Annals of Operation Research manuscript No.
(will be inserted by the editor)

A Constructive Approach for Examination Timetabling based on Adaptive Decomposition and Ordering

Syariza Abdul-Rahman · Edmund K. Burke · Andrzej Bargiela · Barry McCollum · Ender Özcan

Received: date / Accepted: date

Abstract In this study, we investigate a constructive local search approach for examination timetabling which employs an adaptive decomposition strategy as its key feature. This strategy automatically divides examinations into difficult and easy sets. The examinations in the difficult set are considered to be hard to place into a timetable slot and hence are listed before the ones in the easy set. Moreover, the examinations within each set are ordered using different strategies based on graph colouring heuristics, separately. Initially, the examinations are placed into the easy set. During the construction process, the examinations that cannot be scheduled are identified as the ones causing infeasibility and are moved forward in the difficult set to ensure earlier assignment than the others for subsequent attempts. On the other hand, the examinations that can be scheduled remain in the easy set. Within the easy set, a new subset called the boundary set is introduced to accommodate shuffling strategies to change the given ordering of examinations. The proposed approach, which incorporates different ordering and shuffling strategies, is explored on the Carter benchmark problems. The empirical results show that its performance is promising and comparable to existing constructive and improvement approaches.

Keywords Timetabling · Decomposition · Graph Colouring · Heuristic · Grouping

S. Abdul-Rahman · E. K. Burke · A. Bargiela · E. Özcan
University of Nottingham, School of Computer Science, Jubilee Campus,
Nottingham NG8 1BB, UK
Tel.: (+44) (0)115 - 8466504
Fax: (+44) (0)115 - 8467877
E-mail: {sax, ekb, abb, exo}@cs.nott.ac.uk

B. McCollum
Queen's University Belfast, School of Electronics, Electrical Engineering and
Computer Science, University Road Belfast, BT7 1NN, Northern Ireland, UK
E-mail: b.mccollum@qub.ac.uk

1 Introduction

The focus of this study is the university examination timetabling problem. In general, the examination timetabling problem is concerned with the scheduling of a list of examinations into a restricted number of time-slots while satisfying a predefined set of constraints. *Hard constraints* must be satisfied for creating a feasible solution, for example, no student should take two examinations at the same time. *Soft constraints*, on the other hand, can be broken but it is desirable to satisfy them as much as possible. The degree to which these soft constraints are satisfied provides an indication of the overall quality of a given solution. In relation to examination timetabling, evaluating the average cost of student spread in the timetable as an indicator of how 'good' a given solution is was introduced by Carter and Laporte (1996). More overview of the examination timetabling problem and associated constraints can be found in (Carter and Laporte (1996); Carter et al (1996); Petrovic and Burke (2004); Qu et al (2009)).

If we were to only consider the requirement that no student should sit two examinations at the same time, then the formulation of the examination timetabling problem is analogous to the graph colouring problem. Ülker et al (2007) discusses a grouping representation for this type of examination timetabling problems. The vertices and edges of a graph denote the examinations and the conflicting examinations that should not be scheduled at the same time, respectively. The colour of a vertex denotes a time-slot in the timetable. Heuristic ordering methods for graph colouring have been used to construct an examination timetable (often as the initial step in an improvement process). There are several heuristic ordering methods commonly used in examination timetabling i.e. largest degree, saturation degree, largest weighted degree, largest enrolment and colour degree (Carter (1986); Carter and Laporte (1996), Burke et al (2004b)).

A wide variety of approaches have been applied to examination timetabling. The approaches vary from exact methods to meta-heuristic approaches. Recent applications of search methodologies, such as hyper-heuristics that perform search over the heuristics space (Burke et al (2003); Özcan et al (2008)) and case-based reasoning approaches aim to work at a higher level of generality than typical implementations of meta-heuristics. An illustration of some examples of methodologies employed for examination timetabling is provided in Table 1.

Some recent studies in timetabling have focused on constructive approaches for obtaining high quality solutions. Graph colouring heuristics have been 'customized' with adaptive approaches to order the examinations based on their difficulty of timetabling (Burke and Newall (2004)). This utilises the framework of 'squeaky wheel optimisation' (Joslin and Clements (1999)). In this work, the difficulty indicator of scheduling an examination was subsequently increased based on a certain parameter to enable it be scheduled earlier in the next iteration. In 2009, Abdul Rahman et al (2009) extended this study by introducing more strategies for choosing an examination to be scheduled and the time-slots. In another adaptive approach, Casey and Thompson (2003) developed a GRASP algorithm for solving the examination timetabling problems. In their approach, the next examination to be scheduled is chosen from the top items in the list (called the candidate list) using roulette wheel selection and then assigned to the first available time-slot.

The study by Qu and Burke (2007) describes an adaptive decomposition approach for constructing an examination timetable. This paper draws upon the

Table 1 Some representative methodologies for solving examination timetabling problems (this is not exhaustive)

methodology	reference(s)
cluster-based/decomposition	Balakrishnan et al (1992), Burke and Newall (1999), Qu and Burke (2007)
tabu search	Di Gaspero and Schaerf (2001), White and Xie (2001)
simulated annealing	Thompson and Dowsland (1998), Merlot et al (2003)
great deluge algorithm	Burke et al (2004a)
variable neighbourhood search	Burke et al (2010)
large neighbourhood search	Abdullah et al (2007)
iterated local search	Caramia et al (2001)
GRASP	Casey and Thompson (2003)
genetic algorithm	Burke et al (1995), Ülker et al (2007)
memetic algorithm	Burke and Newall (1999), Özcan and Ersoy (2005), Ersoy et al (2007)
ant algorithm	Eley (2007)
exact method	Boizumault et al (1996), David (1998), Merlot et al (2003)
multi-objective	Petrovic and Bykov (2003), Ülker et al (2007)
hyper-heuristic	Bilgin et al (2007), Ersoy et al (2007), Pillay and Banzhaf (2009)
case-based reasoning	Burke et al (2006)
fuzzy approaches	Asmuni et al (2009)
neural network	Corr et al (2006)
constructive approaches	Burke and Newall (2004), Qu and Burke (2007), Abdul Rahman et al (2009)

research on similar adaptive approaches that make use of a decomposition strategy. We propose a methodology which divides the problem into two sub-problems. We adopt the same naming convention introduced by Qu and Burke (2007) for these sets as difficult and easy. In this study, the problem is decomposed into difficult and easy sets at each iteration. A timetable is constructed based on the associated heuristic ordering for each set. We also introduce an additional set of examinations which is located in between the difficult and easy sets. This is referred to as the boundary set. This study describes several mechanisms associated with the boundary set in order to vary the search space of solutions. In Section 2, we present the details of our approach based on adaptive decomposition and ordering for examination timetabling. Section 3 describes the experimental data and discusses the results. Finally, the conclusion is provided in Section 4.

