A Comprehensive Study of Educational Timetabling - a Survey

Kristiansen, Simon; Stidsen, Thomas Jacob Riis

Publication date:
2013

Document Version
Publisher's PDF, also known as Version of record

Citation (APA):
A Comprehensive Study of Educational Timetabling, a Survey

Simon Kristiansen
Thomas Riis Stidsen

November 2013

Report 8.2013

DTU Management Engineering
A Comprehensive Study of Educational Timetabling
- a survey

Simon Kristiansen\textsuperscript{a,b,*}, Thomas R. Stidsen\textsuperscript{a}

\textsuperscript{a}Section of Operations Research, Department of Management Engineering, Technical University of Denmark, Building 436, Produktionstorvet, DK-2800 Kgs. Lyngby, Denmark
\textsuperscript{b}MaCom A/S, Vesterbrogade 48, 1., DK-1620 Copenhagen V, Denmark

Abstract

Educational timetabling is one of the most researched subjects within the range of timetabling problems. There has been a significant increase in efficient planning problems within educational timetabling the last couple of decades. In this paper we will highlight some of the main trends and research achievements within educational planning problems. Furthermore it is an aim to make a differentiation between the different planning problems. This survey is concentrated on the four main education planning problems; University Course Timetabling, High School Timetabling, Examination Timetabling and Student Sectioning. Firstly a presentation of educational timetabling and the main components is given. For each problem a description is given with appertaining benchmark data and recent research. The literature presented is mainly solution tested on real-life data or better yet implemented. Summarizing tables for each section are presented to give an overall view of all the literature of this survey.

Keywords: Educational Timetabling, University Course Timetabling, High School Timetabling, Examination Timetabling, Student Sectioning, Benchmark datasets, Search methodologies
1. Introduction

Timetabling has been a challenging and important problem within operations research for several decades and it still is. In Wren [1996] timetabling is defined as:

"Timetabling is the allocation, subject to constraints, of given resources to objects being placed in space time, in such way as to satisfy as nearly as possible a set of desirable objectives."

Due to the broad definition the timetabling problems covers various forms of real-world problems, including Employee Timetabling (e.g. Balakrishnan and Wong [1990] and Meisels and Schael [2003]), Rostering Problems (e.g. Cheang et al. [2003] and Burke et al. [2004b]), Sports Timetabling (e.g. Easton et al. [2003] and Kendall et al. [2010]) and Educational Timetabling (e.g. Burke and Newall [1999], Melício et al. [2005] and Kristiansen et al. [2011]).

This survey will be concentrated on Educational Timetabling and the different problems within this area. Educational timetabling encompasses problems such as University Course Timetabling, High School Timetabling, Examination Timetabling and Student Sectioning, and is encountered in all institutions in the educational system throughout the world and it is one of the mostly studied timetabling problems from a practical viewpoint. The problems are very difficult and important for the institutions, and several approaches have been used to create good and feasible solutions. The purpose of this paper is to give a comprehensive overview of the different planning problems within education timetabling with the recent developments and trends. Table 1 lists the educational timetabling problem which we are going to discuss in this paper.

It should be mentioned that it is not attempted to perform any experimental comparison on the different methods used, only to give an overview of the methods.

Furthermore, in the effort of producing an exhaustive review of all the planning problems in the education system, we are conscious that some references have been left out and we will therefore apologize in advance. However the goal is not to list all scientific papers on this subject, but to create an overview which illustrates the different types of planning problems within the educational system and the diversity of solution methods used on these problems.

1.1. Previous Surveys and Competitions

The literature on educational timetable includes several surveys on educational timetables. A briefly discussion of the scope of the surveys is given here sorted into chronological order.
<table>
<thead>
<tr>
<th>Problem</th>
<th>Short description</th>
<th>Newest survey</th>
</tr>
</thead>
<tbody>
<tr>
<td>University Course Timetabling</td>
<td>Aims to break university courses into sections, and assign times, students and rooms. Either Curriculum-based or Enrollment-based</td>
<td>McCollum (2006) motivates in bridging the gap between theory and practice within University Course Timetabling</td>
</tr>
<tr>
<td>High School Timetabling</td>
<td>Allocating classes to time slots, teachers and rooms to satisfy the restrictions.</td>
<td>Pillay (2013) gives a definition and an overview of the High School Timetabling problem.</td>
</tr>
<tr>
<td>Examination Timetabling</td>
<td>Scheduling exams for students under the limited resources.</td>
<td>Qu et al. (2009) is a comprehensive study of Examination Timetabling.</td>
</tr>
<tr>
<td>Student Sectioning</td>
<td>Assigning student to sections of courses while respecting the requests of the individual students.</td>
<td>There is no recent survey on Student Sectioning.</td>
</tr>
</tbody>
</table>

Some of the earliest surveys include [Schmidt and Ströhlein (1980), de Werra (1985) and Junginger (1986)]. Schmidt and Ströhlein (1980) provide an annotated bibliography containing more than 200 papers, and hence listing all the papers on the field up to 1979. de Werra (1985) introduces the various problems within education timetabling and provides with different models to the Class-Teacher Timetabling and Course Timetabling based on graph theory, while Junginger (1986) describes the various software products implemented for solving school timetabling in Germany.

The first survey on Examination Timetabling was presented in Carter (1986). None of the approaches mentioned in this survey was implemented in more than one institute. The survey was updated in Carter and Laporte (1996) where the approaches used on Examination Timetabling between 1986 and 1996 is summarized. The criteria for discussion of this paper, was that the solution methods should either have been implemented in a real world application or tested on real life data.

Bardadym (1996) considers computer-aided timetabling for high schools and universities from 1960 to 1995. The paper discusses the core items in
the different problems and gives a small discussion on open issues for future research within timetabling.

In Carter and Laporte (1998) the authors has changed their focus from Examination Timetabling to Course Timetabling. The major components of the University Course Timetabling problem are described in details as well as a discussion of the primary solution methods used. The paper summarize by listing up papers which contains approaches which are either implemented or tested on real data.

In Schaefer (1999a) the difference between Course Timetabling and Examination Timetabling is observed as being relatively small and that the problems could be modeled using the same model. Various formulations of School Timetabling, Course Timetabling and Examination Timetabling is surveyed along with the techniques used for solving the problems.

Burke and Petrovic (2002) give an overview of some of the recent developments that have been carried out in the Automated Scheduling, Optimization and Planning Research Group (ASAP) at the University of Nottingham.

Burke et al. (2004c) gives an introduction to the field of Educational and Sport Timetabling. The paper discusses the application of graph coloring methods to the timetabling problems and thereby highlights the fact that graph coloring have been an important part of timetabling problems in several decades.

McCollum (2006) provides information regarding research on University Course Timetabling up to 2006. The aim is to motivate researchers to bridge the gap between research and practice within University Course Timetabling. The paper rounds up by listing the major challenges working within Examination Timetabling and University Course Timetabling.

Qu et al. (2009) is the far newest survey on Examination Timetabling. The basis of the paper is Carter and Laporte (1996) and it is therefore concentrated on papers published between 1996 and 2009. The different algorithm approaches are classified and discussed. Furthermore the paper renames the existing problem datasets to avoid a significant amount of confusion which have been a problem for many years. The paper rounds off with an estimate on future research direction within examination timetabling.

Pillay (2013) is the latest survey and the first survey concentrated only on school timetabling. The survey gives a definition of the school timetabling, the different hard and soft constraints and it gives a detailed overview on the solution methods used.

The book Automated Scheduling and Planning From Theory to Practice contains a chapter on Educational Timetabling (Kingston (2013a)). The chapter gives a brief introduction to the different educational planning problems.

Some conferences are dedicated to the art of timetabling, such as the
International Conference on the Practice and Theory of Automated Timetabling (PATAT) and the Multidisciplinary International Scheduling Conference: Theory & Application (MISTA) Both conferences are held every second year.

Besides the mentioned surveys and conferences there has been three International Timetabling Competitions (ITC) on educational timetabling problems. These competitions have contributed to an increased focus on educational timetabling problems and thus the research within.

The First International Timetabling Competition was held in 2003 (ITC2003) and was regarding University Course Timetabling. The description of the problem can be found on the website of the competition (Paechter et al. (2002)). The results were presented at PATAT in 2004.

The Second International Timetabling Competition of 2007 had three tracks; Enrollment-based University Course Timetabling, Curriculum-based University Course Timetabling and Examination Timetabling (Gaspero et al. (2007), McCollum et al. (2010)) The winner of each track were announced at PATAT in 2008. Problem description can be found at the competition website (McCollum (2007)).

The third and most resent is the ITC2011 with focus on High School Timetabling (Post et al. (2011c)) and the results were presented at PATAT in 2012. The problem description and competition rules can be found at the website of the competition (Post et al. (2011)).

The results of the different competitions are described later in this paper.

From the above listing, it is seen that there exist several excellent surveys on educational timetabling. This paper concentrates on articles tested on real data or better yet implemented. I.e. it tries to follow the approach introduced in Carter and Laporte (1996) and Carter and Laporte (1998). We try to restrict the paper to only contain newly published papers, i.e. papers published since 2003, however we are aware that some of the educational problems are so sparse represented in the literature that earlier papers might be considered for these problems.

The surveys and competitions described above are listed in Table A.2 and Table A.3 in Appendix A.

1.2. Outline of the paper

The educational timetabling problems are in this survey divided into four main categories: University Course Timetabling, High School Timetabling, Examination Timetabling and Student Sectioning. However, as aforementioned, other surveys have a different view and it can be discussed whether some of the categories could be combined.

The article is organized as follows. In Section 2 a description of the general timetabling problem in the education system is given. In the following four sections, each section treats their own planning problem. Univer-
2. Planning Problems and the Components

The definition timetabling is often used when solving problems with some form of personnel allocation, such as University Course Timetabling. Due to the personnel allocations, many people are affected by the timetables created. For the combined planning problems in the educational system four main stakeholders are identified. Each with their own set of aims and wants.

- **The administration**: Solves the planning problems and sets the minimum standards that the timetables must conform to. The capacity of the room, consecutive exams, etc.

- **The students**: This is by far the largest in volume of the four stakeholders. The students seldom have much influence on the outcome of the different planning problems. And due to the number of students involved it is difficult to declare what is the best timetable for the students as they all might have different wants and needs.

- **The teachers**: The teachers have a little more influence on the outcome than the students and they do also requests a compressed timetables. Furthermore they might have some restrictions on which day and time they want to give lessons or exams.

- **The departments/sections**: Often only relevant for universities as the high schools rarely are organized with departments or sections. The department can have some restrictions or requests for the exam and course timetabling.

For all the various planning problems some restrictions always apply, these are the so-called "first-order conflicts". E.g. no person can be at more than one place at any time. These first-order constrains are always present in some form. Besides these constraints, there is a great variation between the problems, but also within a given problem there is great variation due to different universities, high schools and/or the Ministries of Education of the respective countries. Hence the other constraints within educational timetabling are many and varied. In the following some of the most common types are listed.

- **Resource assignment**: Resources are rarely preassigned to lectures and in these cases the assigning of resources is a part of the decision
problem. The most common resources to assign are rooms. Some lectures prefer or require to be held in a particular class. Furthermore the number of students in a given room must not exceed the capacity of the room.

- **Continuity**: Constraints that ensure that certain features of the timetables are constant or predictable. E.g. lectures of the same course should be scheduled in the same room or at the same time of days.

- **Compactness**: These constraints are designed to produce a more compact timetable for both students and teachers. For the students a compact timetable is one with a low number of idle time slots in between lectures. A teacher might prefer to have all his lectures in fewer days and thereby have some days off.

- **Spreading**: Often used as contrary to compactness. Meetings should be spread out in time. However it is also used on single meeting. If the same meeting should be repeated twice or more times in the timetable it is often preferable if these similar meetings a spread out.

