MISTA 2009

Hybridizing Integer Programming Models with an Adaptive Decomposition Approach for Exam Timetabling Problems

Rong Qu • Fang He • Edmund K. Burke

Abstract The idea of decomposition has been successfully applied to address large combinatorial optimization problems across a range of applications. However, in timetabling, it has not been widely applied. One major difficulty of course, is that early assignment in one sub-problem may lead to later conflict in solving interrelated sub-problems. In our previous work, timetabling problems were adaptively decomposed into a *difficult set* and an *easy set* of exams. They were generated by using the information gathered from previous iterations during the problem solving. The approach obtained promising results and showed somewhat unsurprisingly, that a small set of difficult exams contributed to a much larger portion of the total cost of the solution constructed. An interesting issue, which is explored in this paper, is to investigate the effect of constructing complete solutions based on an optimal solution of this difficult sub-problem.

In this paper, we first present an IP formulation for solving the difficult sub-problem. To have a tighter initial formulation, a well-known inequality called the Clique Inequality is utilized. We then examine the combinatorial properties of the problem to introduce a new family of cutting planes. These are shown to be helpful in obtaining a solution which is within a gap of less than 10% of optimal for the sub-problem, based on which the final solution is constructed. Promising results have been obtained on several benchmark exam timetabling problems in the literature.

1 Introduction

Exam timetabling is an important issue in universities and schools all over the world. It has been very well studied in the last five decades [31]. The basic task of exam timetabling has similarities across very different institutions although the requirements and needs can differ markedly [5]. We need to assign exams to a limited number of time periods so that no hard constraint violation occurs (e.g. no student is assigned more than one exam at a time). A common soft constraint is to spread students' exams out as far as possible to allow them time between exams. Soft constraints tend to be institution specific, and are given different priorities (weightings) in different institutions. The quality of a timetable is usually determined by the exact soft constraint violation. For an overview of timetabling in general see [10, 13, 16, 21]. For a comprehensive specific review of exam timetabling methodologies, see a recent survey in [31]. Due to their importance in practice and inherent scientific challenge, exam timetabling problems have been widely investigated across both the operational research and the artificial intelligence community.

Rong Qu
Automated Scheduling, Optimization and Planning (ASAP) Group, School of Computer Science, University of Nottingham, Nottingham, NG8 1BB, UK
E-mail: rxq@cs.nott.ac.uk

Fang He
Automated Scheduling, Optimization and Planning (ASAP) Group, School of Computer Science, University of Nottingham, Nottingham, NG8 1BB, UK

Edmund K. Burke
Automated Scheduling, Optimization and Planning (ASAP) Group, School of Computer Science, University of Nottingham, Nottingham, NG8 1BB, UK

Due to the large size and complex features of exam timetabling problems, IP models have not been widely investigated in recent research. The only recent work on exam timetabling that we are aware of is in [23], where an IP model was built to solve a specific problem. Some IP formulations have been studied in course timetabling, where Branch-and-Bound [17] and Branch-and-Cut [2] were employed. Different cutting planes were introduced to speed up the Brand-and-Bound process.

In most cases, real-world exam timetabling problems are very large and difficult. Heuristic approaches have been employed to solve the problem efficiently. Simple graph based heuristics have been investigated [8, 14], and are still being employed and adapted within some hybridized methods [7, 9, 30]. Meta-heuristics have become the state-of-the-art [31]. Local search based techniques such as Simulated Annealing [4, 22, 27] and Tabu search [18, 26, 32] have been explored with some success. Evolutionary algorithms have also been investigated throughout of the years [5, 19, 28]. Multi-objective techniques [15] have also been investigated. In recent research, hybrid approaches [11, 22] have played a prominent role in timetabling research, integrating a variety of different techniques including constraint based methods and graph based heuristics.

The basic idea of decomposition is to "divide and conquer", as optimal solutions of smaller sub-problems may be much easier to obtain by using relatively simple approaches or even exact methods. However, the task of decomposition can represent a major challenge. It is usually problem specific and little work exist during the years in the complex timetabling problems. Carter and Johnson [12] considered a subset of quasi-clique (dense sub-graphs) as the foundation upon which to generate complete solutions. Their results showed that the quasi-clique initialization can significantly improve the quality of complete solutions. Qu and Burke [29] adaptively decomposed the problem into two sets of exams (a *difficult set* and an *easy set*). The level of difficulty of exams in specific problems is adaptively adjusted by information gathered from previous iterations of the problem solving procedure. The approach obtained promising results compared against the best results from a number of approaches in the literature at that time. In particular, it was showed that the adaptive difficult set is usually of small size, but makes a major contribution towards the total cost of the constructed solution. A simple random greedy technique was used to find a good enough solution to the difficult set sub-problem in [29].