2 Automated Decomposition and Ordering of Examinations

Most of the timetabling approaches in the literature do not make use of information obtained from the process of building an infeasible timetable. The examinations causing the infeasibility of a solution provide an indication that those examinations are difficult to place and should perhaps be treated in different ways. We propose a general constructive framework as presented in Pseudocode 1 for solving the examination timetabling problem based on the automated decomposition of a set of examinations into two sets i.e. difficult and easy.

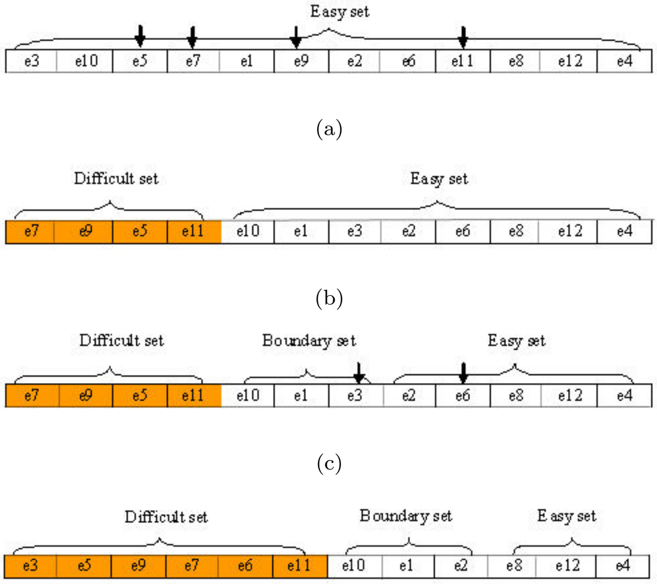
Algorithm 1 Construction of a timetable based on automated decomposition and ordering of examinations.

```
$E = \{e1, e2, ., eN\}$
$BoundarySetSize = \delta$
$EasySet = E$; $DifficultSet = \phi$; $BoundarySet = \phi$; $TempSet = \phi$
Divide $E$ into subsets
for $i = 0$ to MAXIter do
  $OrderExamsWithinSubsets(DifficultSet, EasySet)$
  $BoundarySet = CreateBoundarySet(DifficultSet, EasySet)$
  while there are examinations to be scheduled do
    Consider changing the ordering of examinations using Shuffling-Strategy
    Employ Selection-Strategy to choose an unscheduled exam, $e$
    if $e$ can be scheduled then
      $TempSet = TempSet \cap \{e\}$
      Schedule e in the time-slot with the least penalty
      In the case of the availability of multiple time-slots with the same penalty, choose one randomly
    else
      Move exam $e$ to $DifficultSet$
    end if
    $EasySet = TempSet$
  end while
  Evaluate solution, store if it is the best found so far
end for
```

During each iteration, a new solution is constructed from an ordered list of examinations. The difficult set consists of the examinations that cannot be placed into a time-slot within the timetable due to some conflicts with other examinations from the previous iteration. These examinations need to be associated with a large penalty imposed on the unplaced examinations. On the other hand, the examinations in the easy set cause no violations during the timetabling. In our approach, all the examinations that contribute to the infeasibility in a solution are given priority. They are moved forward in the ordered list of examinations and treated first. Such examinations are detected and included in the difficult set at each iteration and a predefined ordering strategy is employed before their successive assignment to the available time-slots. The remaining examinations (that generate no feasibility issues) are placed into the easy set and the original ordering of those examinations is maintained. In order to incorporate a stochastic component for the selection of examinations from the generated ordering, some shuffling strategies are utilised. The following subsections discuss these strategies.

2.1 Interaction between Difficult and Easy Sets through a Boundary Set

An adaptive decomposition approach is experimented with using two graph colouring heuristics for generating the initial ordering of examinations. We have tested the largest degree heuristic that orders the examinations decreasingly with respect to the number of conflicts with each examination and the saturation degree heuristic that dynamically orders the unscheduled examinations based on the number of available time-slots for each during the timetable construction. The reason for testing these two graph colouring heuristics is to compare their achievement in terms of solution quality and the contrubution of infeasible examinations to the size of difficult set, as they represent static and dynamic ordering heuristics. Initially, all the examinations are considered to be a member of the easy set (as illustrated in Figure 1(a)).


Easy set
e3 e10 e5 e7 e1 e9 e2 e6 e11 e8 e12 e4
Difficult set
Easy set
e7 e9 e5 e11 e10 e1 e3 e2 e6 e8 e12 e4
Difficult set
Boundary set
Easy set
e7 e9 e5 e11 e10 e1 e3 e2 e6 e8 e12 e4
Difficult set
Boundary set
Easy set
e3 e5 e9 e7 e6 e11 e10 e1 e2 e8 e12 e4

(a)

(b)

(c)

(d)

Fig. 1 (a) All examinations are in the easy set in the first iteration and examinations that cause infeasibility are marked, (b) difficult and easy sets after an iteration resulting with an infeasible solution, (c) boundary set with a prefixed size is added to the difficult set after an iteration and reordering is performed, (d) the step in (a) is repeated and the infeasible examinations are placed in the difficult set, the size of the difficult set increased.

During each iteration, the examinations causing infeasibility are identified. As in Figure 1(a), all such examinations are marked as a member of the difficult set to be moved forward towards the top of the list of examinations (Figure 1(b)), while the examinations that caused no violation during the assignment to a time-slot remain in the easy set. In Figure 1(c), the boundary set is created between the difficult and the easy set and is merged with the difficult set before a reordering is performed to the difficult set. In the next iteration, more infeasible examinations are detected and included in the difficult set. Consequently, the size of the difficult set is increased from one iteration to another.