- **Time assignment**: These constraints are used for assigning meeting to a time. This can be used to specify days on which teachers are unavailable or if one particular meeting must take place after another one, e.g. exercises should be placed right after the corresponding lectures. It is also used for meetings that should be held simultaneous.

These types of constraints are somewhat presented in educational timetabling as we will see in the following sections.

### 2.1. Annual Cycle of Educational Timetabling Problems

Whether it applies to high schools or universities, all the planning problems at an educational institute are recurring annual administrative tasks. There are differences in the planning problems that the various institutions have in their annual cycle, but in general the annual cycle of educational planning problems contains three main subjects; *Student Sectioning* followed by *University Course Timetabling* or *High School Timetabling* and finally the *Examination Timetabling*. Figure 1 illustrates the annual cycle of planning problems of a typical educational institute.

For some institutes some of the planning problems might be combined or swap positions in the cycle, or additional planning problems might be added to the cycle. At some universities Student Sectioning is used before and after the course timetabling.
Figure 1: A typical annual cycle of planning problems at an educational institute. Each year starting with Student Sectioning, then Timetabling (either University Course Timetabling or High School Timetabling) and finally the Examination Timetabling before repeating for the next school year.
3. University Course Timetabling

University Course Timetabling problem is the task of assigning lectures of courses to time slots, rooms and other resources, subject to constraints applied for the single student. The large size and structure of universities often makes the problem of solving the course timetabling so complicated that it is solved by several experts scattered across different faculties and departments. Due to the complexity of the problem, it is quite common that the universities re-use the solution from previous years when solving the problem.

The University Course Timetabling is often divided into two general approaches; the Curriculum-based University Course Timetabling and the Enrollment-based University Course Timetabling.

Courses which can be taken in combination, because they are needed to satisfy the degree rules of the given study, are called curricula. And assigning courses to time slots based on this is called Curriculum-based Course Timetabling. The problem consists of creating the weekly schedules of lectures where conflicts between the courses are set accordingly to the curriculum specified by the university. [Gaspero et al. (2007)] gives a short description of the Curriculum-based Course Timetabling Problem applied for Track 2 of ITC2007. Several research papers have been working on this university course approach. (Eg. Lach and Lübbeck (2012); Lü et al. (2011); Burke et al. (2012); Cacciani et al. (2013))

The second approach is the Post Enrollment-based Course Timetabling. Based on the enrollment data of each individual student, it is determined where courses are placed in the timetable such that all students can attend the events on which they are enrolled. This approach has been given various names in the literature including the Class Timetabling Problem and Event Timetabling. [Lewis et al. (2007)] gives a description of the Post Enrollment-based Course Timetabling Problem applied for Track 3 of ITC2007. Papers working on the Post Enrollment-based Course Timetabling include Cam-bazard et al. (2008); Ceschia et al. (2012) and Nothegger et al. (2012).

The two approaches are not necessary mutually exclusive. The universities can use the curriculum based approach in an early state to get the basic structure of the timetable, and then later on use the enrollment-based approach to improve the timetable.

Though the literature often is divided into the two approaches, many of the constraints remain the same. Below some of the most common constraints for University Course Timetabling are listed.

- Primary hard constraints
  - No 1st-order conflicts for students and teachers.
  - Only one event is put into each room in any time slot.
- Rooms should satisfy all of the features required by the event.

- Primary soft constraints
  - The capacity of the room should be respected.
  - Compact timetable of each student or curricula.
  - Room stability. Lectures of same course should be assigned the same room.
  - Time and/or room preferences.

3.1. International Timetabling Competition

In the First International Timetabling Competition in 2003 (ITC2003), the focus was Post Enrollment-based University Course Timetabling. The winner became Kostuch (2004) with a Simulated Annealing based heuristic.

The most recent competition on University Course Timetabling is the Second International Timetabling Competition in 2007 (ITC2007) (McCollum et al. (2010)). The competition consisted of three tracks where two were upon University Course Timetabling, with enrollment-based in Track 2 and curriculum-based in Track 3. Track 1 treated the Examination Timetabling problem.

The winner of Track 2 was a team from Ireland solving the Enrollment-based University Course Timetabling using a hybridized local search algorithm with a Constraint Programming approach in a Large Neighborhood Search scheme to address the hardness of finding feasible solutions (Cambazard et al. (2008)).

Track 3 was won by Tomáš Müller using a hybrid heuristic using three phases (Müller (2009)). Firstly a feasible initial solution is found using an Iterative Forward Search algorithm. In phase two the local optimum is found using a Hill Climbing algorithm and third phase is using Great Deluge techniques. This heuristic solver was among the finalist for all three tracks, and won Track 1 and 3.

3.2. Benchmark Data

There is a major lack of benchmark data for University Course Timetabling. ITC2007 contains datasets for both the Enrollment-based Course Timetabling and the Curriculum-based Course Timetabling. The datasets are available at the website of the competition McCollum (2007).

- ITC2007 Track2 - Enrollment-based Course Timetabling

Twenty-four datasets are available; all were created using an automated problem generator designed by the competition organizers, and all are known to feature at least one perfect solution - that is, a solution with no hard or soft constraint violations. The drawback of this benchmark is that it is generated data and not real-life.
• **ITC2007 Track3 - Curriculum-based Course Timetabling**
  Twenty-one instances were released for this track. All instances are real data and come from the University of Udine and for all instances there exists at least one feasible solution.

  Bonutti et al. (2012) collects the twenty-one instances of ITC2007 and four instances from Gaspero and Schaerf (2003) together with seven new instances mainly from Italian universities. The datasets are all real cases from various universities, mainly from University of Udine. In the following these datasets are denoted as one benchmark called *The Udine benchmark data* UdineBenchmark.

• **The Purdue benchmark data**
  On the website of the educational scheduling system UniTime, there is access to real data from the University of Purdue, United States. The benchmark consists of data sets for each department and one for the combined problem of three years (PurdueBenchmark).

3.3. Recent Research

University Course Timetabling is one of the most researched subjects within educational timetabling. The last decade many of the used methodologies represent some sort of hybridization of a number of techniques. In the following the scientific papers on University Course Timetabling of the last decade are categorized into the main methodologies used.

3.3.1. Swarm Intelligence Algorithms

Swarm intelligence algorithms belong to the family of population based techniques. Based on some sort of agents interacting locally with one another and with their environment. E.g. in Ant Colony algorithms, the ants search the shortest path to food by laying pheromones on the way. The shortest path is the path with the strongest level of pheromones.

In Socha et al. (2003) two different Ant Colony methods is used to solve the University Course Timetabling; Ant Colony System and MAX-MIN Ant System. In each step of both algorithms, every ant constructs a complete assignment of events to time slots using heuristics and pheromone information. The timetables are then improved using a local search procedure.

Ant Colony is also the method used in Nothegger et al. (2012) to solve the Post Enrollment-based Course Timetabling problem. The algorithm uses two distinct but relatively compact pheromone matrices in combination with an effective procedure to exploit their information in the heuristic solution construction. The algorithm was tested on the benchmark datasets from ITC2007 with the same terms as the competition. It had the best solution for 11 out of the 24 instances compared to the 5 finalists, including ties, which would have given a fourth place. However the algorithm did show a
large variation in the solution quality as it produced both the best and worst solution for several instances.

In Turabieh et al. (2010) a simulation of a fish swarm is applied to the same University Course Timetabling definition used in Socha et al. (2003). The idea of Fish Swarm Intelligent is to simulate the behavior of fishes while searching for food. The movements of the fishes are based on a Nelder-Mead simplex algorithm. Two types of local search were applied to enhance the quality of the solution. A multi decay rate Great Deluge and a steepest descent algorithm. The same method was later used to solve the Examination Timetabling problem (Turabieh and Abdullah (2011b)).

Another Swarm Intelligence is the Particle Swarm optimization, which is used on University Course Timetabling in Shiau (2011) and in Chen and Shih (2013). The Particle Swarm Intelligence algorithm consists of a swarm of particles in the space. The position of a particle is indicated by a vector which presents a solution and the movements of the particles are guided by their own best known position in the search-space as well as the entire swarm’s best known position. Shiau (2011) solves the course timetabling problems for a typical university in Taiwan by applying a hybrid particle swarm optimization. Each particle is updated on the basis of continuous particle swarm optimization formulas and local search. Chen and Shih (2013) evaluates on two different kind of particle swarm heuristics used on the problem, an inertia weight version and a constriction version. An interchange heuristic was applied to enhance the quality of the solution. The algorithms were tested on a single dataset where the Constriction Particle Swarm with interchange heuristic performed best.

3.3.2. Evolutionary Algorithms

Evolutionary algorithms do also belong to the family of population based techniques. The most common evolutionary algorithm is the Genetic Algorithm. Genetic algorithms are search heuristic that mimics the process of natural selection.

In Nurmi and Kyngas (2008) the Curriculum-based Course Timetabling problem is solved using an evolutionary algorithm. Firstly the Curriculum-based Course Timetabling is turned into the School Timetabling problem using a conversion scheme. The problem is then solved using a genetic algorithm (Nurmi and Kyngas (2007)). The algorithm consists of one mutation operator and no recombination operators. The most important features of the algorithm are a greedy hill-climbing mutation operator and a adaptive genetic penalty method.

Suyanto (2010) describes an informed generic algorithm for the University Course Timetabling and Student Sectioning. Firstly a greedy heuristic creates some feasible solution were all the hard constraints are satisfied. Then a directed mutation scheme is used to reduce the violations of the soft constraints while keeping the solution feasible.
3.3.3. Local Search Algorithms

Local search based techniques are classified as meta-heuristics and the algorithms move from solution to solution in the search space by applying local changes. Local search algorithms include methods such as Tabu Search, Simulated Annealing (SA) and Greedy Randomized Adaptive Search Procedures (GRASP). Within educational timetabling local search has been widely used and in the following some of the local search based techniques used on University Course Timetabling in the last decade are presented.

Gaspero and Schaerf (2003) propose a set of multi-neighborhood search strategies to improve local search capabilities. It is showed how different neighborhoods operators can be combined. The operators chosen are Tabu Search and Hill-Climber. The combinations used are neighborhood-union, neighborhood-composition and token-ring search. A perturbation operator called "kicker" is used to help improve the search and hence avoid local optima. The algorithm is tested on four real instances from the School of Engineering at Udine University.

Another neighborhood analysis is made in Lü et al. (2011). Four neighborhoods are investigated based on three evaluation criteria: percentage of improving neighbors, improvement strength and search steps. The neighborhoods in this paper is SimpleMove, SimpleSwap and two kinds of KempeMove. To understand the behavior of the neighborhoods and the combinations, they are tested using a steepest descent algorithm. To further evaluate the impact, a series of experiments are conducted using three algorithms: Tabu Search, Iterate Local Search and Adaptive Tabu Search.

Simulated Annealing is another local search algorithm. The idea is to search a wider area of the search space by allowing the algorithm to accept worse moves with a higher probability in the beginning of the search. The acceptance criterion is controlled with a temperature which decades based on a cooling schedule. Kostuch (2004) participated in ITC2003 and won using a heuristic based on Simulated Annealing. Firstly a feasible timetable is constructed, and the timetable is then improved using SA.

Murray et al. (2007) transforms the University Course Timetabling problem at the Purdue University into a constraint satisfaction and optimization problem (CSOP). CSOP consists of a set of variables having finite domain, a set of hard constraints and an objective function. The solver used an iterative forward search algorithm. The paper is a part of the research embedded in the system UniTime.

In Ceschia et al. (2012) the Post Enrollment-based Course Timetabling problem is solved using SA. The SA used is with probabilistic acceptance and a geometric cooling scheme. The algorithm is tested on the benchmark of ITC2003, ITC2007 and the data used in Lewis and Paechter (2005).