In this paper, we carried out an in-depth analysis of these adaptively identified difficult sets. The effect of the optimal solution for the crucial sub-problems within the complete solution is studied. The difficult set sub-problems are formulated into a set packing problem with side constraints, and solved by employing an exact IP model to obtain the optimal solution (within a gap of < 10%). We study the structure of exam timetabling problems and introduce new valid inequalities based on the specific combinatorial properties of the problem. Computational results show that the quality of solutions for the difficult set sub-problems contributes towards building high quality final complete solutions constructed.

This paper is organized as follows: Section 2 presents details of the benchmark exam timetabling problems we are solving and overviews our hybrid adaptive decomposition approach. Section 3 presents the IP formulation of the problem and the problem specific cutting planes introduced in the IP model. Section 4 reports the computational results and analysis, followed by Section 5, which presents the conclusion and future work.

2 Benchmark Exam Timetabling Problem and the Hybrid Adaptive Decomposition Approach

2.1 Benchmark exam timetabling instances

The exam timetabling instances we investigate in this paper concern the University of Toronto problem, firstly introduced by Carter et al. [14] and publicly available at ftp://ftp.mie.utoronto.ca/pub/carter/testprob. Over the years they were widely employed as testbeds in exam timetabling research. However, there has been an issue with different instances circulating under the same name. To encourage future scientific comparisons, the issues were clarified and different versions of the instances in the dataset were renamed as version *I*, version *II* and version *IIc* (see more details in [31]). We use version *I* of the dataset and present the characteristics of these problems in Table 1. More details of the problems and an API evaluation function are provided at http://www.asap.cs.nott.ac.uk/resources/data.shtml.

Table 1 Characteristics of the benchmark exam timetabling problems in [14], also see [31]

Problem instance	No. of Exams	No. of Students	No. of Time Periods	Conflict Density
ear83 I	190	1125	24	0.27
hec92 I	81	2823	18	0.42
lse91	381	2726	18	0.06
sta83 I	139	611	13	0.14
tre92	261	4360	23	0.18
yor83 I	181	941	21	0.29

In the problem dataset, to indicate the density of conflicting exams (i.e. those exams with common students thus cannot be scheduled in the same time period), a *Conflict Matrix C* was defined where each element c_{ij} = 1 if exam *i* has common students with exam *j*, or c_{ij} = 0 otherwise. The *Conflict Density* represents the ratio between the number of elements c_{ij} of value "1" to the total number of elements. A high conflict density indicates that the problem difficult to solve. The objective is to minimize the average cost per student, calculated by an evaluation function which is predicated around spreading conflicting exam out (rather than having them allocated close to each other). For students sitting two exams *s* time periods apart, a proximity cost w_s is assigned, i.e. $w_1 = 16$, $w_2 = 8$, $w_3 = 4$, $w_4 = 2$, $w_5 = 1$ and $w_s = 0$ for $s > 5$. A more formal description of the problem can be seen in [14].

2.2 The hybrid adaptive decomposition approach

In our previous adaptive decomposition approach [29], an initial ordering of the exams is first obtained by using the Saturation Degree graph heuristic. Exams are ordered increasingly by the number of remaining feasible time periods during solution construction. Earlier exams with less number of feasible time periods in the ordering can be seen as more difficult and are scheduled to the timetable first. In some cases, the ordering may need to be randomly adjusted until a feasible solution can be obtained. Based on this initial ordering, this list of exams is then decomposed adaptively into two subsets (named the *difficult set* and the *easy set*). The pseudo-code presented in **Algorithm 1** outlines the adaptive decomposition process to identify the *difficult set*.

The approach is an iterative process where one solution is built at each iteration by scheduling the exams in the *difficult set* and the *easy set* in the ordering at that iteration. The exams in the *difficult set* are adaptively adjusted at each iteration by using information derived from the solution construction in previous iterations i.e. the aim is to identify troublesome exams, which can be dealt with higher priority in future iterations of the solution construction. Problematic exams which cannot be scheduled in the previous iteration are added into *difficult set*, and the quality of the previous solution is used to adjust its size.

After this set is identified by the above process, the ordering of exams in the *easy set* is then randomly adjusted, while the *difficult set* is fixed. It was observed that detecting the problematic exams in the *difficult set* is crucial in order to build high quality solutions, i.e. better solutions after identifying the *difficult set* usually led to better final solutions after adjusting the *easy set*. However, adjusting exams in the *easy set* also contributed towards building good solutions, i.e. solutions after the difficult set is identified still can be greatly improved in some cases. We refer to the work in [29] for more analysis on the identified *difficult set*.

