant of Simulated Annealing, called the Great Deluge algorithm (Dueck 1993). The search accepts worse moves as long as the decrease in the quality is below a certain *level*, which is originally set as the quality of the initial solution and gradually lowered by a decay factor. The decay factor and an estimate of desired quality represent the parameters in this approach. The authors noted that such parameters can be set by users, who are usually not experts on meta-heuristics. The initial solutions, however, need to be feasible to calculate the decay factor so a Saturation Degree was run a number of times, from which the best solutions were employed as the starting points. This Great Deluge approach was superior to a Simulated Annealing method. It was shown to be effective and generated some of the best results on the Toronto and Nottingham datasets when compared with other approaches. Comprehensive experiments were also carried out to analyse the trade-off between the time and solution quality on problems of different size. The approach was further studied in (Burke and Newall 2003) where it was initialised by the adaptive ordering method in (Burke and Newall 2004).

### 2.3.3 Other local search based techniques

Recently, along with the study of different ways of escaping from local optima in local search based techniques, some researchers turned to investigating the effect of designing different neighbourhoods and have obtained some success on timetabling problems. This has demonstrated that not only the method of search, but also the structure of the neighbourhood can have a significant impact on the searching al-

gorithms. For example, Kempe chain neighbourhood structures as mentioned above were investigated by a number of researchers in exam timetabling (see Casey and Thompson 2003; Côté et al. 2005; Merlot et al. 2003). The idea is that chains of conflicting exams are swapped between timeslots. Reasons for why this neighbourhood structure worked well were analysed (Thompson and Dowsland 1998). Other approaches concerned multiple neighbourhood structures (Di Gaspero 2002). Compared with standard moves on single exams, this brought more flexibility in the navigation of search spaces for different problems.

Abdullah et al. (2007a) developed a large neighbourhood search based on the methodology of improvement graph construction originally developed by Ahuja et al. (2001) for different optimisation problems. Instead of just considering traditional pair-wise exchange based operators, a tree-based neighbourhood structure was designed to carry out cyclic exchanges among all of the timeslots. The approach has provided the best results on a number of Toronto dataset problems at the time of publication. However, a large amount of computational time was needed. It was further developed in (Abdullah et al. 2007b) where the improvement moves were kept in a tabu list for capacitated exam timetabling problems (Toronto *c* in Sect. 3.2).

Another technique concerning different neighbourhoods is Variable Neighbourhood Search (e.g. Hansen and Mladenović 2001; Mladenović and Hansen 1997). This approach systematically varies a number of neighbourhood structures. The aim is to escape from local optima by switching from the search space defined by one neighbourhood to another. However, not much work has been done in exam timetabling using this approach. Burke et al. (2006a) investigated variants of Variable Neighbourhood Search and obtained the best results in the literature across some of the problems in the Toronto datasets. The results were further improved by using a standard Genetic Algorithm to intelligently select a subset of neighbourhoods. The latter approach has a strong link to the work in hyper-heuristics and indicated promising directions on developing more general approaches on neighbourhoods rather than directly on solutions. In hyper-heuristics, Variable Neighbourhood Search was also employed where graph heuristics rather than neighbourhoods were searched (Qu and Burke 2008). We present more details in Sect. 2.6 on “Hyper-heuristics”.

In addition to designing different neighbourhood structures within local search based techniques, some researchers have also looked into how iterative techniques can help in solving complex problems. In Iterated Local Search (Lourenço et al. 2003), the search restarts after a certain criterion is met. The motivation is to explore more areas of the search space within a short time. It was first applied to the graph colouring problem (Paquete and Stützle 2002).

Caramia et al. (2001, 2008) developed a fine-tuned local search method where a *greedy scheduler* assigned exams

into the least possible number of timeslots and a *penalty decreaser* improved the timetable without increasing the number of timeslots. When no improvement could be made, the number of timeslots was increased gradually by a *penalty trader*. The process was repeated, employing a permutation technique to reassign the priorities of exams. This approach still holds the best results reported in the literature on several instances of the Toronto datasets.

Casey and Thompson (2003) investigated a Greedy Randomised Adaptive Search Procedures (GRASP) approach (Resende and Ribeiro 2003), which is a relatively new technique in timetabling. In GRASP, a local search algorithm is started iteratively after local optima are reached based on the initial solutions generated by a greedy approach. In (Casey and Thompson 2003), the initial solution in each iteration was generated by a modified Saturation Degree, where one exam from the first  $n$  (experimentally set as 2–6) exams ordered was assigned to the timetable. Backtracking was employed in conjunction with a tabu list to forbid indefinite cycles. A limited form of Simulated Annealing with high starting temperature and fast cooling was used in the improvement phase. Kempe chain moves were employed on exams that particularly contributed to the cost. The approaches applied to the Toronto datasets reported competitive results on some of the instances at the time.

In summary, during the last decade, local search based techniques have been very heavily studied and have obtained a marked level of success in timetabling. All of the work discussed above was either tested on benchmark data or implemented in real applications. Different ways of accepting the moves (i.e. moving strategies, acceptance strategies and selection strategies) were studied to enable escape from local optima. However, one significant drawback of these approaches is the effort required to tune the parameters for specific problems to obtain high quality solutions.

## 2.4 Population based algorithms

### 2.4.1 Evolutionary algorithms

Genetic Algorithms have been the most studied Evolutionary Algorithms in terms of exam timetabling research. In particular, hybridisations of genetic algorithms with local search methods (sometimes called memetic algorithms) have led to some success in the field.

Genetic Algorithms represent an analogy with the evolutionary process in nature by manipulating and evolving populations of solutions within the search space (see Glover and Kochenberger 2003; Reeves 1993; Sastry et al. 2005). Solutions are coded as *chromosomes* and are evolved by a reproduction process using crossover and mutation operators, with the aim of getting better and better solutions through a series of generations. Parameters and operators in Ge-

netic Algorithms need to be defined and set properly, making the approach (usually) more complicated than that of local search based searching methods. The strategy in Genetic Algorithms is fundamentally different from the local search based approaches discussed above in the sense that several solutions (a population of solutions) are dealt with at once rather than just one solution being improved through a series of iterations.

Corne et al. (1994) provided a brief survey on using Genetic Algorithms in general educational timetabling and addressed some issues and future prospects. One contribution of the work showed that the use of direct representation in Genetic Algorithms was incapable of dealing with certain problem structures in some specially generated graph colouring problems. Ross et al. (2003) updated the above work on evolutionary representations and algorithms used on various kinds of timetabling problems.

Ross et al. (1996) showed that transition regions exist in solvable timetabling problems by experimenting upon specially generated graph colouring problems of different connectivity and homogeneity. The authors indicated that the study can assist the understanding of how different algorithms perform on complex timetabling problems. Ross et al. (1998) provided further evidence for the weakness of the use of direct coding in Genetic Algorithms. They observed the failure of a number of (evolutionary and non-evolutionary) approaches to solve special classes of graph colouring problems and suggested that Genetic Algorithms should search for algorithms rather than actual solutions. Indeed hyper-heuristics (where a set of low level heuristics is searched by a high level algorithm; see more details in Sect. 2.6) do exactly this.

Terashima-Marin et al. (1999b) designed a clique-based crossover operator on timetabling problems that was transferred into graph colouring problems. Different recombination strategies were tested in the reproduction processes to maintain the cliques in parents into their offspring. They pointed out the same problem with direct representation in Genetic Algorithms as discussed above in (Ross et al. 1998). They suggested alternatives for future work. In (Terashima-Marin et al. 1999c), they also studied the penalty function on both random and real timetabling problems employing Hardness Theory, which predicts where the hardest instances are within timetabling problems. However, they observed that adding this measure is not helpful in guiding Genetic Algorithms toward promising areas of the search space. Based on the above work, they investigated the non-direct coding in Genetic Algorithms (Terashima-Marin et al. 1999a), where solution construction strategies and heuristics rather than the actual solutions were coded (e.g. configurations of constraint satisfaction methods, ordering of nodes being assigned and heuristics dealing with constraints). Promising results obtained by this approach on

the Toronto datasets indicated the potential of non-direct representations in Genetic Algorithms.

Erben (2001) developed a grouping Genetic Algorithm where appropriate encoding, specially designed crossover and mutation operators, and fitness functions were studied. Genes were grouped for each colour in graph colouring problems (which model the exam timetabling problems with only hard constraints). Although the results were not competing with the best, the approach requires less computational time than some of the methods in the literature. The paper also studied, from a different aspect, the important issue of representations in designing Genetic Algorithms.

Sheibani (2002) built a special mathematical model and developed a standard Genetic Algorithm for solving exam timetabling problems in training centres with the objective of maximising the intervals between the exams. An activity-on-arrow network was employed to estimate the closeness between exams, which was used in the fitness function in the Genetic Algorithm.

Wong et al. (2002) discussed some issues concerning their implementation of a Genetic Algorithm for solving an exam timetabling problem at Ecole de Technologie Supérieure, which was modelled as a Constraint Satisfaction problem. Tournament selection was used to choose parents, and repairing strategies were incorporated with mutation to produce better candidates. Côté et al. (2005) investigated a bi-objective evolutionary algorithm with the objectives of minimising timetable length and spacing out conflicting exams. Two local search operators (a classic Tabu Search and a simplified Variable Neighbourhood Descent), instead of recombination operators, were employed to deal with hard and soft constraints. The approach obtained competitive results on a number of benchmark problems against some of the methods in the literature (e.g. Carter et al. 1996; Burke et al. 1996b; Merlot et al. 2003). The paper also provided a review on state-of-the-art approaches on the Toronto datasets at the time.

Ulker et al. (2007) developed a Genetic Algorithm where Linear Linkage Encoding was used as the representation method. Different crossover operators were investigated in conjunction with this representation on benchmark graph colouring and exam timetabling problems with hard constraints (Toronto variant *a*). Promising results indicated that this encoding with appropriate genetic operators was a viable search methodology.

#### 2.4.2 Memetic algorithms

Memetic Algorithms (Moscato and Norman 1992) are an extension of Genetic Algorithms where the basic idea is that individuals in a population can be improved during their lifetime (i.e. within a generation). This is often implemented by employing local search methods (in the form of hill climbing or repairing strategies, etc.) on individual members of



a population between generations. Burke and Landa Silva (2004) discussed a number of issues concerning the design of memetic algorithms for scheduling and timetabling problems. Recent research ideas and future directions on this topic were presented.

The ability to explore the search space by employing a population based method and exploit a region of it by using a local search enables such methods to deal effectively with large complex problems. However, there is usually a price to pay in terms of the computational time required. Also the right balance between *exploration* and *exploitation* needs to be established (Burke and Landa Silva 2004). There exists a number of in-depth studies on memetic algorithms concerning structures of the search space and different ways of hybridising over a range of combinatorial optimisation problems (see Krasnogor and Smith 2005; Osman and Laporte 1996).

Burke et al. (1996b) developed a memetic algorithm where light and heavy mutation operators were employed to reassign single exams and sets of exams, respectively. Neither of these mutations on their own provided substantial improvement on the solution quality. Hill climbing was used to improve the individuals and to improve the quality of timetables although a larger amount of computational time was required. Another contribution of this paper was the introduction of a new set of benchmark exam timetabling problems (named as the Nottingham data and described in Sect. 3.2). These have been widely used by a number of researchers in later work (see Tables 8 and 14). The same authors also investigated the effects of diversity in initial populations in memetic algorithms (Burke et al. 1998b). To generate a high level of diversity in the initial population, randomness was introduced by using different selection strategies on graph heuristics (see Burke et al. 1998c). Three diversity measures were also developed to study the trade-off between the quality and diversity. It was shown that the study of diversity in initialisation offered great potential benefits for memetic algorithms. Burke and Newall (1999) presented a heuristic methodology for decomposing an exam timetabling problem into a series of sub-problems. They used the memetic algorithm of (Burke et al. 1996b) to address the sub-problems. This approach is discussed in more detail in the section on decomposition (Sect. 2.7).

#### 2.4.3 Ant algorithms

Ant Algorithms (Dorigo and Blum 2005; Merkle and Middendorf 2005) belong to the family of population based techniques. They simulate the way ants search for the shortest route to food by laying pheromone on the way. The shortest trails generate stronger levels of pheromone over a period of time. In the algorithm, each ant is used to construct a solution and information gained during the search is maintained

as pheromone, which is used to help generate solutions in the next stage. In exam timetabling, ant algorithms represent relatively recently explored techniques and have not been particularly widely studied. Relevant work does, however, exist on graph colouring problems (Costa and Hertz 1997) where the frequency of the colours assigned for the vertices in the solution construction were employed as the pheromone.

Naji Azimi (2004) implemented an Ant Colony System and compared it with Simulated Annealing, Tabu Search and a Genetic Algorithm under a unified framework for solving systematically designed exam timetabling problems. Initial solutions for the Ant Colony System were generated heuristically and improved by local search afterwards. The results analysed over the running time indicated that the Ant Colony approach performed the best (although not on all of the problems) and Tabu Search had the highest level of improvement upon the randomly generated initial solution. Three variants of hybridisation on Tabu Search and the Ant Colony method were then studied in (Naji Azimi 2005). It was observed that the hybrid approaches work better than each single algorithm, and the sequential Ant Colony System followed by Tabu Search obtained the best results. However, only randomly generated data was used to test these algorithms.

Dowsland and Thompson (2005) developed Ant Algorithms based on the graph colouring model studied in (Costa and Hertz 1997) for solving the Toronto *a* version of the exam timetabling problem without soft constraints (i.e. to find the lowest number of timeslots). Extensive experiments were carried out to measure the performance of the algorithm with different configurations. These include the initialisation methods (i.e. recursive Largest Degree and Saturation Degree), trail calculations, three variants of fitness functions and different parameter settings. The results obtained were competitive to the others on the same dataset. It was also observed that the initialisation methods had significant influence on the solution quality. Extensions of the algorithm to incorporate other constraints (i.e. time windows, seating capacities and second-order conflicts) were also discussed.