Variable Neighborhood Search (VNS) algorithms are used to solve the
Curriculum-Based Course Timetabling in \cite{Nguyen2011}. Eight Variable Neighborhood Search algorithms with different implementing strategies were created. A basic VNS and seven popular variants, where the Fleszar-Hindi extension of the basic VNS in general was the most effective in solving the 14 instances from University of Science in Vietnam.

\cite{Chiarandini2006} uses a hyper-heuristic which combines various construction heuristics, Tabu Search, Variable Neighborhood Descent and Simulated Annealing. The algorithm is divided in two parts. In the first part construction heuristics are used to solve the hard constraints of the problem. In the second phase the penalty of soft constraints are improved by first using Variable Neighborhood Search, and then using Simulated Annealing. The authors make use of the racing procedure, the \textit{F-race} method, to automatically tune the parameters of the algorithm. The algorithm is tested on the benchmark of ITC2003.

### 3.3.4. Graph Coloring Algorithms

In \cite{Burke2007} a graph-based hyper heuristic is developed for University Course Timetabling and Examination Timetabling. A Tabu Search approach is employed to search for permutations of graph heuristics which are used for constructing timetables. For the University Course Timetabling the approach is tested on the datasets from ITC2007 and the results were competitive with the previous approaches reported in the literature.

Many papers work with a two-phase approach for solving University Course Timetabling. A construction phase and an improvement phase. Where most papers are concentrated on the improvement phase, \cite{Azlan2013} focus on creating as good initial solutions as possible to solve the Curriculum-based Course Timetable. Two different construction heuristics based on Graph Coloring are created; largest degree and largest weighted degree. The two heuristics are tested on the datasets from ITC2007 and both are able to find feasible solutions for most of the cases. Largest weighted degree being the one performing best.

### 3.3.5. Exact Methods

The drawback of using some sort of heuristic to solve timetabling problems is the lack of capability to issue certificates of optimality or the quality of the solutions found. This is made possible using \textit{Integer Programming} methods such as decomposition techniques.

The University Course Timetabling at Ohio University's College of Business is solved in \cite{Martin2004} using Integer Programming (IP). The problem is solved using CPLEX.

In \cite{Daskalaki2004} an Integer Programming formulation for the University Course Timetabling is developed. The formulation is based on the timetabling problem in Engineering Schools of Greek universities. It
was possible to solve the single real case of the problem with success using CPLEX. Daskalaki and Birbas (2005) use a two stage relaxation procedure to solve the problem. In first stage the relaxation is performed and it concerns the constraints that ensure consecutiveness for the multi-period sessions assigned to a given course. These constraints are recovered during second stage where a subproblem for each day is solved for local optima.

Avella and Vasil’Ev (2005) describes a Branch-and-Cut algorithm for the University Course Timetabling problem and two cutting planes are derived, Clique inequalities and Lifted Odd-Hole inequalities, to tighten the initial formulation.

Qualizza and Serani (2005) propose an Integer Programming approach based on Column Generation. Each column represents a weekly timetable of a single course. The master problem then contains all the constraints referring to classroom occupancy and non-overlapping in time of courses. Each subproblem contains the constraints related to a single course timetable and hence creates weekly timetable for a single course. A Branch-and-Bound method is used to ensure feasibility of the solution.

Al-Yakoob and Sherali (2007) takes its origin at Kuwait University. The paper formulates a Mixed Integer Programming (MIP) model for the University Course Timetabling problem which is able to design class timetables that have a number of desirables features related to minimizing class conflicts, providing good patterns, and enhancing traffic flow. Moreover, gender policies are considered when creating sections of courses.

Schimmelpfeng and Helberger (2007) solves the University Course Timetabling problem at the School of Economics and Management at Hannover University by modeling the problem as an IP model and solved using a state-of-art MIP solver. An anonymous satisfactory survey among the faculty was initiated and in general the response was positive.

In Lach and Lübbbecke (2012) the Udine benchmark datasets on Curriculum-based Course Timetabling are solved using a two stage decomposition approach. The theoretical background of the decomposition is given in Lach and Lübbbecke (2008). The outline of the two stage decomposition is firstly to assign lectures to time slots (Stage 1) and then assign rooms to the lectures (Stage 2). The approach has proven to be an effective method for solving the Curriculum-based Course Timetabling.

Hao and Benlic (2011) generates lower bound for the ITC2007 benchmark data on the Curriculum-based Course Timetabling problem. They present a new partition-based approach based on the divide and conquer principle consisting of three phases. Firstly, courses are partitioned into a fixed number of subproblems using an iterative Tabu Search. In second stage the subproblems are formulated as integer linear problems using the formulation of Lach and Lübbbecke (2012). And finally the subproblems are solved using an IP solver. Better lower bounds were found for all instances except two. The found lower bounds can be used to estimate the quality of the solutions.
obtained with some of the various heuristic approaches aforementioned.

In Burke et al. (2012) a Branch-and-Cut procedure is suggested for the Udine benchmark datasets. The Integer Programming formulation is the same as for Burke et al. (2010b). The procedure reduces the number of variables necessary to formulate the soft constraints. Using the Branch-and-Cut it is possible to achieve good lower bounds must faster than solving the initial IP model.

Cacciani et al. (2013) is the latest research paper working on improvements of lower bounds on the University Course Timetabling problem. The bounds are obtained by splitting the objective function into two parts and formulate an integer linear programming models for both. The solution of each is obtained by using a Column Generation procedure. The global bound is then obtained by summing up the corresponding optimal values. By comparing the results with results from previous research on the instances used at ITC2003 and ITC2007, it is proven that this method is able to improve some best-known lower bounds and that for some instances the best known solutions is indeed optimal (or close to).

3.3.6. Software systems

Only few systems which solve the University Course Timetabling have been published in detail. Carter (2001) solves the Enrollment-based Course Timetabling for the University of Waterloo, Canada, and has been used since 1985. The algorithm behind is a three-phase method. First, the Student Sectioning problem is solved aiming at minimizing the number of pairs of sections with students in common. Secondly, the sections are assigned time slots and finally the timetabling is improved for each student using single student timetabling, where the timetable of each student is improved individually.

Dimopoulou and Miliotis (2001) has created a system to solve the University Course Timetabling and the Examination Timetabling at Athens University of Economics and Business. The course timetabling problem is modeled and solved using a Branch-and-Bound based computer code. In most of the tests performed the optimal solution was found in less than one minute.

Another system is UniTime. UniTime is an open-source comprehensive educational scheduling system for universities. It offers both Curriculum-based and Enrollment-based Course Timetabling, as well as Examination Timetabling and Student Sectioning, and is used at several universities. The website of UniTime features all the papers published for the system (UniTime).

Rudova and Murray (2003) is some of the earliest work of UniTime. Of later research Rudova et al. (2011) solves the Post Enrollment-based Course Timetabling using the Iterative Forward Search (IFS) algorithm described in Müller et al. (2005). In Müller and Rudova (2012) the Curriculum-based
Course Timetable is solved by transforming its model into the enrollment model and using a local search algorithm for generating the corresponding enrollments. The algorithms of both papers are implemented in UniTime.

All the references mentioned in this section are listed in Table A.4 in Appendix A.
4. High School Timetabling

High School Timetabling is the problem of allocating classes to time slots, teachers and rooms to satisfy the hard and soft constraints. At the high schools the students are grouped into classes prior to the timetabling problem. The students of the classes are usually occupied together for all the lectures of their given class.

The most important components of the High School Timetabling problem are quite similar of those of University Course Timetabling. However there are some significant differences. Firstly, Student Sectioning is considered as part of the University Course Timetabling, whereas at the high schools it is considered as a separate problem. Secondly we have the grouping of students. High schools are timetabling classes, whereas at universities the students are often timetabled individually and at high schools clashes are not acceptable for the classes, whereas it can be acceptable if some students have clashes at universities. Another difference is the use of teachers. At the high schools the teachers are teaching full time, whereas the universities the teaching is often a small part of the professors/lectures workload.

Some of primary hard and soft constraints of High School Timetabling are listed below:

- Primary hard constraints
  - No 1-order conflicts. A student cannot attend two courses that are overlapping in time.
  - Classes/events must be scheduled for the required number of times for each subject.
  - Classes/events must be assigned a resource/room.
  - Room capacity. Classes can only be assigned rooms of which the capacity suits the class size

- Primary soft constraints
  - Limit idle periods for student and/or teachers.
  - Lessons spreading.
  - Resource/times preferences

The XHSTT format, which is described in the next section, operates with 16 different constraints, which can either be denoted hard or soft.

4.1. International Timetabling Competition

As mentioned the Third International Timetabling Competition (ITC2011) had the High School Timetabling problem as topic. The concept of the idea
was to increase the focus on High School Timetabling and the XHSTT format. The competition was launched in October 2011 and features three rounds.

1. In the first round of the competition the competitors should construct all-time-best solutions to the instances of the XHSTT-2012 archive.
2. The second round of the competition will work on the instances of the XHSTT-ITC2011 archive and 15 hidden instances. The participant could use any programming language and free third party software (excludes CPLEX, Gurobi etc.). A time limit of 100 seconds is given.
3. Third round consists of constructing all-time-best solutions to the hidden instances.

Deadline for submission of the contribution was May 2012 and the results was announced at PATAT in August 2012.

4.2. Benchmark Data

Access to public datasets of High School Timetabling has been very limited. Previously research was done for new unique problems each time or on artificial instances. The first accessible real life instances are the Beligiannis data sets.

- Beligiannis data sets - Greek high schools
  The archive consists of 7 datasets from Greek high schools and is used in Beligiannis et al. (2008). Each dataset contains a requirements matrix specifying the number of times each teacher must teach each class.

Due to the lack of exchangeable benchmarks in a uniform format, a group of researchers agreed on developing an XML-standard for the High School Timetabling problem, this resulted in the XHSTT format.

- XHSTT
  The project on creating XHSTT is described in the paper of Post et al. (2012a), and the format was as mentioned used for ITC2011. The objective is to minimize the number of violation of hard and soft constraints. The description of the format of XHSTT is available at the homepage (Post, 2013) along with nearly 40 instances. An evaluator for instances is available at the webpage of Jeffrey Kingston (Kingston, 2013b).

4.3. Recent Research

Research articles on High School Timetabling are mostly limited to a single high school or a single country. In this section we will look at some of the research papers which have been conducted. For the High School Timetabling, the papers are categorized in order of the origin of the problems.
This does not necessary means that the problems at the different countries are not the same, but that the solution methods only have been tested on instances of the given country.

Pillay (2013) is another comprehensive survey which is advised for additional details on publications on High School Timetabling. In Pillay (2013) the literature is categorized by the used methodologies.

4.3.1. Country Specific Research

Australia.
The Australian case is solved in Boland et al. (2008) by creating two integer linear programming models to solve the course blocking and population problem.

Kingston (2005) solves seven instances of Australian high schools using a tilling algorithm with hill-climbing. The problem is of allocating meetings (teachers and classes) to times. Resources are added to the meeting using an alternating path algorithm. The XHSTTT archive consists of three instances from Australia and they are the only instances where teachers workload are constrained.

Brazil.
The Brazilian case of School Timetabling is one of the most represented cases in the literature. Filho and Lorena (2001) use a constructive genetic algorithm to solve two Brazilian high schools, whereas Souza et al. (2003) use a greedy algorithm to find a good initial solution and then a Tabu Search for improving this solution.

Santos et al. (2004, 2005) creates a Tabu Search algorithm with two different memory based diversification strategies. The papers show that the diversification strategies improve the robustness of the Tabu Search.