2.2 Swapping the Examinations Between Difficult and Boundary Sets

This strategy shuffles the difficult set and the boundary set by swapping the examinations in between them randomly. Occasionally, the examination causing infeasibility is not necessarily the one that is very difficult to schedule. The infeasibility may happen due to the previous assignment and ordering. This strategy introduces the opportunity for some of the examinations in the difficult set to be chosen later in the timetable. There is also a possibility that the examinations in the boundary set are swapped back to the original set because this process is done randomly. Figure 2 illustrates how the swapping of examinations between two sets might take place.

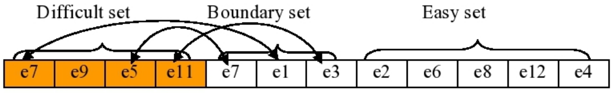


Fig. 2 The boundary set is swapped with the difficult set and is reordered before assigning examinations to the time-slots.

2.3 Roulette Wheel Selection for Examinations

We utilised a roulette wheel selection strategy that incorporates a stochastic element in choosing examinations before assigning them to the time-slots. If there is no improvement evident for a certain time, a list of examinations of size n was chosen from the ordered list in the difficult set from which an examination is chosen based on a probability. The probabilities of an examination being chosen were calculated based on a score, s_i of each examination in the list of size n. The new size of the difficult set will be the set which includes the size of the boundary set whenever there is improvement to the solution quality. The score value, s_i is a dynamic measure that is obtained from the largest and saturation degree values (as in equation 1), where Num_clash_i is the number of examinations in conflict with the examination i, Max_clash is the maximum number of conflicts with all examinations, Sat_degree_i is the saturation degree value for the examination i and Num_slots is the number of time-slots given to the specified problem. The Sat_degree value in this problem is initialised as 1.

$$s_i = \frac{Num\_clash_i}{Max\_clash} + \frac{Sat\_degree_i}{Num\_slots} \qquad (1)$$

The probability, p_i of an examination being chosen from n list of examinations is,

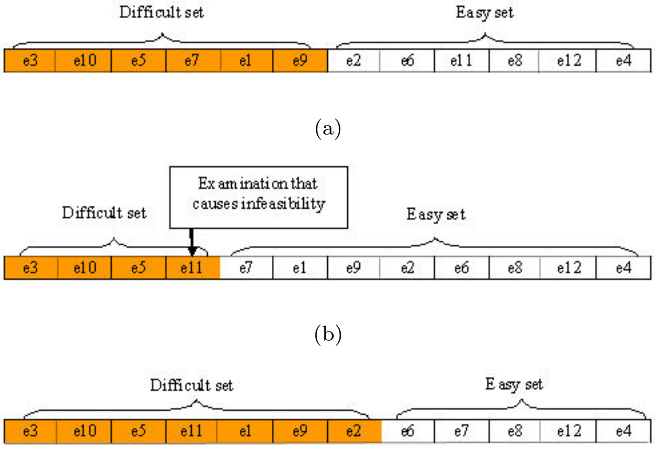
$$p_i = \frac{s_i}{\sum_{i=0}^{n-1} s_i} \qquad (2)$$

A random number from $(0, 1)$ is obtained in order to choose an examination from a list of examinations of size n. An examination with higher score value, s_i will have greater chance to be chosen.

2.4 Comparison of Our Approach to a Previous Study

Qu and Burke (2007) previously proposed an adaptive decomposition approach to construct examination timetables. Their approach starts with an initial ordering of examinations using a graph colouring heuristic, namely saturation degree. In the approach, a perturbation is made by randomly swapping two examinations in order to obtain a better ordering. Examinations are then decomposed into two sets: difficult and easy.

The initial size of the difficult and easy sets is prefixed as half of the number of examinations in a given problem as shown in Figure 3(a). At each iteration, the size of the difficult set is modified according to the feasibility of the solution. If the solution is infeasible after the adjustment of the ordering of examinations then the first examination that causes infeasibility (for example, e11) is moved forward for a fixed number of places (for example, five as illustrated in Figure 3(b)). The size of the difficult set is then re-set to the point where the difficult examination is placed. Otherwise, if a feasible solution or an improved solution is obtained, then the size of the difficult set is increased (Figure 3(c)).


Difficult set
Easy set
e3 e10 e5 e7 e1 e9 e2 e6 e11 e8 e12 e4
Examination that causes infeasibility
Difficult set
Easy set
e3 e10 e5 e11 e7 e1 e9 e2 e6 e8 e12 e4
Difficult set
Easy set
e3 e10 e5 e11 e1 e9 e2 e6 e7 e8 e12 e4

(a)

(b)

(c)

Fig. 3 Difficult and easy sets (a) in the first iteration, (b) after an iteration is over (a) resulting with an infeasible solution, (c) after an iteration is over (a) resulting with a feasible solution.

Our approach initialises with the easy set including all the examinations and the difficult set is formed during each construction phase at each iteration. The size of the difficult set depends on the number of unscheduled examinations that cannot be assigned to any time-slot from all previous iterations. The size of the difficult set never decreases and after a certain number of iteration, the number of examinations in the difficult set might be sustained. On the other hand, in the previous approach, the size of the difficult set is prefixed and increased when the feasible solution or improved solution is obtained statically. The set is also allowed to shrink. Additionally, the previously proposed approach uses an initial ordering and reorders all the examinations without using a heuristic, which is not the case in our approach. Although we have used the same approach for reordering the examinations in difficult and easy sets separately, examinations in different sets can be reordered based on a different heuristic at each iteration.

3 Experiments

Pentium IV 1.86 GHz. Windows machines having 1.97 Gb memory were used during the experiments. 25 runs (trials) were performed for each experiment. The same stopping condition as in Qu and Burke (2007) was chosen for a fair comparison and each run is terminated after 2 000 iterations. Our previous study showed that increasing the number of iterations has no significant effect on the solution quality. So, we decided to increase the number of runs and reduce the number of iterations in order to keep the number of states visited during the search process fixed. The experiments were performed over the benchmark problems introduced by Carter et al (1996) which are also publicly available at `ftp://ftp.mie.utoronto.ca/pub/carter/testprob/`. In this study, we used version I of the 13 problems that were adapted from Qu and Burke (2007) to differentiate various versions of the problem. Two types of graph coloring heuristics used for ordering of the examinations during the experiments: largest degree (LD) and saturation degree (SD). The heuristics used in a given approach will be denoted by a triplet as [*heuristic used for the initial ordering - heuristic used for ordering the examinations in the difficult set - heuristic used for ordering the examinations in the easy set*] from this point onwards.Since initially all examinations are considered to be in the easy set, the same heuristic for the easy set is used as for the initialisation to order the examinations during the search process in this study. This yields four settings: {[LD-LD-LD], [SD-LD-SD], [LD-SD-LD], [SD-SD-SD]}.