Eley (2007) compared two modified ant algorithms based on the Max-Min ant system for course timetabling (Socha et al. 2003), and the ANTCOL algorithm for graph colouring problems (Costa and Hertz 1997). It was observed that the simple ant system ANTCOL outperformed the Max-Min ant system when both algorithms were hybridised with a hill climber. The author also concluded that adjusting parameters can considerably improve the performance of ant systems.

#### 2.4.4 Artificial immune algorithms

Malim et al. (2006) studied three variants of Artificial Immune systems (a Clonal Selection Algorithm, an Immune

Network Algorithm and a Negative Selection Algorithm) and showed that the algorithms can be tailored for both course and exam timetabling problems. However, there was a problem with the results presented in this paper. They were (after publication) found to represent an error in the code and, as such, are invalid.

In summary, evolutionary methods (particularly evolutionary hybrids) have been very effective in providing high quality solutions to exam timetabling problems. Recent research has discussed the issues of encoding to deal with the problem structures that direct coding is not capable of dealing with. This opened up a new research direction in Evolutionary Algorithms and has led to some of the initial work in Hyper-heuristics (see Sect. 2.6). Multi-criteria techniques also form an important research direction in the area of Evolutionary Algorithms for exam timetabling problems. More details are discussed in the next section.

Ant Algorithms (see Dowsland and Thompson 2005; Naji Azimi 2004) and Artificial Immune Algorithms (Malim et al. 2006) have been applied in exam timetabling with some initial observations. As relatively new techniques, they represent some potential and should attract more attention in the exam timetabling domain.

## 2.5 Multi-criteria techniques

In the majority of algorithms/approaches on timetabling, weighted costs of violations of different constraints are summed and used to indicate the quality of the solutions. However, in real world circumstances, the constraints are often considered from different points of view by the different parties involved in the timetabling process (Carter and Laporte 1996). The simple sum of costs on different constraints cannot always take care of the situation in such cases. Multi-criteria techniques have been studied recently in timetabling with the aim of handling different constraints easily by considering a vector of constraints instead of a single weighted sum. In multi-criteria techniques, each criterion can be considered to correspond to a constraint, which has a certain level of importance and is dealt with individually. In some approaches, multiple stages have been employed to deal with different objectives. Landa Silva et al. (2004) provide a review of a large number of scheduling and timetabling applications which employ multi-criteria techniques.

Colijn and Layfield (1995a) applied a multi-stage approach for the exam timetabling problem in the University of Calgary. In the first stage, individual exams and whole sets of exams in timeslots were moved to reduce the number of students who were sitting two exams in a row. In the second stage, students taking three and four exams in a row were considered using a similar process. The authors also considered the cases where timetables have to be modified

in unforeseen circumstances (Colijn and Layfield 1995b) in the second stage of the approach, which was a highly interactive process within a visual interface where exams can be moved, added or removed.

Burke et al. (2001) developed a two-stage multi-criteria approach dealing with nine criteria in exam timetabling problems (e.g. room capacity, proximity of exams, time and order of exams, etc.). In the first stage, Saturation Degree was used to generate a set of feasible solutions, where each criterion was dealt with individually. The second stage then heuristically improved these solutions simultaneously. A multi-criteria method called Compromise Programming (Zeleny 1974) was used where the quality of the solutions was evaluated by the distance between them to an ideal point representing optimal solutions concerning all criteria. This technique was further studied in (Petrovic and Bykov 2003) and integrated with the Great Deluge algorithm (Burke et al. 2004a). A reference point provided by users was used to draw a trajectory in the criteria space. The criteria weights can be dynamically changed to guide the search, starting from random points, towards the reference point. It aims at the ideal point in the criteria space. However, the initial weights needed to be set were dependant on the problems. Also, the search was not guaranteed to converge. Published results from (Burke et al. 1996b) were used as the reference points of the approach and the final results were better on some of the benchmark problems tested. These approaches provided the flexibility for timetablers to obtain desired solutions by managing the weights of different constraints.

As mentioned above, Le Huédé et al. (2006) integrated multi-criteria optimisation in constraint programming to solve exam timetabling problems. A criteria choice heuristic was used to dynamically choose a criterion to improve the last solution found with respect to specific criteria. A fuzzy measure was used to model the importance of each criteria in an aggregate function upon three criteria, namely: the length of the timetable, the number of rooms used and the spreading of the exams. Experimental results on small problems and a selection of modified Toronto datasets indicated that the proposed approach has better control of the search by selecting appropriate parts of the search space.

## 2.6 Hyper-heuristics

Many of the techniques in the literature are dependant upon how domain knowledge (i.e. the hard coding of hard and soft constraints) has been embedded into the technique. This impacts upon meta-heuristic development for examination timetabling. Some of the most effective techniques on the benchmark datasets in the literature are meta-heuristics. However, most of these methods represent a tailor made approach for one particular problem (in this case, exam timetabling). Such methods usually work poorly (or are not

capable of dealing) with other problems. Indeed, it can be the case that such methods do not work consistently across other exam timetabling problem instances. Often, parameter tuning can play a significant role. The effort of tuning parameters to fit new problems can be thought of as being as difficult as that of developing new approaches. Such issues have led a number of researchers to develop new technologies which are aimed at operating at a higher level of generality.

Hyper-heuristics are motivated by such observations and are attracting an increased level of research attention. The term can be seen as representing *heuristics that choose heuristics*, i.e. a search space of heuristics is the focus of attention rather than a search space of solutions (as is the case with most implementations of meta-heuristics, Burke et al. 2003a; Ross 2005). The aim is to develop more *general* approaches rather than to beat the fine-tuned and problem specific approaches which often require much effort on the tuning of parameters and are usually only appropriate for specific problems.

As mentioned above in Sect. 2.4.1, Ross et al. (1998) suggested that a Genetic Algorithm might be successfully employed in searching for good algorithms rather than specific solutions. Terashima-Marin et al. (1999a) investigated using Evolutionary Algorithms to search for solution construction strategies.

Ahmadi et al. (2003) developed a Variable Neighbourhood Search to find good combinations of parameterised heuristics for different exam timetabling problems. Permutations of the low level heuristics (i.e. seven exam selection, two timeslot selection and three room selection heuristics) and their associated parameters (weights) were employed to construct solutions.

Ross et al. (2004) developed a general steady state Genetic Algorithm to search within a simplified search space of problem-state descriptions to construct solutions. The search of the Genetic Algorithm was on heuristics rather than actual solutions. Three different fitness functions were tested. The descriptions of the problem state (corresponding to heuristics) were experimentally studied with respect to these fitness functions. Promising results for both the benchmark course and exam timetabling problems demonstrated potential research directions pertaining to this approach for a range of problems.

Kendall and Hussin (2005a, 2005b) investigated a Tabu Search hyper-heuristic based on the work in (Burke et al. 2003b) where both moving strategies and constructive graph heuristics were employed as low level heuristics. The algorithms were tested on exam timetabling problems from the MARA University of Technology (Hussin 2005) and it was shown that it produced better results compared with solutions that were generated manually.

Burke et al. (2005, 2006b) investigated employing Case-Based Reasoning (see Leake 1996), a knowledge based

technique, as a heuristic selector for solving both course and exam timetabling problems. In (Burke et al. 2006b), knowledge discovery techniques were employed to discover the most relevant features used in evaluating the similarity between problem solving situations. The objective was to choose the best heuristics from the most similar previous problem solving situation to construct good solutions for the problem in hand. The issue of defining the similarity between exam timetabling problems has also been studied in (Burke et al. 2004c) in terms of choosing the best problem solving method. In (Burke et al. 2005), different ways of hybridising the low level graph heuristics (with and without CBR) were compared for solving the Toronto datasets. It was shown that employing knowledge based techniques rather than randomly/systematically hybridising heuristics in a hyper-heuristic framework presented good results. Yang and Petrovic (2005) employed Case-Based Reasoning to choose graph heuristics to construct initial solutions for the Great Deluge algorithm and obtained the best results reported in the literature for several Toronto instances at the time. Attribute graphs studied in (Burke et al. 2000) were employed to model the constraints in the problems so that previous problems with similar constraints were retrieved to solve the problems in hand by reusing the most appropriate graph heuristics.

Burke et al. (2007) investigated using Tabu Search to find sequences of graph heuristics to construct solutions for timetabling problems. A different number of low level graph heuristics were studied in this graph based hyper-heuristic to adaptively assign the most difficult exams at different stages of solution construction. It was observed that the greater the number of intelligent low level heuristics, the better the performance may be. However, the size of the search space will grow significantly. Thus, the computational time may be an issue. The results on both course and exam timetabling problems were competitive with the best state-of-the-art approaches reported in the literature and demonstrated the simplicity and efficiency of this general approach. Qu and Burke (2008) further investigated the effect of employing different high level search algorithms (i.e. Steepest Descent, Tabu Search, Iterated Local Search and Variable Neighbourhood Search) in the unified graph based hyper-heuristic framework for exam timetabling. Experimental results demonstrated that the hyper-heuristic method employed upon the search space of graph heuristics was not crucial. The characteristics of the neighbourhood structures and search space were analysed. It was shown that the exploration over the large solution space enabled the approach to obtain good results on both the exam and course timetabling problems.

Bilgin et al. (2007) analysed 7 heuristic selection methods and 5 acceptance criteria within a hyper-heuristic by conducting an empirical study on both benchmark functions

and exam timetabling problems. They concluded that different combinations of selection methods and acceptance criteria worked well on different problems, although some combinations worked slightly better than others on the instances tested.

Ersoy et al. (2007) studied hyper-heuristic approaches where three hill climbers were applied in different orderings within a memetic algorithm. During the memetic algorithm, individuals were evaluated to keep track of the violations of each constraint type in the benchmark Toronto dataset. The approaches were compared with self-adaptive memetic algorithm hyper-heuristics with different heuristic selection and acceptance criteria. It was shown that the memetic algorithm hyper-heuristic with a single hill climber at a time performed the best among all the approaches tested.

In summary, various strategies and methodologies have been employed as the high level selection methods in a hyper-heuristic framework to choose appropriate low level heuristics. These low level heuristics might be either construction or improvement heuristics. Such methods are laying the foundations of methodologies to automatically design and adapt timetabling heuristics. This has led to some work on analysing the search space of heuristics (rather than the search space of solutions) with the goal of fundamentally understanding the search processes which underpin this new perspective on timetabling research (Qu and Burke 2008).

## 2.7 Decomposition/clustering techniques

The idea of decomposition is that large problems are broken into small sub-problems, for which optimal or high quality solutions can be obtained by relatively simple techniques as the search spaces of the sub-problems are significantly smaller than that of the original problem (Carter 1983). Although it has had some success (Burke and Newall 1999), decomposition in timetabling has not attracted as much attention as might be expected because of two drawbacks. Firstly, early assignments may lead to later infeasibility, which was also a problem encountered in constructive methods in the early days of timetabling research. Secondly, globally high quality solutions may be missed as certain soft constraints cannot be evaluated when the problems are decomposed. The clustering methods studied in early timetabling research (see Carter and Laporte 1996) can be seen as decomposition approaches in the sense that the exams are decomposed into conflict-free or low-conflict groups. Another way of decomposing the problems is by finding the largest clique in underlying graphs. This was studied by Carter et al. (1994) and employed in their later work (Carter et al. 1996). Carter and Johnson (2001) improved the approach by assigning the exams in all of the *almost-cliques* as they potentially represent the most difficult exams.

Burke and Newall (1999) investigated a decomposition approach by using sequential heuristics to assign the first set of  $n$  exams which were measured as the most difficult ones by graph colouring heuristics (i.e. Colour Degree, Largest Degree, Saturation Degree—see Table 3). Backtracking and look-ahead techniques were employed to avoid making early assignments which lead to later infeasibilities. The exams assigned in previous stages were fixed and the sub-problem at each stage was solved by the Memetic Algorithm developed in (Burke et al. 1996b). The algorithm dramatically reduced the time required and also produced high quality solutions on the Toronto and Nottingham data. At the time of publication, this paper had some of the best results on the capacitated benchmark problems (Toronto  $c$  in Sect. 3.2). The decomposition technique was actually independent of the memetic timetabling algorithm which was used on each of the decomposed subsets.

Lin (2002) developed a multi-agent algorithm where problems were divided into sub-problems and solved by each agent locally. A broker was used to solve the remaining schedules including those that were de-allocated from local schedules. The global solutions were obtained by aggregating all the schedules generated by the agents and the broker. Both the Toronto data and randomly generated exam timetabling problems were tested and compared with the method of (Ross et al. 1998). The approach worked well on sparsely scheduled problems but less well on dense problems.

Qu and Burke (2007) investigated an approach where exams were adaptively decomposed into two sets (*difficult* and *easy*) by how difficult they were to schedule in previous iterations of the solution construction. The complexity of the problem is thus reduced as two smaller search spaces were concerned while the overall quality of the timetables is also considered. The small portion of difficult exams obtained by the approach were found to make a significant contribution to the costs of the timetables generated. The approach obtained the best result on one of the problem instances at the time of publication (see Toronto  $c$  in Sect. 3.2).

## 3 Examination timetabling benchmark data

The high level of research interest in examination timetabling has led to the establishment of a variety of different benchmark problems which have been widely studied. The established benchmarks, with variants of standard defined measures, have provided a mechanism to make meaningful scientific comparisons and to facilitate the exchange of research achievements. However, there has been some confusion in the literature due to the circulation of two different versions of five of these benchmark problems (from the University of Toronto datasets). One of the goals of this paper



is to eradicate this confusion by establishing new names for each of the different versions. This, of course, means that we actually have 18 problems that have been studied in the literature (rather than 13). Another aim of this section of the paper is to summarise which of the methods that have appeared in the literature are the best on these benchmarks. This is particularly important given the confusion mentioned above.