Algorithm 1: Adaptive Decomposition by Identifying the Difficult Set

Build a feasible initial ordering of exams based on Saturation Degree heuristic
Initial size of difficult set Sd = number of exams / 2 // $Sd \in$ [1, number of exams -1]
MaxNoIterations = 10,000; iteration = 0
while iteration < MaxNoIterations **do**
    easy set = {e_{Sd+1}, e_{Sd+2}, ... e_e}
    reorder the exams in the *difficult set* {e_1, e_2, ... e_{Sd}}
    construct a solution by using ordered exams in both *difficult set* and *easy set*
    if a feasible solution or an improved solution is obtained
        $Sd = Sd + 1$ // include more potential exams in the difficult set
    else
        move forward the difficult exam causing the infeasibility in the ordering
        set the size of the *difficult set* to where the difficult exam was
    store the *difficult set* and its size S_{best} if the best solution is obtained
    iteration = iteration + 1
end while

At each iteration, the exams are scheduled one by one according to their ordering in the *difficult set* and the *easy set* into the time period leading to the least cost. In the case of ties, i.e. assigning the exam to two time periods leads to the same cost, the exam is randomly scheduled to one of the time periods. It was observed in our previous work that assigning the exam to the earliest or randomly selected time period does not produce significantly different timetables.

In this work, the difficult set sub-problem detected by the above adaptive decomposition is input to the IP model (see Section 3) to obtain the optimal solution, based on which the complete solution is constructed by scheduling the exams in the *easy set*. A simple steepest descent method is then applied to quickly improve the complete solution built based on the optimal solution of the difficult set sub-problem. The aim here is to demonstrate and analyze the effectiveness of building solutions based on optimal solutions of crucial sub-problems. Different advanced meta-heuristics could of course be employed to further improve the complete solution constructed and to provide better results given more computational time.

3 The IP formulation for the sub-problem

3.1 The IP formulation

We define the following notations:

Set:

E, set of exams
S, set of students registered to different exams in E
T, set of time periods for scheduling all exams

Data:

GS_e, the group of students enrolled to exam e, $e \in E$
GE_s, the group of exams to which student s enrolled, $s \in S$

Variables:

Two core binary variables and five additional counting variables are defined:

X_{et} = 1 if exam *e* is scheduled in time period *t*, 0 otherwise.
Y_{st} = 1 if student *s* is scheduled an exam in time period *t*, 0 otherwise.
C_1 is the number of violations in conflicting type 1, i.e. two conflicting exams of a student are scheduled one time period away.
$C_1 = \sum_s \sum_t C_{1st}$, C_{1st} = 1 if student *s* has adjacent exams starting from *t*, 0 otherwise.
C_2 - C_5 are defined in the same way, i.e. the number of violations in conflicting types 2-5, i.e. two conflicting exams scheduled with one-four time periods in between
Different types of conflicting are given different importance, thus a weight is defined as:
$w_1 = 16$, $w_2 = 8$, $w_3 = 4$, $w_4 = 2$, $w_5 = 1$ for two exams scheduled adjacently, and with one, two, three and four time periods in between, respectively

Objective function:

$$\text{Min } \sum w_i C_i$$

$$\text{s.t. } \sum_t X_{et} = 1, \forall e \in E; \quad (1)$$

$$\sum_{e \in GEs} X_{et} \leq 1, \forall s \in S, \forall t \in T; \quad (2)$$

$$X_{et} \leq Y_{st}, \forall e \in E, \forall t \in T, \forall s \in GS_e; \quad (3)$$

$$Y_{st} + Y_{st+1} - C_{1st} \leq 1; \quad (4)$$

$$Y_{st} + Y_{st+2} - C_{2st} \leq 1; \quad (5)$$

$$Y_{st} + Y_{st+3} - C_{3st} \leq 1; \quad (6)$$

$$Y_{st} + Y_{st+4} - C_{4st} \leq 1; \quad (7)$$

$$Y_{st} + Y_{st+5} - C_{5st} \leq 1; \quad (8)$$

The objective is to space out conflicting exams within the limited number of time periods *T*. Constraint (1) requires that one exam can only be scheduled into exactly one time period; Constraint (2) states that a student cannot take two exams at the same time; Constraint (3) forces the following condition to define the relationship between exams and students, if X_{et} = 1, then $\forall s \in GS_e$, $Y_{st} = 1$, otherwise $Y_{st} = 0$. Constraints (4)-(8) are used to count the number of violations for students, i.e. for student *s* and time period *t*, $C_{1st} = 1$ if two exams are allocated in adjacent time periods (i.e. $Y_{st} = 1$ and $Y_{st+1} = 1$), otherwise $C_{1st} = 0$.

3.2 Clique Inequality cutting planes

To derive valid inequalities for exam timetabling problems, we examine the relationship between our problems and the Set Packing problem. The aim is to exploit the connection between these two problems from the polyhedral description of the Set Packing polytope [3], whose structure has been extensively investigated.