3.1 Parameter Tuning Experiments

We have tested our approach using six different boundary set sizes {0, 3, 5, 10, 15, 20} in order to identify the best parameter setting. Figure 4 illustrates the effect of the boundary set size on the average cost over all experiments combining all heuristic ordering choices with adding and swapping strategies for all problems. The results indicate that the boundary set size 3 is the best choice with a lower standard deviation when compared to the other choices. The approach using this parameter value delivers a statistically significant performance variation within a confidence interval of 95% on average when a pairwise comparison is performed to each choice in {0, 10, 15, 20}. Hence, we set the boundary set size to 3 in our approach.

We also investigated the utilisation of a shuffling strategy based on roulette wheel selection in our approach for different list sizes of n in {0, 3, 5, 10, 15}. Figure 5 illustrates the performance of our approach using the roulette wheel selection for different list sizes. The average performance of the approach does not significantly vary between the choices of $n = 3$ and $n = 5$. The list size of 3 is a slightly better choice than 5. A pairwise comparison between $n = 3$ and each value in {0, 10, 15} based on the student's t-test shows that the performance of the approach using

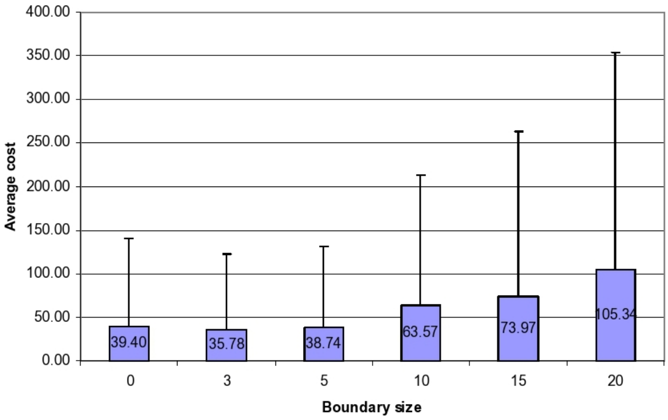


Fig. 4 Average cost over all experiments (and the associated standard deviation) for different boundary set size choices.

$n = 3$ is significantly better than the rest of the choices within a confidence interval of 95%. Hence, the list size for the roulette wheel selection is set to 3.

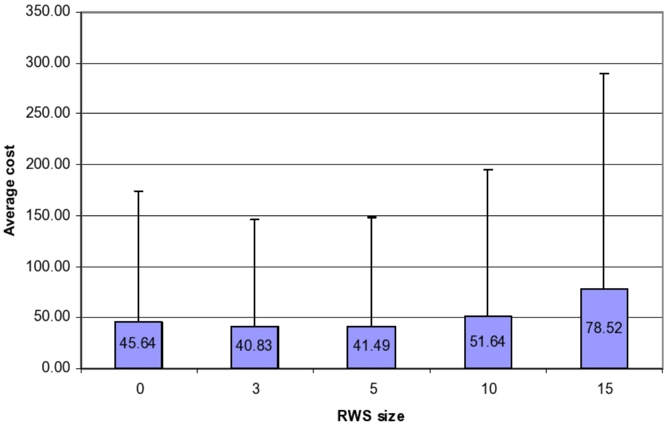


Fig. 5 Average cost over all experiments (and the associated standard deviation) for different boundary set sizes.

3.2 Best Performance Comparison for Different Strategies

Table 2 summarises the best results obtained using our approach with boundary set size fixed as 3 for different combinations of ordering heuristic choices and the strategies of add and swap for each Carter's benchmark problem instance. We observe that the adding boundary set strategy performs better considering that it outperforms the swapping boundary set strategy in eight problem instances. In the overall, saturation degree performs better than largest degree for initial ordering. When the best heuristic ordering for the difficult and the easy sets are analysed, it is also observed that the adding boundary set strategy using the saturation degree initial ordering performs slightly better than using the largest degree initial ordering in seven out of the thirteen problem instances. Similarly, the swapping strategy using the saturation degree initial ordering performs better than using the largest degree initial ordering in nine out of thirteen problem instances. Moreover, from our observation the best combination ordering for the adding boundary set strategy is [LD-SD-LD]???? while the swapping boundary set strategy performed the best with [SD-LD-SD]. An overall comparison of our approach using different strategies based on their best performances reveal that the swapping strategy can improve the solution quality more as compared to the adding strategy. However, this strategy produces higher standard deviation and has chances of generating infeasible timetable.

Table 2 Comparing the best solution quality for the combination of (a) [LD-LD-LD], (b) [SD-LD-SD], (c) [LD-SD-LD], (d) [SD-SD-SD] and the strategies of adding the boundary set into the difficult set and swapping examinations between the boundary and difficult sets with $\delta = 3$. (LD = largest degree; SD = saturation degree) (Bold font indicates the best for different ordering and strategy and italic is the best of all for each problem instance).

	Add the boundary set ($\delta = 3$) into the difficult set				Swap examinations between the boundary ($\delta = 3$) and difficult sets			
Problem	(a)	(b)	(c)	(d)	(a)	(b)	(c)	(d)
car91	5.69	5.70	***5.27***	5.41	5.77	5.77	5.33	**5.32**
car92	4.85	***4.74***	4.79	4.75	4.99	4.81	**4.75**	4.81
ear83 I	41.15	42.27	***41.12***	42.02	41.83	42.24	42.18	**41.34**
hec92 I	12.66	**12.35**	12.69	12.70	12.98	***12.05***	12.53	12.51
kfu93	16.35	16.45	16.11	***16.01***	16.30	**16.25**	16.43	16.33
pur93	6.44	6.56	6.15	***6.05***	6.44	6.45	**6.09**	6.14
lse91	13.44	12.87	**12.44**	12.74	13.43	12.85	***12.41***	12.87
rye93	10.52	**10.30**	10.38	10.39	10.63	***10.24***	10.48	10.36
sta83 I	160.73	***159.03***	160.98	160.05	160.55	**159.62**	160.29	160.29
tre92	9.46	***9.07***	9.49	9.35	9.36	9.51	**9.27**	9.41
ute92	29.15	29.27	28.81	**28.63**	28.96	28.88	29.11	***27.75***
uta92 I	3.88	3.79	**3.73**	***3.72***	3.89	3.82	3.77	3.78
yor83 I	44.70	**44.23**	45.27	44.48	*inf.*	44.93	***44.19***	44.94