### 3.1 University of Toronto benchmark data

Carter et al. (1996) introduced a set of 13 real-world exam timetabling problems from three Canadian high schools, five Canadian universities, one American university, one British university and one university in Saudi Arabia. Over the years they were widely employed as testbeds in exam timetabling research. As mentioned above, there has been an issue concerning the circulation of different sets under the same name. This is discussed at length below.

In the problem, to indicate the density of the conflicting exams in each of the instances, a *Conflict Matrix*  $C$  was defined where each element  $c_{ij} = 1$  if exam  $i$  conflicts with exam  $j$  (i.e. they have common students), or  $c_{ij} = 0$  otherwise. The *Conflict Density* represents the ratio between the number of elements of value “1” to the total number of elements in the conflict matrix.

Two variants of objectives were defined in the original dataset:

- to minimise the number of timeslots needed for the problem (graph colouring) (named as Toronto  $a$  in Table 5); and
- to minimise the average cost per student (named as Toronto  $b$  in Table 5).

For the first objective, the aim is to find feasible timetables of the shortest length. For the second objective, an evaluation function was defined to calculate the cost of the timetables generated. For students sitting two exams  $s$  timeslots apart, the cost was assigned using proximity values  $w_s$ , i.e.  $w_1 = 16$ ,  $w_2 = 8$ ,  $w_3 = 4$ ,  $w_4 = 2$  and  $w_5 = 1$ . The aim is to space out the conflicting exams within a limited number of timeslots. The authors also introduced seven real world applications with side constraints (i.e. maximum room capacity per timeslot, pre-assigned exams, maximum number of exams per timeslot, no  $x$  exams in  $y$  timeslots, etc.). This objective was modified later and tested by a number of approaches (see below).

During the years, however, two versions of the data were circulated and were tested by different approaches. To distinguish between the datasets tested and to build a standard benchmark for future use in timetabling, we have carefully examined the data that has appeared in two different forms

under the same name for five of these benchmark problems. It was found that in the second version of the data files, duplicate exams have been assigned to some students in three instances (sta 83, yor 83 and ear 83). In order to correct these datasets, we have removed all the duplicate exams and verified the total number of exams assigned to each student in the data files. There was also some confusion concerning the instance pur93, as it has conflicts on the total number of enrolments (i.e. a different number of enrolments defined in two different data files for this instance).

We list the characteristics of these two versions of data in Table 4 and post-fix “I” and “II”, respectively, to the circulated datasets to distinguish between them. We have also listed the characteristics of a corrected version II of the relevant datasets and post-fixed “IIc” (the ‘c’ stands for ‘corrected’). The post-fix “I” has been used for the problem instance which we believe has appeared most often in the literature. To avoid any further confusion, the definitive versions of these datasets are available at <http://www.asap.cs.nott.ac.uk/resources/data.shtml> (together with all the other datasets discussed in this paper). Later on in this section, we attempt to cast light on the question of which technique has been applied to which version of these instances in the literature.

Burke et al. (1996b) addresses a modified problem where the objective of the six real world problems introduced in (Carter et al. 1996) considers the maximum room capacity per timeslot, and adjacent exams on the same day. In (Burke et al. 1998b), timeslots in the problems were distinguished by setting three timeslots a day from Monday to Friday and one timeslot on Saturday. The objective is to minimise the students sitting two consecutive exams on the same day and overnight. These two variants are named as Toronto  $c$  and Toronto  $d$  in Table 5. Terashima-Marin et al. (1999a) modified the dataset by assigning, to each problem instance, an estimated number of timeslots and to each timeslot an estimated maximum seats/capacity. This variant is named Toronto  $e$  in Table 5.

The approaches developed and tested on different variants of the Toronto datasets during the years are listed in Table 6 (ordered by the year in which the work was published). The values in “( )” following the variants of the data give the number of problem instances tested by the corresponding approaches. Most of the work did not specify the exact characteristics of the data tested, and in many of the papers it is impossible to determine which version (I or II) of the data was tested (for the 5 problematical instances). Note that so far there is no work yet on version IIc of the dataset. We have attempted, in Tables 10 and 11 (by contacting the authors), to clarify which versions of the datasets were used in each paper. If the entries are written in italics, we are not absolutely sure that the information with respect to this issue is correct. Otherwise, we have had the situation confirmed by the authors concerned.

**Table 4** Characteristics of the two versions and the corrected version II of the Toronto benchmark datasets

Problem instance	Exams	Students	Enrolments	Conflict density	Timeslots
car91	682	16925	56877	0.13	35
car92	543	18419	55522	0.14	32
ear83 I	190	1125	8109	0.27	24
ear83 II	189	1108	8092	0.27	24
ear83 IIc	189	1108	8057	0.27	24
hec92 I	81	2823	10632	0.42	18
hec92 II	80	2823	10625	0.42	18
kfu93	461	5349	25113	0.06	20
lse91	381	2726	10918	0.06	18
pur93 I	2419	30032	120681	0.03	42
pur93 II	2419	30032	120686/120681	0.03	42
rye92	486	11483	45051	0.07	23
sta83 I	139	611	5751	0.14	13
sta83 II	138	549	5689	0.14	13
sta83 IIc	138	549	5417	0.19	35
tre92	261	4360	14901	0.18	23
uta92 I	622	21266	58979	0.13	35
uta92 II	638	21329	59144	0.12	35
ute92	184	2749	11793	0.08	10
yor83 I	181	941	6034	0.29	21
yor83 II	180	919	6012	0.29	21
yor83 IIc	180	919	6002	0.30	21

**Table 5** Variants of the Toronto benchmark datasets

	Variants	Objectives
Toronto <i>a</i>	graph colouring	to minimise the number of timeslots needed
Toronto <i>b</i>	un-capacitated with cost	to space out conflicting exams within limited (fixed number of) timeslots
Toronto <i>c</i>	capacitated with cost	to minimise students sitting two exams in a row on the same day
Toronto <i>d</i>	capacitated with modified cost	same as above, and to minimise students sitting two exams overnight
Toronto <i>e</i>	estimated capacity and timeslots	to minimise students sitting two adjacent exams on the same day

### 3.2 University of Nottingham benchmark data

Burke et al. (1996b) also introduced the 1994 exam timetabling data at the University of Nottingham as a benchmark. It was used later by a number of researchers to test different approaches. Table 7 presents the characteristics of the dataset. We know that 23 is the least possible number of timeslots due to the limitations on the room capacity. The objective is to minimise the students sitting two consecutive exams on the same day. The data can be downloaded from <http://www.asap.cs.nott.ac.uk/resources/data.shtml>.

In (Burke et al. 1998b), the above problems were further constrained by modifying the objective function to consider also consecutive exams overnight. In Table 7 we highlight these variants as Nottingham *a* and *b*. Table 8 presents the approaches applied to these datasets and the University of Melbourne datasets (see Sect. 3.3 below) in the literature.

### 3.3 University of Melbourne benchmark data

Merlot et al. (2003) introduced exam timetabling datasets from the University of Melbourne at the PATAT conference in 2002. Two datasets were introduced. For these datasets,

**Table 6** Approaches to the Toronto benchmark datasets

Reference	Approach/technique	Problem and the number of instances tested
Carter et al. (1996)	Graph heuristics with clique initialisation and backtracking	a(13), b(13)
Carter and Johnson (2001)	Almost cliques with sufficient density as the initialisation for graph heuristics	a(13)
Burke et al. (1996b)	Memetic algorithm with hill climbing and light and heavy mutation	c(5)
Burke et al. (1998b)	Different initialisation strategies in Memetic Algorithms measured by diversity	d(3)
Burke et al. (1998c)	Randomness introduced by selection strategies in graph heuristics	d(3)
Burke and Newall (1999)	Multi-stage Evolutionary Algorithm based on a memetic algorithm	d(3)
Terashima-Marin et al. (1999a)	Genetic Algorithm with in-direct coding of constructive strategies and heuristics	e(12)
Caramia et al. (2001)	Iterated algorithm with novel improving factors	a(13), b(13), c(5)
Di Gaspero and Schaerf (2001)	Adaptive tabu list and cost function in Tabu Search	b(11), c(5), d(3)
Di Gaspero (2002)	Multiple neighbourhood Tabu Search	b(7), d(3)
White and Xie (2001)	Tabu Search with long term memory	b(2)
White et al. (2004)	Relaxation on long and short term tabu lists	b(7)
Paquete and Stützle (2002)	Tabu Search with Lex-tie and Lex-seq strategies in the objective function	b(8)
Merlot et al. (2003)	Constraint programming as initialisation for Simulated Annealing and hill climbing	a(12), b(12), c(5), d(2)
Casey and Thompson (2003)	GRASP with modified Saturation Degree initialisation and Simulated Annealing improvement	b(10)
Burke and Newall (2003)	Great Deluge with adaptive ordering as the initialisation	b(11)
Burke and Newall (2004)	Graph heuristics with adaptive heuristic modifier to dynamically order the exams	b(11), d(3)
Burke et al. (2004a)	Time-predefined Great Deluge and Simulated Annealing	b(13), d(2)
Asmuni et al. (2005)	Fuzzy rules with Largest Degree, Saturation Degree and Largest Enrolment	b(12)
Asmuni et al. (2007a, 2007b)	Fuzzy evaluation function with multiple criteria and tuning on fuzzy rules on improved model	b(12)
Ross et al. (2004)	Genetic Algorithm evolving constructive strategies and heuristics	e(12)
Burke et al. (2005)	Hybridising graph heuristics in hyper-heuristic by CBR and systematic strategies	b(4)
Côté et al. (2005)	Bi-objective Evolutionary Algorithm with local search operators in the recombination process	b(12)
Kendall and Hussin (2005a)	Tabu Search based hyper-heuristic	b(8)
Yang and Petrovic (2005)	Similarity measure using a fuzzy set on selecting hybridisations of Great Deluge and graph heuristics	b(12)
Burke et al. (2006a)	Genetic Algorithms on selecting subset of neighbourhoods in Variable Neighbourhood Search	b(11)
Burke et al. (2006b)	Case based heuristic selection for the solution construction	b(11)
Malim et al. (2006)	Three variants of artificial immune algorithms	a(5)
Abdullah et al. (2007a)	Large neighbourhood search with tree-based neighbourhood structure	b(12), c(5)
Abdullah et al. (2007b)	Tabu Search based large neighbourhood search	c(5)
Burke et al. (2007)	Graph based hyper-heuristic using Tabu Search	b(11)
Qu and Burke (2008)	Graph based hyper-heuristic framework with different high level search algorithms	b(11)
Qu and Burke (2007)	Adaptive decomposition on the ordering of exams to two sets of different difficulty level	b(11)
Eley (2007)	Ant algorithms with hill climbing operators	b(13)
Ulker et al. (2007)	Genetic algorithms with linear linkage encoding and specialised crossover operators	a(12)
Ersoy et al. (2007)	Memetic algorithm hyper-heuristics with three hill climbers chosen adaptively or deterministically	b(6)

**Table 7** Characteristics of the Nottingham benchmark datasets

	Nottingham <i>a</i>	Nottingham <i>b</i>
Exams	800	800
Students	7896	7896
Timeslots	23, 26	23
Enrolments	34265	34265
Conflicts	10034	10034
Capacity	1550	1550
Density	0.03	0.03
Objective	minimise adjacent exams on the same day	minimise adjacent exams on the same day and overnight

**Table 8** Approaches on the Nottingham and Melbourne benchmark datasets

Reference	Approach/technique	Problem
Burke and Newall (2004)	Adaptive ordering based on graph heuristics using heuristic modifier	Nottingham <i>b</i>
Merlot et al. (2003)	Constraint logic programming as initialisation for Simulated Annealing and hill climbing	Nottingham <i>a, b</i> Melbourne <i>I, II</i>
Di Gaspero and Schaerf (2001)	Tabu Search using exhaustive and biased selection by the costs of exams	Nottingham <i>a, b</i>
Burke et al. (2004a)	Great Deluge with a number of runs of Saturation Degree	Nottingham <i>b</i>
Caramia et al. (2001)	Iterative approach where the number of timeslots was gradually increased after greedy improvement	Nottingham <i>a</i>
Ahmadi et al. (2003)	Variable neighbourhood search to find permutations of heuristics and their weights	Nottingham <i>a</i>
Côté et al. (2005)	Evolutionary algorithms with bi-objective constraint satisfaction	Nottingham <i>a</i> Melbourne <i>I</i>
Burke et al. (1996b)	Memetic methodology with light & heavy mutations and graph heuristic initialisation	Nottingham <i>a</i>
Burke and Newall (1999)	Multi-stage evolutionary algorithm initialised by graph heuristics with backtracking	Nottingham <i>b</i>
Burke et al. (2001)	Multi-criteria approach dealing with 9 criteria based on initial solutions by Saturation Degree	Nottingham <i>b</i>

**Table 9** Characteristics of the Melbourne benchmark datasets

	Exams	Timeslots	Students	Enrolls	Objective
<i>I</i>	521	28	20656	62248	minimise adjacent exams on the same day or overnight
<i>II</i>	562	31	19816	60637	same as above

there were two timeslots on each day for each of the five workdays, and the capacity for each session varied. The availability of sessions for some of the exams was restricted. In one problem instance, this meant that no feasible solutions existed, so an alternative data set was created which allowed feasible solutions. These datasets can also be downloaded from <http://www.asap.cs.nott.ac.uk/resources/data.shtml>. The Melbourne datasets are summarised in Table 9.