The set packing problem belongs to combinatorial optimization problems. Given a finite set *S*, the problem is to find the maximum number of pairwise disjoint subsets in *S*, i.e. no two subsets intersect. The problem can be defined as the following 0-1 integer linear programming formulation:

$$\text{Max } \{CX, AX \leq e^T, x_i = 0 \text{ or } 1 \text{ for } i = 1, \dots n\} \quad (9)$$

where *A* is an m×n matrix with elements from {0, 1}; *X* is a set of *n* variables and *C* is a cost vector.

Constraint (2) in the above formulation take the form of a Set Packing problem (9), and define a relaxation of our exam timetabling problem without considering other constraints (3)-(8). We therefore use this to derive a family of Clique Inequalities [20, 25], which are very effective in tightening the formulation for our exam timetabling problems.

Let $G(V, E)$ be the intersection graph associated with the Set Packing relaxation and let $P(G)$ denote the Set Packing polytope, i.e., the convex hull of the incidence vectors of the stable sets in $G(V, E)$. Let $K \subset V$ be a subset of nodes inducing a clique. It is well known that the Clique Inequality

$$\sum_{k \in K} x_k \leq 1 \qquad (10)$$

is valid for $P(G)$ and it is facet-inducing if K is a maximal clique [2]. This Clique Inequality is implemented within our above IP formulation in CPLEX 10.0.

3.3 Problem specific inequalities (cutting planes)

We introduce two classes of valid inequalities to obtain a further tighter formulation for exam timetabling problems.

1. Exam/time period cuts: if exams e and e' involve the same students, then $X_{et} + X_{e't} \leq 1$.
   Proposition 1: $X_{et} + X_{e't} \leq 1$, $\forall e \in E$, $\forall t \in T$, if $GS_e = GS_{e'}$, hold for exam timetabling problems.
   Proof. If exams e and e' involve the same students s, then exams e and e' must be included in the list GE_s. By constraint (2), we have $X_{et} + X_{e't} \leq 1$. □

2. Implied bound cuts: for the cost counting constraints (4)-(8), we can obtain the bound of C_i according to the number of exams each student takes.
   Proposition 2: we denote n_1 as the number of time periods; n_2 as the number of exams student s take; $n_2 \leq n_1$; we then have the implied bound of C_i as

   $\max\{0, n_2 - (\lfloor n_1 / 2 \rfloor + 1)\} \leq C_1 \leq n_2 - 1;$

   $\max\{0, n_2 - (\lfloor n_1 / 3 \rfloor + 1)\} \leq C_2 \leq n_2 - 1;$

   $\max\{0, n_2 - (\lfloor n_1 / 4 \rfloor + 1)\} \leq C_3 \leq n_2 - 1;$

   $\max\{0, n_2 - (\lfloor n_1 / 5 \rfloor + 1)\} \leq C_4 \leq n_2 - 1;$

   $\max\{0, n_2 - (\lfloor n_1 / 6 \rfloor + 1)\} \leq C_5 \leq n_2 - 1;$

   Proof. The upper bound of C_i indicates the worst case scenario (consecutive exams to the student). For example, Figure 1 shows the maximum violations of type 1 for a student. The lower bound of C_i is the best case scenario (exams are spread as much as possible). Figure 2 shows the minimum violations of type 1 for a student. The upper and lower bounds of types 2-5 can be obtained in the same way. □

Figure 1. The maximum violations of type 1 a student may take ("X" represents the student has an exam in the time period. $n_1 = 13$, $n_2 = 9$, $C_1 \leq 8$)

X	X	X	X	X	X	X	X	X				

Figure 2. The minimum violations of type 1 the student take ("X" represents student has exam at the time period. $n_1 = 13$, $n_2 = 9$, $2 \leq C_1$)

X	X	X	X	X		X		X		X		X

4 Computational results

We report upon our analysis and computational experience on applying the IP formulation with cutting planes in the adaptive decomposition approach for solving some representative instances of the Toronto benchmark problem given in Section 2.1. Computational experiments have been carried out on a personal computer with Intel Core

1.86GHz processor and 1.97GB RAM. We have used the callable library ILOG CPLEX as LP solver using LP-based Branch-and-Bound technique for solving integer linear programs. CPLEX provides functionalities to manage the tree search. ILOG CPLEX Barrier was used as the LP solver, and cutting planes (valid Clique inequalities) were generated throughout the search tree [24]. The problem specific cutting planes were only added statically at root node of the Branch-and-Cut tree. All other CPLEX settings are set to default (i.e. CPLEX automatically chooses the heuristic strategy, node and variable selection strategies, etc).

4.1 Analysis of the adaptive difficult set

In [12], a phase-based approach was employed where the first phase is to allocate a large set of mutually conflicting exams (i.e. cliques in the associated graph). A solution to a maximum clique, arbitrarily chosen among a number of cliques, was used as the initial solution. It was shown that an improvement of the clique initialization in exam timetabling can be made by building larger sets of exams which have a certain level of "density" and which can be thought of as "*almost* a clique".