In Bello et al. (2008) the High School Timetabling problem is treated as a Graph Coloring problem and solved using a Modified Tabucol, where Tabucol is a Tabu Search for Graph Vertex Coloring. The method was tested on five Brazilian high schools. Moura and Scaraficci (2010) solves the High School Timetabling for three Brazilian high schools using a basic GRASP heuristic, followed by a path-relinking improvement. Santos et al. (2012) present Column Generation as an approach for establishing bounds for a set of datasets originating from Brazil. It was a Brazilian team that won the ITC2011, Fonseca et al. (2012).

Currently, the XHSTTT archive consists of seven datasets from Brazil.

Denmark.
Denmark is one of the new entrants within High School Timetabling. Sørensen and Stidsen (2013) is the first paper working on Danish high school and it describes a complex MIP model of the problem and establishes computational results for 100 real-life instances using Adaptive Large Neighborhood
Search. The solution approach is implemented in cloud-based administrative software and is available for the majority of all Danish high schools.

Sørensen and Dahms (2014) suggest a two-phase decomposition (created by Lach and Lübbbecke (2012)) for solving the high schools in Denmark. In the first stage lectures are assigned to time slots and in the second stage rooms are assigned to the lectures. The results show that the approach is more effective than solving the original IP model in terms of both the obtained solutions and the obtained lower bounds.

At present four instances of Danish high schools are available in the XHSTT archive.

Germany.

In Bufé et al. (2001) the German high schools are treated with an evolutionary heuristic. Firstly an evolutionary algorithm searches the space of all permutations of the events from which the high school timetable is created. The solution is then improved using local search with specific mutation operators.

Lohnertz (2002) uses a hybrid approach with a combination of Tabu Search and Graph Vertex Coloring. Whereas Wilke et al. (2002) uses a genetic algorithm for solving the German high school timetabling.

A Tabu Search algorithm for the German timetabling problem is presented in Jacobsen et al. (2006). The initial solution is created using a construction heuristic with a graph coloring algorithm. The solution is then improved with the Tabu Search algorithm.

So far as we know none of the German instances are available in an online archive.

Greece.

The Greek high schools are well researched and the XHSTT archive is presented with seven instances from Greece.

In Valouxis and Housos (2003) constraint programming are used in combination with local search for Greek high schools. Papoutsis et al. (2003) creates an Integer Programming model of the problem and solves it using Column Generation.

In Beligiannis et al. (2008) an evolutionary algorithm has been created for the High School Timetabling problem in Greece and this system has been extended with a user-friendly interface (Moschopoulos et al. (2009)). In later work Beligiannis et al. (2009) implement a genetic algorithm to solve the same case. Both algorithms is inherent adaptive, which means that the user is able to assign weights to the different constraints. Both algorithms have been tested on seven real timetables used at schools in the city of Patras and has been made available online (See Beligiannis benchmark, Section 4.2).

In Raghavjee and Pillai (2009) and Raghavjee and Pillai (2010) a generic algorithm is used to solve the high school timetabling problem in Greece.
and South Africa, respectively. The Greek datasets are from the Beligiannis benchmark datasets.

Birbas et al. (2009) uses a hybrid approach to solve the problem at a secondary Hellenic school. The first phase solves the Shift Assignment Problem where teachers are allocated to shifts, and the second phase solves the High School Timetabling. Both phases are solved using Integer Programming.

Zhang et al. (2010) uses a simulated based algorithm for the High School Timetabling problem. The algorithm is tested on two sets of benchmark instances. One randomly created and one real-life from Greece.

Valouxis et al. (2012) describe a two-phase approach based on MIP used to solve the Greek case of the High School Timetabling problem. This includes two instances which are part of the XHSTT project, which were both solved to optimality (solutions were found with an objective value of 0).

Greece is represented in the XHSTT archive with seven instances.

Italy.

Colomi et al. (1998) compares different solution methods for solving two Italian instances of the High School Timetabling problem. The different methods tested are various versions of Simulated Annealing and Tabu Search, and genetic algorithms with and without local search. The genetic algorithm with local search and the Tabu Search based on temporary problem relaxations outperforms the other Simulated Annealing approach and handmade solutions. The genetic algorithm with Tabu Search being the best.

Schaerf (1999b) has implemented a hybrid approach. After the initial timetable is created a randomized non-ascendant search is applied to improve the solution. When no better solution can be found, a Tabu search is applied. The two methods are repeated sequentially until there are no further improvements of the timetable. The solution method is tested on one artificial school and two Italian high schools.

Avella et al. (2007) uses a hybrid algorithm consisting of a Simulated Annealing for the initial solution and Very Large Neighborhood Search for improvements. It is tested on two real life instances from Italy.

Italy is represented with one instance in the XHSTT archive.

Netherlands.

Post and Ruizenaar (2004) use a combination of clustering and Branch-and-Bound algorithms to solve the school timetabling problem for a Netherlands secondary school. The first step is to construct clusterschemes. A clusterscheme contains clusterlines with optional subjects that can be taught in parallel. The second step is the Branch-and-Bound algorithm which constructs the school timetable.

In de Haan et al. (2007) a three-phase approach is use to generate a timetable for a Netherlands high school. The first phase constructs cluster
schemes for optional subjects, in the second phase a feasible schedule is constructing by assigning all lessons to time slots, such that there are no clashes. Finally the third part makes improvements of the timetable.

Post et al. (2012b) presented a cyclic transfer algorithm for the High School Timetabling problem. The methods are tested on four Dutch high schools and one English high school.

In the XHSTT archive the Netherlands are represented with four instances.

Portugal.
In Fernandes et al. (1999) the Portuguese High School Timetabling problem is solved using evolutionary algorithms. The genes which cause hard constraint violations are denoted bad genes and by introducing bad genes mutation it is possible to improve the algorithm in both speed and solution. Melício et al. (2006) developed the software tool THOR. THOR consists of a graphical user interface, an automatic scheduler and a relational database. The automatic scheduler is using a greedy algorithm to establish an initial solution and then improving that using Simulated Annealing. The system is used by more than 100 Portuguese high schools.

Other country specific methods.
Besides the mentioned countries, High School Timetabling has been briefly touched in other countries as well. Wood and Whitaker (1998) solves the timetabling problem for secondary schools in New Zealand where student requests are an important part of the problem formulation. The problem is formulated as a non-linear goal program and solved in several stages using different heuristics such as Simulated Annealing and the Hungarian assignment algorithm. Yigit (2007) uses a hybrid genetic algorithm for solving the High School Timetabling problem for the Technical and Vocational High Schools in Turkey. In Ribic and Konjicija (2010) a two-phase approach to modeling the timetable problem is presented. In the first phase, the classes are allocated to days, and during the second phase, each class in a day are allocated into time slots. The approach is tested on a test case from a Croatian secondary school.

Nurmi and Kyngas (2007) use an extension of a hybrid hill-climbing genetic algorithm. The extension of the algorithm is a Simulated Annealing for choosing periods "intelligently". The algorithm is tested on data from Finnish high schools. Finland is represented in the XHSTT archive with six instances.

The timetabling problem for the Vietnamese high schools were solved in Minh et al. (2010) using a greedy algorithm for the initial solution and then improve the solution using Tabu Search.
4.3.2. The XHSTT-format

Most of the studies on High School Timetabling have resulted in successful solvers, however a big drawback is that many of the papers have only been applied to one problem or one country. One of the reasons for this is the lacking of benchmark data. Another reason is the desire of the single school to get its own problem solved and hence the papers are not interested in a more general approach. By the XHSTT format with corresponding benchmark datasets it is now possible to model a single problem in a standard format and to tests the solution approach on several cases.

As mentioned ITC2011 considered the High School Timetabling problem based on instances of the XHSTT format (Post et al. (2012c)). Four teams made it to the final round: The overall winner (Team Goal) used Simulated Annealing and Iterated Local Search to perform local search around a generated initial solution (Fonseca et al. (2012)). Participant from the University of Nottingham (HySTT) used a method based on Hyper-heuristics (Kheiri et al. (2012)). Team Lectio from Denmark used Adaptive Large Neighborhood Search (ALNS) (Sørensen et al. (2012)) and Romrös and Homberger (2012) (Team HFT) from Germany used an Evolutionary Algorithm. The specification of ITC 2011 and the results are described in Post et al. (2013).

Pimmer and Raidl (2013) describe a 'timeslot-filling' heuristic for XHSTT, which iteratively fills selected timeslots with sets of events. Two state-of-the-art solutions were found for instances of the archive XHSTT-2012.

Fonseca et al. (2013) have made some improvements on their work from ITC2011 which exceeds their previous work (Fonseca et al. (2012)). The new algorithm is a stagnation free Late-Acceptance Hill Climbing algorithm. By combining the new approach with the Simulated Annealing from Fonseca et al. (2012) it provides the best results for some XHSTT datasets of ITC2011.

Kristiansen et al. (2013b) is the first paper working on an exact method for solving problems of the XHSTT format. The paper shows that the complex XHSTT format can be formulated as a Mixed Integer Programming model and solved using the state-of-the-art MIP solver, Gurobi. It was possible to find two new optimal solutions and prove optimality of four previously known solutions. Furthermore, lower bounds were established for 11 datasets.

All the references mentioned in this section are listed in Table A.5 and in Table A.6 in Appendix A.
5. Examination Timetabling

Examination Timetabling is the problem of scheduling a given number of exams to a limited number of time slots. Each course has one event representing the exam. The main problem is to avoid clashes in each student’s examination timetable and to make sure that they have sufficient preparation time for each exam.

It is mainly the university course timetabling which are discussed in the literature and in 
Schaerf (1999a), the difference between Examination Timetabling and University Course Timetabling is observed to be relatively small, as both is of assigning event/exams to time slots and resources. However, it is broadly accepted to distinguish between the two due to the characteristics of the examination timetabling. University Course Timetabling pursuit a compact timetable whereas Examination Timetabling pursue more spreading between events for each student. The time between two exams is the preparation time the student has for the next exam. Furthermore, there is only one exam per course and there can be more than one exam in a single room. And a student can only attend one exam a day.

Below is listed some of the most common constraints used in Examination Timetabling.

- Primary hard constraints
  - 1-order conflicts cannot be accepted in Examination Timetabling.
  - Resources needs to be sufficient for the examinations (e.g. room capacity, enough rooms).

- Primary soft constraints
  - Spreading versus compact.
  - Time requirements. Exams can/cannot be in certain time slot.
  - Consecutive exams.
  - Resource requirements
  - Limited number of students and/or exams in any time slot
  - Ordering of exams must be satisfied
  - Only exams with same length can be located in the same room in the same time slot
  - Exams required taking place at the same time, on the same day or at the same location
  - As early/late as possible
  - Splitting the exams over similar locations
5.1. International Timetabling Competition

Aforementioned, the International Timetabling Competition in 2007 consisted, as mentioned, of three tracks, where the first track was regarding Examination Timetabling (McCollum et al. (2010)).

Eight datasets were given to the competitors, four were given immediately and four were given two weeks before competition deadline. The solution methods submitted were then tested on four hidden datasets. As mentioned in Section 3 it was the contribution of Müller (2009) that won Track 1 and Track 3 of the competition. The algorithm was a hybrid heuristic consisting of three parts, an Iterative Forward Search, a Hill Climbing and the Great Deluge techniques.

The datasets were provided by the EventMap research group and eight of the datasets are made available as benchmark, at the ITC2007 website (McCollum (2007)).

5.2. Benchmark Data

As Examination Timetabling has a high research interest it had let to some establishment of a variety of different benchmark problems. These benchmarks have made it possible to create scientific comparison between different approaches to the Examination Timetabling and thereby exchange of research achievements. The goal of this section is only to give a brief description of the known benchmark of Examination Timetabling. This paper will use name of the benchmarks as they were renamed in Qu et al. (2009) to prevent further confusion between papers. The majority of the benchmarks are available online (ExamBenchmarks).