### 3.4 The second international timetabling competition

Building on the success of the First International Timetabling Competition, with the goal of creating a better understanding between researchers and practitioners in

timetabling, the 2nd International Timetabling Competition was announced in August 2007 (see <http://www.cs.qub.ac.uk/itc2007/>). The competition was divided into three tracks (one on exam and two on course timetabling). It provided researchers with problem models which incorporated more real-world constraints than the models from the literature. Rules and initial results, along with the associated algorithms have been provided to encourage scientific comparisons. At the time of completing this paper, the final results have just been released at the competition web site and the winners of each track will be formally announced at the PATAT 2008 conference in August 2008. Detailed information about this competition can be seen in (McCollum et al. 2008).



**Table 10** Results in the literature on the two versions of the Toronto Dataset *b* (see Table 5). Values in italics indicate that we are unsure about the accuracy in terms of the versions of the datasets used. Val-

ues in bold represent the best results reported. “–” indicates that the corresponding problem is not tested or a feasible solution cannot be obtained

Data set version	Carter et al. (1996)	Caramia et al. (2001)	Di Gaspero and Schaefer (2001)	Di Gaspero (2002)	Paquete and Stützle (2002)	Burke and Newall (2003)	Casey and Thompson (2003)	Merlot et al. (2003)
I/II	I	I	I	I	I	I	II	I
car91	7.1	<i>6.6</i>	<i>6.2</i>	<i>5.2</i>	–	4.65	5.4	5.1
car92	6.2	<i>6.0</i>	<i>5.2</i>	–	–	4.1	4.2	4.3
ear83	36.4	<b>29.3</b>	<i>45.2</i>	<i>39.2</i>	<i>38.2</i>	37.05	34.2	35.1
hec92	10.8	<b>9.2</b>	<i>12.2</i>	<i>10.2</i>	<i>11.2</i>	11.54	10.2	10.6
kfu93	14.0	<i>13.2</i>	<i>18.2</i>	–	<i>16.2</i>	13.9	14.2	13.5
lse91	10.5	<b>9.6</b>	<i>15.2</i>	<i>12.2</i>	<i>13.2</i>	10.82	14.2	10.5
rye92	7.3	<b>6.8</b>	–	–	–	–	–	–
sta83	161.5	<i>158.2</i>	<i>160.2</i>	<i>157.2</i>	<i>168.2</i>	168.73	<b>134.9</b>	157.3
tre92	9.6	9.2	<i>10.2</i>	–	9.2	8.35	8.2	8.4
uta92	3.5	3.2	4.2	4.2	–	3.2	–	3.5
ute92	25.8	<b>24.4</b>	<i>27.8</i>	–	<i>29.2</i>	25.83	25.2	25.1
yor83	41.7	<b>36.2</b>	<i>41.2</i>	<i>39.2</i>	<i>38.2</i>	37.28	37.2	37.4

**Table 11** Results in the literature on the two versions of the Toronto Dataset *b*. The value marked with \* here is different from that in (Yang and Petrovic 2005), as a different objective function was used in (Yang and Petrovic 2005)

Data set version	Burke and Newall (2004)	Burke et al. (2004a)	Asmuni et al. (2005)	Côté et al. (2005)	Kendall and Hussan (2005a)	Yang and Petrovic (2005)	Abdullah et al. (2007a)	Burke et al. (2007)	Burke et al. (2006a)	Eley (2007)
I/II	I	I	I	I	I	I	I	I	I	I
car91	5.0	4.8	5.19	5.2	5.37	<b>4.5</b>	5.2	5.36	4.6	5.2
car92	4.3	4.2	4.51	4.2	4.67	<b>3.93</b>	4.4	4.53	4.0	4.3
ear83	36.2	35.4	36.64	<i>34.2</i>	40.18	33.7	34.9	37.92	32.8	36.8
hec92	11.6	10.8	11.6	<i>10.2</i>	11.86	10.83	10.3	12.25	10.0	11.1
kfu93	15.0	13.7	15.34	<i>14.2</i>	15.84	13.82	13.5	15.2	<b>13.0</b>	14.5
lse91	11.0	10.4	11.35	<i>11.2</i>	–	10.35	10.2	11.33	10.0	11.3
rye92	–	8.9	10.05	8.8	–	8.53	8.7	–	–	9.8
sta83	161.9	159.1	160.79	<i>157.2</i>	157.38	158.35*	159.2	158.19	159.9	157.3
tre92	8.4	8.3	8.47	8.2	8.39	7.92	8.4	8.92	<b>7.9</b>	8.6
uta92	3.4	3.4	3.52	3.2	–	<b>3.14</b>	3.6	3.88	3.2	3.5
ute92	27.4	25.7	27.55	25.2	27.6	25.39	26.0	28.01	24.8	26.4
yor83	40.8	36.7	39.79	<i>36.2</i>	–	36.35	<b>36.2</b>	41.37	37.28	39.4

### 3.5 Results on the benchmark problems

As mentioned above, there has been a large number of papers published which have worked with the datasets discussed above. In addition, the difficulties surrounding the publication of the Toronto datasets have led to some confusion over which methods were tackling which problems. Tables 10, 11, 12 and 13 attempt to clarify this. They list all of the methods which have addressed the

Toronto problems and they attempt to illustrate which methods used which problems. This has been a difficult task and the authors would welcome additional information which we will use to keep an updated version of the table at <http://www.asap.cs.nott.ac.uk/resources/data.shtml>. We would also like to add more methods as they appear.

Table 14 presents the results from different approaches in the literature applied to the Nottingham datasets *a* and *b* (see Table 7) and the Melbourne datasets *I* and *II* (see Table 7).

**Table 12** Results in the literature on the Toronto Dataset I, variants *a* (left column) and *e* (right column)

Data set	Carter et al. (1996)	Caramia et al. (2001)	Merlot et al. (2003)	Ulker et al. (2007)	Terashima-Marin et al. (1999a)	Ross et al. (2004)
car91	<b>28</b>	<b>28</b>	30	36	<b>130</b>	283
car92	<b>28</b>	<b>28</b>	31	36	<b>285</b>	542
ear83	<b>22</b>	<b>22</b>	24	23	<b>723</b>	958
hec92	<b>17</b>	<b>17</b>	18	<b>17</b>	<b>154</b>	224
kfu93	<b>19</b>	<b>19</b>	21	20	<b>223</b>	226
lse91	<b>17</b>	<b>17</b>	18	<b>17</b>	<b>221</b>	263
rye92	<b>21</b>	<b>21</b>	22	23	<b>671</b>	832
pur93	<b>35</b>	36	–	–	–	–
sta83	<b>13</b>	<b>13</b>	<b>13</b>	<b>13</b>	<b>821</b>	1058
tre92	<b>20</b>	<b>20</b>	21	21	<b>586</b>	604
uta92	32	<b>30</b>	32	38	<b>594</b>	855
ute92	<b>10</b>	<b>10</b>	11	<b>10</b>	<b>902</b>	967
yor83	<b>19</b>	<b>19</b>	23	20	<b>708</b>	758

**Table 13** Results in the literature on the Toronto Dataset I, variants *c* (upper part) and *d* (lower part). Values marked with † are for car91 with 51 timeslots; values marked with ‡ and + are for car92 with 40 timeslots and 31 timeslots, respectively

Dataset	car91	car92	kfu93	tre92	uta92	pur93
Burke et al. (1996b)	81†	331‡	974	3	772	–
Caramia et al. (2001)	74†	268‡	912	2	680	–
Di Gaspero and Schaerf (2001)	88†	424+	512	4	554	–
Merlot et al. (2003)	31†	158+	247	0	334	–
Abdullah et al. (2007a)	37†	278+	548	0	300	–
Abdullah et al. (2007b)	47†	525+	206	4	310	–
Burke et al. (1998c)	–	2218	3256	–	<b>2440</b>	–
Burke and Newall (1999)	–	1665	1388	–	–	<b>63824</b>
Di Gaspero and Schaerf (2001)	–	3048	1733	–	–	123935
Merlot et al. (2003)	–	1744	<b>1082</b>	–	–	–
Burke and Newall (2004)	–	1775	1422	–	–	97237
Burke et al. (2004a)	–	<b>1506</b>	1321	–	–	–

Tables 10–14 also illustrate which of the methods are most effective in terms of solution quality. The very best results are presented in bold. We have not listed computational times for the following reasons. Firstly, many of these papers do not report the relevant times. Secondly, comparisons across very different platforms over the years are impossible. Thirdly, examination timetabling is a problem which is almost always tackled weeks or months before the timetable will be used. As such, it is definitely not a time critical problem and there are many real world scenarios where it would be perfectly reasonable to leave an algorithm running overnight or even over a weekend.

## 4 Conclusions

### 4.1 Overview of recent research

Timetabling research started with simple sequential techniques in the 1960s. Constraint based techniques appeared

later and still play a significant role in timetabling today. Recent research in exam timetabling is dominated by meta-heuristics and their integrations/hybridisations with a variety of methodologies, including many of the early techniques. Local search based techniques, multi-criteria techniques and approaches which aim to be more general than the state of the art have also represented important research outcomes.

Recent innovations have utilised different mechanisms in exam timetabling and they cover a variety of new techniques including Variable Neighbourhood Search, Iterated Local Search, GRASP, and hyper-heuristics. These approaches have represented a variety of motivating goals which include the aim of developing of more powerful, efficient, effective and more general approaches.

In the following section, we will draw upon the above discussion to highlight a number of conclusions and to present some ideas for future research which are generated by these conclusions. It is worth noting that McCollum (2007) and Burke et al. (2004b) outline some future research directions

**Table 14** Results on the Nottingham Datasets *a* and *b* and the Melbourne Datasets *I* and *II*. Values in bold represent the best results reported. “–” indicates that the corresponding problem is not tested or a feasible solution cannot be obtained

	Nottingham	Burke et al. (1996b)	Di Gaspero and Schaerf (2001)	Caramia et al. (2001)	Merlot et al. (2003)	Abdullah et al. (2007b)
<i>a</i> (26 slots)		53	11	44	<b>2</b>	18
<i>a</i> (23 slots)		269	123	–	<b>88</b>	–
	Nottingham	Burke and Newall (1999)	Di Gaspero and Schaerf (2001)	Merlot et al. (2003)	Burke and Newall (2004)	Burke et al. (2004a)
<i>b</i>		519	751	401	545	<b>384</b>
	Melbourne	Merlot et al. (2003)				
<i>I</i>		1072				
<i>II</i>		1115				

in university timetabling and nurse rostering, respectively. There is, of course, some synergy with issues discussed in both these papers and these are alluded to here.

#### 4.2 Future research challenges

We will now outline some overall messages from our analysis of the literature and how these may influence future research directions. The following discussion presents a (non-exhaustive) list of strategic research challenges.

##### 4.2.1 Meta-heuristic development

Meta-heuristics have attracted the most attention in exam timetabling research, especially in recent years. They will continue to attract attention in the near future across a number of different perspectives. In particular, it is important to explore these research directions within the context of real world problem solving environments.

In addition to a comprehensive treatment of Tabu Search, Simulated Annealing, Genetic Algorithms and various hybrids, some new exam timetabling techniques have recently been presented. For example, GRASP and Iterated Local Search build on the idea of exploring wider areas of the search space by using a multi-start greedy search technique to reduce the risk of being stuck in local optima. Variable Neighbourhood Search escapes from local optima by switching between the search spaces defined by different neighbourhood structures. Large neighbourhood search extends the flexibility of moves within the search space. In summary, these techniques build upon the idea of helping the search to escape from local optima in a variety of ways and have obtained promising progress on exam timetabling problems.

The development of these new techniques has opened up a wide variety of new research directions such as exploring alternative neighbourhood structures, new multi-start techniques, hybridisation issues, alternative operators and many others. One of the key research goals is to provide an appropriate balance between *exploration* and *exploitation* in the development of search algorithms.

Extensive study is also required to understand how to determine appropriate parameter settings for meta-heuristic methods. The determination of suitable initialisation methods and in-depth analysis of the effects of initialisation on a range of meta-heuristics is another important exam timetabling research topic. Theoretical issues (such as phase transition) and multi-criteria techniques underpin other important directions in meta-heuristic exam timetabling research. Evolutionary methods and other population based techniques represent a significant proportion of the meta-heuristic literature on exam timetabling. There are many research directions generated by considering the hybridisation of meta-heuristic methods particularly between population based methods and other approaches. A study of coding issues represents a promising direction in both evolutionary algorithms and hyper-heuristic research.

##### 4.2.2 Raising the level of generality of search methodologies

A goal which has been addressed in some of the recent exam timetabling research literature is the exploration and development of more general and adaptive methodologies (which range across timetabling and other search problems). This underpins the challenge of automating the design of search techniques.

Hyper-heuristics are concerned with searching for appropriate heuristics rather than concentrating on the problem specific details of actual solutions, which have been the focus of traditional search algorithms. This opens up a new direction of research and represents much potential in both practical applications and theoretical study. Adaptive techniques have also recently emerged where information collected during the problem solving is used to guide the search. Some work has been carried out on knowledge based techniques where the experience gained from previous problem solving drives the search. Further investigation of these (and other) issues has the promise to underpin the development of fundamentally more general approaches. The goal is to deal automatically with different problems in a dynamic way so that extra human effort is not needed to fine-tune the approach.

#### 4.2.3 Understanding search spaces

We do not understand why and how search methods work on complex problems. A wide ranging analysis of heuristics/techniques concerning the nature of search spaces could deepen our understanding of the strengths and limitations of search methodologies. It is generally accepted that little is known about the nature of search spaces, especially for complex real-world problems such as exam timetabling. A deeper understanding of search spaces and fitness landscape analysis (Reeves 2005) offers the possibility of understanding why certain algorithms work well on certain problems (or even instances) and yet work poorly on others. This is not restricted to exam timetabling but covers a much broader remit of problems.

#### 4.2.4 Inter-disciplinarity

Hybridisations of different techniques have been very widely investigated in recent exam timetabling research.