In our approach, the *difficult set* which consists of troublesome exams is identified adaptively by the information obtained from online problem solving in the previous iterations. We first exam the adaptive *difficult set* and compare it with the maximum clique for the specific problem instance. Our aim of decomposition is twofold: on the one hand, we try to identify as many troublesome exams as possible. We consider not only the exams in maximum cliques, but also the exams that form "almost cliques". On the other hand, we only need real difficult exams which cause trouble in our timetables to be included in our difficult set. The inclusion of non-crucial easy exams in the *difficult set* could significantly increase the computational time in searching for the optimal solution by using the IP model for the difficult set sub-problem, and not contributing to the high quality final complete solution.

Table 2 presents the size of the original problem, the size of their maximum clique and the size of the *difficult set* in terms of the number of students and the number of exams. We can see that the size of the *difficult set* is larger than the size of the maximum clique so it includes more troublesome exams. Also the adaptive *difficult set* forms almost clique (see an example in Table 3). Note that the *difficult set* is adaptively obtained rather than by setting an accurate threshold value of the density.

Table 2. Sizes of the original problem, its maximum clique and the difficult set identified

		ear83 I	**hec92 I**	**lse91**	**sta83 I**	**tre92**	**yor83 I**
Original problem	**no. students**	1125	2823	2726	611	4360	941
	no. time periods	24	18	18	13	23	21
	no. exams	190	81	381	139	261	181
	max-clique size	21	17	17	13	20	18
Difficult set	**no. students**	999	1592	1717	269	1677	788
	no. exams	54	46	54	13	28	53

Table 3 presents the elements and the density of the *difficult set* for an example instance (*sta83 I*). It is interesting to see that the adaptive difficult set identifies the clique, i.e. it includes all elements in a clique. Also the adaptive difficult set and the quasi-clique reported in [12] have a large overlap of elements (with only one element being different), indicating that those exams that are identified as being more troublesome form sufficiently dense sub-graphs. The quality of the final complete solution built based on the clique, quasi-clique and adaptive difficult set is also presented, showing that the complete solution built based on difficult set is the best compared against those built based on clique and quasi-clique sets. This indicates that cliques of higher density, which are statically measured, may not form a good basis for final complete solutions.

Table 3. Properties of different sub-problems: set size, set elements, density of the set and the best complete solution for the original problem based on the sub-problems.

Sets	Size	Set elements	Density	Final best solution
Clique	12	{4,18,27,39,59,72,86,107,121,133,136,139}	1.00	167.84
Quasi-clique	13	{4,18,27,39,59,72,86,107,121,122,133,136,139}	0.93	157.51
Difficult set	13	{4,18,27,39,59,72,79,86,107,121,133,136,139}	0.86	157.12

4.2 Analysis of the solutions to the difficult set sub-problems

The difficult set sub-problems are solved by our IP model with Clique inequalities and problem specific cutting planes to reduce the number of branches searched. Cutting planes can be added either at the root node or at other nodes. In our approach, we only add cutting planes at the root node. Table 4 presents more details of the model with and without the added inequalities and cutting planes. The second column gives the size of the original problem formulation. The third column gives the number of Clique Inequalities generated by CPLEX. Note that while the cuts added by CPLEX reduce the running time for most problems, they may occasionally have the opposite effect. If CPLEX added too many cuts at the root node and the objective value does not change significantly, we turn off the added cuts. The fourth column is the reduced problem size after pre-processing and aggregating. The "Barrier LP" column indicates the lower bound of the LP relaxation. The last column is the optimal integer solution with the specific gap.

Table 4. LP relaxations of the problem formulation, with Clique Inequalities added automatically and implied bounds added statically at the root node. Columns 2-6 present the problem size of the original formulation (row*column); number of cliques; reduced problem size after reductions in CPLEX (row*column, non-zeros); LP relaxation result and the best IP result (within the gap of 10% to the optimal).

Sub-problem	Original LP size	Clique table members	Reduced LP size		Barrier LP	IP results (gap<10%)
ear83 I	215976*130435	19631	103823*104730	360534	0.00	15.4985
hec92 I	200414*144137	16830	83280*83226	270372	0.00	6.94221
lse91	220754*156549	20104	96004*95088	321299	0.00	7.03902
sta83 I	29394*18072	3119	15069*15213	50202	3.13	16.0335
tre92	243043*188046	11413	60613*61160	188425	0.00	2.36017
yor83 I	141126*88133	12212	64322*65382	219267	15.50	16.3185

The quality of the solutions to the sub-problem plays an important role in the finial complete solutions. Table 5 gives the solutions built based on the sub-problems of different quality (by setting different gaps to the optimal). A steepest descent method is also applied to quickly improve the complete solution. It shows that the better the solution of the IP sub-problem (see the column "Initial"), the better the final complete solution would be (see the column "Improved"). Δ% presents the improvement of the solution quality to the next smaller gap, i.e. improvement from gap = 70% to gap = 60%, improvement from. gap = 60% to gap = 10%. It shows that the smaller the gap of the optimal solution to the sub-problem, the better the final complete solution. However, it is also observed by experiments that a smaller gap of optimal solution to the sub problem, say gap < 1%, dramatically increases the computational time.