- **The Toronto benchmark data** 
  Carter et al. (1996) introduced a set of 13 real-world Examination Timetabling problem (3 from high schools and 10 from universities).

  The Toronto benchmark data have two variants of objectives: 1: Minimize the number of used time slots needed for the problem. 2: Minimize the average cost per student.

- **The Nottingham benchmark data**
  Burke et al. (1996) a modification of the objective on six of the data sets from Carter et al. (1996) were introduced as benchmark along with the Examination Timetabling data from 1994 at the University of Nottingham.

  The objective is to minimize the number of students having two consecutive exams.

- **The Melbourne benchmark data**
  Merlot et al. (2003) introduced two new datasets from the University
of Melbourne at the fourth conference of Practice and Theory of Automated Timetabling (PATAT2003). The benchmark consists of two datasets which have two time slots for each of the five workdays.

The objective is to minimize adjacent exams on the same day or overnight.

- **The Purdue benchmark data**
  The newest benchmark within Examination Timetabling is given by Müller (2013). Nine datasets from Purdue University are introduced, and each dataset have 29 examination periods, all 2 hours long. PurdueExamBenchmark

### 5.3. Recent Research

Within educational timetabling, Examination Timetabling is a much researched subject. Qu et al. (2009) is an excellent survey of examination papers from 1996 to 2009 and we recommend the reader to read this paper for additional details on the research of Examination Timetabling.

It is noted that one of the current trends in the literature of operations research is the use of some sort of hybridization of different solution techniques. The same is applicable for the Examination Timetabling.

In the following the literatures are divided into the same classification as used for University Course Timetabling in Section 3 i.e. based on the main techniques used.

#### 5.3.1. Swarm Intelligence Algorithms

Ant Colony algorithms has been used to solve Examination Timetabling in Dowsland and Thompson (2005) and Eley (2007). In Dowsland and Thompson (2005) an Ant Colony algorithm based on the ANTCOL algorithm for graph coloring from Costa and Hertz (1997) is developed to solve the Toronto benchmarks. A number of enhancements and modifications to the algorithm are introduced. These include an initialization method, using recursive Largest Degree and Saturation Degree, and trail calculations. Eley (2007) compares the ANTCOL algorithm with the MAX-MIN Ant System for University Course Timetabling from Socha et al. (2003). The algorithms are tested on the Toronto benchmarks and it is showed that the ANTCOL system outperforms the MAX-MIN Ant System.

In Turabiéh and Abdullah (2011b) the Examination Timetabling problem is solved using the Fish Swarm algorithm developed in Turabiéh et al. (2010) for University Course timetabling. The results show that the algorithm performs well on the Toronto benchmark datasets. In later work of the same authors (Turabiéh and Abdullah (2011a)) a Great Deluge algorithm within an electromagnetic-like mechanism is employed for Examination Timetabling. This mechanism method shares the same concepts as the Particle Swarm, where the position is changed based on the total force that
affects the particle in the search space. The algorithm was tested on the Toronto benchmark and ITC2007 competition datasets.

Sabar et al. (2009) uses a honey bee mating algorithm to solve the Toronto Benchmark datasets. The honey bee mating process is a typically swarm-based approach, were the algorithm is inspired by the process of the mating of honey bees. The queen (current best solution) leaves the nest to perform a mating flight during which the drones (new solutions) follows the queen and mate with her. In the beginning of the flight, the queen’s speed is high and therefore the probability of mating is also high, which is also the case when the fitness of the drones is as good as the queen’s. The queen then moves between different solutions in the solution space, according to her speed, and mates with the drones. After a successful mating, a new brood is generated, the fittest replaces the queen, and the rest become the new drones. Solutions show that the honey bee mating algorithm can produce good quality solutions for the Toronto benchmarks.

5.3.2. Evolutionary Algorithms

Wong et al. (2002) present an exam timetable generator which implemented for Ecole de Technologie Superieure at the University of Quebec. The approach of the generator is a Generic Algorithm to construct the exam timetable. A binary tournament selection is used to produce candidates for the algorithm.

Mansour et al. (2011) developed an evolutionary heuristic based on the scatter search approach. Scatter search operates on maintaining and evolving a population of small candidate solutions. It is then possible to find good suboptimal solutions for the problem. The algorithm is compared with other heuristics (genetic algorithm, Simulated Annealing, and 3-phase Simulated Annealing) on real data of the Lebanese American University and the results shows that the adaptive scatter search approach generates the best timetables.

5.3.3. Local Search Techniques

In Casey and Thompson (2003) a Greedy Randomized Adaptive Search Procedure (GRASP) is used for solving the Examination Timetabling problem for the Toronto benchmark datasets. The GRASP algorithm consists of a construction phase, where feasible timetables are created, and an improvement phase. In the constructing phase a limited form of a Tabu Search and efficient ordering of the exams are used. The improvement phase makes use of a neighborhood based on Kempe chains and a limited form of Simulated Annealing. In order to enhance the solution space the algorithm makes use of memory functions. The GRASP algorithm produced is simple to understand and performs robustly across all the instances.

In Ahmadi et al. (2003) a hyper heuristic is developed with a Variable Neighborhood Search to find good combinations of parameterized heuristics.
Permutations of twelve low level heuristics are employed to create solutions. Seven of the heuristics are exam selection, two time slot selection and three room selection heuristics. The hyper-heuristic is tested on instances from University of Nottingham.

Merlot et al. (2003) employ constraint programming to create a feasible initial solution for the Examination Timetabling problem. To improve the solution Simulated Annealing and Hill Climbing are used. The hybrid method has been tested on the University of Melbourne, two variants of Toronto instances and the Nottingham benchmark.

Burke et al. (2004a) make use of two variants of local search; a time-predefined variant of Simulated Annealing and an adaptation of the Great Deluge method. The Great Deluge has the same advantage as Simulated Annealing by accepted worse moves during its run. The algorithms are tested on the Toronto and Nottingham benchmarks and it is showed that the Great Deluge approach was superior to the Simulated Annealing approach.

White and Xie (2001) developed a four phase Tabu Search called OT-TABU. In each phase more constraints are considered. The OTTABU algorithm was tested on the Examination Timetabling problem at the University of Ottawa, Canada. The approach was extended in White et al. (2004) where the Tabu lists are dynamically relaxed after a certain solution time with no improvements. This extended approach was tested on the Toronto benchmark.

Abdullah et al. (2007) developed a Large Neighborhood Search based on the search methodology originally developed in Ahuja et al. (2001). The key features of the approach are the combination of the very large neighborhood tree-based structure with the technique of identifying improvement moves by addressing negative cost partition-disjoint cycles. The approach was able to find some new best solutions for the Toronto benchmarks. In Abdullah et al. (2010) a hybridized approach of tabu-based and memetic algorithms is developed. The construction phase of the algorithm is based on a saturation degree graph coloring heuristic and the improvement phase makes use of the hybrid heuristic. A tabu list is used to penalize neighborhood structures that are unable to generate better solutions after the crossover and mutation operators have been applied to the selected solutions from the population pool.

In Caramia et al. (2008) four variant of a local search based algorithm are tested on the Toronto benchmarks and compared with solution from other papers. The algorithm consists of three building blocks; a greedy scheduler to find timetables with small length, a penalty decreaser to reduce the penalty of a schedule without changing the length and a penalty trader to reduce the penalty of a schedule by increasing the number of used time slots. The four variants are different depending on the type of checkpointing used, and on whether bridging priorities were used. The algorithms perform more robust than those of the comparison. The best variation being a hybridization of
an approach using adaptive checkpointing and bridging priorities with an approach using constant checkpointing but no bridging priorities.

Pilla y and Banzhaf (2009) suggest a hyper-heuristic with hierarchal combination of heuristics. A Tabu Search is used to search the heuristic space for a heuristic list that produces the best quality exam timetable. Each list contains two of the six low level heuristics. The algorithm is tested on the Toronto benchmark and provides all instances with feasible timetables.

Gogos et al. (2012) developed a multi-stage algorithmic process to solve the datasets of ITC2007. The top level heuristic is a GRASP and low level methods consist of several optimization algorithms, heuristics and meta-heuristics. The approach has a construction phase and an improvement phase. Each phase consists of stages that are consumed in a circular fashion.

5.3.4. Graph Coloring Algorithms

Graph bases methods are widely used on Examination Timetabling problems. The algorithms are often hyper-heuristics where the lower level heuristics are graph coloring heuristics, such as largest degree and saturation degree.

In Asmuni et al. (2005) a Fuzzy Logic algorithm was employed to order exams to be scheduled based on graph coloring heuristics. The fuzzy weight of an exam is used to represent how difficult it is to schedule. The method cannot compete with other solution methods on the Toronto benchmark, but the potential of it is demonstrated. Different or more Fuzzy functions are needed to be able to produce best known solutions. In later work of the same authors, a Fuzzy system was developed to evaluate the quality of the exam timetables (Asmuni et al. (2007)). The quality of exam timetables is measured considering two criteria: the average penalty per student and the highest penalty imposed on any of the students.

Yang and Petrovic (2005) used a hyper-heuristic with a Case-Based Reasoning as high level heuristics to choose graph heuristics to construct a feasible initial solution. A Great Deluge algorithm is then employed to improve the solution. Burke et al. (2005) also make use of a Case-Based Reasoning in a hyper-heuristic. Two different ways of hybridizing the low level graph heuristics were compared for solving the Toronto benchmark data. One were Case-Based Reasoning is used as higher level heuristics and one with a Tabu Search. The Tabu Search approach performed a little better than the Case-Based Reasoning approach, however it was significant slower.

The Tabu Search as higher level heuristic for low level graph coloring heuristics were investigated once more in Burke et al. (2007). It was observed that the more different graph-based heuristics used in the lower level the better the performance might by. The drawback is however the enlargement of the search space which can influence the solving time. The approach was used on both the Examination Timetabling and University
Course Timetabling and in both cases it was competitive with other solution methods from the literature.

The Examination Timetabling problem from Universiti Malaysia Pahang, Malaysia, is presented in Kahar and Kendall (2010). The paper compares the Examination Timetabling problem at the university with known benchmark data, and two new constraint concerning splitting exams into different rooms are introduced. When splitting an exam the rooms must be in the same building (hard constraint) and the distance between the rooms should be as close as possible (soft constraint). The algorithm presented in the paper is based on graph coloring heuristics and produce superior solutions compared to the existing software at the university.

Sabar et al. (2012) developed a graph coloring constructive hyper-heuristic algorithm. At the higher level of the hyper heuristic the difficulty level of examinations is calculated by using hybridizations of four graph coloring heuristics. At the low level four graph coloring algorithms are used; largest degree, saturation degree, largest colored degree and largest enrollment. The results show that the approach is a simple and an efficient tool to produce competitive results for the Toronto and the ITC2007 benchmark data.

Rahman et al. (2014) employ adaptive linear combinations of graph coloring heuristics with a heuristic modifier to address the Examination Timetabling problem for the Toronto and the ITC2007 benchmark data. The approach makes use of two graph coloring heuristics, largest descent and saturation degree.

5.3.5. Other Heuristic Methods

This section contains literature which application method does not fit in the previous heuristic categories.

In Petrovic and Bykov (2003) the multi-criteria called compromised programming is used. The method requires the user to specify a reference solution. To improve the values of the reference objectives a trajectory is drawn in the criteria space and a Great Deluge is conducted using the specified trajectory. The criteria weights can be dynamically changed to guide the search, starting from random points, towards the reference point.

Duong and Lam (2004) present a two phase heuristic for the Examination Timetabling problem. In the first phase constraint programming is used to generate a feasible initial solution and in the second phase a Simulated Annealing with Kempe chain neighborhood improves that solution. The algorithm is tested on real data of Ho Chi Minh City University of Technology.