Although different authors have favoured different approaches, it has been observed that hybrid approaches are usually superior to *pure* algorithms. For example, all of the recent work on constraint based techniques represents hybridisation with other techniques (see Sect. 2.2). However, in most of the cases, methodologies are hybridised in a sequential way rather than being efficiently integrated. More work needs to be done to not just simply combine techniques but rather to more meaningfully integrate different methodologies efficiently and effectively. Such research should draw upon the research theme outlined in Sect. 4.2.3 above. For example, in memetic algorithms, local search is used co-operatively after each generation. In a hyper-heuristic, one approach taken is that low level heuristics are searched and combined adaptively during the problem

solving. Further in-depth analysis and investigation can underpin the design and development of more powerful techniques.

In order to fully explore hybridisation issues and indeed the effective development of search methods, it is important that an interdisciplinary approach is taken which draws upon research developments across the disciplinary boundaries of Operational Research, Computer Science and other fields. Our analysis of the literature shows that methodologies from across these disciplinary divides can be integrated and hybridised to produce highly effective exam timetabling techniques.

#### 4.2.5 Closing the gap between theory and practice

Over the last ten years, new benchmark examination timetabling problems have been formed and thoroughly tested. Reformulations of problems will better reflect more constraints in real world environments. Indeed, the 2007 International Timetabling Competition has very recently introduced more complex and realistic exam timetabling problems to the research community.

Recent state-of-the-art approaches in exam timetabling research have carried out comparisons on the benchmark problems (see Sect. 3) that have appeared over the last ten years. This has led to fundamental developments in exam timetabling research. However, these problems still represent simplified versions of the problem. In the wider context of scheduling research, there has been much recent debate about the “gap between theory and practice”. The same is true for exam timetabling research. A major research direction is represented by exploring the wide range of research issues that are opened up by considering the high levels of complexity that are generated by real world problems (McCollum 2007). In addition, there still is no widely accepted universal data format and standard timetabling languages. The establishment of quality measures by standard techniques on both solution quality (objective functions) and computational time for exam timetabling problems also requires much work and is crucial in conjunction with the formation of benchmarks. The requirements for the development of automatic tools to support timetabling staff to save significant development time still exists. To encourage such development, we are building up an archive where benchmark exam timetabling problems are collected, together with a categorised updated timetabling bibliography (after 1995). We welcome contributions to this exam timetabling archive. It is held at <http://www.asap.cs.nott.ac.uk/resources/ETTPbibliography.shtml>.

#### 4.3 Summary

In summary, it is possible to draw a number of conclusions from an in-depth survey of the examination timetabling lit-



erature in the last ten years. Firstly, there has been a significant number of research successes in that time. Secondly, the current state-of-the-art provides a strong platform for a range of important research challenges. Thirdly, future research requires a particular emphasis on the complexity of real world issues and this requires the establishment of more benchmarks that are drawn from real world problems. Fourthly, raising the level of generality of decision support systems (including for exam timetabling) represents an emerging theme. Finally, it is worth noting that successful papers in exam timetabling have been authored by researchers from across a range of disciplinary backgrounds and particularly at the interface of Operational Research and Computer Science. Such interdisciplinary collaboration is crucial to scientific progress in the area. It is clear from this analysis of the literature that the future of exam timetabling research is interdisciplinary.

#### **Appendix A: PhD theses related to university timetabling**

- J. Pendlebury. A computer-aided timetabling system for use in FE/HE. PhD Thesis. Department of Operational Research, University of Lancaster, UK. July 1989.
- H. Fang. Genetic Algorithms in Timetabling and Scheduling. PhD Thesis, Division of Informatics, University of Edinburgh, 1994.
- R.F. Weare. Automated Examination Timetabling. PhD Thesis, School of Computer Science and Information Technology, University of Nottingham, UK, June 1995.
- H. Terashima-Marin. Combinations of GAs and CSP Strategies for Solving the Examination Timetabling Problem. PhD Thesis, Instituto Tecnológico y de Estudios Superiores de Monterrey, August 1998.
- J.P. Newall. Hybrid Methods for Automated Timetabling. PhD Thesis, School of Computer Science and Information Technology, University of Nottingham, UK, May 1999.
- H. Rudova. Constraint Satisfaction with Preferences. Ph.D. Thesis, Faculty of Informatics, Masaryk University, 2001.
- R. Qu. Case Based Reasoning for Course Timetabling Problems. PhD Thesis, School of Computer Science and Information Technology, University of Nottingham, UK, August 2002.
- L. Di Gaspero. Local Search Techniques for Scheduling Problems: Algorithms and Software Tools. PhD Thesis, Computer Science, Università degli Studi di Udine, October 2002.
- R.J. Willems. School timetable construction Algorithms and complexity. Technische Universiteit Eindhoven, 2002.

- Y. Bykov. Time-Predefined and Trajectory-Based Search: Single and Multi-objective Approaches to Exam Timetabling. PhD Thesis, School of Computer Science and Information Technology, University of Nottingham, UK, November 2003.
- Y. Yang. Solving Examination Timetabling Problems by Case Based Reasoning. PhD Thesis, School of Computer Science and Information Technology, University of Nottingham, UK, September 2004.
- L. Merlot. Techniques for Academic Timetabling. PhD Thesis, Department of Mathematics and Statistics and Department of Computer Science and Software Engineering, University of Melbourne, Australia, August 2005.
- N. Hussin. Tabu Search Based Hyper-heuristic Approaches for Examination Timetabling. PhD Thesis, School of Computer Science and Information Technology, University of Nottingham, UK, November 2005.
- A. Eckersley. Novel Knowledge based and Heuristic Approaches to University Timetabling. PhD thesis, School of Computer Science and Information Technology, University of Nottingham, UK, May 2006.
- S. Abdullah. Heuristic Approaches for University Timetabling Problems. PhD Thesis, School of Computer Science and Information Technology, University of Nottingham, UK, July 2006.

#### **Appendix B: Summarising tables categorised by techniques in examination timetabling**

In this appendix, the research methods in the exam timetabling literature are categorised in Tables 15–26 by the approaches and techniques they employed. Work by the same authors was grouped in the tables to represent the continuous development of a line of investigation. The content of the tables is ordered by the year of publication to represent the development of related techniques over the years. In the tables, the term “practical” indicates that the corresponding work was tested/implemented on real world problems. “–” indicates that the corresponding properties were not presented in the paper. “Toronto”, “Nottingham” and “Melbourne” refer to the benchmark problems described in Sect. 3.

**Table 15** Graph colouring heuristics for examination timetabling

Reference	Techniques	Problems
Carter et al. (1996)	Largest cliques as the initialisation for graph heuristics with backtracking	Toronto, random
Burke et al. (1998c)	Biased and tournament selection that introduce nondeterminism to graph heuristics	Toronto
Carter and Johnson (2001)	Almost cliques that are sufficiently dense as the initialisation for graph heuristics	Toronto
Burke and Newall (2004)	Adaptive heuristic modifier based on Largest Degree, Colour Degree and Saturation Degree that dynamically order the exams	Toronto, Nottingham

**Table 16** Constraint based techniques

Reference	Techniques	Problem	Notes
David (1998)	Iterative approach, repairing strategies on partial solutions generated by constraint satisfaction model	Practical	
Reis and Oliveira (1999)	Constraint satisfaction model, set variables	Random, practical	ECLiPSe package
Merlot et al. (2003)	Constraint logic programming as initialisation for Simulated Annealing and hill climbing, labelling: by the size of domain	Toronto, Nottingham Melbourne	OPL language
Duong and Lam (2004)	Constraint programming with Simulated Annealing, backtracking & forward checking, Labelling: by the size of domain, number of students, etc.	Practical	
Le Huédé et al. (2006)	Constraint programming with multicriteria optimisation, variable ordering by largest degree, specific labelling for each criteria	Random, modified Toronto	Criteria choice heuristic during search

**Table 17** Tabu search

Reference	Moving strategies	Tabu list	Problem	Notes
Di Gaspero and Schaerf (2001)	Exhaustive and biased selection on exams causing cost	Adaptive	Toronto, Nottingham	Greedy initialisation Dynamic cost function, employing the EASYLOCAL++ tool
Di Gaspero (2002)	Token ring search with 3 neighbourhood structures	Adaptive	Toronto	
White and Xie (2001, 2004)	Moves on single exams	Long and short term memory	practical, Toronto	4-stage Tabu Search Largest enrollment initialisation
Paquete and Stützle (2002)	Moves on single exams causing costs	Adaptive	Toronto	Ordered priorities for constraints

**Table 18** Simulated annealing

Reference	Moving strategies	Initialisation	Problem	Notes
Thompson and Dowsland (1998)	Kempe chain	Random	Practical and derived problems	Two-phase approach
Bullnheimer (1998)	Slot&exam moves	–	Practical	Adapted model of QAP
Merlot et al. (2003)	Kempe chain	Constraint programming	Toronto, Melbourne, Nottingham	Geometric cooling schedule
Duong and Lam (2004)	Kempe chain	Constraint programming	Practical	Components set experimentally
Burke et al. (2004a)	Moves on single exams	A number of runs of Saturation Degree	Toronto, Nottingham	Time-predefined Great Deluge

**Table 19** Local search based techniques (other than tabu search and simulated annealing)

Reference	Techniques	Problem
Caramia et al. (2001)	Iterative process, the number of timeslots was gradually increased after greedy improvement	Toronto, Nottingham
Ahmadi et al. (2003)	Search on permutations of heuristics and their weights by Variable Neighbourhood Search	Extended Nottingham
Casey and Thompson (2003)	GRASP: Saturation Degree as initialisation, backtracking, improvement by modified Simulated Annealing with Kempe chain neighbourhood	Toronto
Abdullah et al. (2007a)	Large neighbourhood search with cyclic exchanges of exams among timeslots	Toronto
Qu and Burke (2008)	Variable Neighbourhood Search as the high level search upon graph heuristics	Toronto

**Table 20** Evolutionary algorithms

Reference	Operators	Problem	Notes
Ross et al. (1996)	Standard	Special graph colouring problems	Phase transition, compared with stochastic hill climbing
Ross et al. (1998)	As above	Special graph colouring problems	Issues of direct coding in Genetic Algorithms
Terashima-Marin et al. (1999b)	Clique-based crossover	Special classes of graph colouring problems	Issues of direct coding in Genetic Algorithms
Terashima-Marin et al. (1999a, 1999c)	Standard two-point crossover	As above Toronto	Hardness theory Indirect coding of heuristics rather than actual solutions
Erben (2001)	Specialised operators	Special graph colouring problems	Specialised fitness function
Sheibani (2002)	Standard	Derived problems	Special mathematical model to estimate costs
Wong et al. (2002)	Mutation with heuristic repairing strategies	Practical	Modelled as constraint satisfaction problem
Côté et al. (2005)	Local search operators	Toronto, Melbourne, Nottingham	Bio-objective constraint satisfaction problems
Burke et al. (1996b)	Light and heavy mutations	Toronto, Nottingham	Memetic algorithm with hill climbing, graph heuristics as the initialisation
Burke et al. (1998b)	As above	Toronto	Initial populations with different diversities in memetic algorithm
Burke and Newall (1999)	As above	Toronto, Nottingham	Multi-stage Evolutionary Algorithm initialised by graph heuristics with backtracking
Ulker et al. (2007)	Specialised crossover	Toronto graph colouring	Linear linkage encoding in multi-objective GA framework

**Table 21** Ant algorithms and artificial immune algorithms

Reference	Initialisation	Problems	Notes
Naji Azimi (2004)	Heuristic method	Derived Toronto	Local improvement, compared with Simulated Annealing, Tabu Search and Genetic Algorithms
Naji Azimi (2005)	Heuristic method	As above	Hybridisations with Tabu Search
Dowsland and Thompson (2005)	Recursive Largest Degree and Saturation Degree	Toronto	Modified fitness function based on (Costa and Hertz 1997), trail calculation, parameter settings
Eley (2007)	Inverse of Saturation Degree	Toronto	Hybridisations with hill climbing to further improve the solutions
Malim et al. (2006)	Random construction	Toronto & course timetabling	Clonal selection, immune network and negative selection algorithms on both course and exam timetabling



**Table 22** Multi-criteria techniques

Reference	Techniques	Problems	Notes
Colijn and Layfield (1995a)	Multi-stage process for different objectives	Practical	Matrices used to record the violations
Colijn and Layfield (1995b)	Interactive interface for unforeseen circumstances	Practical	
Burke et al. (2001)	Hill climbing and heavy mutation on initial solutions by Saturation Degree	Nottingham	Deal with 9 criteria
Petrovic and Bykov (2003)	Search by Great Deluge towards ideal point guided by predefined trajectory in the criteria space	Nottingham Toronto	Random initial solutions, dynamic weights
Le Huédé et al. (2006)	Multi-criteria optimisation in constraint programming	Random Modified Toronto	Specific strategy for each criteria