An issue to be investigated in our future work is to set a good balance between the quality of the sub-solution and the final complete solution, and the computational time. Due to inter-related constraints between the sub-problem and the other exams in the problem, the optimal solution of gap 0 to the sub-problem may not be in the complete global optimal solution. It is therefore not necessary to spend a large computational time to obtain zero-gap optimal solutions for the sub-problems, i.e. there is no need to find the optimal solution of gap 0 for the sub-problems identified.

Table 5 Complete solutions from optimal solution of different gaps for the difficult set sub-problem. Δ% = (result of gap 1 - result of gap 2) / result of gap 2; "Initial" indicates the initial solution built based on the IP sub-solution; "Improved" indicates the improved solution after the simple steepest descent method. Results are average values from 10 runs.

		IP (gap = 70%)		**IP (gap = 60%)**		**IP (gap = 10%±5)**	
		Results	**Δ%**	**Results**	**Δ%**	**Results**	**Δ%**
ear83 I	Initial	47.56	--	46.28	2.7	43.14	7.2
	Improved	40.30	--	38.53	4.5	37.92	1.6
hec92 I	Initial	13.50	--	13.26	1.8	12.30	7.8
	Improved	12.28	--	11.91	3.1	11.74	1.4
lse91	Initial	12.76	--	11.85	7.6	11.12	6.5
	Improved	11.89	--	11.48	3.5	11.11	3.3
sta83 I	Initial	168.01	--	167.06	0.5	160.78	3.9
	Improved	167.84	--	160.61	4.5	157.50	1.9
tre92	Initial	11.12	--	10.60	4.9	9.92	6.8
	Improved	9.92	--	9.55	3.9	9.38	1.8
yor83 I	Initial	52.30	--	50.76	3.0	45.76	10.9
	Improved	45.48	--	43.70	4.0	42.62	2.5

4.3 Analysis of the complete solutions

To give more insights into the performance of our IP based approach, we compared statistically the results with the previous adaptive decomposition approach in [29]. Table 6 presents the statistical distributions of the results from these two approaches.

Table 6. Results from our previous work [29] and our IP approach. The best average, best result and the lowest standard deviation (s.d.) are given in italics and bold.

		ear83 I	hec92 I	lse91	sta83 I	tre92	yor83 I
[29]	Average	37.84	12.09	11.31	157.60	***8.98***	43.27
	Best	***36.15***	***11.38***	***10.85***	157.21	***8.79***	42.2
	s.d.	1.40	0.43	0.38	0.28	0.15	1.03
Our IP approach	Average	***37.72***	***11.74***	***11.11***	***157.5***	9.38	***42.69***
	Best	37.30	11.48	11.07	***157.12***	9.24	***41.97***
	s.d.	***0.68***	***0.18***	***0.04***	***0.13***	***0.11***	***0.51***

It can be seen that our IP based approach obtained better average results on 5 out of 6 problems being considered. It also has smaller standard deviation for all the problems, meaning that its performance is more stable. This is because in [29] a simple random greedy technique is used to build the solution to the sub-problem, and thus the effectiveness of the approach was not warranted. In our IP based approach, solutions to the crucial sub-problems are solved optimally.

Table 7 compares our approach with a number of other approaches in the literature. The best results are highlighted. The first three approaches are all constructive based approaches where improvement meta-heuristics are not the main approach to generate the solutions for the problems. Our approach obtained similar results. The last three approaches generate at least one of the best results in the current literature for the problem instances being considered. Our IP based approach obtained the best result in the literature for one instance *sta83 I*. It should be noted that so far the best results for the benchmark problem instances are generated from a range of state-of-the-art approaches in the literature. No single approach can be seen as the best for solving all problem instances.