Ozcan et al. (2009) uses a late acceptance hyper-heuristic to solve the Toronto benchmark data. Late acceptance strategy is a memory based technique that maintains the history of objective values from the last \(L\) previous solutions. The new solution is compared to a previous solution obtained at the \(L\)th step and the acceptance decision is made accordingly. The results show that Simple Random performs the best when combined with late
Abdullah et al. (2009) present a hyper-heuristic of an electromagnetic-like mechanism and the Great Deluge algorithm. Electromagnetic-like mechanism is implemented to calculate the force for each solution. The force value later will be used in the Great Deluge algorithm to calculate the force decay rate.

Another hyper-heuristic for Examination Timetabling is the Monte Carlo based hyper-heuristic developed in Burke et al. (2010a). A number of new and previous suggested Monte Carlo based selection hyper-heuristics are investigated on the Toronto benchmark data. As heuristic selection methods a simple random algorithm, a greedy algorithm, a Choice Function algorithm and a learning scheme are utilized. The hyper-heuristic make use of four low level heuristics.

In Demeester et al. (2012) a tournament based hyper-heuristic is presented. The hyper-heuristic framework of Özcan et al. (2008) is extended using a tournament factor. At each iteration, the selected heuristic generates a predefined number, namely the tournament factor, of random moves. The low level heuristics are Simulated Annealing, Great Deluge and steepest descent. The hyper heuristic is tested on the Toronto benchmark, the instances from ITC2007 and datasets from KAHO Sint-Lieven, Belgium.

Anwar et al. (2013) propose a harmony search based hyper-heuristic method for the Examination Timetabling. The harmony search algorithm is a fairly new meta-heuristic method inspired by musical improvisation process. The algorithm of this paper employed harmony search algorithm at the high level to evolve a sequence of improvement low-level heuristics. At the low level, two different neighborhood structures are used. Swap and move. The algorithm is tested on the ITC2007 benchmark.

5.3.6. Exact Methods

The amount of research using some sort of exact methods, such as decomposition, is quite sparse for Examination Timetabling.

In Qu and Burke (2007) a new decomposition technique is developed for the Examination Timetabling. The idea is to decompose the problem into two sub-sets of events; a difficult and an easy sub-set. In the first step the exams of the difficult set is ordered to find the best feasible solution. In second step the solution from step one is fixed and the easy events are ordered. The two steps are then repeated in a cycle. The approach was tested on the Toronto benchmark data.

Al-Yakoob et al. (2010) considers two exam related problems at Kuwait University; the Examination Timetabling problem and the proctor assignment problem. A Mixed Integer Programming model has been created for both problem and solved using the State-of-art MIP solver CPLEX.
results obtained by solving the MIP models are of significant improvements compared to the existing manual approach at the given university.

5.3.7. Software Systems

Many of the systems mentioned in Section 3.3.6 are for both University Course Timetabling and Examination Timetabling. In Dimopoulou and Miliotis (2001), the initial solution for the Examination Timetabling is based on the University Course Timetabling. This is then adjusted repeatedly by a heuristic approach.

The newest paper from the system UniTime on Examination Timetabling is Müller (2013). The algorithm presented consists of several phases. A construction phase where a complete solution is found using an Iterative Forward Search (Müller et al. (2005)). The next phase uses a Hill Climber to find a local optimum. Once a solution cannot be improved further a Great Deluge technique is used.

Other systems which only solves the Examination Timetabling problem includes Hansen and Vidal (1995) and Thompson and Dowsland (1998).

Hansen and Vidal (1995) is a system for timetabling oral and written examinations at more than 200 high schools in Denmark. The system uses a four phase process dealing with different objectives using different techniques. Phase one is the subject draft which determine which exams a student is assigned. Examination chains are generated in phase two. Phase three creates the examination timetables and finally phase four assigns censorships to the exams.

Thompson and Dowsland (1998) create an Examination Timetabling system at Swansea University in Wales. The problem is divided in two phases both solved using Simulated Annealing. The first phase seeks out a feasible solution and the second finds an improvement in terms of meeting the secondary objectives and soft constraints.

All the references mentioned in this section are listed in Table A.7 in Appendix A.
6. Student Sectioning

The previous mentioned educational timetabling problems are all considering the problem of assigning some events to times. Student Sectioning usually resides outside this categorization as it involves assigning students to sections and not times.

A course might be split into sections/classes, i.e. copies of the same course, each with its own time, room and teachers. Student Sectioning is the problem of assigning students to sections of courses while respecting the requests of the individual student. Some of primary hard and soft constraints of Student Sectioning problems are listed below:

- Primary hard constraints
  - No 1-order conflicts. A student cannot attend two courses that are overlapping in time.
  - Limitations on class sizes.
  - Resource limitation. E.g. only two Physic classes in each cluster.

- Primary soft constraints
  - Equally distribution between sections of same course (spreading)
  - Minimize the number of sections used.

6.1. Recent Research

Literature concerning Student Sectioning is very sparse, none of the previous surveys listed in Section 1.1 use much effort on this research subject and in many papers where Student Sectioning is mentioned the papers are on University Course Timetabling or High School Timetabling and not specific on Student Sectioning.

The surveys of Carter and Laporte (1998) and Schaefer (1999a) provides a good overview of previous work within practical course timetabling and automated timetabling problems. Both papers give a short description of student sectioning problem and some of the earliest work on the subject. Kingston (2013a) makes a short description on the subject, however the book chapter only refers to few articles. Other surveys only briefly mentioned that many articles on timetabling problems have a preprocessing problem of assigning students to classes.

Student Sectioning arises at both universities and high schools. In both cases it is often a prepossessing problem for the timetabling problem.

Of the two cases, Student Sectioning at high schools is the least studied and are in general smaller in size compared to university student sectioning. In de Haan et al. (2007) optional subject for the students is used when constructing timetables at high schools in Netherlands, and cluster schemes
are created to maintain the students' optional courses. Due to a new educational system at the test case high school the program in the paper is only used operational for the Student Sectioning (constructing the cluster schemes) and not on the timetabling.

Kristiansen and Stidsen (2013) use the comparative term electives. The problem is to assign 2nd and 3rd year students to electives given their requests while minimizing the number of classes created and producing a fair distribution. An Adaptive Large Neighbor Search algorithm has been created to solve the problem and the results are in average 0.5% from the best known lower bound. The algorithm developed in Kristiansen and Stidsen (2013) is implemented in the cloud-based high school administration software Lectio.

In Kristiansen et al. (2013a) the problem is concerned the first year students at Danish high schools and is bipartite. First the students are grouped into cohorts in which they are going to attend the same mandatory courses, secondly the cohorts are assigned time slots to satisfy the students requests for two electives. The problem is solved using the MIP solver Gurobi.

There exist more papers on Student Sectioning at universities, however in many cases you must search in papers on University Course Timetabling to find discussion on the subject, e.g. Rudova and Murray (2003) and Suyanto (2010).

Carter (2001) and Rudova and Murray (2003) describe a demand-driven timetabling where student selections of courses are utilized to create a timetable that satisfy as many requests as possible. In Carter (2001) the University Course Timetabling is created first by assigning time slots to sections and the students is then assigned individually to the sections that maximizes timetable satisfaction and balance section sizes. Rudova and Murray (2003) uses student course selections to construct timetables that attempt to maximize the number of satisfied course requests.

In Sönmez and Ünver (2010) they make use of a bidding system. The students make bids on which courses they want to participate in, and based on these bids the courses are placed in class rooms which size reflects the number of bids for the given course. This paper looks on the mechanisms within the course bidding.

The approach of using "bidding/requests" and solving Student Sectioning as a subproblem to the university course timetabling is the most common method. (Aubin and Ferland (1989); Sampson et al. (1995); Robert and Hertz (1996))

In other papers the students are clustered to avoid conflicts between the students' choices of courses. Banks et al. (1998) formulates the timetabling problem as a Constraint Satisfaction Problem (CSP) where the algorithm iteratively adds subset constraints to the CSP formulation. Amintooosi and Haddadnia (2005) proposed a fuzzy clustering algorithm to create an initial sectioning prior to timetabling a set of classes. The same approach is used in Alvarez-Valdes et al. (2000). The students select course in an interactive
process in the first phase and in the second phase a Tabu search algorithm is used for constructing the timetable.

In Müller and Murray (2010) Student Sectioning is solved as a part of the University Course Timetabling where it is considered during and after the creation of the timetabling. In Suyanto (2010) used a two stage approach for solving the university course timetabling, where batch student sectioning is done by allowing the first stage timetable to change.

All the references mentioned in this section are listed in Table A.8 in Appendix A.
7. Conclusion

This paper has been provided in an in-depth survey on educational timetabling literature in the last decade. The survey provides a comprehensive overview of methodologies used for each of the four main subjects within educational timetabling problems; University Course Timetabling, High School Timetabling, Examination Timetabling and Student Sectioning. For each planning problem a description is given along with benchmarks and recent research.

It is possible to draw a few conclusions for the complete educational timetabling problems. Firstly, within the last decade the amount of successful literature on this subject has been increasingly and many of the used solution approach is of some kind of hybridization of multiple heuristics.

Secondly, in many cases the quality of a solution is only compared to previous solutions and not on the optimal or lower bound. The main approach is some sort of heuristic, it could be advantage to research in the use of more exact methods to create some good lower bounds or better yet, optimal solutions to the benchmark data.

Finally, there is still a problem of closing the gap between theory and practice. The different planning problems are still in need of some generalized format and description and more benchmark data from the real world. The XHSTT format for High School Timetabling is an excellent example on a generalized description on a educational planning problem with corresponding benchmark consisting of real world data from a range of countries.

Appendix A. Summary Tables

This section contains summarization tables on the literature mentioned in each section, listed in order of appearance in this survey. For each section, the references are sorted according to the year of publication. Some of the mentioned papers in the tables might cover more than one subject and these papers are therefore listed in all the corresponding tables.

Each table consists of the description of author(s), research area and comments.

First we have the tables related to previous surveys and competitions within educational timetabling in Table A.2 and Table A.3 respectively.