**Table 23** Hyper-heuristics

	High level heuristic	Low level heuristic	Problem	Notes
Terashima-Marin (1999a)	Evolutionary Algorithms	Solution construction strategies	Toronto	Study of indirect codings
Ahmadi et al. (2003)	Variable Neighbourhood Search	Constructive heuristics & weights	Extended Nottingham	Perturbations of low level heuristics and their weights in heuristic space
Petrovic and Yang (2005)	Case based reasoning	Graph heuristics	Toronto	Reuse graph heuristics as initialisation methods
Asmuni et al. (2005)	Fuzzy techniques	3 graph heuristics	Toronto	Different fuzzy membership functions
Kendall and Hussin (2005a)	Tabu Search	Constructive, moving strategies	Practical	Hybridise Tabu Search with hill climbing and Great Deluge
Ross et al. (2004)	Steady state Genetic Algorithm	Events and timeslots picking heuristics	Toronto, course timetabling problems	Simplified problem-state descriptions mapped to constructive heuristics, fitness functions
Burke et al. (2006b)	Case based reasoning	Graph heuristics	Toronto, course timetabling	Heuristics mapped and reused to problem solving situations
Burke et al. (2005)	Case based reasoning	As above	Toronto	Hybridise graph heuristics by different methodologies
Burke et al. (2007)	Tabu Search	Graph heuristics	Toronto	Different number of low level graph heuristics, multi-stage approach
Qu and Burke (2008)	Iterated Local Search, Variable Neighbourhood Search, etc.	As above	Toronto	Different neighbourhood and high level heuristics in the graph based hyper-heuristic
Bilgin et al. (2007)	Different (meta-) heuristics & acceptance criteria	Variants of hill climbers	Modified Toronto, functions	Pairs of heuristic selection and acceptance criteria for different problems
Ersoy et al. (2007)	As above	Hill climbers in Memetic Algorithms	Toronto	Three hill climbers applied in order within a Memetic Algorithms

**Table 24** Decomposition approaches

Reference	Techniques	Problem	Notes
Burke and Newall (1999)	Sequential methods to partition the problems	Toronto Nottingham	Sub-problems solved by Memetic Algorithms
Lin (2002)	Multi-agent algorithm	Toronto random	Aggregate schedules from agents and a broker
Qu and Burke (2007)	Adaptive ordering and decomposition	Toronto	Exams iteratively partitioned to two sets of different difficulty

**Table 25** Timetabling systems

Reference	Techniques	Problem	Notes
Hansen and Vidal (1995)	4-phase process with different objectives	Danish high schools	Centralised planning on oral and written exams, GIA-system
Lim et al. (2000)	Constructive heuristic with 3 measures, arc consistency algorithm	National Uni of Singapore	Exam and course timetabling, 3-tier client/server application, UTTS system
Ho et al. (2001)	Tabu Search with Push Forward Insertion heuristic	As above	Maximise exam paper spreading
Dimopoulou and Miliotis (2001)	Integer Programming, grouped courses assigned to grouped timeslots	Athens Uni of Economic & Business	Initial exam timetables based on course timetables and improved by heuristics

**Table 26** Timetabling languages and tools

Reference	Languages/tools	Notes
Burke et al. (1998a)	Similar to the Z specification language	Standard data format for easy exchange of results and comparisons in timetabling research
Tsang et al. (1999)	EaCL	Easy specification of constraint satisfaction problems to support building Constraint Satisfaction systems
Reis and Oliveira (2001)	UniLang using a list of synonyms	Timetabling problem specification in the ECLiPse package
Di Gaspero and Schaerf (2001)	EASYLOCAL++	Object-oriented software tool to support the development of local search algorithms
De Causmaecker et al. (2002)	Semantic Web, XML	Machine accessible way of easy identification for timetabling problems

## References

- Aarts, E. H. L., & Korst, J. (1989). *Simulated annealing and Boltzmann machines*. New York: Wiley.
- Aarts, E. H. L., Korst, J., & Michiels, W. (2005). Simulated annealing. In E. K. Burke & G. Kendall (Eds.), *Search methodologies: introductory tutorials in optimisation and decision support techniques* (pp. 187–211). Berlin: Springer. ISBN: 0387234608.
- Abdullah, S., Ahmadi, S., Burke, E. K., & Dror, M. (2007a). Investigating Ahuja–Orlin large neighbourhood search for examination timetabling. *OR Spectrum*, 29(2), 351–372.
- Abdullah, S., Ahmadi, S., Burke, E. K., Dror, M., & McCollum, B. (2007b). A tabu based large neighbourhood search methodology for the capacitated examination timetabling problem. *Journal of Operational Research*, 58, 1494–1502.
- Ahmadi, S., Barone, R., Cheng, P., Cowling, P., & McCollum, B. (2003). Perturbation based variable neighbourhood search in heuristic space for examination timetabling problem. In *Proceedings of multidisciplinary international scheduling: theory and applications (MISTA 2003)* (pp. 155–171), Nottingham, 13–16 August, 2003. ISBN: 0-9545821-2-8.
- Ahuja, R. K., Orlin, J. B., & Sharma, D. (2001). Multi-exchange neighbourhood search algorithm for capacitated minimum spanning tree problem. *Mathematical Programming*, 91, 71–97.
- Ajili, F., & Wallace, M. W. (2003). Hybrid problem solving in ECLiPSe. In M. Milano (Ed.), *Constraint and integer programming: toward a unified methodology* (pp. 169–201). Dordrecht: Kluwer Academic.
- Asmuni, H., Burke, E. K., Garibaldi, J., & McCollum, B. (2005). Fuzzy multiple ordering criteria for examination timetabling. In E. K. Burke & M. Trick (Eds.), *Lecture notes in computer science: Vol. 3616. Practice and theory of automated timetabling V: selected papers from the 5th international conference* (pp. 334–353). Berlin: Springer.
- Asmuni, H., Burke, E. K., Garibaldi, J., & McCollum, B. (2007a). A novel fuzzy approach to evaluate the quality of examination timetabling. In E. K. Burke & H. Rudova (Eds.), *Lecture notes in computer science: Vol. 3867. Practice and theory of automated timetabling VI: selected papers from the 6th international conference* (pp. 327–346). Berlin: Springer.
- Asmuni, H., Burke, E. K., Garibaldi, J., & McCollum, B. (2007b). Determining rules in fuzzy multiple heuristic orderings for construction examination timetables. In *Proceedings of the 3rd multidisciplinary international conference on scheduling: theory and application* (pp. 59–66), Paris, France, August 2007.
- Bardady, V. A. (1996). Computer-aided school and university timetabling: The new wave. In E. K. Burke & P. Ross (Eds.), *Lecture notes in computer science: Vol. 1153. Practice and theory of automated timetabling I: selected papers from the 1st international conference* (pp. 22–45). Berlin: Springer.
- Bilgin, B., Özcan, E., & Korkmaz, E. E. (2007). An experimental study on hyper-heuristics and exam timetabling. In E. K. Burke & H. Rudova (Eds.), *Lecture notes in computer science: Vol. 3867. Practice and theory of automated timetabling VI: selected papers from the 6th international conference* (pp. 394–412). Berlin: Springer.
- Boizumault, P., Delon, Y., & Peridy, L. (1996). Constraint logic programming for examination timetabling. *Journal of Logic Programming*, 26(2), 217–233.
- Brailsford, S. C., Potts, C. N., & Smith, B. M. (1999). Constraint satisfaction problems: Algorithms and applications. *European Journal of Operational Research*, 119, 557–581.
- Brelaz, D. (1979). New methods to colour the vertices of a graph. *Communication of the ACM*, 22(4), 251–256.
- Broder, S. (1964). Final examination scheduling. *Communications of the ACM*, 7, 494–498.
- Bullnheimer, B. (1998). An examination scheduling model to maximise students study time. In E. K. Burke & M. W. Carter (Eds.), *Lecture notes in computer science: Vol. 1408. Practice and theory of automated timetabling II: selected papers from the 2nd international conference* (pp. 78–91). Berlin: Springer.
- Burke, E. K., & Carter, M. W. (Eds.). (1998). *Lecture notes in computer science: Vol. 1408. Practice and theory of automated timetabling II: selected papers from the 2nd international conference*. Berlin: Springer. ISBN: 3-540-64979-4.
- Burke, E. K., & De Causmaecker, P. (Eds.). (2003). *Lecture notes in computer science: Vol. 2740. Practice and theory of automated timetabling IV: selected papers from the 4th international conference*. Berlin: Springer. ISBN: 3-540-40699-9.
- Burke, E. K., & Erben, W. (Eds.). (2001). *Lecture notes in computer science: Vol. 2079. Practice and theory of automated timetabling III: selected papers from the 3rd international conference*. Berlin: Springer. ISBN: 3-540-42421-0.
- Burke, E. K., & Kendall, G. (Eds.). (2005). *Search methodologies: introductory tutorials in optimisation and decision support techniques*. Berlin: Springer. ISBN: 0387234608.
- Burke, E. K., & Landa Silva, J. D. (2004). The design of memetic algorithms for scheduling and timetabling problems. In W. E. Hart, N. Krasnogor, & J. E. Smith (Eds.), *Studies in fuzziness and soft computing: Vol. 166. Recent advances in memetic algorithms and related search technologies* (pp. 289–312). Berlin: Springer.
- Burke, E. K., & Newall, J. P. (1999). A multi-stage evolutionary algorithm for the timetable problem. *IEEE Transactions on Evolutionary Computation*, 3(1), 63–74.
- Burke, E. K., & Newall, J. P. (2003). Enhancing timetable solutions with local search methods. In E. K. Burke & P. De Causmaecker (Eds.), *Lecture notes in computer science: Vol. 2740. Practice and theory of automated timetabling IV: selected papers from the 4th international conference* (pp. 195–206). Berlin: Springer.
- Burke, E. K., & Newall, J. P. (2004). Solving examination timetabling problems through adaptation of heuristic orderings. *Annals of Operational Research*, 129, 107–134.
- Burke, E. K., & Petrovic, S. (2002). Recent research directions in automated timetabling. *European Journal of Operational Research*, 140(2), 266–280.
- Burke, E. K., & Ross, P. (Eds.). (1996). *Lecture notes in computer science: Vol. 1153. Practice and theory of automated timetabling I: selected papers from the 1st international conference*. Berlin: Springer. ISBN: 3-540-61794-9.
- Burke, E. K., & Rudova, H. (Eds.). (2007). *Lecture notes in computer science: Vol. 3867. Practice and theory of automated timetabling VI: selected papers from the 6th international conference*. Berlin: Springer. ISBN: 978-3-540-77344-3.
- Burke, E. K., & Trick, M. (Eds.). (2005). *Lecture notes in computer science: Vol. 3616. Practice and theory of automated timetabling V: selected papers from the 5th international conference*. Berlin: Springer. ISBN: 3-540-30705-2.
- Burke, E. K., Elliman, D. G., & Weare, R. F. (1994). A genetic algorithm for university timetabling. In *Proceedings of the AISB workshop on evolutionary computing*, University of Leeds, UK, 11–13 April 1994.
- Burke, E. K., Elliman, D. G., Ford, P. H., & Weare, R. F. (1996a). Examination timetabling in British universities: a survey. In E. K. Burke & P. Ross (Eds.), *Lecture notes in computer science: Vol. 1153. Practice and theory of automated timetabling I: selected papers from the 1st international conference* (pp. 76–90). Berlin: Springer.
- Burke, E. K., Newall, J. P., & Weare, R. F. (1996b). A memetic algorithm for university exam timetabling. In E. K. Burke & P. Ross (Eds.), *Lecture notes in computer science: Vol. 1153. Practice and theory of automated timetabling I: selected papers from the 1st international conference* (pp. 241–250). Berlin: Springer.