Table 7. Best results from our IP based approach, compared with other approaches in the literature.

	ear83 I	hec92 I	lse91	sta83 I	tre92	yor83 I
graph heuristics [14]	36.4	10.8	10.5	161.5	9.6	41.7
adaptive ordering [7]	36.16-38.55	11.61-12.82	10.96-12.53	161.91-170.53	8.79-8.96	40.77-42.97
adaptive decomposition [29]	36.15	11.38	10.05	157.21	8.79	42.2
Our IP based approach	37.30	11.48	11.07	***157.12***	9.24	41.97
large neighborhood search [1]	34.87	10.28	10.24	159.2	***8.4***	***36.2***
hybrid local search [11]	***29.3***	***9.2***	***9.6***	158.2	9.2	***36.2***

4.4 Discussions and future work

It would be interesting, in our future work, to extend the study in several directions. Firstly, the solutions to the sub-problem are not optimal (i.e. with a gap of 0%). It would be interesting to investigate the trade-off between obtaining sub-problem solutions closer to the real optimal and the computational time required. Due to the interrelations between exams in the *difficult set* and the *easy set*, the optimal sub-solution may not remain the same in the global complete solution. The task is not necessarily to search for the real optimal solution for the difficult set sub-problem, but rather to find a high quality solution which may form a good basis for a complete solution which is closer to the global optimal solution. Secondly, as the solutions of sub-problems are fixed during the construction the complete solution, the final complete solution is restricted within some specific regions of the search space. It would be interesting to investigate how much this restriction should be relaxed so that potential global optimal solutions are not excluded from the search. Thirdly, it is worth further investigation to examine the size of the *difficult set* with regard to the "troublesomeness" of the exams included. Due to the interrelated constraints between exams in the timetabling problem, the static measurements such as the degree of difficult exams, the maximum cliques and dense enough quasi-cliques are not enough to provide good evaluations on scheduling the exams into the timetable. Our future work would include designing more robust decomposition methods to include very troublesome exams in the crucial sub-problems. Finally, more efficient Branch-and-Cut procedures will be investigated to obtain solutions with closer gap to the optimal for larger size sub-problems.

5 Conclusions

Due to the large computational time of exam methods, in the literature there is little work on IP for exam timetabling problems. In this paper, we developed and evaluated an IP model for solving sub-problems of exam timetabling problems identified by an adaptive decomposition approach. The difficult sets which are adaptively decomposed from the original large problems are solved optimally by the IP model we proposed. To get a tighter formulation, we not only induce Clique Inequalities deduced from Set Packing problems, but we also introduce problem specific cutting planes. Better average complete solutions can be built based on the solution of the difficult set sub-problem compared with the previous adaptive decomposition approach. Closer investigations of the identified difficult set sub-problem with cliques and quasi-cliques of higher density, and complete solutions built based on optimal solutions of different gaps for the sub-problems are also carried out.

The quality of the complete solution lies in the size and elements of the sub-problem identified and their solutions from the IP model. Due to the observation that the IP formulations of real-world timetabling problems have a huge number of variables and constraints, in future work we will investigate more problem specific properties to obtain more efficient formulations, Branch-and-Cut procedures and alternative LP solution techniques. Larger crucial sub-problems can be addressed and robust techniques can be employed to further improve the adaptive decomposition approach.