Table 3 summarises the best solution quality obtained by our approach with different combinations of ordering heuristic choices for the sets, where boundary set and roulette wheel selection list sizes are fixed as $\delta = 3$ and $n = 3$, respectively, for each problem instance. As it can be observed from the results, the adding boundary set strategy with roulette wheel selection performs better when compared to the swapping strategy with roulette wheel selection in eight problem instances. Moreover, the adding boundary set and roulette wheel selection strategy performs the best with [SD-SD-SD], while the swapping with roulette wheel selection strategy performs the best with either [SD-LD-SD] or [SD-SD-SD]. Comparing the best results obtained from the strategies without roulette wheel selection in Table 2 and the strategies with roulette wheel selection in Table 3, we can observe that the incorporation of the shuffling strategy improves the performance of the approach in most cases.

Table 3 Comparing the best solution quality for the combination of (a) [LD-LD-LD], (b) [SD-LD-SD], (c) [LD-SD-LD], (d) [SD-SD-SD] and shuffling strategies of adding the boundary set into the difficult set and swapping examinations between the boundary and difficult sets with $\delta = 3$ and includes roulette wheel selection for examinations with list size $n = 3$. (LD = largest degree; SD = saturation degree) (Bold font indicates the best for different ordering and strategy and italic is the best of all for each problem instance).

	Add the boundary set ($\delta = 3$) into the difficult set				Swap examinations in the boundary ($\delta = 3$) and difficult sets			
Problem	(a)	(b)	(c)	(d)	(a)	(b)	(c)	(d)
car91	5.75	5.74	**5.30**	5.31	5.74	5.76	***5.17***	***5.17***
car92	4.86	4.82	4.88	***4.74***	5.02	4.79	4.82	**4.76**
ear83 I	41.15	41.85	42.14	***40.91***	42.20	41.84	42.77	**41.33**
hec92 I	***12.26***	12.44	12.43	12.36	**12.47**	12.52	12.55	12.84
kfu93	16.27	16.35	16.27	16.31	**16.23**	16.01	16.42	***15.85***
pur93	6.42	6.48	**6.07**	**6.07**	6.41	6.48	***5.87***	6.02
lse91	12.93	12.77	***12.58***	12.84	12.69	12.73	**12.67**	13.01
rye93	10.72	**10.22**	10.39	10.40	10.61	***10.11***	10.46	10.41
sta83 I	160.51	***158.12***	161.59	159.20	159.62	**158.55**	160.29	160.29
tre92	9.33	9.40	9.37	***9.30***	9.60	**9.49**	9.58	9.57
ute92	***27.71***	28.37	27.87	27.80	28.54	**27.89**	28.58	28.37
uta92 I	3.91	3.82	3.77	**3.74**	3.92	3.88	***3.65***	***3.65***
yor83 I	46.30	45.00	44.44	***43.98***	*inf.*	45.39	**44.26**	44.55

3.3 Observations

The overall results once again highlight the importance of the methodology used to change the ordering of difficult examinations, particularly the ones causing infeasibility. In our approach, the ordering of the examinations within the difficult set with respect to the others appears to be vital combined with the assignment strategy. As shown in Figure 6, for the experiments adding and swapping the boundary set and difficult set with shuffling strategy of roulette wheel selection, the average number of the examinations in the difficult set varies with different ordering strategies. The approach using the largest degree ordering generates infeasibility more often for a given solution during the time-slot assignments as compared to the one using the saturation degree ordering. This causes larger difficult set sizes at the end of the search process. On the other hand, saturation degree ordering might easily produce a feasible solution for some problem instances (for example car91 and uta92 I). However, using saturation degree alone does not guarantee a good solution quality.

In some cases, using the saturation degree ordering may easily produce a feasible solution and in general, a constructive approach stops. In our approach, the use of adding an swapping strategies acts as a diversification mechanism causing

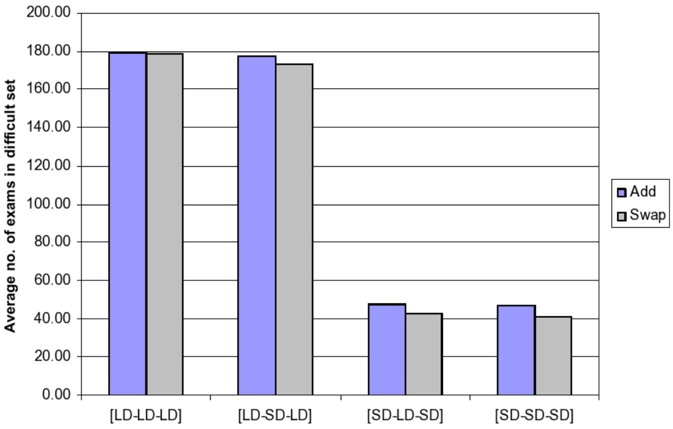


Fig. 6 Average number of examinations in the difficult set (its size) over all problems considering all shuffling strategies using different initialisation and reordering heuristics. (LD=largest degree, SD=saturation degree, Add=adding strategy, Swap=swapping strategy).

the search process to continue. Of course, this may cause infeasible examinations (increasing the number of examinations in the difficult set), emphasizing the importance of ordering for the difficult set. Yet, this provides the advantage of not getting stuck during the search process and searching other possible feasible solution which might be even better than the best solution found so far.

Figure 7 illustrates the size of the difficult set and the solution quality at each 100 iteration for different combination of initial ordering and reordering heuristics for kfu93 problem instance. It shows that using the largest degree initial ordering causes an increase in the number of examinations (generating infeasible examinations) when compared to the saturation degree initial ordering. The plot in Figure 7 shows that there is a significant decrease in the solution quality when the difficult set size is increased for different heuristic combination. [LD-SD-LD], however, does not show any improvement in the solution quality for some time and then starts to show improvement after the shuffling strategy of roulette wheel selection is incorporated while, [SD-LD-SD] shows a slight movement and remain steady for a certain time even though there is small increased in the number of examinations in the difficult set. Meanwhile, [SD-SD-SD] shows a drastic change in the solution quality which is consistent with increasing size of the difficult set. It is interesting to observe that the increasing size of difficult set by [LD-LD-LD] in this Figure 7 give higher possibility of getting a good solution quality with the help of the boundary size and the shuffling strategy of roulette wheel selection.