- Burke, E. K., Jackson, K., Kingston, J. H., & Weare, R. (1997). Automated university timetabling: the state of the art. *The Computer Journal*, 40(9), 565–571.
- Burke, E. K., Kingston, J., & Pepper, P. A. (1998a). A standard data format for timetabling instances. In E. K. Burke & M. W. Carter (Eds.), *Lecture notes in computer science: Vol. 1408. Practice and theory of automated timetabling II: selected papers from the 2nd international conference* (pp. 215–224). Berlin: Springer.
- Burke, E. K., Newall, J. P., & Weare, R. F. (1998b). Initialisation strategies and diversity in evolutionary timetabling. *Evolutionary Computation*, 6(1), 81–103.
- Burke, E. K., Newall, J. P., & Weare, R. F. (1998c). A simple heuristically guided search for the timetable problem. In *Proceedings of the international ICSC symposium on engineering of intelligent systems (EIS98)* (pp. 574–579).
- Burke, E. K., MacCarthy, B., Petrovic, S., & Qu, R. (2000). Structured cases in CBR—Re-using and adapting cases for timetabling problems. *Knowledge-Based Systems*, 13(2–3), 159–165.
- Burke, E. K., Bykov, Y., & Petrovic, S. (2001). A multi-criteria approach to examination timetabling. In E. K. Burke & W. Erben (Eds.), *Lecture notes in computer science: Vol. 2079. Practice and theory of automated timetabling III: selected papers from the 3rd international conference* (pp. 118–131). Berlin: Springer.
- Burke, E. K., Hart, E., Kendall, G., Newall, J., Ross, P., & Schulenburg, S. (2003a). Hyper-heuristics: an emerging direction in modern search technology. In F. Glover & G. Kochenberger (Eds.), *Handbook of meta-heuristics* (pp. 457–474). Dordrecht: Kluwer.
- Burke, E. K., Kendall, G., & Soubeiga, E. (2003b). A tabu-search hyper-heuristic for timetabling and rostering. *Journal of Heuristics*, 9, 451–470.
- Burke, E. K., Bykov, Y., Newall, J. P., & Petrovic, S. (2004a). A time-predefined local search approach to exam timetabling problems. *IIE Transactions*, 36(6), 509–528.
- Burke, E. K., De Causmaecker, P., Vanden Berghe, G., & Van Landeghem, H. (2004b). The state of the art of nurse rostering. *Journal of Scheduling*, 7(6), 441–499.
- Burke, E. K., Eckersley, A. J., McCollum, B., Petrovic, S., & Qu, R. (2004c). Analysing similarity in examination timetabling. In: E. K. Burke, M. Trick (Eds.), *Proceedings of the 5th international conference on the practice and theory of automated timetabling* (pp. 89–106), Pittsburgh, PA, USA, August 2004.
- Burke, E. K., Kingston, J. H., & de Werra, D. (2004d). Applications to timetabling. In J. Gross & J. Yellen (Eds.), *The handbook of graph theory* (pp. 445–474). London: Chapman Hall/CRC.
- Burke, E. K., Dror, M., Petrovic, S., & Qu, R. (2005). Hybrid graph heuristics in hyper-heuristics applied to exam timetabling problems. In B. L. Golden, S. Raghavan, & E. A. Wail (Eds.), *The next wave in computing, optimisation, and decision technologies* (pp. 79–91). Maryland: Springer.
- Burke, E. K., Eckersley, A. J., McCollum, B., Petrovic, S., & Qu, R. (2006a). *Hybrid variable neighbourhood approaches to university exam timetabling* (Technical Report NOTTCS-TR-2006-2). School of Computer Science, University of Nottingham.
- Burke, E. K., Petrovic, S., & Qu, R. (2006b). Case-based heuristic selection for timetabling problems. *Journal of Scheduling*, 9, 115–132.
- Burke, E. K., McCollum, B., Meisels, A., Petrovic, S., & Qu, R. (2007). A graph based hyper-heuristic for exam timetabling problems. *European Journal of Operational Research*, 176, 177–192.
- Caramia, M., Dell’Olmo, P., & Italiano, G. F. (2001). New algorithms for examination timetabling. In S. Naher & D. Wagner (Eds.), *Lecture notes in computer science: Vol. 1982. Algorithm engineering 4th international workshop, proceedings WAE 2000* (pp. 230–241). Berlin: Springer.
- Caramia, M., Dell’Olmo, P., & Italiano, G. F. (2008). Novel local-search-based approaches to university examination timetabling. *INFORMS Journal of Computing*, 20(1), 86–99.
- Carter, M. W. (1983). *A decomposition algorithm for practical timetabling problems* (Technical Paper 83-06). Department of Industrial Engineering, University of Toronto.
- Carter, M. W. (1986). A survey of practical applications of examination timetabling algorithms. *Operations Research*, 34(2), 193–202.
- Carter, M. W., & Johnson, D. G. (2001). Extended clique initialisation in examination timetabling. *Journal of Operational Research Society*, 52, 538–544.
- Carter, M. W., & Laporte, G. (1996). Recent developments in practical examination timetabling. In E. K. Burke & P. Ross (Eds.), *Lecture notes in computer science: Vol. 1153. Practice and theory of automated timetabling I: selected papers from the 1st international conference* (pp. 3–21). Berlin: Springer.
- Carter, M. W., Laporte, G., & Chinneck, J. W. (1994). A general examination scheduling system. *Interfaces*, 24, 109–120.
- Carter, M. W., Laporte, G., & Lee, S. Y. (1996). Examination timetabling: algorithmic strategies and applications. *Journal of Operational Research Society*, 47(3), 373–383.
- Casey, S., & Thompson, J. (2003). GRASping the examination scheduling problem. In E. K. Burke & P. De Causmaecker (Eds.), *Lecture notes in computer science: Vol. 2740. Practice and theory of automated timetabling IV: selected papers from the 4th international conference* (pp. 232–244). Berlin: Springer.
- Chand, A. (2005). A constraint based genetic model for representing complete University timetabling data. In E. K. Burke & M. Trick (Eds.), *Lecture notes in computer science: Vol. 3616. Proceedings of the 5th international conference on the practice and theory of automated timetabling* (pp. 125–150). Berlin: Springer.
- Colijn, A. W., & Layfield, C. (1995a). Conflict reduction in examination schedules. In E. K. Burke & P. Ross (Eds.), *Proceedings of the 1st international conference on the practice and theory of automated timetabling*. (pp. 297–307), 30 August–1 September 1995. Edinburgh: Napier University.
- Colijn, A. W., & Layfield, C. (1995b). Interactive improvement of examination schedules. In E. K. Burke & P. Ross (Eds.), *Proceedings of the 1st international conference on the practice and theory of automated timetabling* (pp. 112–121), 30 August–1 September 1995. Edinburgh: Napier University.
- Cooper, T. B., & Kingston, J. H. (1996). The complexity of timetable construction problems. In E. K. Burke (Ed.), *Lecture notes in computer science: Vol. 1153. Practice and theory of automated timetabling I: selected papers from the 1st international conference* (pp. 283–295). Berlin: Springer.
- Corne, D., Ross, P., & Fang, H. (1994). Evolutionary timetabling: Practice, prospects and work in progress. In P. Prosser (Ed.), *Proceedings of UK planning and scheduling SIG workshop*.
- Corr, P. H., McCollum, B., McGreevy, M. A. J., & McMullan, P. (2006). A new neural network based construction heuristic for the examination timetabling problem. In *The international conference on parallel problem solving from nature (PPSN 2006)* (pp. 392–401), Reykjavik, Iceland, September 2006.
- Costa, D., & Hertz, A. (1997). Ant can colour graphs. *Journal of Operational Research Society*, 48, 295–305.
- Côté, P., Wong, T., & Sabouri, R. (2005). Application of a hybrid multi-objective evolutionary algorithm to the uncapacitated exam proximity problem. In E. K. Burke & M. Trick (Eds.), *Lecture notes in computer science: Vol. 3616. Practice and theory of automated timetabling V: selected papers from the 5th international conference* (pp. 151–168). Berlin: Springer.
- David, P. (1998). A constraint-based approach for examination timetabling using local repair techniques. In E. K. Burke & M. W. Carter (Eds.), *Lecture notes in computer science: Vol. 1408. Practice and theory of automated timetabling II: selected papers from the 2nd international conference* (pp. 169–186). Berlin: Springer.
- De Causmaecker, P., Lu, Y., Demeester, P., & Vanden Berghe, G. (2002). Using web standards for timetabling. In E. K. Burke &



- P. De Causmaecker (Eds.), *Proceedings of the 4th international conference on practice and theory of automated timetabling* (pp. 238–257), KaHo St.-Lieven, Gent, Belgium, 21–23 August 2002.
- de Werra, D. (1985). An introduction to timetabling. *European Journal of Operational Research*, 19, 151–162.
- de Werra, D. (1997). Restricted colouring models for timetabling. *Discrete Mathematics*, 165/166, 161–170.
- de Werra, D., Asratian, A. S., & Durand, S. (2002). Complexity of some special types of timetabling problems. *Journal of Scheduling*, 5, 171–183.
- Di Gaspero, L. (2002). Recolour, shake and kick: A recipe for the examination timetabling problem. In: E. K. Burke & P. De Causmaecker (Eds.), *Proceedings of the 4th international conference on practice and theory of automated timetabling* (pp. 404–407), KaHo St.-Lieven, Gent, Belgium, 21–23 August 2002.
- Di Gaspero, L., & Schaerf, A. (2001). Tabu search techniques for examination timetabling. In E. K. Burke & W. Erben (Eds.), *Lecture notes in computer science: Vol. 2079. Practice and theory of automated timetabling III: selected papers from the 3rd international conference* (pp. 104–117). Berlin: Springer.
- Di Gaspero, L., & Schaerf, A. (2003). EasyLocal++: an object-oriented framework for flexible design of local search algorithms. *Software, Practice & Experience*, 33(8), 733–765.
- Dimopoulou, M., & Miliotis, P. (2001). Implementation of a university course and examination timetabling system. *European Journal of Operational Research*, 130, 202–213.
- Dorigo, M., & Blum, C. (2005). Ant colony optimisation theory: a survey. *Theoretical Computer Science*, 344(2–3), 243–278.
- Dowsland, K. A. (1996). Simulated annealing solutions for multi-objective scheduling and timetabling. In V. J. R. Smith, I. H. Osman, C. R. Reeves, & G. D. Smith (Eds.), *Modern heuristic search methods* (pp. 155–166). New York: Wiley.
- Dowsland, K. A., & Thompson, J. (2005). Ant colony optimisation for the examination scheduling problem. *Journal of Operational Research Society*, 56, 426–438.
- Dueck, G. (1993). New optimization heuristics: the great deluge and the record-to-record travel. *Journal of Computational Physics*, 104, 86–92.
- Duong, T. A., & Lam, K. H. (2004). 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)* (pp. 205–210), Hanoi, Vietnam, 2–5 February, 2004.
- Easton, K., Nemhauser, G., & Trick, M. (2004). Sports scheduling. In J. Leung (Ed.), *Handbook of scheduling: algorithms, models, and performance analysis*. Boca Raton: CRC Press, Chap. 52.
- Eley, M. (2007). Ant algorithms for the exam timetabling problem. In E. K. Burke & H. Rudova (Eds.), *Lecture notes in computer science: Vol. 3867. Practice and theory of automated timetabling VI: selected papers from the 6th international conference* (pp. 364–382). Berlin: Springer.
- Erben, W. (2001). A grouping genetic algorithm for graph colouring and exam timetabling. In E. K. Burke & W. Erben (Eds.), *Lecture notes in computer science: Vol. 2079. Practice and theory of automated timetabling III: selected papers from the 3rd international conference* (pp. 132–156). Berlin: Springer.
- Ersoy, E., Özcan, E., & Etaner, A. S. (2007). Memetic algorithms and hyperhill-climbers. In *Proceedings of the 3rd multidisciplinary international conference on scheduling: theory and applications* (pp. 159–166), Paris, France, August 2007.
- Freuder, E. C., & Wallace, M. (2005). Constraint programming. In E. K. Burke & G. Kendall (Eds.), *Search methodologies: introductory tutorials in optimisation and decision support techniques* (pp. 239–272). Berlin: Springer.
- Gendreau, M., & Potvin, J. Y. (2005). Tabu search. In E. K. Burke & G. Kendall (Eds.), *Search methodologies: introductory tutorials in optimisation and decision support techniques* (pp. 165–186). Berlin: Springer. ISBN: 0387234608.
- Glover, F., & Kochenberger, G. A. (2003). *Handbook of meta-heuristics*. Dordrecht: Kluwer.
- Glover, F., & Laguna, M. (1993). Tabu search. In C. R. Reeves (Ed.), *Modern heuristic techniques for combinatorial problems*. Oxford: Scientific Publications.
- Goltz, H. J., & Matzke, D. (1999). University timetabling using constraint logic programming. In *Lecture notes in computer science: Vol. 1551. Practical aspects of declarative languages* (pp. 320–334). Berlin: Springer.
- Hansen, P., & Mladenović, N. (2001). Variable neighbourhood search: principles and applications. *European Journal of Operational Research*, 130, 449–467.
- Hansen, M. P., & Vidal, R. V. V. (1995). Planning of high school examinations in Denmark. *European Journal of Operational Research*, 87, 519–534.
- Hansen, M. P., Lauersen, V., & Vidal, R. V. V. (1995). Nationwide scheduling of examinations: lessons from experience. In E. K. Burke & P. Ross (Eds.), *Proceedings of the 1st international conference on the practice and theory of automated timetabling* (pp. 468–473), 30 August–1 September 1995. Edinburgh: Napier University.
- Ho, W. K., Lim, A., & Oon, W. C. (2001). Maximising paper spread in examination timetabling using a vehicle routing method. In *Proceedings of 13th IEEE international conference on tools with artificial intelligence (ICTAI01)* (pp. 359–366).
- Hussin, N. (2005). *Tabu search based hyper-heuristic approaches for examination timetabling*. PhD thesis, Department of Computer Science, University of Nottingham, UK, November 2005.
- Kendall, G., & Hussin, N. M. (2005a). A tabu search hyper-heuristic approach to the examination timetabling problem at the MARA university of technology. In E. K. Burke & M. Trick (Eds.), *Lecture notes in computer science: Vol. 3616. Practice and theory of automated timetabling V: selected papers from the 5th international conference* (pp. 199–218). Berlin: Springer.
- Kendall, G., & Hussin, N. M. (2005b). An investigation of a tabu search based hyper-heuristic for examination timetabling. In G. Kendall, E. Burke, & S. Petrovic (Eds.), *Selected papers from multidisciplinary scheduling: theory and applications* (pp. 309–328).
- Kingston, J. H. (1995). Bibliography on practice and theory of automated timetabling. <http://liinwww.ira.uka.de/bibliography/Misc/timetabling.html>.
- Kingston, J. H. (2001). Modelling timetabling problems with STTL. In E. K. Burke & W. Erben (Eds.), *Lecture notes in computer science: Vol. 2079. Practice and theory of automated timetabling III: selected papers from the 3rd international conference* (pp. 309–321). Berlin: Springer.
- Krasnogor, N., & Smith, J. E. (2005). A tutorial for competent memetic algorithms: model, taxonomy and design issues. *IEEE Transactions on Evolutionary Computation*, 9(5), 474–488.
- Kwan, R. (2004). Bus and train driver scheduling. In J. Leung (Ed.), *Handbook of scheduling: algorithms, models, and performance*. Boca Raton: CRC Press, Chap. 51.
- Landa Silva, J. D., Burke, E. K., & Petrovic, S. (2004). An introduction to multi-objective meta-heuristics for scheduling and timetabling. In X. Gandibleux, M. Sevaux, K. Sorensen, & V. Tkindt (Eds.), *Lecture notes in economics and mathematical systems: Vol. 535. Multiple objective meta-heuristics* (pp. 91–129). Berlin: Springer.
- Le Huédé, F., Grabisch, M., Labreuche, C., & Savéant, P. (2006). MCS—a new algorithm for multicriteria optimisation in constraint programming. *Annals of Operational Research*, 147, 143–174.
- Leake, D. B. (1996). *Case based reasoning: experiences, lessons, and future directions*. Menlo Park: AAAI Press/MIT Press.
- Lewis, R. (2008). A survey of metaheuristic-based techniques for university timetabling problems. *OR Spectrum*, 30(1), 167–190.