References

1. Abdullah S., Ahmadi S., Burke E.K. and Dror M. Investigating Ahuja–Orlins large neighbourhood search for examination timetabling. *OR Spectrum*, 29(2), 351-372, 2007
2. Avella P. and Vasil'ev I. A computational study of a cutting plane algorithm for university course timetabling. *Journal of Scheduling* 8, 497-514, 2005
3. Borndorfer R. and Weismantel R. Set packing relaxations of some integer programs. *Mathematical Programming*. 88, 425-450, 2000
4. Burke E.K., Y. Bykov, J.P. Newall, and S. Petrovic. A time-predefined local search approach to exam timetabling problems. *IIE Transactions*, 36(6), 509-528, 2004
5. Burke E.K., Elliman D.G., Ford P.H. and Weare R.F. Examination timetabling in British universities: A survey. In: Burke E.K. and Ross P. (eds). Practice and Theory of Automated Timetabling: Selected Papers from the 1st International Conference. Lecture Notes in Computer Science, vol. 1153. 76-90, 1996
6. Burke E.K. and Newall J.P. A multi-stage evolutionary algorithm for the timetable problem. *IEEE Transactions on Evolutionary Computation*, 3(1), 63-74, 1999
7. Burke E.K. and Newall J.P. Solving examination timetabling problems through adaptation of heuristic orderings. *Annals of OR*, 129, 107-134, 2004
8. Burke E. K., Newall, J. P. and Weare, R. F. A simple heuristically guided search for the timetable problem. In Proceedings of the international ICSC symposium on engineering of intelligent systems (EIS98), 574–579, 1998.
9. Burke E. K., Hart E., Kendall G., Newall J., Ross P. and Schulenburg S. Hyper-heuristics: An emerging direction in modern search technology. In F. Glover & G. Kochenberger (eds.), *Handbook of Meta-heuristics*, 457-474, 2003
10. Burke E.K. and Petrovic S., Recent research trends in automated timetabling. *European Journal of Operational Research*, 140(2), 266-280, 2002
11. Caramia M., Dell'Olmo P. and Italiano G.F. Novel local search-based approaches to university examination timetabling. *INFORMS Journal on Computing*, 20(1), 86-99, 2008
12. Carter M.W. and Johnson D.G. Extended clique initialization in examination timetabling. *Journal of Operational Research Society*, 52, 538-544, 2001
13. Carter M.W. and Laporte G. Recent developments in practical examination timetabling. In: Burke E.K. and Ross P. (eds), Practice and Theory of Automated Timetabling: Selected Papers from the 1st International Conference. Lecture Notes in Computer Science, vol. 1153. 3-21, 1996
14. Carter M.W., Laporte G. and Lee S.Y. Examination timetabling: Algorithmic strategies and applications. *Journal of Operational Research Society*, 47(3), 373-383, 1996
15. Cote P., Wong T. and Sabouri R. Application of a hybrid multi-objective evolutionary algorithm to the uncapacitated exam proximity problem. In: Burke E.K. and Trick M. (eds.) Practice and Theory of Automated Timetabling V: Selected Papers from the 5th International Conference. Lecture Notes in Computer Science, vol. 3616. 151-168, 2005
16. de Werra D. The combinatorics of the timetabling. *European Journal of Operational Research*, 96, 504-513, 1997
17. Dascalki S.T. Birbas, and E. Housos. An integer programming formulation for a case study in university timetabling. *European Journal of Operational Research*, 153, 117-135, 2004
18. Di Gaspero L. Recolour, shake and kick: A recipe for the examination timetabling problem. In: Burke E.K. and De Causmaecker P. (eds.). Proceedings of the 4th International Conference on Practice and Theory of Automated Timetabling. 21st-23rd August 2002. KaHo St.-Lieven, Gent, Belgium. 404-407, 2002
19. Erben W. A grouping genetic algorithm for graph colouring and exam timetabling. In: Burke E.K. and Erben W. (eds). Practice and Theory of Automated Timetabling III: Selected Papers from the 3rd International Conference. Lecture Notes in Computer Science, vol. 2079. 132-156, 2001
20. Hoffman K.L. and Padberg M. Solving airline crew scheduling problems by branch-and-cut. *Management Science*, 39(6), 657-682, 1993
21. Lewis R., A Survey of metaheuristic-based techniques for university timetabling problems. *OR Spectrum*, 30(1), 167-190, 2008

22. Merlot L.T.G., N. Boland, B.D. Hughes and P.J. Stuckey. A hybrid algorithm for the examination timetabling problem. In: Burke E.K. and De Causmaecker P. (eds). Practice and Theory of Automated Timetabling IV: Selected Papers from the 4th International Conference. Lecture Notes in Computer Science, vol. 2740. 207-231, 2003
23. MirHassani S.A. Improving paper spread in examination timetables using integer programming, *Applied Mathematics and Computation* 179, 702-706, 2006
24. Nemhauser G.L. and Wolsey L.A. Integer and Combinatorial Optimization. Wiley-Interscience, 1999
25. Padberg M.W. On the facial structure of the set packing polyhedra, *Mathematical Programming*, 5, 199-215, 1973
26. Paquete L. and T. Stutzle. Empirical analysis of tabu search for the lexicographic optimization of the examination timetabling problem. In: Burke E.K. and De Causmaecker P. (eds), Proceedings of the 4th International Conference on Practice and Theory of Automated Timetabling. 21st-23rd August 2002. KaHo St.-Lieven, Gent, Belgium. 413-420, 2002
27. Thompson J. and K. Dowsland. A robust simulated annealing based examination timetabling system. *Computers & Operations Research*, 25, 637-648, 1998
28. Ross P., Hart E. and Corne D. Some observations about GA-based exam timetabling. In: Burke E.K. and Carter M.W. (eds.) Practice and Theory of Automated Timetabling II: Selected Papers from the 2nd International Conference. Lecture Notes in Computer Science, vol. 1408. 115-129, 1998
29. Qu R. and Burke E.K. Adaptive decomposition and construction for examination timetabling problems. Proceedings of the 3rd Multidisciplinary International Scheduling: Theory and Applications, 418-425, 2007
30. Qu R., Burke E.K. and McCollum B. Adaptive automated construction of hybrid heuristics for exam timetabling and graph colouring problems. *European Journal of Operational Research*, 198(2), 392-404, 2009
31. Qu R., Burke E. K., McCollum B., Merlot L.T.G. and Lee S.Y. A survey of search methodologies and automated system development for examination timetabling. *Journal of Scheduling*, 12(1), 55-89, 2009
32. White G.M., Xie B.S. and Zonjic S. Using tabu search with longer-term memory and relaxation to create examination timetables. *European Journal of Operational Research*, 153(16), 80-91, 2004