3.4 Comparison to the Previous Constructive Approaches

Table 4 compares our best approach to the other previous constructive approaches. The examination scheduling problem by Carter et al (1996) is based on graph colouring problem. It incorporated several sequencing strategies and backtracking procedure is applied when the examinations cannot be scheduled in the time-slot.

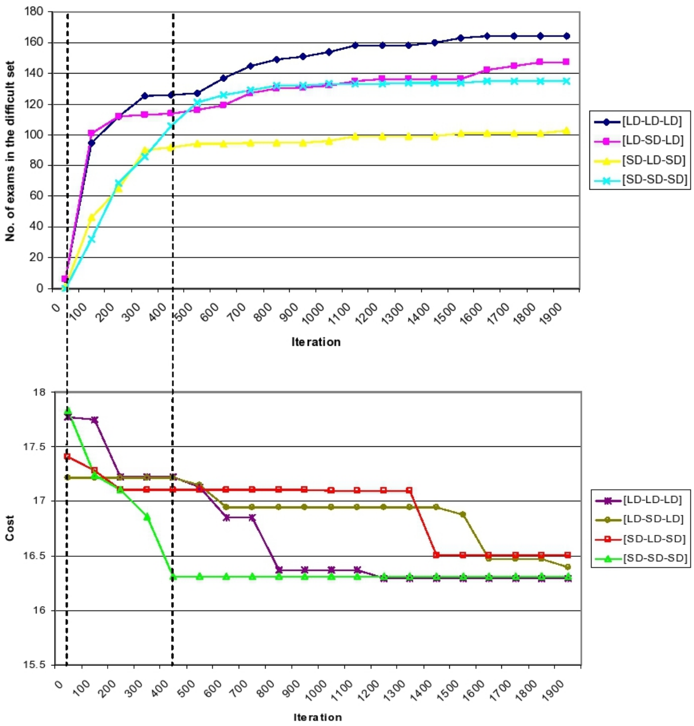


Fig. 7 The change in the size of the difficult set and the solution quality at every 100 iteration during the sample runs for kfu93. (LD= largest degree, SD= saturation degree).

The backtracking procedure worked by unscheduled all the previous assignment by removing the assigned examination in timetable and reschedule them back into new periods after giving priority to the problematic examinations. In order to limit the backtracking procedure, Carter et al (1996) added the tabu search method so that the algorithm will converge. The datasets has been tested with 40 different sequencing strategies and the result presented in Table 4 is the best obtained by them. Burke and Newall (2004) proposed an adaptive heuristic orderings technique that can adapt to any given problem by adding a heuristic modifier to the basic heuristic technique. It works by promoting difficult examinations to be schedule first at each of iteration based on its order. In the next study, Asmuni et al (2009)introduced fuzzy approach by combining two graph colouring heuristics at the same time to order the examinations based on their difficulties. Fuzzy approach is used to represent the knowledge from the heuristics (named as input variables), evaluate them and construct an examination weight as an input variable. The 'bumped back' strategy is employed if examination cannot be scheduled into timetable. The study by Abdul Rahman et al (2009) extended the study by Burke and Newall (2004) by introducing strategies to choose an examination in the ordering with different parameter setting and strategies to increase the difficulty of examinations.

The method of Qu and Burke (2007) as described in Section 2.4 is the closest approach that can be compared to our approach, since they have also implemented a decomposition strategy. Comparing the solutions across all problem instances, it is observed that our approach does not yield the best results. However, it provides a better result when compared to the approach proposed by Qu and Burke (2007) for car91. Moreover, we have obtained better results than the approach in Carter et al (1996) for four problems (car91, car92, sta83 I, tre92), Burke and Newall (2004) for one problem (sta83 I) and Asmuni et al (2009) for two problems (sta83 I and ute92), respectively. Unfortunately, Burke and Newall (2004) and Qu and Burke (2007) did not provide a result for rye93 and the rest did not provided a result for pur93. Burke and Newall (2004) applied their approach to a variant of the pur93 instance which is different than the one used in this study.

Table 4 Comparison of different constructive approaches. ((1) Carter et al (1996), (2) Burke and Newall (2004), (3) Asmuni et al (2009), (4) Abdul Rahman et al (2009), (5) Qu and Burke (2007), (6) [SD-SD-SD] with RWS, (7) Best obtained using heuristic combinations for $\delta = 3$ and $n = 3$) (The bold entries indicate the best results for constructive approaches only, while the italic and bold ones indicate the best results for the decomposition approach).

Problem	(1)	(2)	(3)	(4)	(5)	(6)	(7)
car91	7.10	**4.97**	5.29	5.08	5.45	***5.31***	5.17
car92	6.20	**4.32**	4.54	4.38	***4.5***	4.76	4.74
ear83 I	36.40	36.16	37.02	38.44	**36.15**	40.91	40.91
hec92 I	**10.80**	11.61	11.78	11.61	***11.38***	12.36	12.26
kfu93	**14.00**	15.02	15.80	14.67	***14.74***	16.31	15.85
pur93	3.90	-	-	-	-	***6.07***	5.87
lse91	**10.50**	10.96	12.09	11.69	***10.85***	12.84	12.58
rye93	**7.30**	-	10.38	9.49	-	***10.40***	10.11
sta83 I	161.50	161.90	160.40	157.72	**157.21**	159.20	158.12
tre92	9.60	**8.38**	8.67	8.78	***8.79***	9.30	9.30
ute92	**25.80**	27.41	28.07	26.63	***26.68***	27.80	27.71
uta92 I	3.50	**3.36**	3.57	3.55	***3.55***	3.74	3.65
yor83 I	41.70	40.88	**39.80**	40.45	***42.2***	43.98	43.98