Table A.8 lists all the papers related to Student Sectioning.
Table A.2: Surveys on educational timetabling

<table>
<thead>
<tr>
<th>References</th>
<th>Title</th>
<th>Research area</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Schmidt and Ströhlein</td>
<td>Timetable construction – an annotated bibliography</td>
<td>Timetabling</td>
<td>Provide an annotated bibliography including more than 200 entries.</td>
</tr>
<tr>
<td>de Werra (1985)</td>
<td>An introduction to timetabling</td>
<td>Class-teacher and course timetabling</td>
<td>Graph coloring and network flows methods.</td>
</tr>
<tr>
<td>Junginger (1986)</td>
<td>Timetabling in Germany – A Survey</td>
<td>School timetabling</td>
<td>Various software products implemented in Germany.</td>
</tr>
<tr>
<td>Carter and Laporte</td>
<td>Recent developments in practical examination timetabling</td>
<td>Examination timetabling</td>
<td>Algorithms tested on real data or implemented from 1986 to 1996.</td>
</tr>
<tr>
<td>Carter and Laporte</td>
<td>Recent developments in practical course timetabling</td>
<td>Course timetabling</td>
<td>Algorithms tested on real data or implemented.</td>
</tr>
<tr>
<td>Schaerf (1998a)</td>
<td>A Survey of Automated Timetabling</td>
<td>School, course and exam timetabling</td>
<td>Classification of solution techniques particularly from artificial intelligence.</td>
</tr>
<tr>
<td>Burke and Petrovic</td>
<td>Recent research directions in automated timetabling</td>
<td>University timetabling (course and exam)</td>
<td>Tries to explore approaches that can operate at a higher level of generality.</td>
</tr>
<tr>
<td>Burke et al. (2004c)</td>
<td>Application to timetabling</td>
<td>Class-teacher, course, exam and sport timetabling</td>
<td>Application of graph coloring methods to timetabling.</td>
</tr>
</tbody>
</table>

Continued on next page
<table>
<thead>
<tr>
<th>References</th>
<th>Title</th>
<th>Research area</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>McCollum (2006)</td>
<td>University Timetabling: Bridging the Gap between Research and Practice</td>
<td>Latest survey on University Course Timetabling.</td>
<td></td>
</tr>
<tr>
<td>Kingston (2013a)</td>
<td>Educational Timetabling</td>
<td>-</td>
<td>Book chapter introducing the problems of timetabling educational institutions.</td>
</tr>
</tbody>
</table>
### Table A.3: Competitions within Educational Timetabling

<table>
<thead>
<tr>
<th>References</th>
<th>Research area</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gaspero et al. (2007)</td>
<td>University Timetabling and Exam Timetabling</td>
<td>Overview of the competitions and the competitors of the 2nd International Timetabling Competition 2007</td>
</tr>
<tr>
<td>McCollum et al. (2010)</td>
<td>Setting the Research Agenda in Automated Timetabling: The Second International Timetabling Competition</td>
<td></td>
</tr>
<tr>
<td>Post et al. (2012c)</td>
<td>High School Timetabling</td>
<td>Overview of the competitions and the competitors of the 3rd International Timetabling Competition 2011</td>
</tr>
</tbody>
</table>

### Table A.4: University Course Timetabling

<table>
<thead>
<tr>
<th>Authors</th>
<th>Problem</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Carter (2001)</td>
<td>A comprehensive course timetabling and student scheduling system at the university of waterloo</td>
<td>Implemented at University of Waterloo and in 2001 it has been used successfully for 15 years.</td>
</tr>
<tr>
<td>Dimopoulos and Millotis (2001)</td>
<td>Implementation of a university course and examination timetabling system</td>
<td>A system for University Course Timetabling and Examination Timetabling at Athens University of Economics and Business.</td>
</tr>
<tr>
<td>Socha et al. (2003)</td>
<td>Ant Algorithms for the University Course Timetabling Problem with Regard to the State-of-the-Art</td>
<td>-</td>
</tr>
<tr>
<td>Gaspero and Schaefer (2003)</td>
<td>Multi-neighbourhood Local Search with Application to Course Timetabling</td>
<td>Tested on the School of Engineering at Udine University.</td>
</tr>
</tbody>
</table>

Continued on next page
<table>
<thead>
<tr>
<th>Author</th>
<th>Problem</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Martin (2004)</td>
<td>Ohio University’s College of Business Uses Integer Programming to Schedule Classes</td>
<td>Solves the University Course Timetabling at College of Business, Ohio University.</td>
</tr>
<tr>
<td>Daskalaki et al. (2004)</td>
<td>An integer programming formulation for a case study in university timetabling</td>
<td>Engineering School of Greek universities.</td>
</tr>
<tr>
<td>Qualizza and Serafini (2005)</td>
<td>A Column Generation Scheme for Faculty Timetabling</td>
<td>Column Generation with a Branch-and-Bound method to ensure feasibility.</td>
</tr>
<tr>
<td>Lewis et al. (2007)</td>
<td>Post Enrollment based Course Timetabling: A Description of the Problem Model used for Track Two of the Second International Timetabling Competition</td>
<td>Explanation of Post enrollment-based course timetabling used for ITC2007 Track 3.</td>
</tr>
<tr>
<td>Al-Yakoob and Sherali (2007)</td>
<td>A mixed-integer programming approach to a class timetabling problem: A case study with gender policies and traffic considerations</td>
<td>Incorporates gender policies in the University Course Timetabling at Kuwait University.</td>
</tr>
<tr>
<td>Murray et al. (2007)</td>
<td>Modeling and Solution of a Complex University Course Timetabling Problem</td>
<td>Transform the University Course Timetabling problem at Purdue University into a constraint satisfaction and optimization problem. Part of the UniTime system.</td>
</tr>
</tbody>
</table>
Table A.4 – continued from previous page

<table>
<thead>
<tr>
<th>Authors</th>
<th>Problem</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Schimmelfeng and Helber</td>
<td>Application of a real-world university-course timetabling model solved by integer programming</td>
<td>Solves the problem at School of Economics and Management at Hannover University.</td>
</tr>
<tr>
<td>Lach and Lübecke (2008)</td>
<td>Optimal University Course Timetables and the Partial Transversal Polytope</td>
<td>Theoretical background for a two stage decomposition method for the University Course Timetabling.</td>
</tr>
<tr>
<td>Sayanto (2010)</td>
<td>An informed genetic algorithm for university course and student timetabling problems</td>
<td>-</td>
</tr>
<tr>
<td>Turahieh et al. (2010)</td>
<td>Fish Swarm Intelligent Algorithm for the Course Timetabling Problem</td>
<td>Applied to the definition of Socha et al. (2003).</td>
</tr>
<tr>
<td>Li et al. (2011)</td>
<td>Neighborhood analysis: a case study on curriculum-based course timetabling</td>
<td>-</td>
</tr>
<tr>
<td>Rudova et al. (2011)</td>
<td>Complex university course timetabling</td>
<td>Newest research paper from UniTime on Post Enrollment-based Course Timetabling.</td>
</tr>
<tr>
<td>Nguyen et al. (2011)</td>
<td>Variable Neighborhood Search for a Real-World Curriculum-Based University Timetabling Problem</td>
<td>Solves 14 instances from University of Science in Vietnam.</td>
</tr>
<tr>
<td>Shiao (2011)</td>
<td>A hybrid particle swarm optimization for a university course scheduling problem with flexible preferences</td>
<td>Solves the course timetabling at a university in Taiwan.</td>
</tr>
</tbody>
</table>

Continued on next page
Table A.4 – continued from previous page

<table>
<thead>
<tr>
<th>Authors</th>
<th>Problem</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bonatti et al. 2012</td>
<td>Benchmarking curriculum-based course timetabling: formulations, data formats, instances, validation, visualization, and results</td>
<td>Formulations and collection of the Udine benchmark.</td>
</tr>
<tr>
<td>Lach and Lübbecke 2012</td>
<td>Curriculum based course timetabling: new solutions to Udine benchmark instances</td>
<td>Uses the two stage decomposition of Lach and Lübbecke (2008) to solve the Udine Benchmark.</td>
</tr>
<tr>
<td>Nothegger et al. 2012</td>
<td>Solving the post enrolment course timetabling problem by ant colony optimization</td>
<td>Tested on the ITC2007 benchmark with mixed results.</td>
</tr>
<tr>
<td>Chen and Shih 2013</td>
<td>Solving University Course Timetabling Problems Using Constriction Particle Swarm Optimization with Local Search</td>
<td>-</td>
</tr>
</tbody>
</table>

Table A.5: High School timetabling - XHSTT

<table>
<thead>
<tr>
<th>Authors</th>
<th>Problem</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Post et al. 2012a</td>
<td>An XML format for benchmarks in High School Timetabling</td>
<td>The XHSTT format description.</td>
</tr>
</tbody>
</table>

Continued on next page
### Table A.5 – continued from previous page

<table>
<thead>
<tr>
<th>Author</th>
<th>Problem</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Post et al. (2013)</td>
<td>The Third International Timetabling Competition</td>
<td>Description and results of ITC2011.</td>
</tr>
<tr>
<td>Fonseca et al. (2013)</td>
<td>Late Acceptance-Hill Climbing Applied to the High School Timetabling Problem</td>
<td>Improvements of Fonseca et al. (2012)</td>
</tr>
<tr>
<td>Kristiansen et al. (2013b)</td>
<td>Integer Programming for the Generalised (High) School Timetabling Problem</td>
<td>IP model for the XHSTT format and creation of lower bounds.</td>
</tr>
</tbody>
</table>

### Table A.6: High School timetabling - country based

<table>
<thead>
<tr>
<th>Studies</th>
<th>Subject</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wood and Whitaker (1998)</td>
<td>Student Centred School Timetabling</td>
<td>Solves the problem at Secondary schools in New Zealand using Hill Climber and Hungarian Assignment.</td>
</tr>
<tr>
<td>Schaerf (1998b)</td>
<td>Local search techniques for large high school timetabling problems</td>
<td>Solves one artificial and two Italian high schools using a hybrid heuristic with Tabu Search.</td>
</tr>
<tr>
<td>Fernandes et al. (1999)</td>
<td>High school weekly timetabling by evolutionary algorithms</td>
<td>High schools in Portugal.</td>
</tr>
<tr>
<td>Studies</td>
<td>Subject</td>
<td>Comments</td>
</tr>
<tr>
<td>-------------------------</td>
<td>-------------------------------------------------------------------------</td>
<td>--------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Bufé et al. [2001]</td>
<td>Automated Solution of a Highly Constrained School Timetabling Problem - Preliminary Results</td>
<td>Using Tabu Search to solve the German high school timetabling problem.</td>
</tr>
<tr>
<td>Lohnertz [2002]</td>
<td>A timetabling system for the German gymnasium</td>
<td>Combines Tabu Search and Graph Vertex Coloring for German high school.</td>
</tr>
<tr>
<td>Souza et al. [2003]</td>
<td>A GRASP-tabu search algorithm for school timetabling problems</td>
<td>Brazilian high schools.</td>
</tr>
<tr>
<td>Valouxis and Housos [2003]</td>
<td>Constraint programming approach for school timetabling</td>
<td>Greek high schools.</td>
</tr>
<tr>
<td>Papoušek et al. [2003]</td>
<td>A column generation approach for the timetabling problem of Greek high schools</td>
<td>Greek high schools.</td>
</tr>
<tr>
<td>Santos et al. [2004]</td>
<td>An Efficient Tabu Search Heuristic for the School Timetabling Problem</td>
<td>Brazilian high schools.</td>
</tr>
<tr>
<td>Post and Kuizenga [2004]</td>
<td>Clusterschemes in Dutch secondary schools</td>
<td>Clusterschemes are constructed and a Branch-and-Bound approach is then used for Dutch high school timetabling problem.</td>
</tr>
<tr>
<td>Santos et al. [2005]</td>
<td>A Tabu search heuristic with efficient diversification strategies for the class/teacher timetabling problem</td>
<td>Uses memory based diversifications to improve robustness in a Tabu Search. Brazilian high schools.</td>
</tr>
<tr>
<td>Melício et al. [2006]</td>
<td>THOR: A Tool for School Timetabling</td>
<td>Describes the system THOR which in use by more than 100 schools in Portugal.</td>
</tr>
<tr>
<td>Studies</td>
<td>Subject</td>
<td>Comments</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>------------------------------------------------------------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Avella et al. (2007)</td>
<td>A computational study of local search algorithms for Italian high-school timetabling</td>
<td>Using a hybrid heuristic with Variable Neighborhood Search and Simulated Annealing on two Italian high schools.</td>
</tr>
<tr>
<td>Bello et al. (2008)</td>
<td>An Approach for the Class/Teacher Timetabling Problem using Graph Coloring</td>
<td>Tested on five Brazilian high schools.</td>
</tr>
<tr>
<td>Boland et al. (2008)</td>
<td>New integer linear programming approaches for course timetabling</td>
<td>Australian high schools.</td>
</tr>
<tr>
<td>Beligiannis et al. (2008)</td>
<td>Applying evolutionary computation to the school timetabling problem: The Greek case</td>
<td>Greek high schools made available as benchmark data.</td>
</tr>
<tr>
<td>Birias et al. (2006)</td>
<td>School timetabling for quality student and teacher schedules</td>
<td>Solves a secondary Hellenic school. First by assigning teachers to shifts and then solving the High School Timetabling problem.</td>
</tr>
<tr>
<td>Moura and Scarafucci (2010)</td>
<td>A GRASP strategy for a more constrained School Timetabling Problem</td>
<td>Solves the Brazilian high schools' Teacher-class assignment problem.</td>
</tr>
<tr>
<td>Zhang et al. (2010)</td>
<td>A simulated annealing with a new neighborhood structure based algorithm for high school timetabling problems</td>
<td>Tested on two sets of benchmark instances. A randomly generated and instances from Greek high schools in Patras.</td>
</tr>
<tr>
<td>Studies</td>
<td>Subject</td>
<td>Comments</td>
</tr>
<tr>
<td>-------------------------</td>
<td>------------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Santos et al. (2012)</td>
<td>Strong bounds with cut and column generation for class-teacher timetabling</td>
<td>Brazilian high schools Class-teacher assignment problem.</td>
</tr>
<tr>
<td>Post et al. (2012b)</td>
<td>Cyclic transfers in school timetabling</td>
<td>High school timetabling in the Netherlands and England.</td>
</tr>
<tr>
<td>Valouxis et al. (2012)</td>
<td>Decomposing the High School Timetable Problem</td>
<td>Greek high schools.</td>
</tr>
<tr>
<td>Sørensen and Stidsen (2013)</td>
<td>Integer Programming and Adaptive Large Neighborhood Search for Real-World Instances of High School Timetabling</td>
<td>Implemented at available for more than 200 Danish high schools.</td>
</tr>
<tr>
<td>Sørensen and Dahmis (2014)</td>
<td>A Two-Stage Decomposition of High School Timetabling applied to cases in Denmark</td>
<td>Using the approach of Lach and Lübbbecke (2012) on the Danish high school timetabling problem</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>References</th>
<th>Problem</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hansen and Vidal (1995)</td>
<td>Planning of high school examinations in Denmark</td>
<td>System for examination timetabling and censorship assignment at Danish high schools.</td>
</tr>
</tbody>
</table>