- Lim, A., Chin, A. J., Kit, H. W., & Oon, W. C. (2000). A campus-wide university examination timetabling application. In *Proceedings of the 17th national conference on artificial intelligence and 12th conference on innovative applications of artificial intelligence* (pp. 1020–1025).
- Lin, S. L. M. (2002). A broker algorithm for timetabling problem. In E. K. Burke & P. De Causmaecker (Eds.), *Proceedings of the 4th international conference on practice and theory of automated timetabling*, (pp. 372–386), KaHo St.-Lieven, Gent, Belgium, 21–23 August 2002.
- Lourenço, H. R., Martin, O., & Stützle, T. (2003). Iterated local search. In F. Glover & G. A. Kochenberher (Eds.), *Handbook of meta-heuristics* (pp. 321–354). Boston: Kluwer Academic.
- Malim, M. R., Khader, A. T., & Mustafa, A. (2006). Artificial immune algorithms for university timetabling. In E. K. Burke & H. Rudova (Eds.), *Proceedings of the 6th international conference on practice and theory of automated timetabling* (pp. 234–245), Brno, Czech Republic, August 2006.
- McCollum, B. G. C. (2007). A perspective on bridging the gap between theory and practice in university timetabling. In E. K. Burke & H. Rudova (Eds.), *Lecture Notes in Computer Science: Vol. 3867. Practice and theory of automated timetabling VI: selected papers from the 6th international conference* (pp. 3–23). Berlin: Springer.
- McCollum, B., McMullan, P., Burke, E. K., Parkes, A. J., & Qu, R. (2008). *The second international timetabling competition: examination timetabling track* (Technical Report QUB/IEEE/Tech/ITC2007/Exam/v1.0/1). Queen's Belfast University, N. Ireland.
- Mehta, N. K. (1981). The application of a graph colouring method to an examination scheduling problem. *Interfaces*, 11, 57–64.
- Mehta, N. K. (1982). A computer-based examination management system. *Journal of Educational Technology Systems*, 11, 185–198.
- Merkle, D., & Middendorf, M. (2005). Swarm intelligence. In E. K. Burke & G. Kendall (Eds.), *Search methodologies: introductory tutorials in optimisation and decision support techniques* (pp. 401–436). Berlin: Springer.
- Merlot, L. T. G., Boland, N., Hughes, B. D., & Stuckey, P. J. (2003). A hybrid algorithm for the examination timetabling problem. In E. K. Burke & P. De Causmaecker (Eds.), *Lecture notes in computer science: Vol. 2740. Practice and theory of automated timetabling IV: selected papers from the 4th international conference* (pp. 207–231). Berlin: Springer.
- Miles, R. (1975). Computer timetabling: a bibliography. *British Journal of Educational Technology*, 6(3), 16–20.
- Mladenović, N., & Hansen, P. (1997). Variable neighbourhood search. *Computers and Operations Research*, 24(11), 1097–1100.
- Moscato, P., & Norman, M. G. (1992). A “memetic” approach for the travelling salesman problem—implementation of a computational ecology for combinatorial optimisation on message passing systems. In *Proceedings of the international conference on parallel computing and transputer applications* (pp. 177–186). Amsterdam: IOS Press.
- Naji Azimi, Z. (2004). Comparison of metaheuristic algorithms for examination timetabling problem. *Applied Mathematics and Computation*, 16(1–2), 337–354.
- Naji Azimi, Z. (2005). Hybrid heuristics for examination timetabling problem. *Applied Mathematics and Computation*, 163(2), 705–733.
- Osman, I. H., & Laporte, G. (1996). Metaheuristics: a bibliography. *Annals of Operational Research*, 63, 513–628.
- Paquete, L., & Stützle, T. (2002). An experimental investigation of iterated local search for colouring graphs. In S. Cagnoni, J. Gottlieb, E. Hart, M. Middendorf, & G. Raidl (Eds.), *Lecture notes in computer science: Vol. 2279. Applications of evolutionary computing, EvoWorkshops 2002* (pp. 122–131). Berlin: Springer.
- Paquete, L., & Stützle, T. (2002). Empirical analysis of tabu search for the lexicographic optimisation of the examination timetabling problem. In E. K. Burke & P. De Causmaecker (Eds.), *Proceedings of the 4th international conference on practice and theory of automated timetabling* (pp. 413–420), KaHo St.-Lieven, Gent, Belgium 21–23 August 2002.
- Petrovic, S., & Burke, E. K. (2004). University timetabling. In J. Leung (Ed.), *Handbook of scheduling: algorithms, models, and performance analysis*. Boca Raton: CRC Press. Chap. 45.
- Petrovic, S., & Bykov, Y. (2003). A multi-objective optimisation technique for exam timetabling based on trajectories. In E. K. Burke & P. De Causmaecker (Eds.), *Lecture notes in computer science: Vol. 2740. Practice and theory of automated timetabling IV: selected papers from the 4th international conference* (pp. 179–192). Berlin: Springer.
- Pirlot, M. (1996). General local search methods. *European Journal of Operational Research*, 92, 493–511.
- Qu, R., & Burke, E. K. (2007). Adaptive decomposition and construction for examination timetabling problems. In *Multidisciplinary international scheduling: theory and applications (MISTA'07)* (pp. 418–425), Paris, France, August 2007.
- Qu, R., & Burke, E. K. (2008, accepted). Hybridisations within a graph based hyper-heuristic framework for university timetabling problems. *Journal of Operational Research Society*.
- Ranson, D., & Ahmadi, S. (2007). An extensible modelling framework for timetabling problems. In E. K. Burke & H. Rudova (Eds.), *Lecture notes in computer science: Vol. 3867. Practice and theory of automated timetabling VI: selected papers from the 6th international conference* (pp. 383–393). Berlin: Springer.
- Reeves, C. R. (Ed.). (1993). *Modern heuristic techniques for combinatorial problems*. Oxford: Scientific Publications.
- Reeves, C. R. (2005). Fitness landscapes. In E. K. Burke & G. Kendall (Eds.), *Search methodologies: introductory tutorials in optimisation and decision support techniques* (pp. 587–610). Berlin: Springer.
- Reis, L. P., & Oliveira, E. (1999). Constraint logic programming using set variables for solving timetabling problems. In *12th international conference on applications of Prolog*.
- Reis, L. P., & Oliveira, E. (2001). A language for specifying complete timetabling problems. In E. K. Burke & W. Erben (Eds.), *Lecture notes in computer science: Vol. 2079. Practice and theory of automated timetabling III: selected papers from the 3rd international conference* (pp. 322–341). Berlin: Springer.
- Resende, M. G. C., & Ribeiro, C. C. (2003). Greedy randomised adaptive search procedures. In F. Glover & G. A. Kochenberher (Eds.), *Handbook of meta-heuristics* (pp. 219–249). Dordrecht: Kluwer.
- Romero, B. P. (1982). Examination scheduling in a large engineering school: a computer assisted participative procedure. *Interfaces*, 12, 17–23.
- Ross, P. (2005). Hyper-heuristics. In E. K. Burke & G. Kendall (Eds.), *Search methodologies: introductory tutorials in optimisation and decision support techniques* (pp. 529–556). Berlin: Springer. Chap. 17.
- Ross, P., Corne, D., & Terashima-Marin, H. (1996). The phase transition niche for evolutionary algorithms in timetabling. In E. K. Burke & P. Ross (Eds.), *Lecture notes in computer science: Vol. 1153. Practice and theory of automated timetabling I: selected papers from the 1st international conference* (pp. 309–324). Berlin: Springer.
- Ross, P., Hart, E., & Corne, D. (1998). Some observations about GA-based exam timetabling. In E. K. Burke & M. W. Carter (Eds.), *Lecture notes in computer science: Vol. 1408. Practice and theory of automated timetabling II: selected papers from the 2nd international conference* (pp. 115–129). Berlin: Springer.
- Ross, P., Hart, E., & Corne, D. (2003). Genetic algorithms and timetabling. In A. Ghosh & S. Tsutsui (Eds.), *Advances in evolu-*

- tionary computing: theory and applications (pp. 755–771). New York: Springer.
- Ross, P., Marin-Blazquez, J. G., & Hart, E. (2004). Hyper-heuristics applied to class and exam timetabling problems. In *Proceedings of the 2004 congress on evolutionary computation (CEC2004)* (pp. 1691–1698). Washington: IEEE Press.
- Sabin, G. C. W., & Winter, G. K. (1986). The impact of automated timetabling on universities—a case study. *Journal of Operational Research Society*, 37, 689–693.
- Sastry, K., Goldberg, D., & Kendall, G. (2005). Genetic algorithms. In E. K. Burke & G. Kendall (Eds.), *Search methodologies: introductory tutorials in optimisation and decision support techniques* (pp. 97–125). Berlin: Springer. ISBN: 0387234608.
- Schaerf, A. (1999). A survey of automated timetabling. *Artificial Intelligence Review*, 13(2), 87–127.
- Schaerf, A., & Di Gaspero, L. (2007). Measurability and reproducibility in university timetabling research: discussion and proposals. In E. K. Burke & H. Rudova (Eds.), *Lecture notes in computer science: Vol. 3867. Practice and theory of automated timetabling VI: selected papers from the 6th international conference* (pp. 40–49). Berlin: Springer.
- Schmidt, E. A., & Strohlein, T. (1979). Timetable construction—an annotated bibliography. *The Computer Journal*, 23, 307–316.
- Sheibani, K. (2002). An evolutionary approach for the examination timetabling problems. In E. K. Burke & P. De Causmaecker (Eds.), *Proceedings of the 4th international conference on practice and theory of automated timetabling* (pp. 387–396), KaHo St.-Lieven, Gent, Belgium, 21–23 August 2002.
- Simonis, H. (1995). The CHIP system and its applications. In *Lecture notes in computer science: Vol. 976. First international conference on principles and practice of constraint programming* (pp. 643–646). Berlin: Springer.
- Socha, K., Sampels, M., & Manfrin, M. (2003). Ant algorithms for the university course timetabling problem with regard to state-of-the-art. In *Proceedings of the 3rd European workshop on evolutionary computation in combinatorial optimisation* (pp. 334–345), Essex, UK, April 2003.
- Terashima-Marin, H., Ross, P., & Valenzuela-Rendon, M. (1999a). Evolution of constraint satisfaction strategies in examination timetabling. In *Proceedings of the genetic and evolutionary conference* (pp. 635–642), Orlando, FL.
- Terashima-Marin, H., Ross, P., & Valenzuela-Rendon, M. (1999b). Clique-based crossover for solving the timetabling problem with GAs. In *Proceedings of 1999 IEEE congress on evolutionary computation* (pp. 1200–1206). Washington: IEEE Press.
- Terashima-Marin, H., Ross, P., & Valenzuela-Rendon, M. (1999c). Application of the hardness theory when solving the timetabling problem with GAs. In *Proceedings of the congress on evolutionary computation 1999* (pp. 604–611). Washington: IEEE Press.
- Thompson, J., & Dowsland, K. (1996). Variants of simulated annealing for the examination timetabling problem. *Annals of Operational Research*, 63, 105–128.
- Thompson, J., & Dowsland, K. (1998). A robust simulated annealing based examination timetabling system. *Computers & Operations Research*, 25, 637–648.
- Tsang, E., Mills, P., & Williams, R. (1999). A computer aided constraint programming system. In *The 1st international conference on the practical application of constraint technologies and logic programming (PACLP)* (pp. 81–93).
- Ulker, O., Özcan, E., & Korkmaz, E. E. (2007). Linear linkage encoding in grouping problems: applications on graph colouring and timetabling. In E. K. Burke & H. Rudova (Eds.), *Lecture notes in computer science: Vol. 3867. Practice and theory of automated timetabling VI: selected papers from the 6th international conference* (pp. 347–363). Berlin: Springer.
- Van Hentenryck, P. (1989). *Logic programming series. Constraint satisfaction in logic programming*. Cambridge: MIT Press.
- Van Hentenryck, P. (1999). *The OPL optimisation programming language*. Cambridge: MIT Press.
- Welsh, D. J. A., & Powell, M. B. (1967). The upper bound for the chromatic number of a graph and its application to timetabling problems. *The Computer Journal*, 11, 41–47.
- White, G. M. (2000). Constrained satisfaction, not so constrained satisfaction and the timetabling problem. In E. K. Burke & W. Erben (Eds.), *Proceedings of the 3rd international conference on the practice and theory of automated timetabling* (pp. 32–47), Constance, Germany, 16–18 August 2000.
- White, G. M., & Xie, B. S. (2001). Examination timetables and tabu search with longer-term memory. In E. K. Burke & W. Erben (Eds.), *Lecture notes in computer science: Vol. 2079. Practice and theory of automated timetabling III: selected papers from the 3rd international conference* (pp. 85–103). Berlin: Springer.
- White, G. M., Xie, B. S., & Zonjic, S. (2004). Using tabu search with longer-term memory and relaxation to create examination timetables. *European Journal of Operational Research*, 153(16), 80–91.
- Wong, T., Côté, P., & Gely, P. (2002). Final exam timetabling: a practical approach. In *IEEE Canadian conference on electrical and computer engineering (CCECE 2002)* (Vol. 2, pp. 726–731).
- Wood, D. C. (1968). A system for computing university examination timetables. *The Computer Journal*, 11(1), 41–47.
- Wren, A. (1996). Scheduling, timetabling and rostering—a special relationship? In E. K. Burke & P. Ross (Eds.), *Lecture notes in computer science: Vol. 1153. Practice and theory of automated timetabling I: selected papers from the 1st international conference* (pp. 46–75). Berlin: Springer.
- Yang, Y., & Petrovic, S. (2005). A novel similarity measure for heuristic selection in examination timetabling. In E. K. Burke & M. Trick (Eds.), *Lecture notes in computer science: Vol. 3616. Practice and theory of automated timetabling V: selected papers from the 5th international conference* (pp. 377–396). Berlin: Springer.
- Zeleny, M. (1974). A concept of compromise solutions and method of displaced ideal. *Computers & Operations Research*, 1(4), 479–496.