3.5 Comparison to the Previous Improvement Approaches

This section presents a comparison of our approach to some other previously proposed improvement approaches. These approaches usually consist of multiple phases that first attempts to construct a feasible solution before proceeding with the improvement phases. Caramia et al (2001) proposed a multi phase local search algorithms that starts with a greedy scheduler to create a feasible timetable. A penalty-decreaser and penalty-trader then were used to improve the solution quality. Meanwhile, Di Gaspero and Schaerf (2001) and Paquete and Stuetzle (2002) investigate a tabu search approach for the examination timetabling problem. Di Gaspero and Schaerf (2001) used the feature of graph coloring problem and in order to guide the search the study adapted a variable size of tabu list while, Paquete and Stuetzle (2002) used a lexicographic formulation similar to the multi-criteria approaches. Burke and Newall (2003) presented some multi-stage local search methods i.e. hill climbing, simulated annealing and great deluge algorithm to improve the high quality initial solution obtained from an adaptive approach during the construction phase. In another study, Merlot et al (2003) presented a three-phase hybrid algorithm for examination timetabling problem that consisted of constraint programming, simulated annealing and hill climbing. Eley (2007) applied ant systems and Max-Min ant systems to the examination timetabling problem.

Table 5 provides the comparison of improvement approaches to our approach. It is observed that our approach is comparable to some improvement approaches. Our approach is better than Di Gaspero and Schaerf (2001) for eight problem instances (car91, hec92 I, kfu93, lse91, sta83 I, tre92, ute92, uta92 I), Caramia et al (2001) for four problem instances (car91, car92, sta83 I, tre92), Paquete and Stuetzle (2002) for three problem instances (kfu93, lse91, ute92) and a tie with tre92 and Burke and Newall (2003) (sta83 I) and Eley (2007) (car91) for one problem instance. Only two approaches (Caramia et al (2001) and Eley (2007)) provided results for pur93 and only three approaches (Caramia et al (2001), Merlot et al (2003) and Eley (2007)) provided result for rye93 while, Paquete and Stuetzle (2002) did not provide any result for car91, car92 and also for uta92 I.

Table 5 Comparison of different improvement approaches. ((1) Di Gaspero and Schaerf (2001), (2) Caramia et al (2001), (3) Paquete and Stuetzle (2002), (4) Burke and Newall (2003), (5) Merlot et al (2003), (6) Eley (2007), (7) Best obtained using heuristic combinations for $\delta = 3$ and $n = 3$) (The bold entries indicate the best results).

Problem	(1)	(2)	(3)	(4)	(5)	(6)	(7)
car91	6.20	6.60	-	**4.65**	5.10	5.20	5.17
car92	5.20	6.00	-	**4.10**	4.30	4.30	4.74
ear83 I	45.70	**29.30**	38.90	37.05	35.10	36.80	40.91
hec92 I	12.40	**9.20**	11.20	11.54	10.60	11.10	12.26
kfu93	18.00	13.80	16.50	13.90	**13.50**	14.50	15.85
pur93	-	**3.70**	-	-	-	4.60	5.87
lse91	15.50	**9.60**	13.20	10.82	10.50	11.30	12.58
rye93	-	**6.80**	-	-	8.40	9.8	10.11
sta83 I	160.80	158.20	158.10	168.73	**157.30**	**157.30**	158.12
tre92	10.00	9.40	9.30	**8.35**	8.40	8.60	9.30
ute92	29.00	**24.40**	27.80	25.83	25.10	26.40	27.71
uta92 I	4.20	3.50	-	**3.20**	3.50	3.50	3.65
yor83 I	41.00	**36.20**	38.90	37.28	37.40	39.30	43.98

4 Conclusion

This study discusses a novel approach based on adaptive strategies that decomposes the examinations in a given problem into two sets: a set of difficult to schedule and a set of easy to schedule examinations. This decomposition is performed automatically at each iteration, and is augmented with suitable ordering of examinations within each set. In this study, it is observed that by merging or swapping the boundary set with the difficult set we could improve solution quality. A stochastic component based on roulette wheel selection is embedded into the approach in order to shuffle the order of examinations. This mechanism gives a higher chance to an examination with a higher score to be selected for timetabling. Different parameter were tested on the boundary size and roulette wheel selection size and the parameter setting is done based on the statistical analysis. It is observed that using saturation degree could decrease the possibility of creating infeasible solutions and that dynamic ordering gives better ordering of examinations in the list. This preliminary study shows that the proposed approach is simple to implement, yet it is competitive to some other previously proposed constructive as well as improvement approaches. In this study, the same ordering heuristics are used for reordering the examinations in the difficult and easy sets. In fact, the proposed framework allows the use of different strategies. As a direction for future work, different strategies will be investigated for reordering the examinations and choosing the examinations from the difficult set.

References

Abdul Rahman S, Bargiela A, Burke EK, McCollum B, Özcan E (2009) Construction of examination timetables based on ordering heuristics. In: In Proceedings of the 24th International Symposium on Computer and Information Sciences, pp 727–732

Abdullah S, Ahmadi S, Burke EK, Dror M (2007) Investigating ahuja-orlin's large neighbourhood search approach for examination timetabling. OR Spectrum 29(2):351–372

Asmuni H, Burke EK, Garibaldi JM, McCollum B, Parkes AJ (2009) An investigation of fuzzy multiple heuristic orderings in the construction of university examination timetables. Computers and Operations Research 36(4):981–1001

Balakrishnan N, Lucena A, Wong RT (1992) Scheduling examinations to reduce second order conflicts. Computers and Operations Research 19:353–361

Bilgin B, Özcan E, Korkmaz EE (2007) An experimental study on hyper-heuristics and exam scheduling. In: Burke EK, Rudova H (eds) Lecture notes in computer science (Practice and theory of automated timetabling VI: selected papers from the 6th international conference), Berlin: Springer, vol 3867, pp 394–412

Boizumault P, Delon Y, Peridy L (1996) Constraint logic programming for examination timetabling. Journal of Logic Programming 26(2):217–233

Burke EK, Newall JP (1999) A multistage evolutionary algorithm for the timetable problem. IEEE Trans Evolutionary Computation 3(1):63–74

Burke EK, Newall JP (2003) Enhancing timetable solutions with local search methods. In: Lecture notes in computer science: Practice and theory of automated timetabling IV: selected papers from the 4th international conference, Springer-Verlag, vol 2740, pp 195–206

Burke EK, Newall JP (2004) Solving examination timetabling problems through adaptation of heuristic orderings. Annals of Operations Research 129:107–134

Burke EK, Elliman DG, Weare R (1995) A hybrid genetic algorithm for highly constrained timetabling problems. In: In the Procedings of the 6th International Conference on Genetic Algorithms (ICGA'95), San Francisco, CA, USA, Pittsburgh, USA.