Continued on next page
### References

<table>
<thead>
<tr>
<th>References</th>
<th>Problem</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thompson and Dolesland</td>
<td>A robust simulated annealing based examination timetabling system</td>
<td>A system for handling exams at Swansea University, Wales.</td>
</tr>
<tr>
<td>White and Xie</td>
<td>Examination Timetables and Tabu Search with Longer-Term Memory</td>
<td>University of Ottawa, Canada.</td>
</tr>
<tr>
<td>Dimopoulos and Miliotis</td>
<td>Implementation of a university course and examination timetabling system</td>
<td>A system for University Course Timetabling and Examination Timetabling at Athens University of Economics and Business.</td>
</tr>
<tr>
<td>Wong et al.</td>
<td>Final exam timetabling: a practical approach</td>
<td>Solve the examination problem at Ecole de Technologie Superieure, Montreal, Canada, and has been in use since 2001.</td>
</tr>
<tr>
<td>Ahmadi et al.</td>
<td>Perturbation based variable neighbourhood search in heuristic space for examination timetabling problem</td>
<td>Tested on the Nottingham benchmark.</td>
</tr>
<tr>
<td>Merlot et al.</td>
<td>A Hybrid Algorithm for the Examination Timetabling Problem</td>
<td>Created the Melbourne benchmarks.</td>
</tr>
<tr>
<td>Petrovic and Bylov</td>
<td>A Multiobjective Optimisation Technique for Exam Timetabling Based on Trajectories</td>
<td>Tested on the Nottingham and Toronto benchmarks.</td>
</tr>
<tr>
<td>Casey and Thompson</td>
<td>GRASPling the Examination Scheduling Problem</td>
<td>Tested on the Toronto benchmarks.</td>
</tr>
<tr>
<td>Burke et al.</td>
<td>A time-predefined local search approach to exam timetabling problems</td>
<td>Tested on the Nottingham and Toronto benchmarks.</td>
</tr>
<tr>
<td>Duong and Lam</td>
<td>Combining Constraint Programming and Simulated Annealing on University Exam Timetabling</td>
<td>HoChiMinh City University of Technology, Vietnam.</td>
</tr>
<tr>
<td>White et al.</td>
<td>Using tabu search with longer-term memory and relaxation to create examination timetables</td>
<td>University of Ottawa, Canada.</td>
</tr>
<tr>
<td>Burke et al.</td>
<td>Hybrid Graph Heuristics within a Hyper-Heuristic Approach to Exam Timetabling Problems</td>
<td>Tested on the Toronto benchmarks.</td>
</tr>
<tr>
<td>Atniuni et al.</td>
<td>Fuzzy Multiple Heuristic Orderings for Examination Timetabling</td>
<td>Tested on the Toronto benchmarks.</td>
</tr>
</tbody>
</table>

Continued on next page
Table A.7 – continued from previous page

<table>
<thead>
<tr>
<th>References</th>
<th>Problem</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yang and Petrovic 2005</td>
<td>A Novel Similarity Measure for Heuristic Selection in Examination Timetabling</td>
<td>Tested on the Toronto benchmarks.</td>
</tr>
<tr>
<td>Dowland and Thompson 2005</td>
<td>Ant colony optimization for the examination scheduling problem</td>
<td>Tested on the Toronto benchmarks.</td>
</tr>
<tr>
<td>Abdullah et al. 2007</td>
<td>Investigating Ahuja-Orlin's large neighbourhood search approach for examination timetabling</td>
<td>Tested on the Toronto benchmarks.</td>
</tr>
<tr>
<td>Annunzi et al. 2007</td>
<td>A Novel Fuzzy Approach to Evaluate the Quality of Examination Timetabling</td>
<td>Tested on the Toronto benchmarks.</td>
</tr>
<tr>
<td>Burke et al. 2007</td>
<td>A graph-based hyper-heuristic for educational timetabling problems</td>
<td>Tested on the Toronto benchmarks.</td>
</tr>
<tr>
<td>Qiu and Burke 2007</td>
<td>Adaptive decomposition and construction for examination timetabling problems</td>
<td>Tested on the Toronto benchmarks.</td>
</tr>
<tr>
<td>Caramia et al. 2008</td>
<td>Novel Local-Search-Based Approaches to University Examination Timetabling</td>
<td>Tested on the Toronto benchmarks.</td>
</tr>
<tr>
<td>Pillay and Banzhaf 2009</td>
<td>A study of heuristic combinations for hyper-heuristic systems for the uncapacitated examination timetabling problem</td>
<td>Tested on the Toronto benchmarks.</td>
</tr>
<tr>
<td>Ocas et al. 2009</td>
<td>Examination timetabling using late acceptance hyper-heuristics</td>
<td>Tested on the Toronto benchmarks.</td>
</tr>
<tr>
<td>Burke et al. 2010a</td>
<td>Monte Carlo hyper-heuristics for examination timetabling</td>
<td>Tested on the Toronto benchmarks.</td>
</tr>
</tbody>
</table>

Continued on next page
<table>
<thead>
<tr>
<th>References</th>
<th>Problem</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abdullah et al. 2010</td>
<td>A Tabu-Based Memetic Approach for Examination Timetabling Problems</td>
<td>Tested on the Toronto benchmarks.</td>
</tr>
<tr>
<td>Kahar and Kendall 2010</td>
<td>The examination timetabling problem at Universiti Malaysia Pahang: Comparison of a constructive heuristic with an existing software solution</td>
<td>Universiti Malaysia Pahang, Malaysia.</td>
</tr>
<tr>
<td>Al-Yakoob et al. 2010</td>
<td>A mixed-integer mathematical modeling approach to exam timetabling</td>
<td>Kuwait University.</td>
</tr>
<tr>
<td>Mansour et al. 2011</td>
<td>Scatter-search technique for exam timetabling</td>
<td>Lebanese American University, Lebanon.</td>
</tr>
<tr>
<td>Turabieh and Abdullah 2011b</td>
<td>A Hybrid Fish Swarm Optimisation Algorithm for Solving Examination Timetabiling Problems</td>
<td>Tested on the Toronto benchmarks.</td>
</tr>
<tr>
<td>Müller 2013</td>
<td>Real-life Examination Timetabling</td>
<td>Purdue University datasets.</td>
</tr>
</tbody>
</table>
### Table A.8: Student sectioning

<table>
<thead>
<tr>
<th>Author(s)</th>
<th>Problem</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aubin and Ferland</td>
<td>A large scale timetabling problem</td>
<td>Tested on data from a High School in Montreal, Canada.</td>
</tr>
<tr>
<td>Sampson et al.</td>
<td>Class scheduling to maximize participant satisfaction</td>
<td>Using a local search heuristic and is able to meet 94% of the requirements at the Graduate School of Business Administration at the University of Virginia, USA.</td>
</tr>
<tr>
<td>Robert and Hertz</td>
<td>How to decompose constrained course scheduling problems into easier assignment type subproblems</td>
<td>Course timetabling with student requests.</td>
</tr>
<tr>
<td>Banks et al.</td>
<td>A heuristic incremental modelling approach to course timetabling</td>
<td>Constraints are added to the timetabling model to avoid student conflicts. The students are individually scheduled after complete timetabling.</td>
</tr>
<tr>
<td>Alvarez-Valdes et al.</td>
<td>Assigning students to course sections using tabu search</td>
<td>Used at the Faculty of Mathematics, University of Valencia, Spain of the academic year 96/97.</td>
</tr>
<tr>
<td>Carter</td>
<td>A comprehensive course timetabling and student scheduling system at the university of waterloo</td>
<td>Implemented at University of Waterloo and in 2001 it has been used successfully for 15 years.</td>
</tr>
<tr>
<td>Rudova and Murray</td>
<td>University course timetabling with soft Constraints</td>
<td>Tested on a data for the fall semester 2001 at Purdue University, USA.</td>
</tr>
<tr>
<td>Amin toosi and Haddadnia</td>
<td>Feature selection in a fuzzy student sectioning algorithm</td>
<td>Tested on the Mathematical department of Sabzevar University, Iran.</td>
</tr>
<tr>
<td>de Haan et al.</td>
<td>A case study for timetabling in a Dutch secondary school</td>
<td>Is used operationally only for constructing the cluster schemes.</td>
</tr>
<tr>
<td>Stimmer and Caver</td>
<td>Course bidding at business schools</td>
<td>Mechanisms within course bidding.</td>
</tr>
</tbody>
</table>

Continued on next page
Table A.8 – continued from previous page

<table>
<thead>
<tr>
<th>Author(s)</th>
<th>Problem</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Müller and Murray 2010</td>
<td>Comprehensive approach to student sectioning for Purdue University, USA</td>
<td>Student sectioning during course timetabling and batch sectioning after a complete timetable. Implemented in the open source software UniTime.</td>
</tr>
<tr>
<td>Suyanto 2010</td>
<td>An informed genetic algorithm for university course and student timetabling problems</td>
<td>Batch student sectioning within a sparse timetabling model.</td>
</tr>
<tr>
<td>Kristiansen and Stidsen 2013</td>
<td>Elective course student sectioning</td>
<td>Implemented and used for the majority of the Danish high schools from 2012 and forward.</td>
</tr>
<tr>
<td>Kristiansen et al. 2013a</td>
<td>High school student sectioning at Danish high schools</td>
<td>Tested on 25 real life instances from Danish high schools.</td>
</tr>
</tbody>
</table>
References


61


H. Santos, L. Ochi, and M. Souza. An efficient tabu search heuristic for the school timetabling problem. In Celso Ribeiro and Simone Martins, editors,


Educational timetabling is one of the most researched subjects within the range of timetabling problems. This report contains a comprehensive survey of the research in Educational Timetabling problems. The paper highlights some of the main trends and research achievements within this research subject. Educational Timetabling is often divided into four main categories; University Course Timetabling, High School Timetabling, Examination Timetabling and Student Sectioning. For each planning problem a description is given with appertaining benchmark and the recent research.