Burke EK, Hart E, Kendall G, Newall J, Ross P, Schulenburg S (2003) Handbook of Meta-Heuristics, Kluwer, chap Hyper-Heuristics: An Emerging Direction in Modern Search Technology, pp 457–474

Burke EK, Bykov Y, Newall JP, Petrovic S (2004a) A time-predefined local search approach to exam timetabling problem. IIE Transactions 36(6):509–528

Burke EK, Kingston J, de Werra D (2004b) Handbook of Graph Theory, Chapman Hall/CRC Press, chap Applications to timetabling, pp 445–474

Burke EK, Petrovic S, Qu R (2006) Case based heuristic selection for timetabling problems. Journal of Scheduling 9(2):115–132

Burke EK, Eckersley AJ, McCollum B, Petrovic S, Qu R (2010) Hybrid variable neighbourhood approaches to university exam timetabling. European Journal of Operational Research 206(1):46–53

Caramia M, Dell'Olmo P, Italiano GF (2001) New algorithms for examination timetabling. In: Lecture Notes in Computer Science: Selected paper in the WAE '00: Proceedings of the 4th International Workshop on Algorithm Engineering, Springer-Verlag, London, UK, vol 1982, pp 230–242

Carter MW (1986) A survey of practical applications of examination timetabling algorithms. Operational Research 34(2):193–202

Carter MW, Laporte G (1996) Recent developments in practical examination timetabling. In: Selected papers from the First International Conference on Practice and Theory of Automated Timetabling, Springer-Verlag, London, UK, pp 3–21

Carter MW, Laporte G, Lee S (1996) Examination timetabling: Algorithmic strategies and applications. Journal of the Operational Research Society 47(3):373–383

Casey S, Thompson J (2003) Grasping the examination scheduling problem. In: Lecture notes in computer science: Practice and theory of automated timetabling IV: selected papers from the 4th international conference, Berlin: Springer, vol 2740, pp 234–244

Corr P, McCollum B, McGreevy M, McMullan P (2006) A new neural network based construction heuristic for the examination timetabling problem. In: In Parallel Problem Solving from Nature - PPSN IX, Lecture Notes in Computer Science, Springer-Verlag, vol 4193, pp 392–401

David P (1998) A constraint-based approach for examination timetabling using local repair techniques. In: Burke EK, Carter MW (eds) Lecture notes in computer science. Practice and theory of automated timetabling II: selected papers from the 2nd international conference, Berlin: Springer, vol 1408, pp 169–186

Di Gaspero L, Schaerf A (2001) Tabu search techniques for examination timetabling. In: Lecture notes in computer science: Practice and theory of automated timetabling III: selected papers from the 3rd international conference, Berlin: Springer, London, UK, pp 104–117

Eley M (2007) Ant algorithms for the exam timetabling problem. In: Burke EK, Rudova H (eds) Lecture notes in computer science: Practice and theory of automated timetabling VI: selected papers from the 6th international conference, Berlin: Springer, vol 3867, pp 364–382

Ersoy E, Özcan E, Sima Uyar A (2007) Memetic algorithms and hyperhill-climbers. In: Baptiste G Pand Kendall, Kordon AM, Sourd F (eds) Lecture notes in computer science: Multidisciplinary international conference on scheduling: theory and applications: selected papers from the 3rd international conference, pp 159–166

Joslin DE, Clements DP (1999) "squeaky wheel" optimization. Journal of Artificial Intelligence Research 10:353–37

Merlot LTG, Boland N, Hughes BD, Stuckey PJ (2003) A hybrid algorithm for the examination timetabling problem. In: Burke EK, Erben W (eds) Lecture notes in computer science: Practice and theory of automated timetabling V: selected papers from the 4th international conference, Berlin: Springer, vol 2740, pp 207–231

Özcan E, Ersoy E (2005) Final exam scheduler - fes. In: In Proceedings of the IEEE Congress on Evolutionary Computation (CEC'05), vol 2, pp 1356–1363

Özcan E, Bilgin B, Korkmaz EE (2008) A comprehensive analysis of hyperheuristics. Intelligent Data Analysis 12(1):3–23

Paquete L, Stuetzle T (2002) Empirical analysis of tabu search for the lexicographic optimization of the examination timetabling problem. In: The Proceedings of the 4th International Conference on the Practice and Theory of Automated Timetabling (PATAT IV), Gent, Belgium., p 413 420

Petrovic S, Burke E (2004) University timetabling. In: Handbook of Scheduling: Algorithms, Models, and Performance Analysis, chapter 45, Chapman Hall/CRC Press

Petrovic S, Bykov Y (2003) A multiobjective optimisation technique for exam timetabling based on trajectories. In: Burke EK, De Causmaecker P (eds) Lecture notes in computer science: Practice and theory of automated timetabling IV: selected papers from the 4th international conference, Berlin: Springer, vol 2740, pp 179–192

Pillay N, Banzhaf W (2009) A study of heuristic combinations for hyper-heuristic systems for the uncapacitated examination timetabling problem. European Journal of Operational Research 197(2):482–491

Qu R, Burke EK (2007) Adaptive decomposition and construction for examination timetabling problems. In: In Multidisciplinary international scheduling: theory and applications (MISTA'07), Paris, France., pp 418–425

Qu R, Burke EK, Mccollum B, Merlot LT, Lee SY (2009) A survey of search methodologies and automated system development for examination timetabling. Journal of Scheduling 12(1):55–89

Thompson JM, Dowsland KA (1998) A robust simulated annealing based examination timetabling system. Comput Oper Res 25(7-8):637–648

Ülker O, Özcan E, Korkmaz EE (2007) Linear linkage encoding in grouping problems: applications on graph coloring and timetabling. In: Practice and theory of automated timetabling 2006, Springer-Verlag, selected papers, Lecture notes in computer science, vol 3867, pp 347–363

White GM, Xie BS (2001) Examination timetables and tabu search with longer-term memory. In: E K B, Erben W (eds) Lecture notes in computer science. Practice and theory of automated timetabling III: selected papers from the 3rd international conference, Berlin: Springer, vol 2079, pp 85–103